

Linux RedHat 8.0

**Installation de base
Linux station de travail
Installation des services Internet
Compléments système
Installation de bases de données
Développement sous Linux**

Serge TAHÉ, mai 2003

ISTIA - INSTITUT DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR D'ANGERS

Avertissement

Ce document a été écrit en 2003 pour Linux RedHat 8.0 (RH8). Depuis Linux a évolué : diverses distributions matures sont disponibles et les modes d'installation de celles-ci sont différentes de celle expliquée ici. Il y a donc beaucoup d'informations désormais obsolètes dans ce document, et également beaucoup d'informations toujours d'actualité... Le lecteur intéressé par suivre ce document trouvera la distribution Linux RedHat 8.0 à l'Url :

[ftp://archive.download.redhat.com/pub/redhat/linux/8.0/fr/iso/i386].

Le document présente l'installation de RH8 dans un cadre particulier : celui du réseau privé d'une école. En effet, le document a servi de TP à des étudiants pendant de nombreuses années. Le lecteur devra donc adapter ce qui est écrit à son propre environnement et ne pas s'étonner du vocabulaire particulier utilisé : celui d'un TP.

Serge Tahé, août 2006.

Introduction

Le but de ce document et du travail pratique qui l'accompagne est de montrer qu'on peut monter en quelques heures un serveur Linux avec les principaux services internet (web, ftp, telnet, sendmail, pop, ...) qui soit également un serveur de fichiers pour un réseau local de PC/Windows (samba). Linux étant gratuit et fonctionnant sur un simple PC, un administrateur peut se permettre d'installer, de désinstaller puis de réinstaller plusieurs fois Linux. C'est une possibilité inestimable de formation.

Ce document permet de monter une machine Linux qui pourrait être mise en exploitation dans un site réel comme l'ISTIA qui a plusieurs machines Linux semblables à celles que vous allez monter. Si vous êtes novice, vous aurez parfois l'impression de ne pas bien comprendre ce que vous faites. Lorsque vous monterez ultérieurement une deuxième machine, les choses vous paraîtront déjà plus claires. Ce n'est que lorsque vous administrerez réellement une machine Linux que vous comprendrez les détails qui pourront vous échapper lors de cette première installation dont le rôle premier est de vous "mettre le pied à l'étrier".

Dans le monde Unix, on distingue :

- les **utilisateurs**, ceux qui sont amenés à travailler avec une machine Unix comme d'autres avec une machine Windows. Ceux-là connaissent les commandes de base et/ou les interfaces graphiques.
- les **administrateurs**, ceux qui gèrent les différents services présents sur le serveur Unix (web, mail, ftp, comptes utilisateurs, service de noms, service de fichiers, service de bases de données...). Ceux-là font à longueur de journée de l'optimisation, de la surveillance, des installations de nouveaux services, ... Ils sont souvent amenés, dans le cadre de l'optimisation de services, à développer des scripts d'administration. Ils le font dans un des langages de base d'Unix (sh, ksh, csh, bash, ...) ou avec des langages plus performants (Perl, Python,...)
- les **développeurs**, ceux qui écrivent les programmes que vous utilisez sous Unix.

Dans ce document, on fait surtout un travail d'**administrateur**. Il serait sain d'être un utilisateur avancé avant d'être un administrateur. Celui qui connaît les commandes de base d'Unix, qui travaille déjà sous Unix aura bien sûr ici plus de facilités que quelqu'un qui découvre le monde Unix pour la première fois. On a cherché cependant à rendre possible pour tous cette installation de Linux en décrivant les commandes Unix qui doivent être utilisées pour installer les divers services. Il faudra cependant garder à l'esprit pour l'avenir, qu'un administrateur doit être d'abord un utilisateur averti. On peut ajouter qu'un administrateur gagne beaucoup à être également un développeur de scripts averti, point qui n'est nullement abordé ici.

Une fois chez vous, vous pourrez refaire cette installation sur votre ordinateur personnel. Procurez-vous l'un des nombreux livres sur Linux, téléchargez les CD de la distribution linux de votre choix et allez-y !

Serge Tahé

1 Notions de base des réseaux TCP-IP

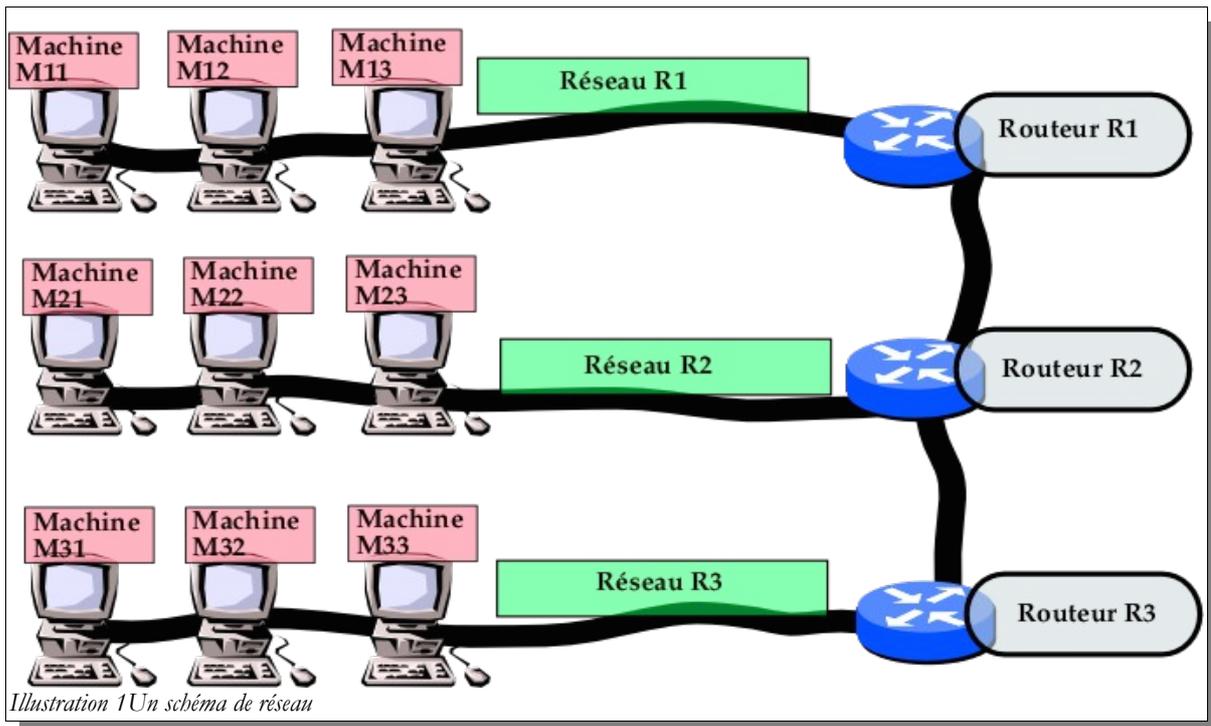
L'installation d'une machine Linux au sein d'un réseau TCP-IP nécessite la connaissance de notions de base sur ce type de réseau. Nous les présentons ici avec des exemples tirés du réseau de l'ISTIA.

1.1 Un réseau ...de ressources

Lorsqu'on désire mettre en commun des ressources en vue de les partager ou mieux les utiliser, on installe un réseau. Un réseau peut être vu comme une collection de machines (pas forcément des ordinateurs) pouvant communiquer entre elles, et se rendant des services. L'ensemble des règles qui régissent la communication est appelé *protocole*. La mise en place d'un réseau passe par une phase d'installation physique (câbles, cartes, etc), une phase de configuration (paramétrage des fichiers) et une phase d'exploitation et de maintenance. Avec une carte Ethernet on peut connecter sa machine Unix au reste du monde en passant par TCP/IP (Transport Control Protocol/Internet Protocol). Une machine Unix équipée pour TCP/IP dispose par exemple des services comme `telnet` ou `ftp` pour la connexion à distance sur une machine et le transfert de fichiers.

1.2 Généralités

Un réseau local utilisant les protocoles de l'Internet est souvent appelé un **Intranet**. Le réseau informatique de l'ISTIA est essentiellement un réseau de PC. Parmi les machines sur le réseau, certaines jouent un rôle particulier : les serveurs. Ce sont des machines auxquelles d'autres machines, appelées machines clientes, demandent des services. On trouve à l'ISTIA, des serveurs NT et Linux.



Le protocole d'échanges de données sur Internet appelé protocole IP est un protocole d'échanges de réseau à réseau. Des machines appelées **routeurs** et représentées ci-dessus par le symbole R, sont chargées de relier les réseaux locaux entre-eux. Au sein d'un réseau local Ri, ce ne sont plus les adresses Internet IP qui font la loi. En effet, on peut trouver différents protocoles réseau au sein d'un réseau local et pas seulement le protocole IP de l'internet. Pour en citer quelques-uns : le protocole IPX/SX de Novell ou Netbios de Microsoft. Il existe différents types de réseaux locaux, le plus répandu étant le réseau **Ethernet**. Au sein d'un tel réseau, une machine est repérée par son **adresse Ethernet** (et non Internet), une adresse inscrite dans la mémoire de la carte réseau qui relie l'ordinateur au réseau Ethernet. Tous les protocoles utilisés sur un réseau Ethernet devront au final référencer une machine par son adresse Ethernet. Comment cela se passe-t-il pour le protocole IP par exemple ?

Si dans un même réseau local Ethernet, une machine d'adresse IP IP1 et d'adresse ethernet ETH1 veut envoyer des données à la machine d'adresse IP IP2, elle procède ainsi :

- La machine IP1 envoie un message à toutes les machines du réseau local où elle se trouve (broadcast) demandant " Qui a l'adresse IP IP2 ? ". Dans cette demande, elle envoie également son adresse ethernet ETH1.
- Toutes les machines reçoivent ce message. Seule la machine d'adresse IP IP2 répond "C'est moi". Elle envoie cette réponse à la seule machine d'adresse ethernet ETH1 en ajoutant sa propre adresse ethernet ETH2.
- La machine ETH1 peut alors envoyer les données à la machine ETH2 selon le protocole ETHERNET.

1.3 Un peu de formalisme : Les réseaux, vus selon l'architecture OSI

7	Application
6	Présentation
5	Session
4	Transport
3	Réseau
2	Liaison de donnée
1	Physique

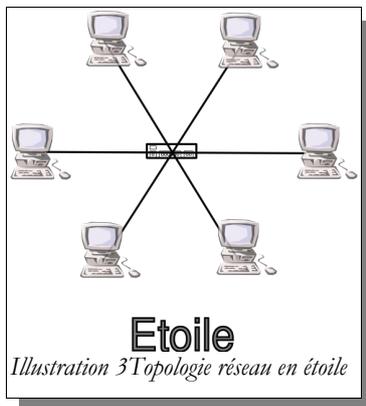
L'ISO (*International Standards Organization*) a défini un modèle architectural des réseaux, appelé architecture OSI (*Open Systems Interconnect*). Cette architecture constitue un modèle de référence à 7 couches, représenté sur la figure 2. Chaque couche du modèle représente une fonction effectuée lorsque des données sont transférées entre des applications en communication.

Ce chapitre étant consacré aux différentes technologies réseau utilisées pour le stage, seules les 4 premières couches seront abordées. Il convient, néanmoins, de noter que l'architecture OSI n'est qu'un *modèle*. A ce titre, il n'est pas forcément implanté de cette façon dans les réseaux

existants. Certaines technologies font des entorses au modèle OSI, d'autres utilisent un modèle de référence très différent du modèle OSI. Il est souvent peu commode de trouver des équivalents entre les deux mondes.

Malgré tout, le choix du modèle OSI est assez pratique dans ce chapitre, car, actuellement, il est toujours fait référence aux 3 premières couches dans la littérature technique. De plus, la signification de ces couches est assez claire pour tout le monde. Cela permet d'avoir une référence commune dès que l'on considère l'ensemble des technologies existantes.

La première couche physique est constituée, pour simplifier, par le câblage. Le réseau de l'ISTIA est câblé en étoile. Tous les PC sont connectés à un matériel central (le commutateur) :



La connexion entre les PC et le commutateur se fait via des câbles en paire torsadée, (RJ45, 10-100 Mbits/s).

Maintenant, il faut normaliser la façon dont ces machines communiquent entre elles pour qu'elles puissent s'entendre. C'est le rôle de la couche 2 (Liaison de données). Les machines communiquent via le protocole **Ethernet**. Les PC sont équipés d'une carte réseau Ethernet. Le commutateur est un commutateur Ethernet. Chaque carte Ethernet dispose d'une adresse unique (donnée par le constructeur). C'est l'adresse Ethernet, appelée aussi adresse MAC. **La couche 2 utilisée est Ethernet.**

Si nos PC étaient uniquement connectés à un seul et unique réseau, cela serait suffisant. Mais ce n'est pas le cas. Chaque PC doit pouvoir se connecter à d'autres machines, un peu partout dans le monde (pour aller consulter un serveur, une page WWW). Il est donc nécessaire de pouvoir localiser un PC et son réseau au niveau mondial. C'est le rôle de la couche 3 (Réseau). **La couche de niveau 3 est ici IP (Internet Protocol).**

La couche 3 assure une communication de machine à machine. Une machine peut héberger divers services (web, ftp, mail, ...). Dans une application client-serveur sur le réseau TCP-IP, le client s'adresse à un service particulier. La communication client-serveur est assurée par la couche 4, dite **couche transport**. Cette couche a connaissance de la notion de n° de service (port) alors que la couche réseau 3 n'a elle connaissance que de la notion d'adresse IP.

Les applications client-serveur s'appuient sur les couches 1-4 pour communiquer. Elles se situent dans les couches 5 à 7.

1.4 Zoom sur la couche de niveau 3 : le protocole IP

Les machines de l'Internet sont repérées par des adresses dites **adresses IP** du nom du protocole qui sous-tend les échanges sur l'Internet (IP=Internet Protocol). Deux machines de l'Internet ont nécessairement des adresses IP différentes. Celles-ci ont pour le moment la forme **I1.I2.I3.I4** où I_i est un nombre entre 1 et 254. Par exemple, l'adresse IP de la machine *istia.univ-angers.fr* est 193.49.146.171. Lorsqu'une machine d'adresse A1 envoie un message à une machine d'adresse A2, celui-ci va passer par un certain nombre de machines intermédiaires. Internet est un réseau de réseaux. Chaque machine, avant d'appartenir à Internet, appartient à un réseau dit réseau local. Internet est le réseau qui connecte ces réseaux locaux entre eux. C'est pourquoi on l'appelle le *réseau des réseaux*.

Introduit par Unix, TCP/IP est l'ensemble des protocoles standardisés pour faire communiquer des machines Unix entre elles. D'autres protocoles existent, ou ont existé (IPX/SPX de Novell, NetBEUI de IBM/Microsoft, AppleTalk d'Apple). TCP/IP s'est imposé au milieu des années 1990 comme LE protocole de communication, et est désormais utilisé par tous les systèmes. La version actuelle de TCP/IP est la version 4. On parle d'IPv4. Une nouvelle version d'IP est disponible et en cours de déploiement. Il s'agit d'IPv6.

Une machine de l'INTERNET dispose d'une adresse IP (unique) permettant de l'identifier et de l'adresser. C'est l'ICANN (organisme international, succédant au NIC) qui attribue les (découpages d')adresses par pays. Cet organisme délègue des blocs d'adresses aux RIR (Regional Internet Registries). Ces RIR sont actuellement au nombre de trois :

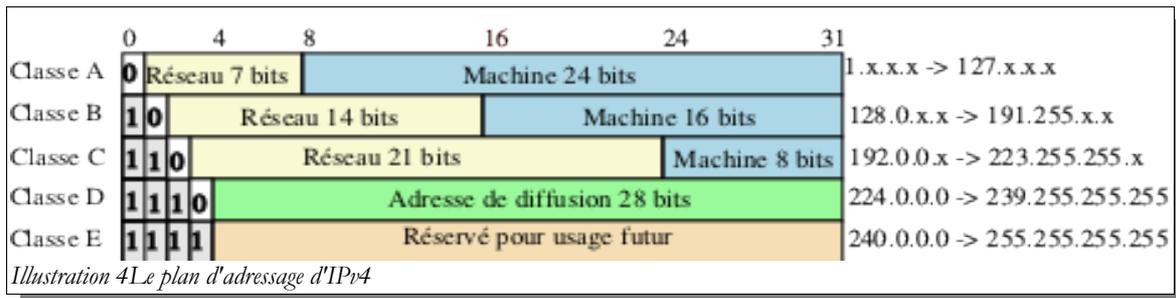
1. l'APNIC (Asia Pacific Network Information Centre)
2. l'ARIN (American Registry for Internet Numbers)
3. le RIPE-NCC: Réseau IP Européen (Network Coordination Centre)

La France dépend donc du RIPE-NCC. Celui-ci délègue à l'AFNIC , qui gère spécifiquement les réseaux français et le domaine **.fr** .

Le protocole IP est un protocole qui a été décrit pour la première fois en 1981. Il a, bien sûr, évolué, et la version actuelle est la version 4. On parle parfois d'IPv4, en opposition à IPv6 (IP version 6) qui est la nouvelle version. La version 6 n'est pas déployée très largement pour l'instant. Certains systèmes d'exploitation ne savent pas encore la traiter.

IPv4 souffre d'un certain nombre de limites. Entre autres, les adresses de source et de destination sont codées sur 32 bits. Ces adresses représentent, de manière *unique* un équipement réseau. De prime abord, cela peut sembler suffisant, puisque $2^{32}=4294967296$, ce qui fait plus de 4 milliards d'adresses disponibles. Malheureusement, les numéros IPv4 ne sont pas attribués séquentiellement. En effet, les ordinateurs sont regroupés en réseau qu'il faut aussi numéroter. Si 16 des 32 bits sont consacrés au réseau, on s'aperçoit que seuls 65536 réseaux peuvent exister sur la planète, ce qui ne serait pas suffisant. Si on consacre 24 bits au réseau, on ne peut plus mettre que 256 ordinateurs par réseau, ce qui est trop faible pour de grosses entreprises mondiales.

Un plan d'adressage a donc été créé pour tenter de résoudre ces problèmes, ce qui a donné 5 classes d'adresses, représentées sur la figure 4,page 6 en fonction du nombre de bits consacré au réseau.



La classe A dont les adresses sont comprises entre 1.0.0.0 et 127.0.0.0, le premier octet désignant l'adresse du réseau.

La classe B entre 128.0.0.0 et 191.255.0.0. Les deux premiers octets désignant le numéro du réseau.

La classe C entre 192.0.0.0 et 223.255.255.0, les trois premiers octets désignant le numéro du réseau.

La classe D est réservée au multicast. Les classes E et F sont des adresses réservées ou expérimentales.

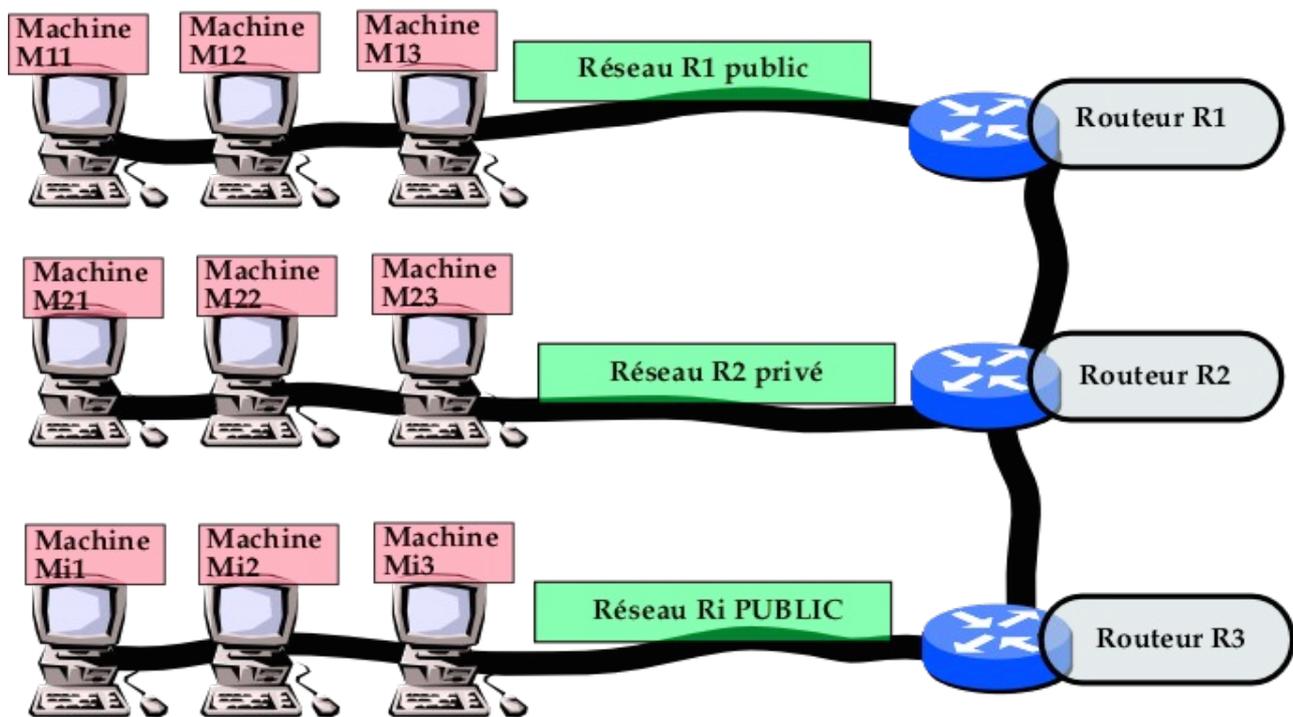
Ce plan d'adressage a atteint ses limites. En effet, la classe B, la mieux adaptée à des établissement disposant de nombreux ordinateurs a été très vite épuisée. D'une manière générale, les adresses IPv4 sont en phase d'épuisement. Il est très difficile de se faire allouer de nouvelles classes d'adresses, même pour des réseaux de classe C.

IPv6 est l'aboutissement du travail du groupe de travail IPng (IP Next generation), et constitue l'évolution logique d'IPv4. Les différents groupes de travail ont fait évoluer IPv4 pour qu'il puisse gérer nettement plus d'adresses, mais aussi pour qu'il soit mieux adapté aux réseaux rapides.

IPv6 dispose d'adresses sur 128 bits. De plus, la notion de classes d'adresses, finalement très limitative et peu souple a disparue et est remplacée par la notion de préfixe : L'adresse d'une machine est séparée en deux parties. La partie gauche représente le réseau, la partie droite identifie la machine sur ce réseau. L'équivalence avec les classes d'adresses est visible. Cependant, l'intérêt est que le nombre de bits réservés au réseau est variable. (Ce principe d'agrégation de réseaux contigus en un seul préfixe a été conçu au départ pour IPv4) (CIDR : *Classless Inter Domain Routing*).

1.5 Illustration

Voyons sur un exemple, comment se passent les échanges entre machines de réseaux locaux différents. Considérons le schéma suivant :



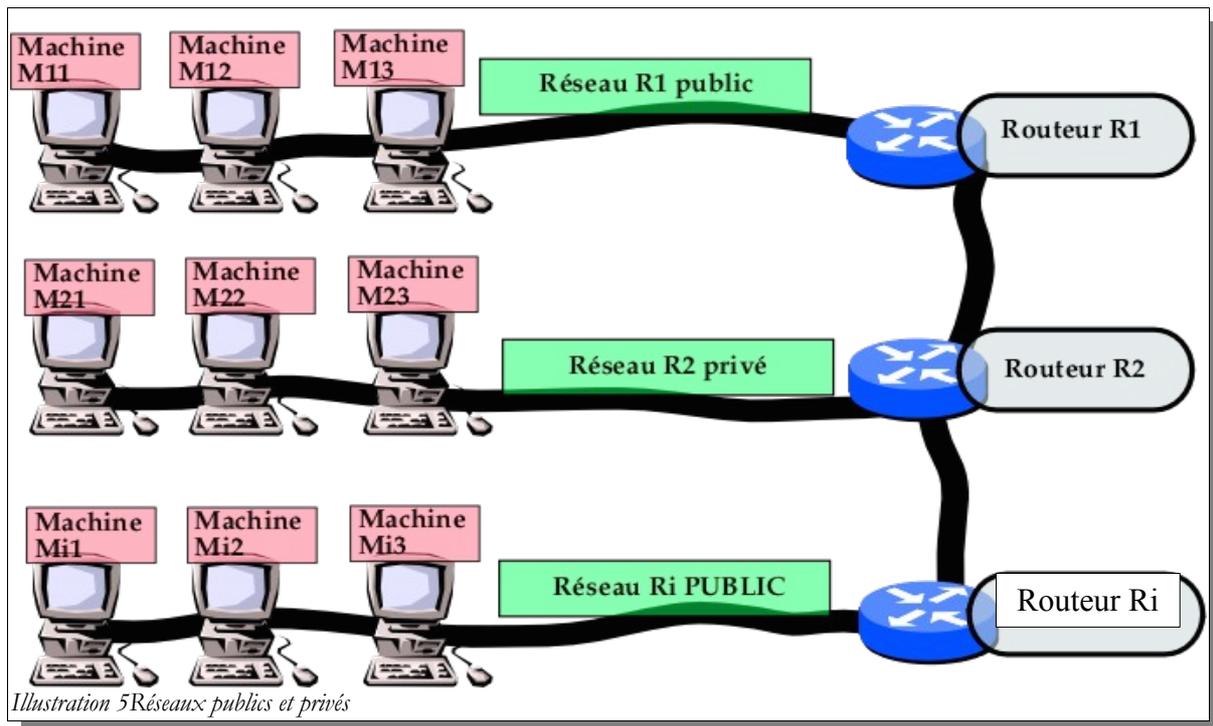
L'envoi par la machine M_{11} du réseau R_1 d'un message à la machine M_{i3} du réseau R_i se fait en plusieurs étapes :

- la machine M_{11} doit connaître l'adresse IP de la machine M_{i3} . Très souvent, elle n'en connaît pas l'adresse mais le nom. Ainsi, si avec votre navigateur, vous demandez l'url <http://www.ibm.com>, vous n'utilisez pas une adresse IP (vous pourriez) mais le nom d'une machine, ici www.ibm.com. Pour envoyer votre demande à la machine www.ibm.com, votre ordinateur devra en demander l'adresse IP car seule cette adresse est utilisée dans les échanges sur Internet. Il existe sur Internet, un service appelé **service de noms** qui fait la correspondance entre un nom de machine et son adresse IP. Ce service est assuré par des machines appelées **serveurs de noms**. Ainsi dans l'exemple précédent, votre ordinateur va demander à un serveur de noms l'adresse IP de la machine www.ibm.com. A quel serveur de noms va-t-il s'adresser ? Cela dépend de la configuration réseau de votre ordinateur. Il y a deux grandes méthodes :
 1. l'adresse IP d'un serveur de noms a été inscrit dans l'un des fichiers de configuration réseau de votre ordinateur.
 2. votre ordinateur est configuré pour demander cette information à chaque fois qu'il redémarre. Il envoie sur le réseau auquel il appartient une demande du genre "Quelqu'un peut-il me donner ma configuration Internet ? ". Un serveur appelé **serveur DHCP** lui donne alors sa configuration : son adresse IP, l'adresse IP du serveur de noms qu'il peut interroger, adresse du routeur qui relie le réseau local sur lequel se trouve votre ordinateur à Internet,...
- une fois obtenue l'adresse IP A_{i3} de la machine M_{i3} , la machine M_{11} envoie ses données au routeur R_1 . Cette machine est la porte de sortie du réseau R_1 vers les autres réseaux. On l'appelle souvent la **passerelle**, **gateway** en anglais, du réseau R_1 vers Internet.
- Le routeur R_1 va transmettre les données à un autre routeur pour que, de proche en proche, celles-ci arrivent au routeur qui dessert le réseau R_i où se trouve la machine M_{i3} . Il existe divers chemins qui mènent du routeur R_1 au routeur R_i . Le choix d'un chemin plutôt qu'un autre est fait par les routeurs selon divers critères (état d'encombrement d'un chemin vis à vis d'un autre, chemin fermé pour cause de routeur en panne,...)
- le routeur R_i va délivrer les données à la machine M_{i3} .

1.6 Réseaux publics, réseaux privés

Parmi toutes les adresses IP possibles de la forme I1.I2.I3.I4, certaines ne sont pas routées. Cela veut dire que si un routeur reçoit des données pour une machine ayant une adresse IP non routable, le routeur déclarera qu'il ne peut pas acheminer ces données à destination. Ceci a donné naissance à deux types de réseaux : les **réseaux publics** et les **réseaux privés**. Les réseaux publics sont constitués de machines ayant des adresses IP accessibles de n'importe quelle autre machine ayant elle aussi une adresse IP accessible, les réseaux privés de machines ayant des adresses IP inaccessibles à partir d'une machine qui n'est pas sur le même réseau privé.

Quelle est l'intérêt d'un réseau *privé* ? Reprenons le schéma déjà étudié :



Le dessin précédent représente trois réseaux :

- deux réseaux **publics** R_1 et R_i
- un réseau **privé**, le réseau R_2

Voyons les caractéristiques d'un poste de ces différents réseaux.

Une machine du réseau public R_1

- peut atteindre et être atteinte par une machine d'un autre réseau public

Une machine du réseau privé R_2

- peut atteindre et être atteinte par toute machine du réseau privé R_2 , notamment par le routeur R_2
- ne peut atteindre ni être atteinte par une machine n'appartenant pas au réseau privé R_2

Un réseau privé est donc un **intranet** coupé de l'Internet. On l'utilise essentiellement pour des raisons de sécurité. Une machine d'un réseau privé ne peut être la cible d'attaques logicielles provenant de machines extérieures au réseau privé puisque celles-ci ne peuvent pas entrer dans le réseau privé. Cependant, il n'est pas intéressant de se couper de l'Internet car on y trouve des ressources dont on a besoin. Si on prend l'exemple du réseau privé R_2 , le routeur R_2 fait l'interface entre le réseau privé et l'Internet. C'est une machine publique et donc, elle peut atteindre toute machine de l'Internet. On l'utilise alors comme machine relais pour permettre à une machine du réseau privé R_2 d'accéder aux ressources de l'Internet. Cela se passe grossièrement de la façon suivante :

- une machine M_{2j} du réseau R_2 veut faire une demande à une machine de l'Internet. Cette demande est alors transmise à la passerelle R_2 .
- la machine R_2 fait alors la demande à la place de la machine M_{2j} . Elle le peut puisqu'elle se trouve sur le réseau public.
- la machine R_2 reçoit une réponse qu'elle transmet alors à la machine M_{2j} qui a fait la demande initiale

On notera divers points :

1. la machine R_2 fait plus qu'un simple routage. Elle fait du relaying de demande. On l'appelle une machine relais ou **proxy**.
2. la connexion vers l'extérieur est initiée par la machine du réseau privé et non par une machine de l'extérieur. La connexion initiée par l'extérieur vers une machine du réseau privé R_2 reste impossible.
3. une fois la connexion établie entre la machine du réseau privé et la machine extérieure via la machine relais, une attaque reste possible de la part de la machine extérieure. Aussi à l'intérieur d'un réseau privé sécurisé, on

cherchera à limiter le type de services que peut demander une machine à l'extérieur, à des services connus pour être inoffensifs.

INSTALLATION DE LINUX RH8

Linux est un système d'exploitation, clone gratuit d'Unix pour des ordinateurs personnels. Linux a été développé au départ par **Linus TORVALDS** à l'Université de HELSINKI en 1991. La base de Linux est un autre système : Minix. Mais rapidement d'autres personnes se sont intéressées à Linux et le système s'est développé. La version 1.0 est sortie au mois de mars 1992. Aujourd'hui Linux est un système véritablement stable et totalement compatible avec les applications du monde Unix. Il s'est tout d'abord dans le monde des serveurs du fait de ses excellentes performances et de sa gratuité. Il commence à devenir une alternative crédible à Windows dans le monde de l'informatique personnelle. On utilise alors linux comme station de travail plutôt que comme serveur.

2 Le travail à réaliser

2.1 Introduction

Vous allez installer des serveurs Linux jouant un rôle analogue à celui des machines Linux décrites plus haut. A l'issue de l'installation, vous aurez acquis les compétences de base d'administration d'une machine Linux et des services Internet les plus utilisés. La machine que vous aurez montée pourrait être mise en production dans une entreprise. A l'université d'Angers, de telles machines ont été installées dans différents sites où à l'origine il n'y avait pas d'informaticiens pour les gérer. Elles étaient allumées et on ne s'en occupait plus sauf pour créer des comptes utilisateurs dessus. Elles ont fonctionné plusieurs années ainsi. A l'ISTIA, la machine SHIVA a été installée en Juin 97 avec tous les services que vous allez installer. On ne l'administre depuis que pour y gérer les comptes utilisateurs (création et suppression) et passer les patches de sécurité. Au fil du temps c'est cette dernière activité qui est devenue la plus fréquente.

Le document a été écrit initialement en 1997 par deux étudiants dans le cadre d'un projet. Ils ont étudié comment avait été installée la machine SHIVA et ont reproduit la même configuration sur une machine de test en notant toutes les étapes afin de produire un document utilisable par le plus grand nombre. Ce document est la base d'un TP pour certains étudiants de l'ISTIA.

Le but recherché est de montrer qu'installer un serveur Linux opérationnel sur Internet n'est pas compliqué. L'apparente simplicité du document, où tout a été décrit précisément, ne doit pas cependant cacher le fait que, si vous êtes amenés à administrer un système Linux, vous aurez à vous former à Unix et à l'administration système. Néanmoins, avec l'aide de ce document, il est plutôt facile de monter rapidement une machine Linux. Ceci permet de monter des machines de test où l'on ne craint pas de détruire le système. Cela donne une grande liberté à l'administrateur. Il peut sans crainte tester les services qu'il souhaite installer ultérieurement sur des serveurs en production.

L'utilisation de ce document n'exige que des connaissances de base d'Unix. Il peut être utilisé même par un néophyte. En revanche, des connaissances réseau générales, notamment sur le rôle des services qui seront installés, seront utiles pour apprécier l'intérêt de ce qui sera fait. On trouve dans les librairies des livres sur Linux avec des CD permettant son installation. C'est à mon avis, une excellente méthode d'approfondir vos connaissances si vous le souhaitez.

2.2 Les différentes versions de Linux

Il existe différentes versions de linux : Linux RedHat, Linux Mandrake, Linux Suse, Linux Debian, Linux Slackware. Une distribution linux, c'est un ensemble de logiciels du domaine public dont l'assemblage forme un système linux. Des distributions linux vont différer :

- par la manière dont se fait l'installation
- par le sous-ensemble de logiciels du domaine public qui a été inclus dans la distribution
- par la manière dont on peut mettre à jour le système en lui ajoutant par exemple de nouvelles applications
- par des compléments propres à la distribution elle-même

Par exemple, l'application MySQL est une base de données relationnelle du domaine public. Une distribution peut décider de l'inclure par défaut, une autre non portant son choix sur la base de données PostGres. On peut donc avoir au départ, un ensemble d'applications différentes.

On peut avoir des procédures d'installation plus ou moins sophistiquées. Celle de la Mandrake est par exemple plus graphique que la Debian et sera considérée pour cela comme plus attrayante par certains.

Tous les linux sont des Unix, donc malgré toutes les différences, il y a un fonds commun important. Néanmoins, les différences de surface existent et ne sont pas négligeables : les différentes distributions peuvent ainsi ne pas mettre leurs fichiers de configuration aux mêmes endroits. Aussi un administrateur système habitué à les trouver à un certain endroit avec une distribution donnée pourra être un certain temps désorienté en passant à une autre distribution.

Le choix de la distribution RedHat a été fait en raison de la grande diffusion de celle-ci. Le déroulement de l'installation de Linux serveur & station de travail sera le suivant :

2.3 Installation de Linux sur un PC

Nous commençons par installer Linux sur un PC. Cette installation suivie par les premières configurations de base (création de groupes, d'utilisateurs) dure environ 2 h. Chaque groupe possède un PC sur lequel est installé le couple Dos/Win9x et qui est relié au réseau de l'ISTIA. Chaque PC possède deux disques. Linux sera installé en général sur le deuxième. L'installation se fait au moyen d'un CD ou d'une disquette.

2.4 Installation des services

Comme il a déjà été dit, le serveur que vous allez installer aura un rôle comparable à celui des machines Linux de l'ISTIA. Vous y installerez les services suivants :

- Service d'envoi/réception du courrier (SMTP)
- Service de relève externe du courrier (POP)
- Service Web
- Service de transfert de fichiers (FTP)
- Service des Quotas qui permet de contrôler l'espace disque des utilisateurs sur le serveur
- Sécurisation du serveur

Un service important, le service de noms (NAMED) ne sera pas installé sur vos serveurs. Ce service assure le lien nom machine <--> adresse IP. Il sera assuré ici par la machine privée **vishnou.istia.uang**. À quoi sert un serveur de noms ? Lorsqu'un étudiant utilise un navigateur sur un PC du réseau étudiant et demande l'URL <http://www.ibm.com>, le navigateur commence par demander l'adresse IP de la machine www.ibm.com car sur Internet, ce sont les adresses IP des machines qui sont utilisées et non leurs noms. À qui cette information est-elle demandée ? à un serveur de noms. C'est la machine **vishnou.istia.uang** qui assure ce service pour les réseaux privés de l'ISTIA. Ce sera également elle qui devra être interrogée par vos serveurs.

Les serveurs Linux que vous allez installer joueront un rôle analogue aux machines linux en exploitation à l'ISTIA. Ils seront sur le **réseau privé étudiant 172.20.81.0**. Ils joueront le rôle de serveurs pour des machines situées elles-aussi sur le réseau étudiant. Vos serveurs Linux seront dépourvus du client Socks qui leur permettrait de sortir sur l'Internet via le serveur Socks de l'université. Ce seront donc des serveurs au sein d'un réseau privé. Cela ne change rien vis à vis d'un serveur public visible sur l'internet. La seule différence est justement cette visibilité qui n'est pas due à une configuration particulière du serveur linux, mais du réseau physique où on l'a placé. Vos serveurs auront néanmoins un accès web via le proxy HTTP eproxy.istia.uang.

2.5 Installation d'applications complémentaires

Une fois installés les principaux services Internet, on peut vouloir installer d'autres applications telles des systèmes de gestion de bases de données par exemple. La durée de l'installation est fonction de la complexité du produit. Une fois celle-ci faite, intervient généralement une phase de personnalisation du produit qui peut être longue car nécessitant de bien comprendre les fonctionnalités de celui-ci. Cette personnalisation se fait souvent tout au long de la vie du produit installé. Ce document vous propose d'installer diverses applications, notamment des gestionnaires de bases de données et des outils de développement.

3 Installation de Linux RedHat 8.0

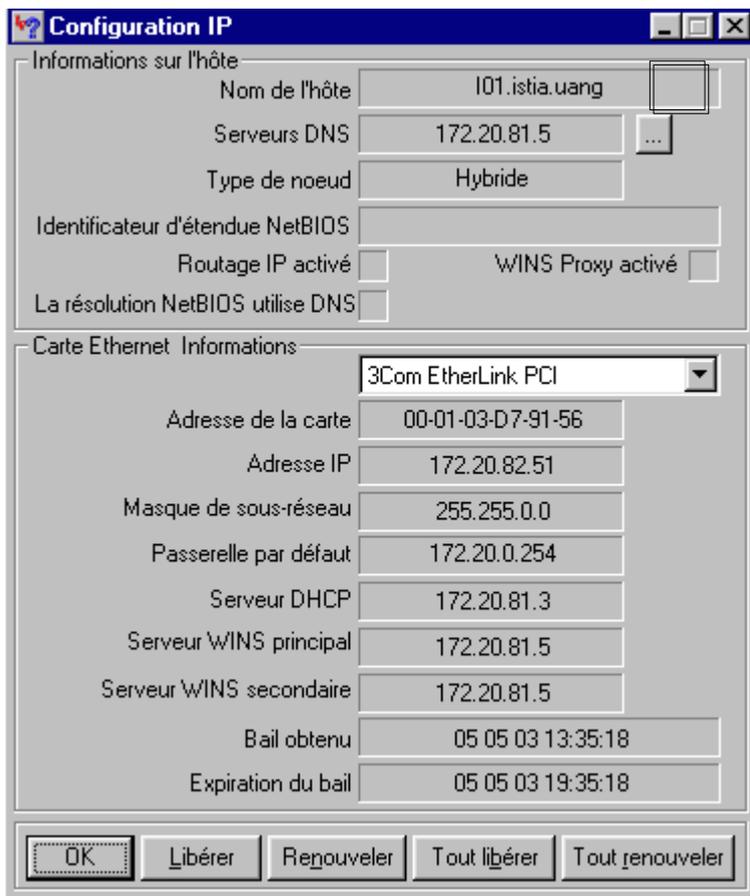
3.1 Configuration du BIOS

Les deux façons les plus courantes d'installer linux est d'utiliser des CD ou le réseau. Dans le cas d'utilisation de CD, il faut pouvoir booter sur CD. Il faut donc que le BIOS de votre machine ait une séquence de boot où le CD vient avant le disque C. Dans le cas d'utilisation du réseau, l'installation commence avec un boot sur disquette. Il faut donc que le BIOS de votre machine ait une séquence de boot où la disquette vient avant le disque C.

3.2 Récupérer l'adresse IP du poste windows

A l'ISTIA, vous allez installer linux sur une machine possédant déjà un système windows (9x ou 2000). La machine a par ailleurs deux disques : le premier est dédié à windows, le second à linux. La machine que vous utilisez a déjà une adresse internet (adresse IP) sur le réseau étudiant de l'ISTIA. Lorsque vous affecterez une adresse IP à votre serveur linux, il faudra utiliser **la même adresse**. Pour l'obtenir, procédez ainsi :

- sous windows, ouvrez une fenêtre DOS et tapez la commande **winipcfg**. L'application **winipcfg** donne un certain nombre de renseignements sur la configuration réseau de votre poste.



- Notez les renseignements suivants :
 - adresse IP du poste
 - nom du poste
 - masque de sous-réseau
 - adresse IP du serveur de noms
 - adresse IP de la passerelle
- sous windows 2000, faites la même chose que ci-dessus et tapez la commande **ipconfig /all** :

```
C:\winnt\system32>ipconfig /all
Configuration IP de Windows 2000

    Nom de l'hôte . . . . . : take
    Suffixe DNS principal . . . . . :
    Type de noud. . . . . : Diffuser
    Routage IP activé . . . . . : Non
    Proxy WINS activé . . . . . : Non

PPP carte Olitec USB ADSL :

    Suffixe DNS spéc. à la connexion. :
    Description . . . . . : WAN (PPP/SLIP) Interface
```

```
Adresse physique. . . . . : 00-53-45-00-78-12
DHCP activé . . . . . : Non
Adresse IP. . . . . : 81.51.154.88
Masque de sous-réseau . . . . . : 255.255.255.255
Passerelle par défaut . . . . . : 81.51.154.88
Serveurs DNS. . . . . : 193.252.19.3
                        193.252.19.4
NetBIOS sur Tcpiip . . . . . : Désactivé
```

Ces renseignements vous seront utiles par la suite. Vérifiez que vous avez noté correctement l'adresse IP de votre poste. Les autres renseignements sont communs à tous les postes d'une même salle.

3.3 installation par FTP ou par CD

Installer linux par FTP consiste à se connecter par ftp à un site possédant la distribution linux que l'on veut installer. Ces sites ont également des images ISO de CD d'installation qu'on peut télécharger pour ensuite graver les CD d'installation. La distribution RH 8.0 a cinq CD d'installation. Parce qu'il n'est pas facile de dupliquer 5 CD pour chaque groupe installant linux lors d'un stage ou un TP, l'installation se fera par FTP. Pour une installation personnelle chez soi, il est plus agréable d'avoir les CD.

Il existe différents sites où on peut trouver les distributions de linux Redhat. Le site principal est <ftp://ftp.redhat.com>. Ce site est souvent surchargé. On pourra utiliser alors l'un des sites miroirs du site. A l'ISTIA, la distribution Redhat à installer a été téléchargée sur un serveur ftp privé. C'est ce serveur que nous utiliserons. Néanmoins, pour rester général nous décrivons les quelques étapes à observer si nous utilisons un serveur ftp public. Le plus simple est de se procurer les images iso des CD d'installation :



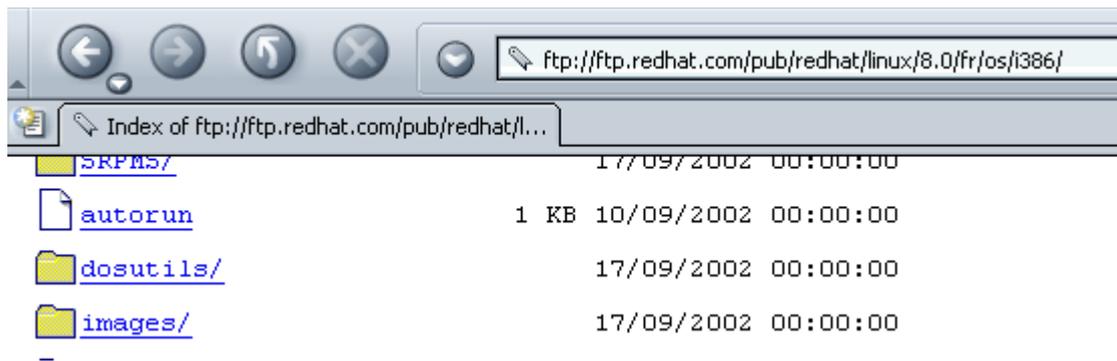
Index of ftp://ftp.redhat.com/pub/redha

[Up to higher level directory](#)

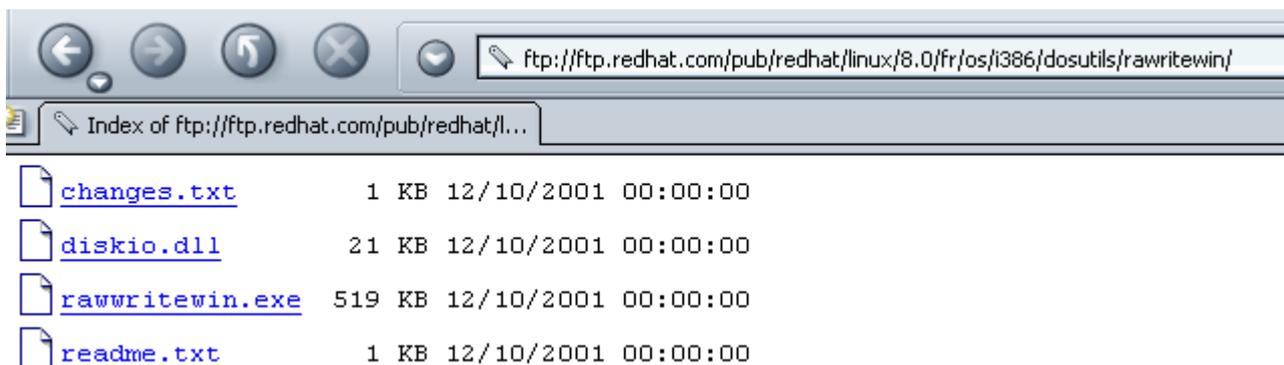
 MD5SUM	1 KB	30/09/2002	00:00:00
 psyche-i386-disc1.iso	659488 KB	10/09/2002	00:00:00
 psyche-i386-disc2.iso	650656 KB	10/09/2002	00:00:00
 psyche-i386-disc3.iso	661728 KB	10/09/2002	00:00:00
 psyche-i386-disc4.iso	611584 KB	10/09/2002	00:00:00
 psyche-i386-disc5.iso	663616 KB	10/09/2002	00:00:00

Dans le répertoire ci-dessus, on trouvera les images des CD d'installation. On peut les télécharger et graver des CD à partir d'elles. Les outils de gravure de CD savent en général utiliser ces images iso. Ensuite on fait une installation avec les CD.

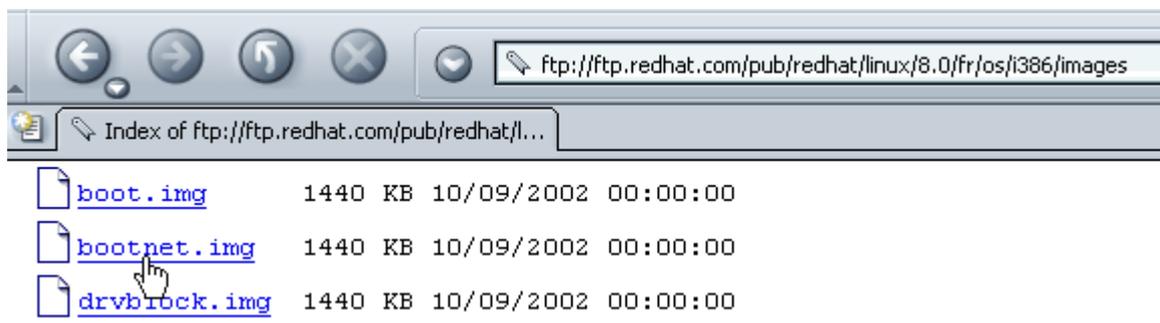
L'installation par ftp nécessite la création d'une disquette spéciale sur laquelle on va booter le poste. Cette disquette possède un mini-système linux qui va connecter votre poste au service ftp que vous allez lui indiquer. Pour créer cette disquette, procédez ainsi :



Le dossier ci-dessus, présente un dossier **dosutils** dans lequel on trouvera un sous-dossier **rawritewin** :

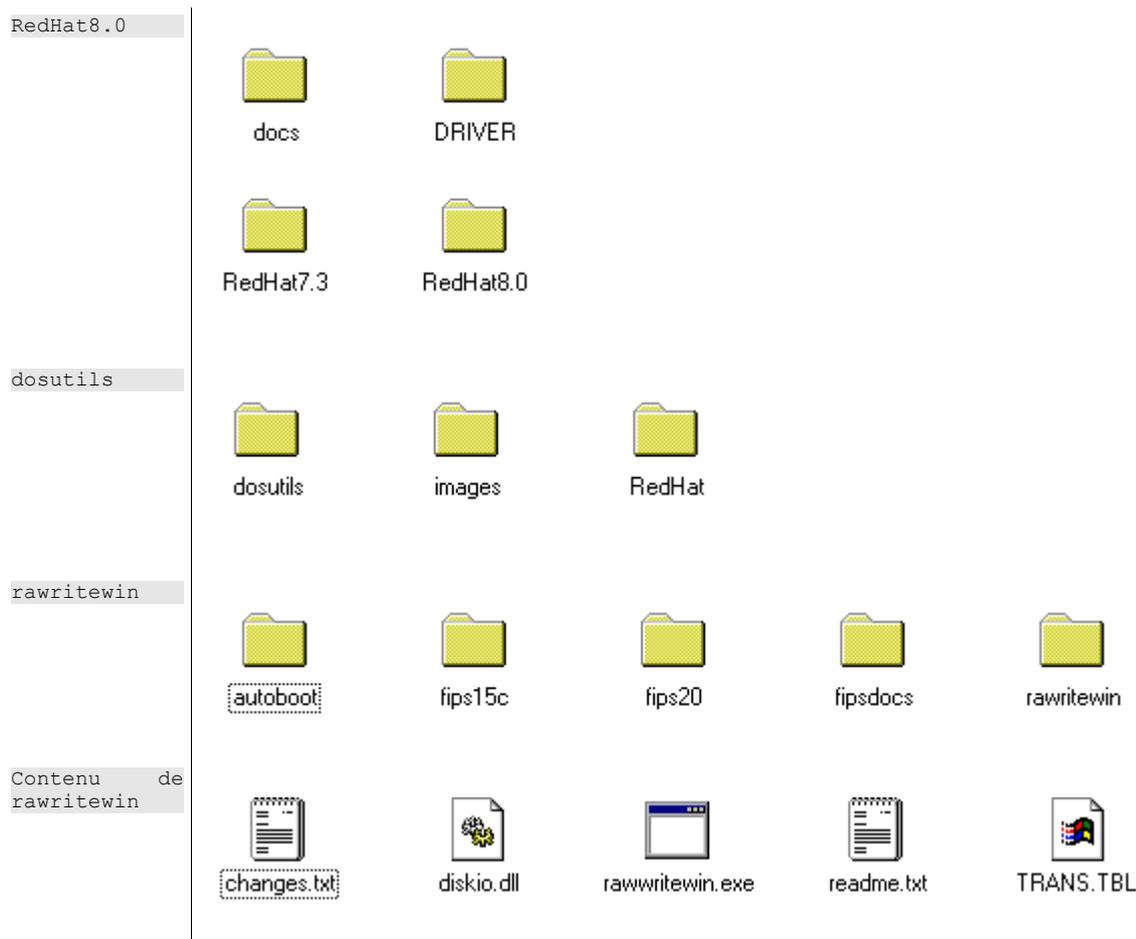


On téléchargera les fichiers **rawritewin.exe** et **diskio.dll** dans un même dossier. Dans le dossier **dosutils/images** on trouve des images de disquettes : téléchargez l'image **bootnet.img**.



Le site officiel de la Redhat est souvent surchargé. La distribution RH8 a donc été téléchargée sur un serveur ftp privé de l'ISTIA. Procédez ainsi :

- prenez un explorateur windows et dans le champ adresse tapez l'URL du serveur ftp contenant la distribution linux RH8. Cette URL vous sera donnée par votre instructeur.
- une fois la connexion au service ftp opérée, allez successivement dans les dossiers suivants :



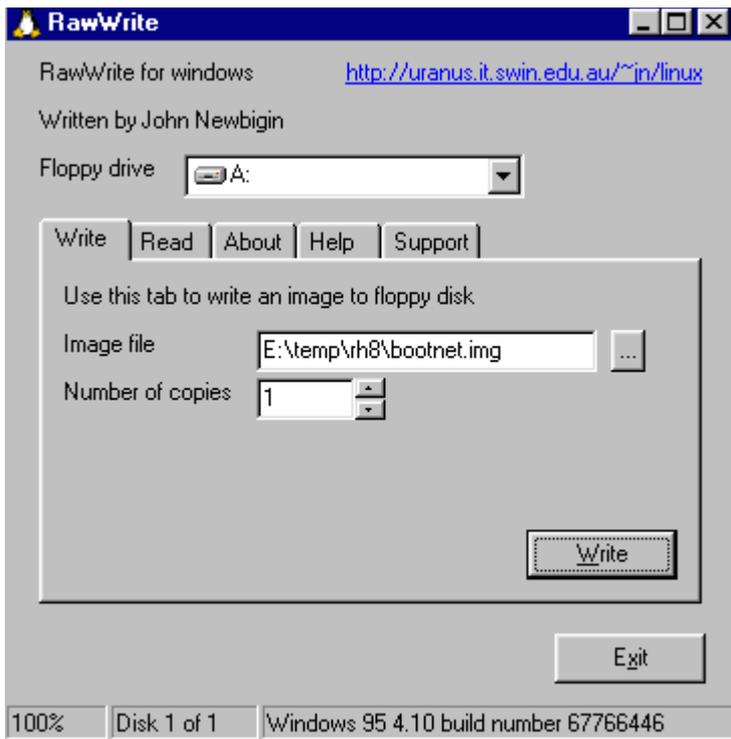
- copiez dans un dossier (e:\temp\rh8 par exemple) les fichiers suivants :
 - du dossier **dosutils/rawritewin** du serveur ftp, copiez (avec la souris) les fichiers **rawritewin.exe** et **diskio.dll**
 - du dossier **Redhat8.0/images** du serveur ftp, copiez le fichier **bootnet.img**

Note : pour copier un fichier, cliquez droit dessus et prenez l'option **[Copier dans dossier]**.

Le contenu du dossier e:\temp\rh8 sera alors le suivant :



Le fichier **bootnet.img** contient l'image de la disquette à créer. L'utilitaire **rawritewin.exe** accompagné de la dll **diskio.dll** permet de copier cette image sur une disquette. Lancez cet utilitaire :



Mettez une disquette dans le lecteur et utilisez le bouton [Write].

Une fois la disquette créée, étiquetez-la '**BOOTNET.IMG**'. Placez cette disquette dans le lecteur et rebootez votre machine dessus. Un écran d'explications apparaît avec la ligne :

boot:

Tapez [entrée]. Un système linux va démarrer et va vous demander les renseignements dont il a besoin pour se connecter au serveur ftp ayant la distribution Redhat. Nous décrivons la suite des écrans qui vont se succéder avec pour chacun d'eux son titre et ce qu'il faut faire. L'installation par FTP n'est pas graphique alors que celle par CD l'est. Il y a alors des écrans qui diffèrent légèrement entre les deux modes d'installation.

titre	saisies & explications
Choose a language	French – OK
Type de clavier	fr-latin1 – OK
Méthode d'installation	FTP – OK
Configuration TCP-IP	<input checked="" type="checkbox"/> Configuration IP dynamique (BOOT/DHCP) – OK // les machines de l'ISTIA peuvent obtenir leur adresse IP en la demandant à un serveur appelé serveur DHCP. C'est cette solution qui est choisie ici
Configuration FTP	Nom du site FTP : donné par l'instructeur Répertoire RedHat : idem <input checked="" type="checkbox"/> Utilisation d'un ftp non anonyme // les informations nécessaires vous seront données par votre instructeur
Further FTP setup	Nom de compte : donné par l'instructeur Mot de passe : idem // parès OK, votre machine va se connecter au servur FTP avec les informations que vous lui avez données. Si la connexion échoue, revoyez ces informations et recommencez. Si la connexion réussit, votre machine va commencer à télécharger les assistants d'installation.
Red Hat Linux	message de bienvenue

Type d'installation

[Personnalisée]

// ce choix nous permet de dire exactement ce que l'on veut

Configuration du partitionnement du disque

[Disk Druid]

// on va faire les partitions du disque nous-mêmes

Avertissement

// on peut avoir ce type d'écran si le programme d'installation ne détecte pas correctement les caractéristiques des partitions déjà en place. On prendra alors l'option **[ignorer]**.// on peut l'avoir également si Disk Druid ne reconnaît pas les partitions d'un disque. Il demande alors l'autorisation de recréer la table des partitions. Dans ce cas, prendre l'option **[oui]**.

Partitionnement

- l'installation se fait sur le 2ième disque /dev/hdb. C'est là que vous créez deux partitions. Il ne faut toucher au disque /dev/hda qui contient un système windows. Un espace disque de 6 à 10 Go est convenable pour une installation Linux. On peut même se contenter de beaucoup moins si on installe peu de services.
- Positionnez le curseur sur l'espace libre du disque hdb. Si vous n'avez pas 6 Go libres, supprimez des partitions existantes sur hdb après avoir obtenu l'accord de votre instructeur.
 - créer la 1ère partition : **[Nouveau]**
 - Point de montage : /
 - Type de systèmes de fichiers : **ext3**
 - Disque : **hdb**
 - Taille de la partition : au moins 6 Go
 - valider

On peut de nouveau avoir un message **[Avertissement]** qu'on ignorera. On crée maintenant la partition de swap : **[Nouveau]**

- Pas de point de montage
- Type de systèmes de fichiers : **swap**
- Disque : **hdb**
- Taille : entre 500 Mo et 1 Go
- valider

Configuration du chargeur de démarrage

[*]Ignorer le chargeur de démarragemessage **[Avertissement]** possible : ignorer

// En situation réelle, nous aurions installé linux sur une machine dédiée et celui-ci aurait été lancé par le chargeur de démarrage GRUB. A cause de considérations techniques propres à l'ISTIA (machine multi-OS avec au démarrage un logiciel de restauration de disques), nous choisissons par prudence de ne pas installer le chargeur qui viendrait modifier des secteurs de boot peut-être déjà utilisés par le logiciel de restauration.

Configuration du réseau

adresse IP : **IP1.IP2.IP3.IP4***Masque réseau* : **255.255.0.0***Passerelle par défaut* : **172.20.0.254***Serveur de noms primaire* : **172.20.81.5****[*] Activation au démarrage**

//

adresse IP : **IP1.IP2.IP3.IP4**

Le chapitre des généralités vous a montré que toute machine de l'internet avait une adresse unique appelée adresse IP. Votre machine fera partie du réseau étudiant de l'ISTIA **172.20.81**. ou **172.20.82**. Ici, vous mettrez l'adresse IP que vous avez notée au début de l'installation (Preliminaires). **Ne vous trompez pas**.

Masque réseau : 255.255.0.0

Supposons que votre machine veuille communiquer avec la machine d'adresse IP I1.I2.I3.I4. Pour savoir si cette adresse fait partie du même réseau qu'elle, votre machine va faire un "Et logique" de cette adresse avec le masque réseau : I1.I2.I3.I4 & **255.255.0.0**. Sans entrer dans les détails, le résultat de cette opération sera I1.I2.0.0. Votre machine va faire de même avec sa propre adresse IP : 172.20.81.xxx & **255.255.0.0** ce qui va donner 172.20.0.0. Si on a I1.I2.0.0=172.20.0.0 donc I1=172 et I2=20, alors la machine destinatrice est sur le même réseau que la machine émettrice. Les données lui seront alors directement envoyées. Si on a I1.I2.0.0<>172.20.0.0 alors la machine destinatrice n'est pas sur le même réseau que la machine émettrice. Les données seront alors envoyées à la passerelle (gateway) qui relie le réseau 172.20 à l'internet.

Passerelle par défaut : 172.20.0.254

L'adresse IP de la passerelle dont on vient de parler.

Serveur de noms primaire : 172.20.81.5

L'adresse IP du serveur de noms que votre machine pourra interroger. Rappelons que le serveur de noms fait la correspondance adresse IP <--> nom de machine dans les deux sens. Celui-ci est sur le réseau étudiant et s'appelle vishnou.istia.uang.

Configuration du nom d'hôte

Ci-dessus, vous avez repris l'adresse IP d'une machine existante et connue du serveur de noms vishnou.istia.uang. Celui-ci va être interrogé par l'installateur qui devrait donc récupérer un nom. Celui-ci sera de la forme X.istia.uang. Si ce n'était pas le cas, il faudra revenir à la page précédente pour vérifier votre configuration réseau.

Configuration du pare-feu

[Pas de pare-feu]

// un pare-feu sert à protéger le serveur. Pour une machine en production, il est nécessaire. Pour une première installation, la présence d'un pare-feu risque de nous amener à des considérations qui dépasseront le débutant. La mise en place du pare-feu peut se faire ultérieurement.

Support supplémentaire
Ajouter utilisateur

langue

[French (France)]

Mot de passe root : **azerty**

Confirmer : **azerty**

//Ici on donne le mot de passe d'un utilisateur particulier appelé *root*. C'est le seul utilisateur qui a tous les droits sur un système Unix. En général seul l'administrateur du système est amené à travailler sous cette identité. Ce sera le cas pour vous pour toutes les installations et configurations d'applications que vous allez faire.

On choisit un mot de passe qu'on va se rappeler. Si on oublie ce mot de passe, il faudra recommencer l'installation.

On vous propose de créer un autre utilisateur que *root*, ce compte ne devant être utilisé que pour des tâches d'administration système. Il est déconseillé de l'utiliser pour une simple utilisation d'unix car alors on peut tout "casser" par mégarde. Voici un exemple que vous adapterez à votre cas :

ID utilisateur : tahe

// pas d'espaces, pas de caractères accentués, en minuscules en général

Nom complet : Serge Tahé

Mot de passe : azerty

Vérification : azerty

Configuration du compte utilisateur

[OK]

// l'installateur affiche ici la liste des comptes utilisateurs créés. On pourrait en créer de nouveaux. Nous ne le faisons pas.

Configuration de l'authentification	<ul style="list-style-type: none"> - accepter les valeurs par défaut <p>// Les utilisateurs sont enregistrés dans le fichier /etc/passwd avec diverses informations les concernant. C'est un fichier texte lisible par tous. Historiquement on trouvait dedans les mots de passe des utilisateurs sous forme cryptée. A partir de ce mot de passe crypté, il est impossible de trouver le mot de passe à partir duquel il a été créé. Pour y arriver, il faut essayer des mots de passe au hasard, les crypter et regarder si le résultat se trouve dans le fichier /etc/passwd. Ce qui au départ pouvait être une opération longue est devenue de plus en plus rapide avec les ordinateurs actuels. On peut donc essayer tous les mots d'un dictionnaire en quelques heures. Si donc un utilisateur a choisi comme mot de passe un mot de ce dictionnaire, celui-ci sera découvert. Il y a là un trou de sécurité.</p> <p>Pour y remédier, on peut placer les mots de passe cryptés non pas dans le fichier /etc/passwd qui est lisible par tous mais dans le fichier /etc/shadow qui n'est lui lisible que par l'utilisateur root. C'est le sens du choix Utiliser les mots de passe masqués. Par ailleurs, historiquement les mots de passe avaient au plus 8 caractères. Le choix Utiliser les mots de passe MD5 permet d'aller au-delà de cette limite.</p>
Sélection du groupe de paquetages	<p>// ici, il nous faut choisir les applications à installer. Prendre les applications suivantes :</p> <p>Administration Tools (oui), Authoring and Publishing (oui), DNS Name Server (non), Development Tools (oui), Editors (oui), Engineering and Scientific (non), FTP Server (oui), GNOME Desktop Environment (oui), GNOME Software Development (non), Games and Entertainment (non), Graphical Internet (oui), Graphics (oui), KDE Desktop Environment (oui), KDE Software Development (non), Kernel Development (oui), Mail Server (oui), News Server (non), Office/Productivity (oui), Printing Support (oui), SQL Database Server (oui), Server Application Tools (oui), Sound and Video (oui), System Tools (oui), Text-Based Internet (oui), Web Server (oui), Windows File Server (oui), X Software Development (non), W Windows System (oui).</p> <p>[*] Sélection individuelle des paquetages</p>
Sélection individuelle des paquetages Installation des paquetages	<ul style="list-style-type: none"> - prendre le paquetage Application/Databases et tout cocher - l'installation des paquetages demandés se fait
Création d'une disquette d'amorçage	<p>[oui]</p> <p>// cette disquette va vous permettre de lancer le système Linux. En situation réelle, il est prudent de dupliquer cette disquette. Toujours en situation réelle, nous aurions installé linux sur une machine dédiée et celui-ci aurait été lancé par l'utilitaire GRUB. A cause de considérations techniques, nous n'avons pas choisi cette option ici.</p>
Configuration de l'interface graphique Configuration de l'écran	<ul style="list-style-type: none"> - doit découvrir le type de carte graphique du PC - doit découvrir le type de l'écran du PC
Personnalisation de X	<ul style="list-style-type: none"> - faites un test graphique avec le bouton [Test], puis faites OK s'il est concluant.
Félicitations	<ul style="list-style-type: none"> - c'est terminé

3.4 installation avec CD

Mettre le CD Linux dans le lecteur et rebootez votre machine. Celle-ci va détecter qu'il y a un CD bootable et démarrer avec. Un premier écran vous propose plusieurs types d'installation. Celle par défaut est une installation graphique. Le programme d'installation va essayer de reconnaître la carte graphique du PC. S'il y arrive, il lancera une installation

graphique permettant notamment d'utiliser la souris. S'il n'y arrive pas, il fera une installation en mode texte. Nous ferons ici une installation en mode graphique. Le programme d'installation a affiché à la fin de son écran d'explications la ligne

boot:

Tapez **[entrée]** pour indiquer que vous voulez une installation en mode graphique. Nous décrivons les premiers écrans de l'installation avec pour chacun d'eux son titre et ce qu'il faut faire.

titre	saisies & explications
CD found	[Skip] // on évite la vérification du CD – le programme d'installation va détecter la carte graphique, l'écran ce qui va permettre l'installation graphique
Welcome	[Next] // lire les explications
Language Selection	[French]
Clavier	[French (latin1)]
Configuration souris	[3 Button Mouse (PS/2)]

L'installation se fait avec une suite d'écrans où l'utilisateur va donner un certain nombre de renseignements au programme d'installation. Nous décrivons ici la suite d'écrans d'installation de Linux RedHat 8.0. Pour d'autres versions de RedHat, les écrans peuvent être différents. Nous donnons pour chaque écran :

- le nom de l'écran présent dans sa barre de titre
- les renseignements à donner
- des explications si nécessaires

titre	saisies & explications
Type d'installation	[Personnalisée] // ce choix nous permet de dire exactement ce que l'on veut
Configuration du partitionnement du disque	[Disk Druid] // on va faire les partitions du disque nous-mêmes
Avertissement	[ignorer] // on peut avoir ce type d'écran si le programme d'installation ne détecte pas correctement les caractéristiques des partitions déjà en place
Partitionnement	<ul style="list-style-type: none"> – l'installation se fait sur le 2ième disque /dev/hdb – trouver un espace disque libre d'au moins 6 G0 pour la partition principale de Linux – trouver un espace de 500 mo pour la partition de swap – au besoin supprimer d'anciennes partitions linux – créer la 1ère partition : [Ajouter partition] <ul style="list-style-type: none"> – Point de montage : / – Type de systèmes de fichiers : ext3 – Disque : hdb – Début du cylindre : ne pas changer la valeur affichée – Fin du cylindre : dimensionner jusque le champ [Taille] ait atteint la valeur désirée (6 go) – valider <p>On peut de nouveau avoir un message [Avertissement] qu'on ignore. On crée maintenant la partition de swap : [Ajouter partition]</p> <ul style="list-style-type: none"> – Type de systèmes de fichiers : swap

- Disque : **hdb**
- Début du cylindre : ne pas changer la valeur affichée
- Fin du cylindre : dimensionner jusque le champ [Taille] ait atteint la valeur désirée (500 mo)
- valider

Configuration du chargeur de démarrage

[Modifier le chargeur de démarrage]
[Ne pas installer le chargeur de démarrage]
 message Avertissement possible : ignorer

// En situation réelle, nous aurions installé linux sur une machine dédiée et celui-ci aurait été lancé par le chargeur de démarrage GRUB. A cause de considérations techniques propres à l'ISTIA (machine multi-OS avec au démarrage un logiciel de restauration de disques), nous choisissons par prudence de ne pas installer le chargeur qui viendrait modifier des secteurs de boot peut-être déjà utilisés par le logiciel de restauration.

Configuration du réseau

adresse IP : **IP1.IP2.IP3.IP4**
Masque réseau : **255.255.0.0**
Passerelle par défaut : **172.20.0.254**
Serveur de noms primaire : **172.20.81.5**

//
adresse IP : **IP1.IP2.IP3.IP4**

Le chapitre des généralités vous a montré que toute machine de l'internet avait une adresse unique appelée adresse IP. Votre machine fera partie du réseau étudiant de l'ISTIA **172.20.81**. Ici, vous mettez l'adresse IP que vous avez notée au début de l'installation (Preliminaires). **Ne vous trompez pas.**

Masque réseau : **255.255.0.0**

Supposons que votre machine veuille communiquer avec la machine d'adresse IP I1.I2.I3.I4. Pour savoir si cette adresse fait partie du même réseau qu'elle, votre machine va faire un "Et logique" de cette adresse avec le masque réseau : I1.I2.I3.I4 & **255.255.0.0**. Sans entrer dans les détails, le résultat de cette opération sera I1.I2.0.0. Votre machine va faire de même avec sa propre adresse IP : 172.20.81.xxx & **255.255.0.0** ce qui va donner 172.20.0.0. Si on a I1.I2.0.0=172.20.0.0 donc I1=172 et I2=20, alors la machine destinatrice est sur le même réseau que la machine émettrice. Les données lui seront alors directement envoyées. Si on a I1.I2.0.0<>172.20.0.0 alors la machine destinatrice n'est pas sur le même réseau que la machine émettrice. Les données seront alors envoyées à la passerelle (gateway) qui relie le réseau 172.20 à l'internet.

Passerelle par défaut : **172.20.0.254**

L'adresse IP de la passerelle dont on vient de parler.

Serveur de noms primaire : **172.20.81.5**

L'adresse IP du serveur de noms que votre machine pourra interroger. Rappelons que le serveur de noms fait la correspondance adresse IP <--> nom de machine dans les deux sens. Celui-ci est sur le réseau étudiant et s'appelle vishnou.istia.uang.

Configuration du pare-feu

[Pas de pare-feu]

// un pare-feu sert à protéger le serveur. Pour une machine en production, il est nécessaire. Pour une première installation, la présence d'un pare-feu risque de nous amener à des considérations qui dépasseront le débutant. La mise en place du pare-feu peut se faire ultérieurement.

Support supplémentaire langue
 Configuration du compte

[French (France)]

Mot de passe root : **azerty**

Confirmer : **azerty**

//Ici on donne le mot de passe d'un utilisateur particulier appelé *root*. C'est le seul utilisateur qui a tous les droits sur un système Unix. En général seul l'administrateur du système est amené à travailler sous cette identité. Ce sera le cas pour vous pour toutes les installations et configurations d'applications que vous allez faire.

On choisit un mot de passe qu'on va se rappeler. Si on oublie ce mot de passe, il faudra recommencer l'installation.

On vous propose de créer un autre utilisateur que *root*, ce compte ne devant être utilisé que pour des tâches d'administration système. Il est déconseillé de l'utiliser pour une simple utilisation d'unix car alors on peut tout "casser" par mégarde. Voici un exemple que vous adapterez à votre cas :

ID utilisateur : **tahe**

// pas d'espaces, pas de caractères accentués, en minuscules en général

Nom complet : **Serge Tahé**

Mot de passe : **azerty**

Vérification : **azerty**

Configuration de l'authentification

— accepter les valeurs par défaut

// Les utilisateurs sont enregistrés dans le fichier `/etc/passwd` avec diverses informations les concernant. C'est un fichier texte lisible par tous. Historiquement on trouvait dedans les mots de passe des utilisateurs sous forme cryptée. A partir de ce mot de passe crypté, il est impossible de trouver le mot de passe à partir duquel il a été créé. Pour y arriver, il faut essayer des mots de passe au hasard, les crypter et regarder si le résultat se trouve dans le fichier `/etc/passwd`. Ce qui au départ pouvait être une opération longue est devenue de plus en plus rapide avec les ordinateurs actuels. On peut donc essayer tous les mots d'un dictionnaire en quelques heures. Si donc un utilisateur a choisi comme mot de passe un mot de ce dictionnaire, celui-ci sera découvert. Il y a là un trou de sécurité.

Pour y remédier, on peut placer les mots de passe cryptés non pas dans le fichier `/etc/passwd` qui est lisible par tous mais dans le fichier `/etc/shadow` qui n'est lui lisible que par l'utilisateur *root*. C'est le sens du choix **Utiliser les mots de passe masqués**. Par ailleurs, historiquement les mots de passe avaient au plus 8 caractères. Le choix **Utiliser les mots de passe MD5** permet d'aller au-delà de cette limite.

Sélection du groupe de paquetages

// ici, il nous faut choisir les applications à installer. Prendre les applications suivantes :

Système X-Window (oui), Environnement de bureau GNOME (oui), Environnement de bureau KDE (oui), Editeurs (non), Engineering et scientifique (non), Internet graphique (oui), Internet basé sur texte (non), Bureau/Productivité (oui), Son et vidéo (non), Authoring et Publishing (oui), Graphismes (oui), Jeux et divertissements (non), Outils de configuration du serveur (oui), Serveur web (oui), Serveur de courrier (oui), Serveur de fichiers windows (oui), Seveur de noms DNS (oui), Serveur FTP (oui), Serveur de bases de données SQL (oui), Serveur de news (non), Serveur de réseau (oui), Outils de développement (oui), Développement du noyau (oui), Développement du logiciel X (non), Développement du logiciel GNOME (non), Développement du logiciel KDE (non), Outils d'administration (oui), Outils de système (oui), Support d'impression (oui).

Installation des paquetages

— l'installation des paquetages demandés se fait

Création d'une disquette d'amorçage

[oui]

// cette disquette va vous permettre de lancer le système Linux. En situation réelle, il est prudent de dupliquer cette disquette. Toujours en situation réelle, nous aurions installé linux sur une machine dédiée et celui-ci aurait été lancé par l'utilitaire GRUB. A cause de considérations techniques, nous n'avons pas choisi cette option ici.

Configuration de l'interface graphique	- doit découvrir le type de carte graphique du PC
Configuration de l'écran	- doit découvrir le type de l'écran du PC
Configuration graphique personnalisée	
Félicitations	- c'est terminé

3.5 Fin de l'installation

L'installation est terminée. La machine va redémarrer. Mettez la disquette d'amorçage et rebootez. Au boot de la disquette, vous devriez voir une invite :

boot :

Validez. Le poste boote sur Linux. Un certain nombre de services sont installés. Au bout de quelques minutes arrivent des fenêtres de post-configuration. Acceptez les propositions par défaut de chacune d'elles sauf pour l'option **[RedHat Update Agent]** où il faut cocher la phrase

[x] Non, je ne veux pas enregistrer mon système avec RedHat Network.

L'enregistrement auprès de RedHat permet d'avoir des mises à jour des paquetages qui ont été installés. C'est important car Redhat diffuse régulièrement des patchs qui corrigent des trous de sécurité. Il y a différents types d'abonnements à ces mises à jour. L'abonnement le plus simple permet d'avoir les mises à jour gratuitement mais de façon non prioritaire. Dans ce cas, si au moment où on veut faire des mises à jour, le serveur est chargé, on recevra le message qu'il nous faut attendre. Moyennant un abonnement peu onéreux, il est possible d'avoir un accès garanti au serveur de mises à jour.

Ici, on refuse l'enregistrement car nous sommes sur un réseau privé et l'assistant de mises à jour ne le sait pas. Il va donc faire un accès direct à Internet et dans la pratique il se « plante » au point qu'on est obligé de rebooter la machine.

En-dehors de cette page particulière, nous acceptons toutes les propositions faites par défaut avec le bouton **[Suivant]**. Lorsqu'arrive la fenêtre d'authentification nous définissons l'environnement de travail souhaité :

- cliquez sur le lien **Session**
- choisissez KDE

Pour vous authentifier, tapez le login **root** puis le mot de passe **azerty**. Il vous sera demandé si vous souhaitez que KDE soit votre environnement par défaut. Répondez oui. Il vous sera toujours possible ultérieurement de faire de GNOME votre environnement par défaut.

LINUX STATION DE TRAVAIL

4 Découvrir l'environnement de travail Linux

4.1 L'environnement de travail KDE

Si tout s'est bien passé, vous êtes maintenant dans l'environnement de travail KDE, un environnement analogue à celui de windows. Nous ne présentons ci-dessous que les quelques outils dont nous aurons besoin par la suite. Il est déconseillé de passer trop de temps à découvrir l'interface KDE. Vous aurez tout le loisir de le faire en-dehors du stage. Nous utiliserons principalement les outils suivants :

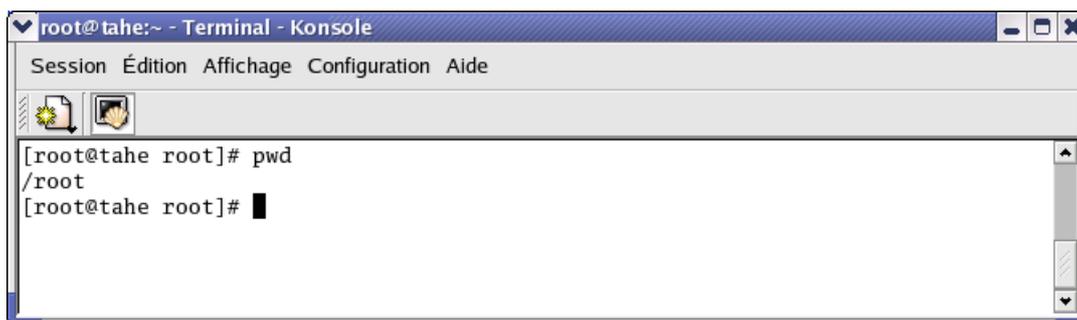
- l'explorateur de fichiers **Konqueror**
- le terminal **Konsole**
- l'éditeur de texte **Kwrite**
- le navigateur **Mozilla**

Nous travaillerons largement à la mode « windows » avec des outils proches ou parfois identiques à ceux qu'on trouve dans le monde windows. Cela permettra au débutant de ne pas être dérouté et d'utiliser ses habitudes « windows ». Dans la réalité, les administrateurs système travaillent peu avec les interfaces graphiques. Ils travaillent en mode ligne en tapant des commandes Unix. C'est le cas lorsque l'administrateur administre son serveur à distance avec une simple fenêtre terminal, ce qui est un cas très fréquent. On ne perdra donc pas de vue qu'un bon administrateur système maîtrise parfaitement les commandes Unix.

Ici, nous alternerons le travail en mode texte et celui en mode graphique. Cela facilitera le travail du débutant et fera découvrir de façon naturelle un aspect non formalisé dans ce document : **linux peut servir de station de travail personnelle** plutôt que de serveur. C'est une tendance actuelle. On voit apparaître, notamment dans les universités et administrations, des salles de PC linux où les étudiants disposent des outils fondamentaux, notamment bureautiques avec OpenOffice, qui leur permettent de travailler. L'intérêt est qu'en termes de logiciels cela ne coûte rien puisque tout est du domaine public.

4.1.1 Le terminal Konsole

Le terminal **Konsole** est obtenu par l'option de menu **Outils système/Terminal**. Le terminal est un outil interactif permettant d'exécuter des commandes Unix :



Il nous servira à émettre des commandes Unix n'ayant pas d'équivalent graphique.

4.1.1.1 Commandes de base d'Unix

Les tests suivants visent à donner les notions de base des commandes Unix. Ceux qui connaissent celles-ci peuvent passer au paragraphe suivant.

Action	Commande Unix	Equivalent Dos
Création d'un répertoire	mkdir répertoire	mkdir répertoire md répertoire
Suppression d'un répertoire vide	rmdir répertoire	rmdir répertoire rd répertoire

Suppression d'un répertoire non vide	rm -r répertoire rm -rf répertoire : pour le supprimer sans confirmation	deltree repertoire
Contenu d'un répertoire	ls -la répertoire	dir repertoire
Se positionner sur un répertoire Copier un fichier	cd répertoire cp fichier1 fichier2 cp fichier1 répertoire	cd repertoire copy fichier1 fichier2 copy fichier1 repertoire
Supprimer un fichier	rm fichier rm -f fichier : pour le supprimer sans confirmation	del fichier
Visualiser le contenu d'un fichier texte	cat fichier more fichier less fichier	type fichier
Visualiser la fin d'un fichier texte	tail -n fichier : pour visualiser les n dernières lignes tail -n fichier more : pour les visualiser page par page	
Chercher une chaîne dans un fichier texte	grep chaîne fichier grep -i chaîne fichier : pour ne pas faire de différence majuscules/minuscules	
Créer un lien de fic2 vers fic1	ln -s fic1 fic2	
Liste des processus actifs	ps aux ps aux grep texte : pour chercher dans les lignes produites par ps, celles qui contiennent texte	
Changer les droits d'un fichier	chmod mode fichier	
Changer le propriétaire d'un fichier	chown utilisateur:groupe fichier	

Vous serez amenés à travailler sous diverses identités. Il est donc important de savoir à un moment donné sous quelle identité vous travaillez. Dans une fenêtre terminal, vous pouvez avoir ce renseignement avec la commande **id** :

```
[root@tahe root]# id
uid=0(root) gid=0(root) groupes=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

Chaque utilisateur a un numéro. Le superutilisateur **root** a le numéro 0. Cet utilisateur particulier a tous les droits. C'est sous ce compte que travaille habituellement l'administrateur du serveur. Chaque utilisateur a un groupe principal défini par un nom et un numéro. Ici c'est le groupe **root** de numéro 0. Un utilisateur peut appartenir à des groupes secondaires, ici les groupes bin, daemon, sys, adm, disk et wheel. Dans un système multi-utilisateurs, les objets (fichiers, dossiers, processus, ...) ont des propriétaires. Ceux-ci peuvent donner des droits d'accès à deux types d'utilisateurs :

- ceux appartenant à un groupe donné
- tous les autres

Rattacher des utilisateurs à plusieurs groupes permet de leur donner les droits d'accès de ces groupes. C'est pourquoi il est parfois nécessaire de rattacher un utilisateur à plusieurs groupes afin de lui donner divers droits d'accès aux objets du système.

Chaque utilisateur a un répertoire de connexion qui lui appartient. Pour connaître celui-ci, tapez les commandes suivantes :

```
[root@tahe root]# cd
[root@tahe root]# pwd
/root
```

cd	Positionne sur le répertoire de connexion
pwd	affiche le répertoire sur lequel on est positionné

Lorsque vous aurez à créer des dossiers ou des fichiers, c'est dans le répertoire de connexion de l'utilisateur sous le nom duquel vous opérez que vous les créez. Aussi aurez-vous toujours besoin de connaître votre identité (commande *id*). En tant qu'utilisateur non *root*, vous ne pourrez créer de dossiers ou de fichiers en-dehors de votre répertoire de connexion. En tant qu'utilisateur *root*, vous pouvez tout faire et vous aurez parfois à changer les droits d'accès de certains objets du système essentiellement des dossiers et répertoires. Découvrons ces droits en listant le contenu du répertoire de connexion de *root* avec la commande **ls** :

```
[root@tahe root]# ls -l
total 104
-rw-r--r--  1 root  root    2390 mar  4 19:23 anaconda-ks.cfg
drwxr-xr-x  2 root  root    4096 mar 20 16:32 autosave
drwx----- 3 root  root    4096 mar 26 07:38 Desktop
drwxr-xr-x  5 root  root    4096 mar 21 16:09 downloads
-rw-r--r--  1 root  root   21896 mar  4 19:18 install.log
-rw-r--r--  1 root  root    4096 mar  4 19:18 install.log.syslog
```

Ce que vous obtenez peut être différent de ce que vous voyez ci-dessus. On voit une liste de fichiers et dossiers. La signification des colonnes est la suivante :

1	droits d'accès au fichier ou dossier
2	nombre de liens sur ce fichier
3	utilisateur propriétaire du fichier ou dossier
4	groupe propriétaire du fichier ou dossier
5	taille en octets du fichier ou dossier
6	date et heure de dernière modification du fichier ou dossier
7	nom du fichier

La colonne 1 représente les droits d'accès au fichier ou dossier. Ceux-ci sont représentés par une suite de 10 caractères :

- le 1er caractère indique le type de l'objet : **d**=directory(dossier), **-**=fichier
- les 3 caractères suivants représentent les droits d'accès de l'utilisateur propriétaire du fichier ou dossier avec les conventions suivantes :
 - ils sont dans l'ordre *rwX*. Pour un fichier : *r*=droit de lire le fichier, *w*=droit d'écrire dans le fichier, *x*=droit d'exécuter le fichier. Pour un dossier, *r*=droit de lister le contenu du dossier, *w*=droit d'ajouter des entrées dans le dossier, *x*=droit de traverser le dossier.
 - le signe *-* signifie l'absence du droit correspondant
- les 3 caractères suivants représentent les droits d'accès du groupe propriétaire du fichier ou dossier avec les mêmes conventions que précédemment.
- les 3 derniers caractères représentent les droits sur l'objet de ceux qui ne sont pas propriétaire de celui-ci et qui n'appartiennent pas non plus au groupe qui en est propriétaire.

Ainsi la ligne

```
-rw-r--r--  1 root  root    4096 mar  4 19:18 install.log.syslog
```

indique qu'on a affaire à un fichier (1er caractère) appartenant à l'utilisateur *root* et au groupe *root*. Les droits *rw-* du propriétaire indiquent que celui-ci peut lire(*r*) et écrire(*w*) dans le fichier. Les droits *r--* du groupe indiquent que les membres de celui-ci ne peuvent que lire le fichier. Les droits *r--* (3 derniers caractères) indiquent que les utilisateurs non concernés par les deux premiers groupes de droits ont le droit de lire le fichier.

Maintenant, affichons les droits du répertoire de connexion de l'utilisateur *root* :

```
root@tahe root]# ls -ld /root
drwxr-x--- 21 root root 4096 mar 26 07:37 /root
```

L'option **-d** utilisée pour la commande **ls** permet d'afficher les propriétés d'un dossier. Sans cette option, ce sont les propriétés des objets contenus dans ce dossier qui auraient été affichés. Ici,

- le 1er caractère **d** indique qu'on a affaire à un répertoire
- il appartient à l'utilisateur *root* et au groupe *root* (colonnes 3 et 4)
- le propriétaire a les droits **rwx** indiquant qu'il peut lister le contenu du dossier (**r**), lui ajouter des entrées (**w**), le traverser (**x**).
- le groupe propriétaire a lui les droits **r-w**. Les membres de ce groupe ne pourront donc pas ajouter d'entrées au dossier
- les autres utilisateurs n'ont aucun droit.

On trouvera ci-dessous quelques exemples que le lecteur est amené à reproduire s'il n'est pas à l'aise avec les commandes Unix.

```
// créer un répertoire
[root@tahe root]# mkdir test

// vérification
[root@tahe root]# ls -l
total 108
-rw-r--r-- 1 root root 2390 mar 4 19:23 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 mar 20 16:32 autosave
-rw-r--r-- 1 root root 56437 mar 22 12:21 capture1.png
drwx----- 3 root root 4096 mar 26 07:38 Desktop
drwxr-xr-x 5 root root 4096 mar 21 16:09 downloads
-rw-r--r-- 1 root root 21896 mar 4 19:18 install.log
-rw-r--r-- 1 root root 4096 mar 4 19:18 install.log.syslog
drwxr-xr-x 2 root root 4096 mar 26 08:48 test

// changer de répertoire courant
[root@tahe root]# cd test

// visualiser le contenu du répertoire courant
[root@tahe test]# ls -l
total 0

// créer un fichier
[root@tahe test]# echo test >fic1

// visualiser le contenu du répertoire courant
[root@tahe test]# ls -l
total 4
-rw-r--r-- 1 root root 5 mar 26 08:50 fic1

// visualiser le contenu du fichier fic1
[root@tahe test]# cat fic1
test

// dupliquer le fichier fic1
[root@tahe test]# cp fic1 fic2

// visualiser le contenu du répertoire courant
[root@tahe test]# ls -l
total 8
-rw-r--r-- 1 root root 5 mar 26 08:50 fic1
-rw-r--r-- 1 root root 5 mar 26 08:50 fic2

// visualiser le contenu du fichier fic2
[root@tahe test]# cat fic2
test

// créer deux dossiers
[root@tahe test]# mkdir rep1 rep2

// contenu du dossier courant
[root@tahe test]# ls -l
total 16
-rw-r--r-- 1 root root 5 mar 26 08:50 fic1
-rw-r--r-- 1 root root 5 mar 26 08:50 fic2
drwxr-xr-x 2 root root 4096 mar 26 08:51 rep1
drwxr-xr-x 2 root root 4096 mar 26 08:51 rep2

// création d'un fichier
[root@tahe test]# echo test2 >rep1/fic3
```

```

// vérification
[root@tahe test]# ls -l repl
total 4
-rw-r--r--  1 root    root          6 mar 26 08:52 fic3

// contenu de fic1
[root@tahe test]# cat repl/fic3
test2

// suppression rep2 qui est vide
[root@tahe test]# rmdir rep2

// vérification
[root@tahe test]# ls -l
total 12
-rw-r--r--  1 root    root          5 mar 26 08:50 fic1
-rw-r--r--  1 root    root          5 mar 26 08:50 fic2
drwxr-xr-x  2 root    root       4096 mar 26 08:52 repl

// suppression repl qui est non vide
[root@tahe test]# rm -rf repl/

// vérification
[root@tahe test]# ls -l
total 8
-rw-r--r--  1 root    root          5 mar 26 08:50 fic1
-rw-r--r--  1 root    root          5 mar 26 08:50 fic2

// suppression fichiers fic1, fic2
[root@tahe test]# rm fic1 fic2
rm: remove regular file `fic1'? y
rm: remove regular file `fic2'? y

// vérification
[root@tahe test]# ls -l
total 0

// création fichier
[root@tahe test]# echo test >test

// vérification
[root@tahe test]# ls -l
total 4
-rw-r--r--  1 root    root          5 mar 26 08:55 test

// droit x pour l'utilisateur propriétaire
[root@tahe test]# chmod u+x test

// vérification
[root@tahe test]# ls -l
total 4
-rwxr--r--  1 root    root          5 mar 26 08:55 test

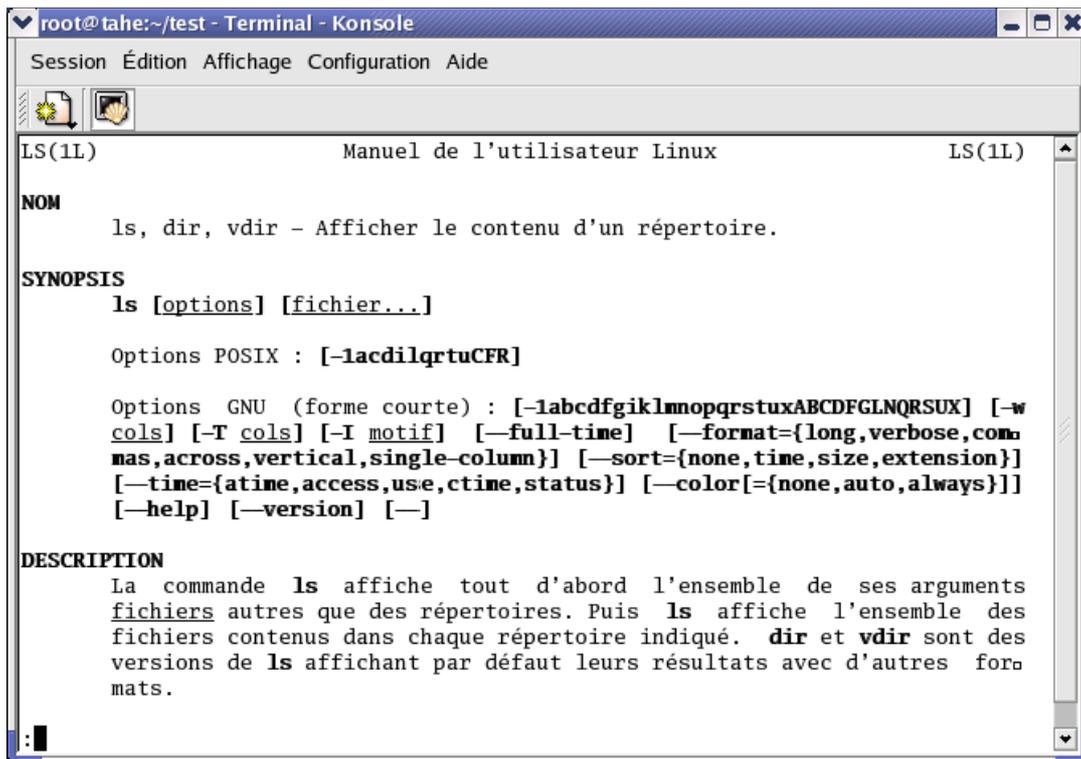
// on enlève le droit x au propriétaire
[root@tahe test]# chmod u-x test

// vérification
[root@tahe test]# ls -l
total 4
-rw-r--r--  1 root    root          5 mar 26 08:55 test

```

4.1.1.2 Aide sur les commandes Unix

Les commandes Unix ont en général de multiples options introduites par le caractère -, par exemple ls -ld. Pour connaître la syntaxe exacte d'une commande Unix, on utilisera la commande man. Voici ce que donne par exemple la commande **man ls** :



Pour se déplacer dans l'aide, on utilisera les touches suivantes :

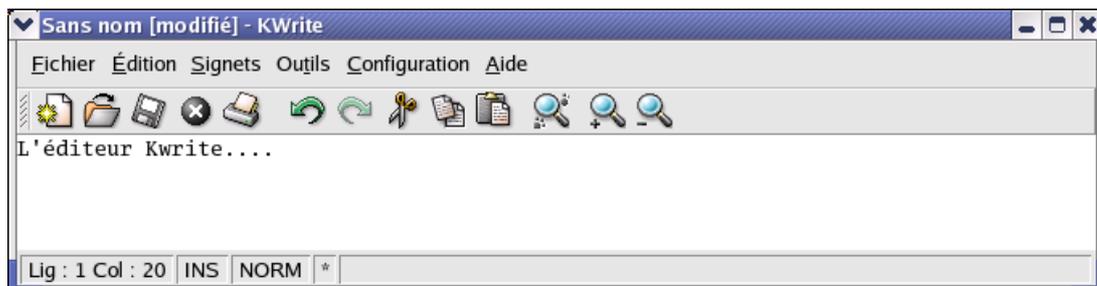
PageUp	: page précédente
PageDown	: page suivante
Flèche haute	: remonte d'une ligne
Flèche basse	: descend d'une ligne
q	: termine l'application man

4.1.2 L'éditeur de texte Kwrite

L'éditeur **Kwrite** est obtenu par l'option de menu **Extras/Accessoires/Kwrite**. En mode ligne, il peut être obtenu par la commande **kwrite** :

```
# kwrite &
```

qui lancera l'éditeur **kwrite** en tâche de fond. Cet éditeur nous permettra de modifier des fichiers texte existants, notamment des fichiers de configuration et/ou parfois en créer de nouveaux.



4.1.2.1 Tests

Ecrivez un texte et sauvegardez-le dans le dossier `/root`. Puis rechargez-le.

4.1.3 Le gestionnaire de fichiers Konqueror

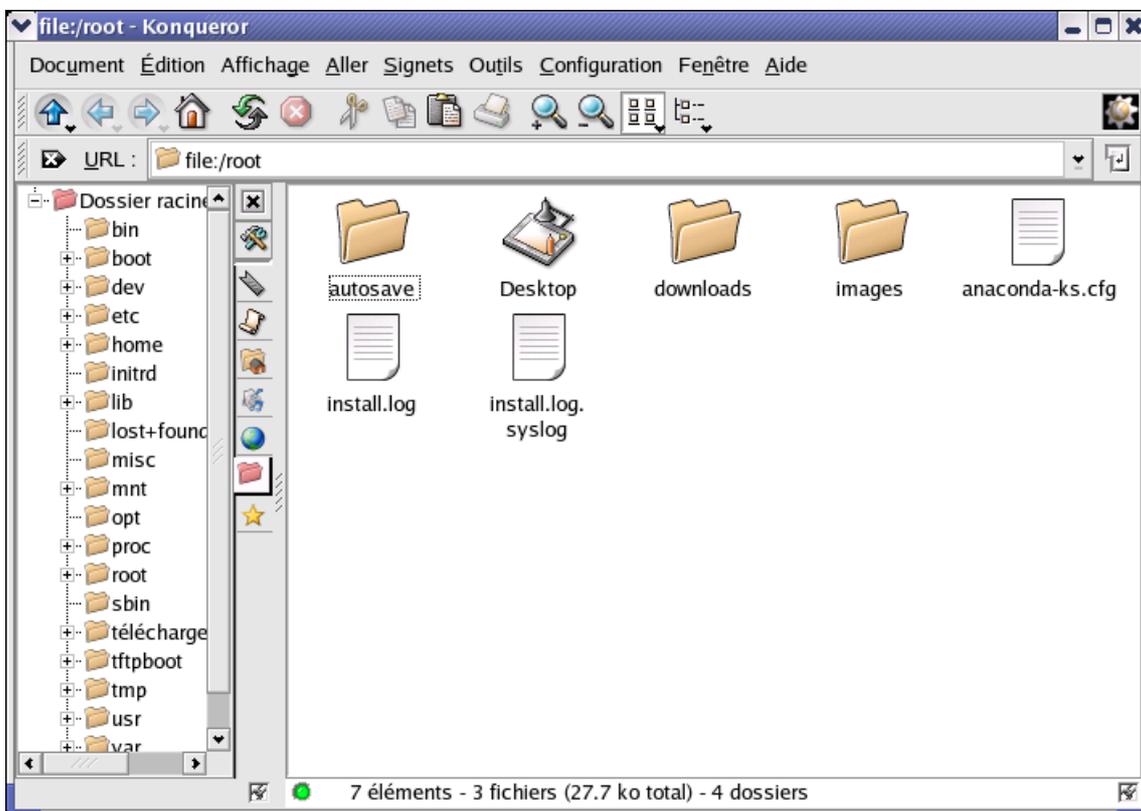
Le gestionnaire de fichiers **Konqueror** est disponible sur le bureau sous le nom **Dossier personnel** :



En mode ligne, il peut être lancé avec la commande **konqueror** :

```
# konqueror &
```

On obtient une fenêtre analogue à celle de l'explorateur windows :

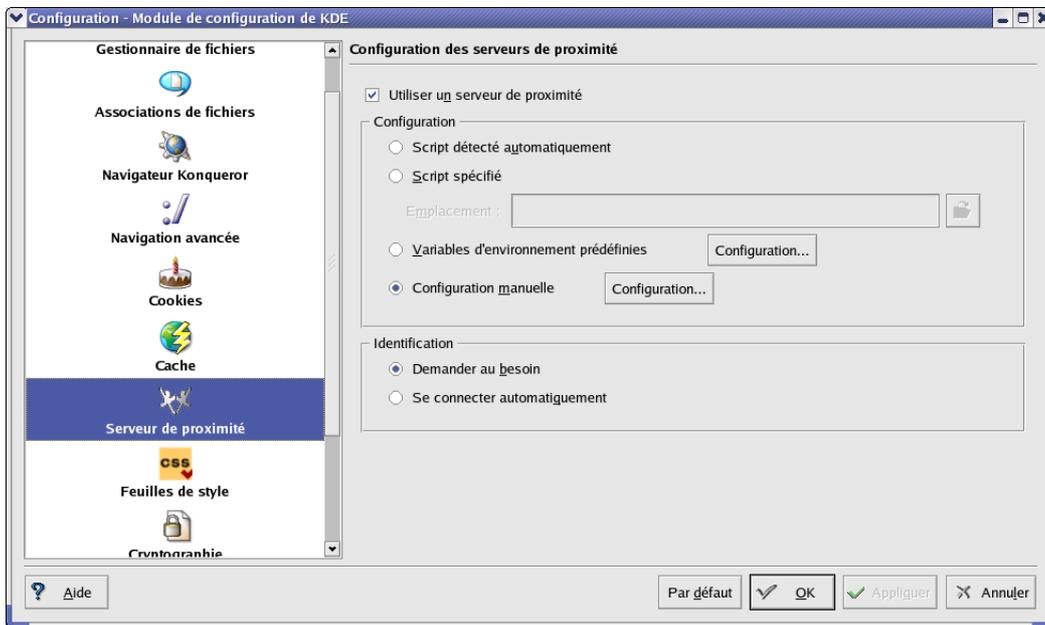


L'explorateur se positionne par défaut sur le dossier personnel de l'utilisateur qui l'a lancé (ici `root`). La touche **F9** permet d'avoir ou non l'arborescence des fichiers dans la partie gauche de la fenêtre (essayez). Cet outil vous permettra de gérer vos fichiers et répertoires sans avoir à taper les commandes Unix équivalentes. Voici quelques actions possibles :

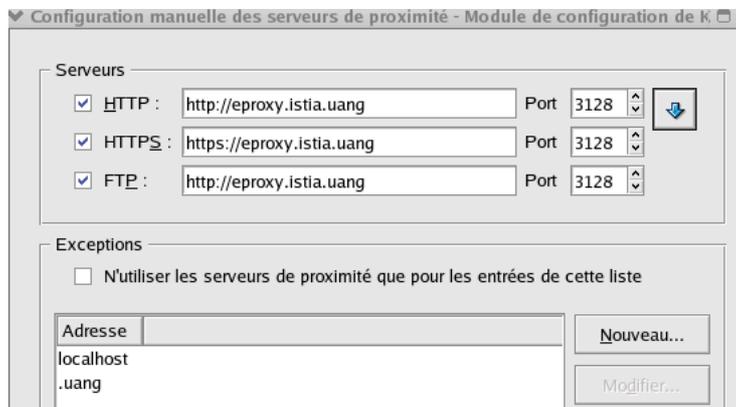
créer un dossier	clic droit/nouveau
aller dans un dossier	double clic sur le dossier
supprimer un dossier ou fichier	clic droit sur objet/supprimer
changer les droits d'un dossier ou fichier	clic droit sur objet/propriétés/permissions
duplication de fichiers	utiliser le copier/coller

Konqueror peut aller sur Internet avec différents protocoles (HTTP, FTP, ...). Pour cela, il devra passer par une machine relais puisque votre machine se trouve sur un réseau privé. Nous le configurons ainsi :

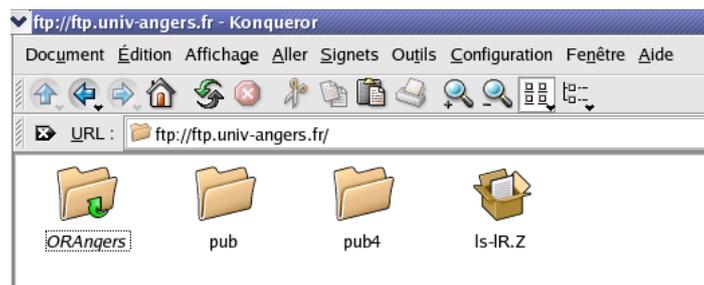
Configuration/Configurer Konqueror :



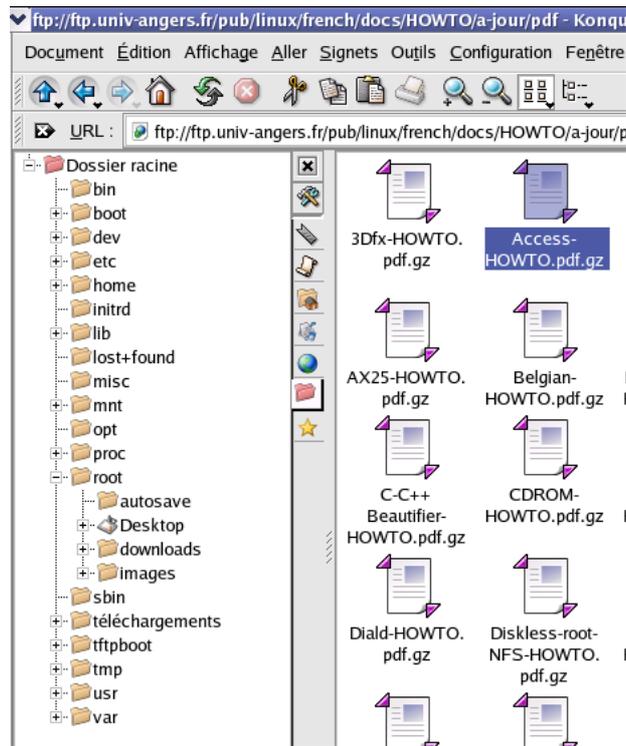
Prendre l'option [Configuration manuelle] et indiquer le nom de la machine relais pour les protocoles HTTP, HTTPS et FTP. Par ailleurs, à l'aide du bouton [Nouveau], préciser les domaines pour lesquels le proxy ne doit pas être utilisé : la machine locale (**localhost**) et les machines du réseau privé (**.uang**) :



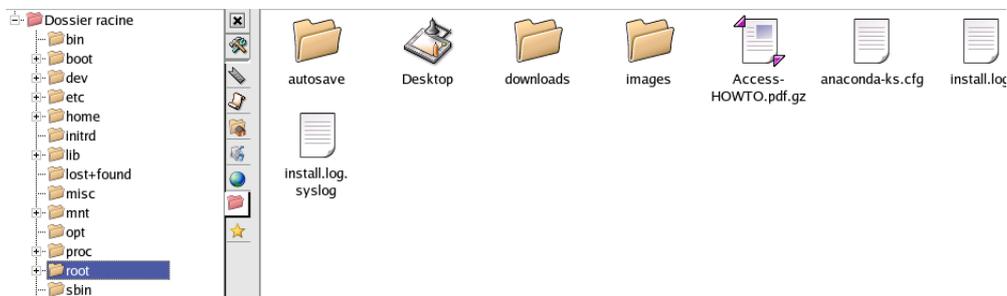
Après avoir validé cette configuration, demandez l'URL <http://istia.univ-angers.fr>. Vous devriez voir la page d'entrée du site web de l'istia. Faites F9 éventuellement pour supprimer le partitionnement en deux de la fenêtre. Essayez diverses URL de votre connaissance. Essayez ensuite l'URL <ftp://ftp.univ-angers.fr> :



Descendez dans l'arborescence du dossier **pub** jusqu'au dossier ftp://ftp.univ-angers.fr/pub/linux/french/docs/HOWTO/a-jour/pdf :



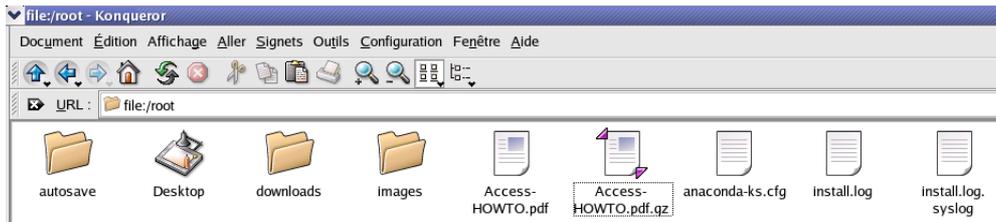
Avec la souris, prendre à droite le fichier Access-HOWTO.pdf.gz et le déposer dans le dossier **/root** à gauche puis double-cliquez sur ce dossier pour voir son contenu :



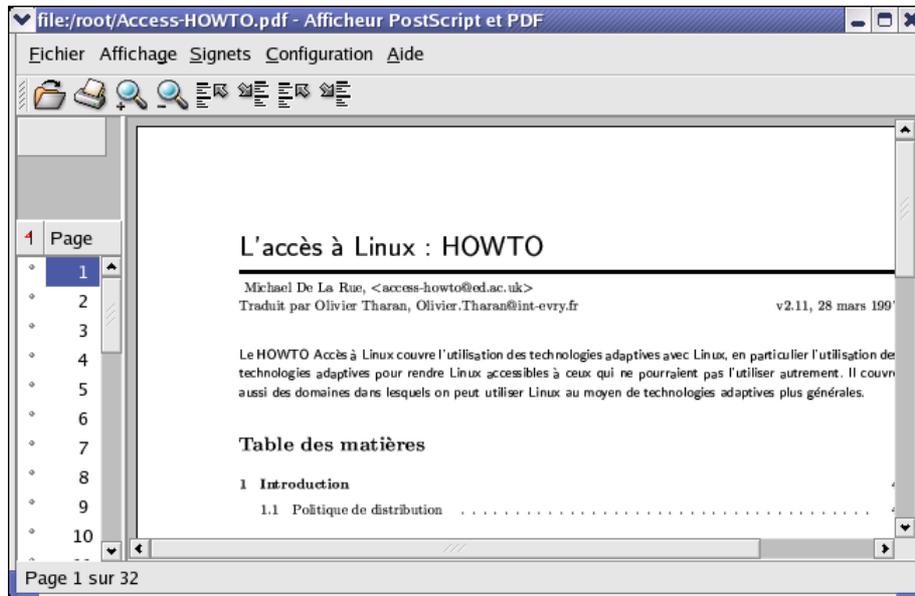
Le fichier téléchargé est un fichier zippé (suffixe gz). Double-cliquez dessus : un utilitaire d'archivage va être automatiquement lancé :



Cliquez droit sur le fichier **Access-HOWTO.pdf** et prenez l'option **[Extraire]**. Extrayez le fichier dans le dossier **/root** :



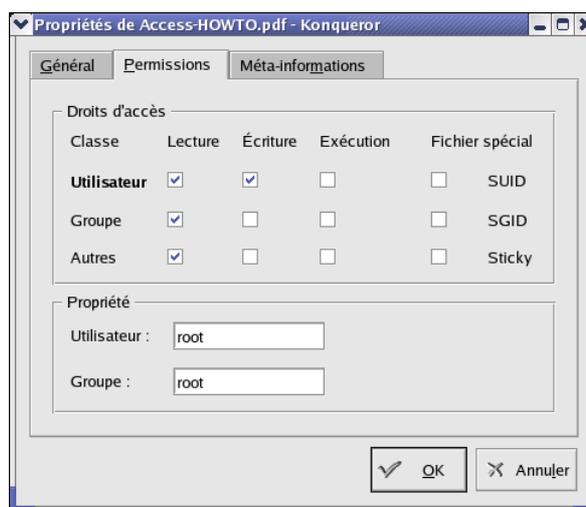
Double-cliquez maintenant sur le fichier **Access-HOWTO.pdf**. Il est alors visualisé par un afficheur PDF :



Les opérations précédentes ont montré la puissance et la convivialité de *Konqueror* et de l'interface KDE :

- gestion facilitée des fichiers et dossiers. Ceux-ci peuvent être sur internet.
- comme pour windows, des utilitaires sont associés à certains suffixes de fichiers et automatiquement lancés lorsqu'on double-clique sur ceux-ci.

Voyons maintenant un dernier point qui nous sera utile : la gestion des droits sur un dossier ou un fichier. Sélectionnez un fichier du dossier */root* et cliquez droit dessus. Prenez l'option **Propriétés** puis l'onglet **Permissions** :



Cette fenêtre permet de :

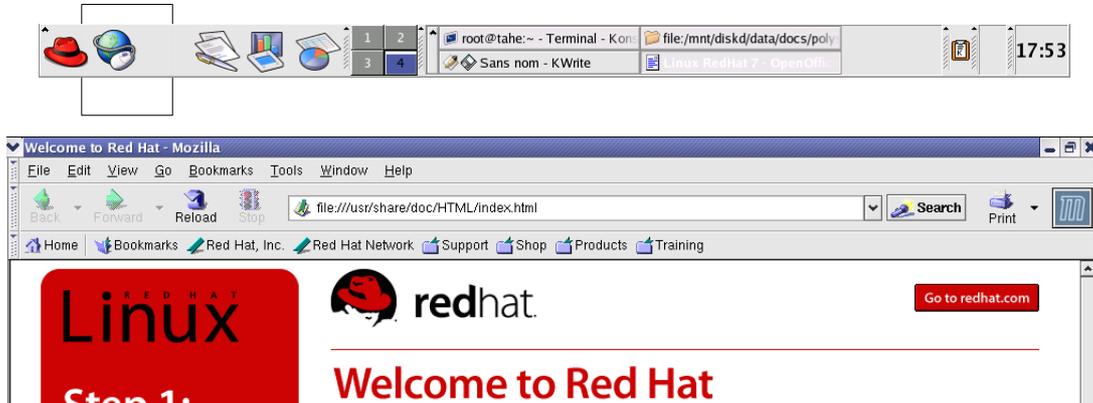
- connaître l'utilisateur propriétaire, le groupe propriétaire et les droits d'accès de l'objet (fichier ou dossier)

- de changer ces mêmes informations si vous en avez le droit (vous êtes root ou propriétaire de l'objet)

Nous serons amenés à utiliser cet outil pour parfois changer des droits d'accès.

4.1.4 Le navigateur Mozilla

Le navigateur Mozilla est disponible dans la barre des tâches :



Le navigateur Mozilla nous permettra d'aller chercher des informations sur le web et de gérer le courrier. C'est surtout pour cette dernière application qu'il nous sera utile. Nous avons vu en effet que Konqueror nous permettait également d'aller sur le web.

Le navigateur Mozilla peut également être lancé à partir d'un terminal :

```
# mozilla &
```

4.1.4.1 Configuration du proxy HTTP

Comme pour *Konqueror*, il nous faut indiquer à *Mozilla* par quelle machine passerelle il doit passer pour aller sur l'internet. Prendre l'option **Edit/Preferences/Advanced/Proxies** :

- cocher [Manual Proxy Configuration]
- HTTP Proxy : eproxy.istia.uang, port : 3128
- FTP Proxy : idem HTTP Proxy
- No proxy for : localhost, istia.uang (ne pas oublier le . devant istia).

Avec cette configuration, le navigateur passera par le proxy HTTP indiqué sauf lorsque l'URL demandée référence la machine locale (**localhost**) ou une machine du réseau privé **istia.uang**. Essayez diverses URL et vérifiez que vous les atteignez.

4.2 Systèmes de fichiers, périphériques et arborescence

4.2.1 Gestion des périphériques

La philosophie d'Unix est de considérer toutes les ressources (essentiellement des périphériques) comme des fichiers accessibles ou non aux utilisateurs. De ce fait, l'installation et l'administration en sont simplifiées (montage de systèmes de fichiers). Ces fichiers « spéciaux » sont rangés dans un répertoire spécial : **/dev**

4.2.2 Gestion des systèmes de fichiers

Un système de fichiers (*filesystem*) désigne l'organisation logique utilisée par le système d'exploitation pour gérer l'espace disque (physique) ou les partitions. Il est nécessaire de créer les systèmes de fichiers avant de pouvoir utiliser les disques et partitions. De façon générale les systèmes Unix utilisent une organisation des systèmes de fichiers en blocs (*superblocs*, *inodes* et blocs de données). On peut créer plusieurs types de systèmes de fichiers : *minix*, *ext2* (le plus populaire sous Linux), *msdos*,

ifat, ntfs... Plus récemment, d'autres systèmes de fichiers plus modernes ont été introduits : *ext3*, *reiserfs*, *xfs* (silicon graphics), *jfs* (IBM). Chacun a des avantages et des inconvénients.

La création d'un système de fichiers se réalise avec la commande **mkfs**. Il est possible de créer un système de fichiers sur une disquette ou sur un disque. En fait, chaque système de fichier dispose de ses propres utilitaires pour créer et vérifier son espace de stockage. **mkfs** est une commande universelle, qui permet d'avoir une vue unifiée, avec des paramètres standards. Elle appelle ensuite le bon utilitaire, en fonction du système de fichiers utilisé. Exemple :

```
mkfs -t ext2 -c /dev/fd0
```

appellera, en fait, **mke2fs** (le programme qui crée un système de fichiers ext2). Pour rendre un système de fichiers accessible aux utilisateurs, il doit être *monté* sur un répertoire. Par exemple, les disques durs disponibles sur une machine Unix sont montés automatiquement au démarrage de la machine. Le système de fichiers appelé *racine* ou *root* est monté ainsi sur le répertoire `/`, le système de fichiers pour les programmes est monté sur le répertoire `/usr`, etc...



Toutes les informations sur le montage (initial) des fichiers sont décrites dans le fichier nommé `/etc/fstab`. C'est ce fichier qui est lu au démarrage (boot) du système pour monter les systèmes de fichiers nécessaires.

Dans une fenêtre [**Kconsole**] visualisez le contenu du fichier `/etc/fstab` :

```
[root@tahe root]# more /etc/fstab
LABEL=/                                /                ext3    defaults        1 1
none                                    /dev/pts         devpts  gid=5,mode=620  0 0
none                                    /proc           proc    defaults        0 0
none                                    /dev/shm        tmpfs   defaults        0 0
/dev/hdb3                               swap            swap    defaults        0 0
/dev/cdrom                               /mnt/cdrom      iso9660 noauto,owner,kudzu,ro 0 0
/dev/cdrom1                             /mnt/cdrom1    iso9660 noauto,owner,kudzu,ro 0 0
/dev/hdd4                               /mnt/zip       auto   noauto,owner,kudzu 0 0
/dev/fd0                                 /mnt/floppy    auto   noauto,owner,kudzu 0 0
```

Sur la machine de test, il y avait un lecteur de CD (`/dev/cdrom`), un graveur de C (`/dev/cdrom1`), un lecteur zip (`/dev/hdd4`), un lecteur de disquettes (`/dev/fd0`). Les quatre lignes correspondantes dans `/etc/fstab` vont provoquer le « montage » de ces quatre systèmes de fichiers au démarrage du serveur. Ils seront montés aux points d'arborescence indiqués :

<code>/dev/cdrom</code>		<code>/mnt/cdrom</code>
<code>/dev/cdrom1</code>		<code>/mnt/cdrom1</code>
<code>/dev/hdd4</code>		<code>/mnt/zip</code>
<code>/dev/fd0</code>		<code>/mnt/floppy</code>

Visualisez le contenu du dossier `/mnt` :

```
[root@tahe root]# ls -la /mnt/
drwxr-xr-x  2 root  root    4096 mar  4 18:27 cdrom
drwxr-xr-x  2 root  root    4096 mar  4 18:27 cdrom1
drwxr-xr-x  2 root  root    4096 mar  4 18:27 floppy
drwxr-xr-x  2 root  root    4096 mar  4 18:27 zip
```

On trouve bien les quatre dossiers référencés (appelés points de montage) par le fichier `/etc/fstab`. Ces dossiers peuvent être vides si au boot il n'y avait rien dans le périphérique correspondant :

```
[root@tahe root]# ls -l /mnt/cdrom
total 0
```

La commande précédente montre qu'il n'y a pas de système de fichiers monté sous le dossier `/mnt/cdrom`. La commande **mount** permet de *monter* un système de fichiers sur un répertoire. Elle est accessible à l'administrateur, mais pas aux autres utilisateurs pour des raisons de sécurité. La syntaxe standard de la commande est :

mount -t type device dir

mais elle s'utilise aussi sous diverses autres formes :

```
mount -a [-fnrvw] [-t vfstype]
mount [-fnrvw] [-o options [...]] device | dir
mount [-fnrvw] [-t vfstype] [-o options] device dir
```

Faites un **man** de la commande **mount** pour découvrir la signification des différentes options. Utilisez cette commande pour accéder au contenu d'un CD. Pour ce faire, il faut monter le système de fichiers de celui-ci. Lorsque le périphérique à monter est décrit dans le fichier `/etc/fstab`, il peut être monté avec une commande *mount* simplifiée : **mount périphérique** ou **mount point_de_montage**. Ainsi pour monter le périphérique `/dev/cdrom` on pourra utiliser la commande :

```
mount /dev/cdrom
```

Nous le faisons. Nous introduisons un CD, nous le montons et affichons son contenu :

```
[root@tahe root]# mount /dev/cdrom
[root@tahe root]# ls -l /mnt/cdrom
total 598455
-r-xr-xr-x  1 root    root      612817854 août  7  2002 oracle.zip
```

Pour démonter un système de fichiers, on utilise la commande `umount` :

```
[root@tahe root]# umount /dev/cdrom
[root@tahe root]# ls -l /mnt/cdrom
total 0
```

On voit ci-dessus, qu'après la commande *umount*, il n'y a plus de système de fichiers sous `/mnt/cdrom` alors que le CD est toujours dans le lecteur. En utilisant la syntaxe complète de *mount*, on peut monter un système de fichiers n'importe où dans l'arborescence comme le montre l'exemple suivant :

```
[root@tahe root]# mkdir /mnt/oracle1
[root@tahe root]# mount -t iso9660 /dev/cdrom /mnt/oracle1/
mount : Périphérique de bloc/dev/cdrom est protégé en écriture. Montage en lecture seule.
[root@tahe root]# ls -l /mnt/oracle1/
total 598455
-r-xr-xr-x  1 root    root      612817854 août  7  2002 oracle.zip
```

Ici, le même CD que précédemment a été monté sous `/mnt/oracle1`. Il sera démonté par :

```
[root@tahe root]# umount /dev/cdrom
```

ou

```
[root@tahe root]# umount /mnt/oracle1/
```

Appliquons ce qui vient d'être vu à une disquette. Nous prenons une disquette vierge et nous la formatons au format `ext3` :

```
[root@tahe root]# mkfs -t ext3 /dev/fd0
mke2fs 1.27 (8-Mar-2002)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 inodes, 1440 blocks
72 blocks (5.00%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
184 inodes per group

Writing inode tables: done

Filesystem too small for a journal
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Montons cette disquette (syntaxe simplifiée parce que périphérique présent dans `/etc/fstab`) :

```
[root@tahe root]# mount /dev/fd0
```

```
[root@tahe root]# ls -l /mnt/floppy/
total 12
drwx-----  2 root    root      12288 mar 28 14:18 lost+found
```

On voit que le formatage a créé un dossier appelé *lost+found*. Maintenant, copions un fichier sur la disquette :

```
[root@tahe root]# cp /etc/passwd /mnt/floppy/fic1
```

```
[root@tahe root]# ls -l /mnt/floppy/fic1
-rw-r--r--  1 root    root      2003 mar 28 14:24 /mnt/floppy/fic1
```

Lorsque nous n'avons plus besoin de la disquette, nous démontons le périphérique **/dev/fd0** :

```
[root@tahe root]# umount /dev/fd0
```

```
[root@tahe root]# ls -l /mnt/floppy/
total 0
```

Maintenant formatons la même disquette au format DOS. Cela permettra de l'utiliser avec un système windows par exemple :

```
[root@tahe root]# mkfs -t msdos /dev/fd0
mkfs.msdos 2.8 (28 Feb 2001)
[root@tahe root]# mount -t msdos /dev/fd0 /mnt/floppy/
[root@tahe root]# cp /etc/passwd /mnt/floppy/fic2
[root@tahe root]# ls -l /mnt/floppy/fic2
-rwxr-xr-x  1 root    root      2003 mar 28 14:28 /mnt/floppy/fic2
```

Ci-dessus, on a :

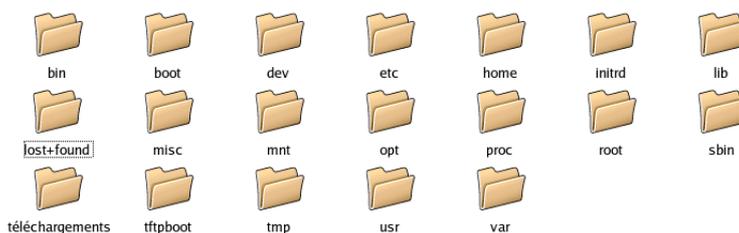
- formaté la disquette au format DOS
- monté la disquette en précisant son type msdos

Ceci fait, la disquette est vue comme le dossier **/mnt/floppy** et est gérée comme le reste de l'arborescence. Seulement, les fichiers qui y sont copiés le sont au format MS-DOS. Il y a là un concept très puissant.

Ce qui vient d'être fait pourrait être répété pour le lecteur Zip. Les gestionnaires de fichiers, intégrés dans les bureaux évolués, comme GNOME ou KDE, permettent le montage ou démontage en un clic de souris - encore faut-il que la ressource soit montable par un utilisateur (il faut que les attributs *noauto*, *user* soient positionnés dans */etc/fstab*). Des mini applications ("applets" peuvent aussi faire ce travail).

4.2.3 Organisation des données sur le système de fichiers

Sur un système Unix, les données du système et des utilisateurs ne sont pas mélangées. L'arborescence d'un système Unix est relativement bien définie. Avec *Konqueror*, placez vous sur la racine de l'arborescence des fichiers du serveur :



- **bin** : les binaires nécessaires au fonctionnement de la base du système.
- **etc** : les fichiers de configuration du système et des programmes (paramétrage par défaut).
- **lib** : les bibliothèques partagées du système (et les modules dynamique du noyau)
- **mnt** : un répertoire dédié pour le montage d'autre systèmes de fichiers.
- **root** : le répertoire où sont stockées les données du super utilisateur.
- **tmp** : sont stockées ici les données temporaires créées par les programmes des utilisateurs. Ce répertoire est normalement vidé à chaque démarrage de la machine.

- **var** : sont stockés ici tous les fichiers créés par les programmes, mais qui doivent être conservés sur une certaine période : Les courriers électroniques arrivés ou en partance, l'historique de connexion des utilisateurs, etc...
- **boot** : répertoire spécifique à Linux, dans lequel sont stockés les noyaux de la machine.
- **dev** : Tous les pseudo fichiers représentant les périphériques du système y sont stockés; /dev/floppy désigne la disquette par exemple.
- **home** : dessous home, il y a d'autres sous répertoires (un par compte Unix) dans lequel sont stockées les données de chaque utilisateur. ex: /home/dupont
- **proc** : répertoire spécifique à quelques Unix (Linux, Digital Unix), sur lequel est monté un pseudo système de fichiers qui représente l'état du système.
- **sbin** : les binaires du système. Y sont stockés tous les binaires nécessitant d'être super utilisateur pour les utiliser : (gestion des interfaces réseaux, montage et démontage de système de fichiers...)
- **usr** : ce répertoire n'est pas à proprement parler nécessaire au bon fonctionnement du système de base. Par contre, tous les binaires et bibliothèques nécessaires aux programmes (autres que ceux qui sont considérés comme partie système) y sont stockés. Ce sous répertoire contient donc une arborescence complète, et est souvent très volumineux (on peut y trouver XWindow, Latex, Gnome, KDE, traitement de textes, navigateurs WWW, etc...)



Les distributions Linux suivent ce standard, et sont encore plus strictes sur l'emplacement des programmes et de leurs données. Elles suivent un document appelé FHS.

4.3 Gestion des comptes utilisateurs

Les utilisateurs d'un système Unix sont répartis en groupes d'utilisateurs. Sur quels critères met-on des utilisateurs dans un groupe ou un autre ? Quelquefois c'est évident. À l'ISTIA, on met les étudiants d'une même filière dans un même groupe. Cela offre de multiples avantages :

- on donne au groupe le nom de la filière. Ainsi on sait quel type d'étudiants sont dans le groupe
- les opérations sur les membres d'un groupe sont nombreuses :
 - affecter le même quota disque à tous les membres du groupe
 - envoyer un mél à tous les membres du groupe
 - déplacer tous les membres d'un groupe dans un autre (changement de filière en fin d'année)

On peut mettre également dans un même groupe des utilisateurs qui pourraient avoir à se partager des fichiers. Ce cas est cependant peu fréquent. Les droits sur les fichiers se répartissent en trois groupes de droits :

1. ceux pour l'utilisateur propriétaire du fichier
2. ceux pour le groupe propriétaire du fichier
3. ceux pour les utilisateurs n'entrant pas dans les cas 1 et 2

On mettra dans un même groupe des utilisateurs ayant à partager des fichiers. Le groupe serait propriétaire des fichiers et ainsi tous les utilisateurs membres du groupe auraient les mêmes droits d'accès aux fichiers appartenant au groupe.

La gestion des utilisateurs se fait sous l'identité **root**. On peut le faire en mode texte ou graphique. Nous le ferons ici en mode graphique. Lancez le gestionnaire d'utilisateurs par **Menu KDE/Paramètres de système/Gestionnaire d'utilisateurs** :



4.3.1 Création de groupes d'utilisateurs

Commençons par créer deux groupes appelés **groupe1** et **groupe2** avec l'icône [Ajouter Groupe] ci-dessus :

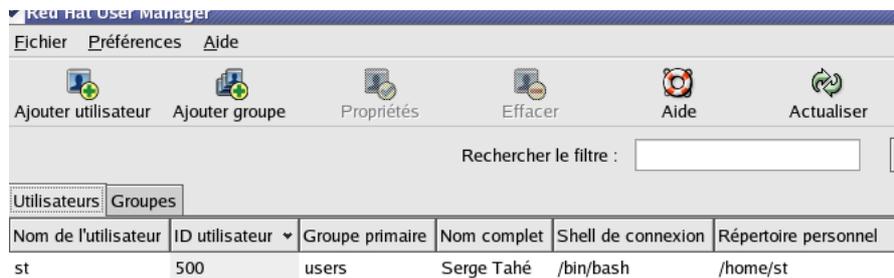


Une fois créés, les groupes apparaissent dans la liste des groupes :

Utilisateurs	Groupes	
Nom du groupe	ID groupe	Men
admin	500	st
groupe1	501	
groupe2	502	

4.3.2 Création d'utilisateurs

Maintenant, sélectionnons l'onglet [Utilisateurs] pour créer de nouveaux utilisateurs :



Avec le bouton [Ajouter Utilisateur] ci-dessus, créons quatre utilisateurs **uti1**, **uti2**, **uti3**, **uti4**. Le mot de passe de chacun d'eux sera **azerty**.



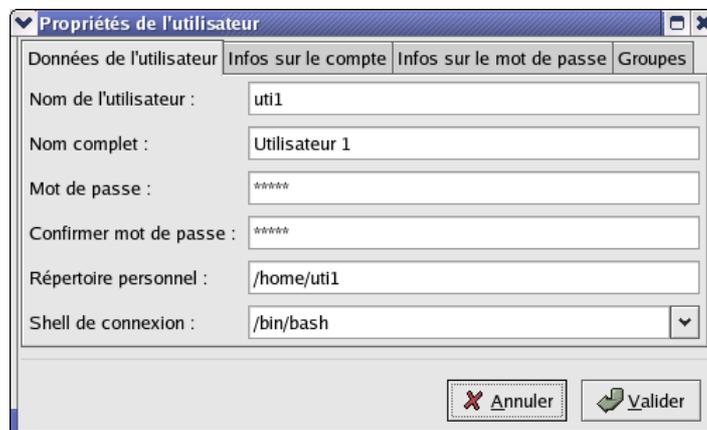
On répétera l'opération quatre fois. On a alors la liste d'utilisateurs suivante :

Nom de l'utilisateur	ID utilisateur	Groupe primaire	Nom complet	Shell de connexion	Répertoire person
st	500	users	Serge Tahé	/bin/bash	/home/st
uti1	501	uti1	Utilisateur 1	/bin/bash	/home/uti1
uti2	502	uti2	utilisateur 2	/bin/bash	/home/uti2
uti3	503	uti3	Utilisateur 3	/bin/bash	/home/uti3
uti4	504	uti4	Utilisateur 4	/bin/bash	/home/uti4

Le gestionnaire a créé un nouveau groupe pour chacun de ces quatre utilisateurs comme le montre la liste des groupes :

Nom du groupe	ID groupe	Membres d
admin	500	st
groupe1	501	
groupe2	502	
uti1	503	uti1
uti2	504	uti2
uti3	505	uti3
uti4	506	uti4

On peut changer les propriétés d'un utilisateur. Revenir sur l'onglet **[Utilisateurs]**, sélectionnez **uti1** et cliquez sur le bouton **[Propriétés]** de la barre d'outils :



On voit qu'on peut changer certains des attributs de l'utilisateur. Pour gérer son appartenance à différents groupes, cliquez sur l'onglet **[groupes]** :



Nous allons faire de **groupe1**, le groupe primaire de l'utilisateur **uti1**. Ce sera par ailleurs son unique groupe. Procédez comme suit :

- cochez le groupe **groupe1** dans la liste des groupes
- décochez le groupe **uti1** dans la liste des groupes
- sélectionnez **groupe1** dans la liste déroulante du groupe primaire
- validez

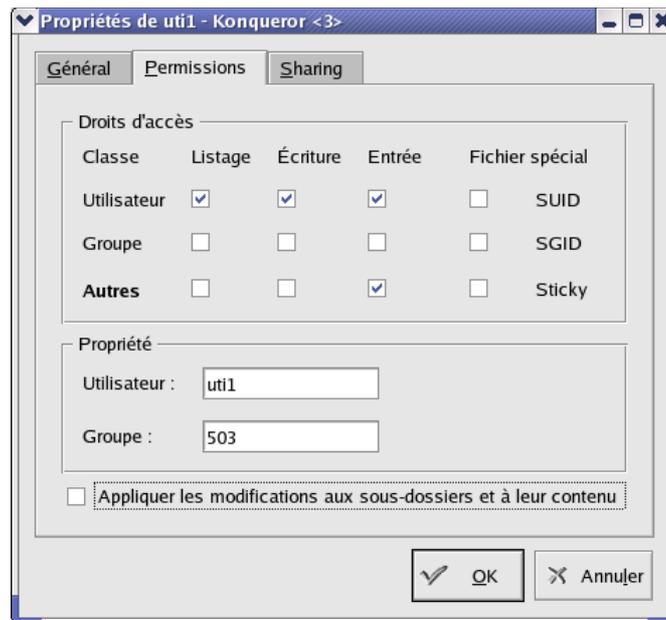
Procédez ainsi pour chacun des quatre utilisateurs : **uti1** et **uti2** iront dans le groupe **groupe1** et **uti3** et **uti4** dans le groupe **groupe2** :

Nom de l'utilisateur	ID utilisateur ▼	Groupe primaire	Nom complet	Shell de connexion	Répertoire personnel
st	500	users	Serge Tahé	/bin/bash	/home/st
uti1	501	groupe1	Utilisateur 1	/bin/bash	/home/uti1
uti2	502	groupe1	utilisateur 2	/bin/bash	/home/uti2
uti3	503	groupe2	Utilisateur 3	/bin/bash	/home/uti3
uti4	504	groupe2	Utilisateur 4	/bin/bash	/home/uti4

Revenez dans l'onglet **[Groupes]** et supprimez les groupes **uti1** à **uti4** avec le bouton **[Effacer]** :

Nom du groupe	ID groupe ▼	Membres du groupe
admin	500	st
groupe1	501	uti1, uti2
groupe2	502	uti3, uti4

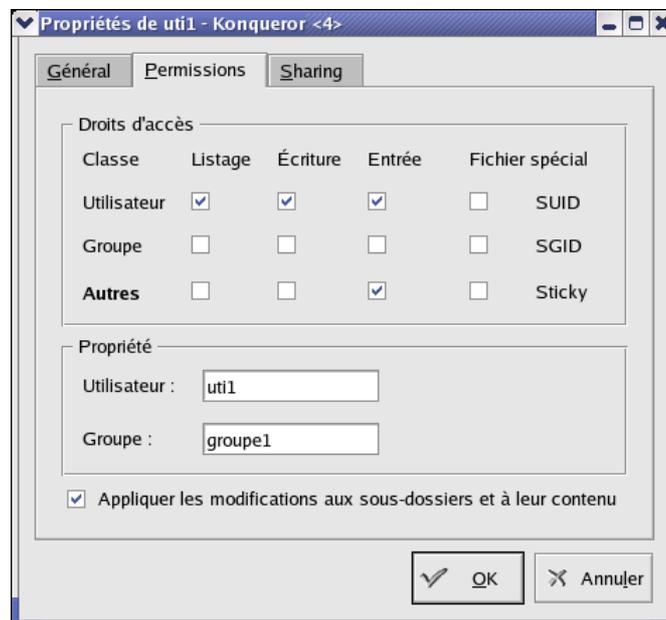
Lorsque l'utilisateur **uti1** a été créé, le groupe **uti1** a été créé et l'utilisateur **uti1** a été mis dedans. Par ailleurs, un dossier **/home/uti1**, appelé répertoire de connexion a été créé. Son propriétaire était l'utilisateur **uti1**, son groupe propriétaire le groupe **uti1**. Le fait de mettre l'utilisateur **uti1** dans le groupe **groupe1** n'a pas changé ces droits. Ainsi maintenant le groupe propriétaire du dossier **/home/uti1** est un groupe qui n'existe plus... Pour le vérifier, utilisez [Konqueror] et visualisez les droits du dossier **/home/uti1** :



Le groupe 503 n'existe pas et c'est pourquoi c'est un numéro qui apparaît et non le nom du groupe. Cela peut se vérifier également en mode texte dans la fenêtre [Konsole] :

```
[root@tahe root]# ls -ld /home/uti1/
drwx-----x  25 utili  503  4096 mar 25 18:39 /home/uti1/
```

Pour changer le groupe propriétaire de **/home/uti1**, mettez **groupe1** dans le champ **Groupe** ci-dessus et cochez la case « Appliquez les ... aux sous-dossiers ... » :



Le groupe propriétaire de **/home/uti1** et de ses sous-dossiers devient **groupe1** comme le montre les résultats de la commande suivante dans [Konsole] :

```
[root@tahe root]# ls -ld /home/uti1/
drwx-----x  25 utili  groupe1  4096 mar 25 18:39 /home/uti1/
```

Faites la même chose pour **uti2 (groupe1)**, **uti3 (groupe2)**, **uti4(groupe2)**. On retiendra de toutes ces manipulations qu'il est préférable de fixer dès la création de l'utilisateur le groupe auquel il doit appartenir.

4.3.3 Les fichiers /etc/passwd, /etc/shadow, /etc/group

Les groupes d'utilisateurs sont enregistrés dans le fichier `/etc/group`. Dans la fenêtre terminal (Kconsole) tapez la commande suivante qui visualise le fichier `/etc/group` page par page :

```
# more /etc/group
```

Vous verrez quelque chose du genre :

```
....
admin:x:500:st
postdrop:x:90:
postfix:x:89:
mailman:x:41:
groupe1:x:501:uti1,uti2
groupe2:x:502:uti3,uti4
```

On retrouve bien les deux groupes `groupe1` et `groupe2` que nous avons créés. Chaque ligne comporte quatre champs :

nom du groupe : mot de passe du groupe : numéro du groupe : membres

Les comptes utilisateurs sont eux regroupés dans le fichier `/etc/passwd`. Tapez la commande :

```
# more /etc/passwd
```

Vous obtiendrez un affichage analogue à ce qui suit :

```
....
webalizer:x:67:67:Webalizer:/var/www/html/usage:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
st:x:500:100:Serge Tah□/home/st:/bin/bash
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
mailman:x:41:41:GNU Mailing List Manager:/var/mailman:/bin/false
uti1:x:501:501:Utilisateur 1:/home/uti1:/bin/bash
uti2:x:502:501:utilisateur 2:/home/uti2:/bin/bash
uti3:x:503:502:Utilisateur 3:/home/uti3:/bin/bash
uti4:x:504:502:Utilisateur 4:/home/uti4:/bin/bash
```

On retrouve les quatre utilisateurs `uti1` à `uti4` que nous avons créés. Chaque ligne de ce fichier a sept champs :

login : mot de passe (codé) : son numéro d'identité : son numéro de groupe : son identité : son répertoire : le shell utilisé

Dans l'exemple ci-dessus, le champ [mot de passe] contient toujours x. Le mot de passe est en fait stocké dans le fichier `/etc/shadow`. Visualisez ce fichier à l'aide de la commande :

```
# more /etc/shadow
```

L'affichage devrait être analogue à ce qui suit :

```
...
uti1:7kySlexQwrPfg:12138:-1:99999:-1:::
uti2:FeEwPwhlG5GUc:12138:-1:99999:-1:::
uti3:BeNj/O9D/xk.k:12138:-1:99999:-1:::
uti4:DtfGVCEUR7bhc:12138:-1:99999:-1:::
....
```

On retrouve les quatre utilisateurs `uti1` à `uti4` avec cette fois-ci leur mot de passe (crypté quand même) et d'autres informations sur la durée de validité de ces mots de passe. Pourquoi les mots de passe ne sont-ils pas stockés dans le fichier `/etc/passwd`? Regardons les droits des deux fichiers :

```
[root@tahe root]# ls -l /etc/passwd
-rw-r--r--  1 root  root      2003 mar 27 17:17 /etc/passwd
```

```
[root@tahe root]# ls -l /etc/shadow
-r-----  1 root  root      1266 mar 27 17:17 /etc/shadow
```

On voit que le fichier `/etc/passwd` est lisible par tout le monde (droits `r--` dans les trois derniers caractères des droits) alors que le fichier `/etc/shadow` n'est lisible que par son propriétaire `root`. Seul `root` a donc accès aux mots de passe

cryptés des utilisateurs. Ces mots de passe étaient autrefois cryptés dans */etc/passwd* lui-même. Des programmes pouvaient alors les lire et les « casser ».

4.3.4 Gestion des utilisateurs en mode texte

Les groupes et les utilisateurs peuvent être gérés à l'aide de commandes Unix :

- **groupadd** pour créer un groupe
- **groupdel** pour supprimer un groupe
- **groupmod** pour modifier les propriétés d'un groupe
- **useradd** pour créer un utilisateur
- **userdel** pour supprimer un utilisateur
- **usermod** pour modifier les propriétés d'un utilisateur

On utilisera la commande **man** pour obtenir la syntaxe exacte de ces différentes commandes.

4.3.5 Droits d'accès attachés aux utilisateurs

Nous allons tenter d'illustrer ici la notion de droits d'accès aux fichiers et répertoires. A la création d'un utilisateur est également créé un dossier qui lui appartient et sur lequel il est placé lorsqu'il se connecte. C'est ce qu'on appelle son répertoire de connexion (home directory). Les copies d'écran précédentes ont ainsi montré que le répertoire de connexion de l'utilisateur **util** était **/home/util**. Regardons les droits d'accès de celui-ci :

```
[root@tahe root]# ls -ld /home/util
drwx----- 26 util groupe1 4096 mar 28 15:49 /home/util
```

On voit que le propriétaire **util** a tous les droits (1er groupe de droits=**rw**x), que les membres de son groupe (**groupe1**) n'en ont aucun (2ième groupe de droits=**---**), et que les autres utilisateurs n'en ont aucun également (3ième groupe de droits=**---**). Ces droits sont positionnés par défaut pour une sécurité maximale. Pour les besoins de notre démonstration, changeons-les. Donnons les droits **r-x** aux membres du groupe **groupe1**, et aux autres les droits **--x** :

```
[root@tahe root]# chmod g+r,g+x,o+x /home/util
```

```
[root@tahe root]# ls -ld /home/util
drwxr-x--x 26 util groupe1 4096 mar 28 15:49 /home/util
```

La commande **chmod** permet de changer les droits d'accès d'un fichier ou dossier. Seul root ou le propriétaire du fichier ou dossier peut changer les droits d'accès de celui-ci. Il y a, nous l'avons vu trois groupes de droits :

1. les droits pour le propriétaire du fichier ou dossier, représentés par les trois premières permissions
2. les droits pour le groupe propriétaire du fichier ou dossier, représentés par les trois permissions suivantes
3. les droits des autres utilisateurs, représentés par les trois dernières permissions

Le 1er groupe est représenté par la lettre **u** (user), le second par la lettre **g** (group), le troisième par la lettre **o** (others). Pour ajouter un droit **D** au propriétaire, on écrira **chmod u+D**. Pour lui enlever ce même droit, on écrira **chmod u-D**. On peut changer plusieurs droits en même temps. On écrit alors **chmod u+/-D1,u+/-D2,...**

Pour donner les droits **r** et **x** au groupe propriétaire (lettre **g**) et le droit **x** aux autres utilisateurs (lettre **o**), on écrira donc **chmod g+r,g+x,o+x** :

```
[root@tahe root]# chmod g+r,g+x,o+x /home/util
```

```
[root@tahe root]# ls -ld /home/util
drwxr-x--x 26 util groupe1 4096 mar 28 15:49 /home/util
```

On peut, de façon plus simple, utiliser **[Konqueror]** pour changer ces droits. Affichez pour cela les propriétés du dossier **/home/util** :

Droits d'accès			
Classe	Listage	Écriture	Entrée
Utilisateur	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Groupe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Autres	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

L'utilisation de *Konqueror* n'est possible qu'en mode interactif. L'intérêt des commandes Unix est qu'elles peuvent être incluses dans un fichier batch. Maintenant, lancez une autre application [Konsole] et prenez l'identité de **uti1** :

```
[root@tahe root]# su uti1
```

```
[uti1@tahe root]$ id
uid=501(uti1) gid=501(groupe1) groupes=501(groupe1)
```

La commande **su** permet de changer d'identité. Ici c'est **root** qui change d'identité, aussi aucun mot de passe n'est demandé. Sinon, un mot de passe aurait été demandé. Lancez deux autres applications [Konsole] et connectez-vous successivement comme les utilisateurs *uti2* et *uti3*. On se rappelle que *uti1* et *uti2* sont dans le même groupe *groupe1* alors que *uti3* est dans le groupe *groupe2* :

Dans le terminal n° 3 :

```
[root@tahe root]# su uti2
```

```
[uti2@tahe root]$ id
uid=502(uti2) gid=501(groupe1) groupes=501(groupe1)
```

Dans le terminal n° 4 :

```
[root@tahe root]# su uti3
```

```
[uti3@tahe root]$ id
uid=503(uti3) gid=502(groupe2) groupes=502(groupe2)
```

On remarquera que dans le message d'attente du système, le nom de l'utilisateur connecté est indiqué. Cela vous permettra de savoir qui est l'utilisateur dans chaque fenêtre terminal. Les utilisateurs *uti1*, *uti2*, *uti3* vont tenter d'afficher le contenu de */home/uti1* :

Terminal **uti1** :

```
[uti1@tahe root]$ ls -l /home/uti1
drwx-----x  3 uti1  groupe1    4096 mar 25 18:38 Desktop
drwx-----x  2 uti1  groupe1    4096 mar  7 18:01 downloads
-rw-----  1 uti1  groupe1  342125 mar 25 15:52 gnome
-rw-----  1 uti1  groupe1  342096 mar 25 15:53 gnome.ui
-rw-----  1 uti1  groupe1  342123 mar 25 15:52 gtk
-rw-----  1 uti1  groupe1  342126 mar 25 15:52 signal
-rw-----  1 uti1  groupe1 4789869 mar 25 15:52 sys
```

Terminal **uti2** :

```
[uti2@tahe root]$ ls -l /home/uti1
drwx-----x  3 uti1  groupe1    4096 mar 25 18:38 Desktop
drwx-----x  2 uti1  groupe1    4096 mar  7 18:01 downloads
-rw-----  1 uti1  groupe1  342125 mar 25 15:52 gnome
-rw-----  1 uti1  groupe1  342096 mar 25 15:53 gnome.ui
-rw-----  1 uti1  groupe1  342123 mar 25 15:52 gtk
-rw-----  1 uti1  groupe1  342126 mar 25 15:52 signal
-rw-----  1 uti1  groupe1 4789869 mar 25 15:52 sys
```

Terminal **uti3** :

```
[uti3@tahe root]$ ls -l /home/uti1
ls: /home/uti1: Permission non accordée
```

Pour pouvoir lister le contenu de `/home/uti1`, il faut disposer du droit de lecture (`r`) sur ce dossier. Rappelons les droits d'accès du dossier :

```
[root@tahe root]# ls -ld /home/uti1/
drwxr-x--x  25 uti1  groupe1  4096 mar 25 18:39 /home/uti1/
```

L'utilisateur **uti1** a le droit **r** en tant que propriétaire. L'utilisateur **uti2** l'a également en tant que membre du **groupe1** qui a le droit de lecture. L'utilisateur **uti3** tombe dans les autres cas (3ième groupe de droits `-x`) et n'a pas le droit de lecture. Ce qui explique qu'il n'a pas pu lister le contenu de `/home/uti1`. Maintenant ces droits montrent que le groupe propriétaire **groupe1** n'a pas de droit d'écriture (**w**) sur `/home/uti1`. Il ne peut donc pas ajouter d'entrées au dossier (un fichier ou un répertoire). L'utilisateur **uti1** a lui un droit d'écriture. Vérifions ces deux points :

Terminal **uti1** :

```
[uti1@tahe root]$ mkdir /home/uti1/test1

[uti1@tahe root]$ ls -ld /home/uti1/test1
drwxr-xr-x  2 uti1  groupe1  4096 mar 28 15:46 /home/uti1/test1
```

Terminal **uti2** :

```
[uti2@tahe root]$ mkdir /home/uti1/test2
mkdir: Ne peut créer le répertoire `/home/uti1/test2'.: Permission non accordée
```

Tout se passe comme attendu. Terminons par le droit `x`. Créons un fichier dans `/home/uti1` lisible par tous. Dans le terminal *uti1* :

```
[uti1@tahe root]$ echo test >/home/uti1/fic1

[uti1@tahe root]$ ls -l /home/uti1/fic1
-rw-r--r--  1 uti1  groupe1  5 mar 28 15:49 /home/uti1/fic1

[uti1@tahe root]$ more /home/uti1/fic1
test
```

Rendons le fichier **fic1** lisible par tous :

```
[uti1@tahe root]$ chmod +r /home/uti1/fic1

[uti1@tahe root]$ ls -l /home/uti1/fic1
-rw-r--r--  1 uti1  groupe1  5 mar 28 15:49 /home/uti1/fic1
```

Faisons lire ce fichier par **uti2** et **uti3** :

Terminal **uti2** :

```
[uti2@tahe root]$ more /home/uti1/fic1
test
```

Terminal **uti3** :

```
[uti3@tahe root]$ more /home/uti1/fic1
test
```

L'utilisateur **uti3** qui ne pouvait lister le contenu de `/home/uti1` (absence du droit **r** sur `/home/uti1`) a pu lire le fichier **fic1** pour deux raisons :

1. il a le droit d'entrer dans le dossier `/home/uti1` (droit **x** sur `/home/uti1`)
2. il a le droit de lire *fic1* (droit **r** sur *fic1*)

Revenons dans le terminal **uti1** pour enlever le droit `x` à **uti3** sur le dossier `/home/uti1` :

```
[uti1@tahe root]$ chmod o-x /home/uti1

[uti1@tahe root]$ ls -ld /home/uti1
drwxr-x---  26 uti1  groupe1  4096 mar 28 15:49 /home/uti1
```

Dans le terminal **uti3**, essayons de nouveau d'afficher le contenu de **fic1** :

```
[uti3@tahe root]$ more /home/uti1/fic1
/home/uti1/fic1: Permission non accordée
```

Maintenant que **uti3** n'a plus le droit **x**, il ne peut plus « entrer » dans le dossier **/home/uti1**.

Le raisonnement qui a été tenu pour le dossier **/home/uti1** peut être étendu à un fichier, le droit **x** signifiant alors le droit d'**exécuter** le fichier. Les manipulations de droits faites avec la commande Unix **chmod** peuvent être faites avec Konqueror ce qui évite de se rappeler la syntaxe de la commande **chmod**.

Pour terminer, rendons au dossier **/home/uti1** ses droits initiaux dans le terminal **uti1** :

```
[root@tahe root]# chmod u+r,u+w,u+x,g-r,g-w,g-x,o-r,o-w,o-x /home/uti1
```

```
[root@tahe root]# ls -ld /home/uti1
drwx----- 26 uti1      groupe1      4096 mar 28 15:49 /home/uti1
```

De ces manipulations, on se rappellera les deux points suivants :

- l'utilisateur **root** peut tout faire
- un autre utilisateur peut généralement tout faire dans son dossier personnel et rien dans ceux des autres utilisateurs.

Guide d'installation des services réseau sous Linux

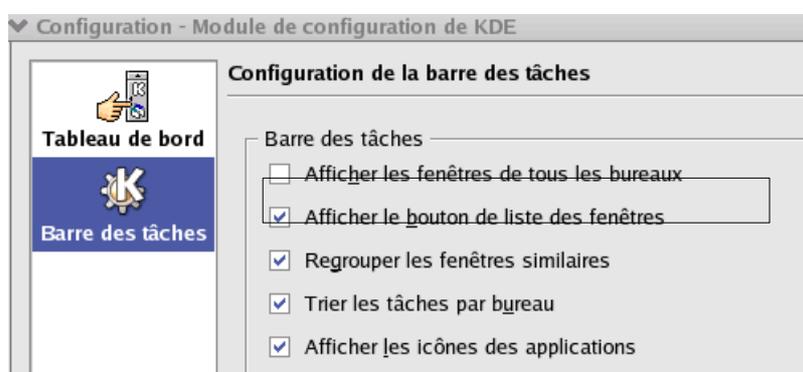
5 Introduction à l'installation de services

5.1 Mode opératoire

L'interface KDE offre la possibilité de travailler avec quatre bureaux qui peuvent être considérés comme quatre écrans différents. Les quatre bureaux sont présents dans la barre des tâches :



Il suffit de cliquer sur l'un des quatre bureaux pour y travailler. Par défaut, c'est le bureau n° 1 qui est actif. Par défaut toujours, les applications lancées dans les différents bureaux sont affichées dans la barre des tâches. Il est alors difficile de savoir à quel bureau appartient telle tâche. Nous allons configurer la barre des tâches pour qu'elle n'affiche que les tâches du bureau actif à un moment donné. Cliquez droit sur la barre des tâches. Un menu contextuel s'affiche. Choisissez l'option [Configuration] :



Faites en sorte que l'option [Afficher les fenêtres de tous les bureaux] soit désactivée comme ci-dessus. Validez cette configuration par [OK]. Dans la suite de ce document, nous utiliserons les bureaux suivants :

- bureau 1, propriété de l'utilisateur **uti1**
- bureau 2, propriété de l'utilisateur uti1 mais délégué à **uti2**
- bureau 4, propriété de l'utilisateur uti1 mais délégué à **root**

Par la suite, nous les appellerons bureau1(uti1), bureau2(uti2) et bureau4(root) pour rappeler sous quelle identité vous devez oeuvrer dans ces bureaux. Dans chacun d'eux, au moins quatre outils seront utilisés :

- une fenêtre terminal (**Konsole**)
- un éditeur de texte (**Kwrite**)
- un explorateur de fichiers (**Konqueror**)
- un navigateur (**Mozilla**)

Tous ces outils devront être **propriété de l'utilisateur à qui a été dédié le bureau**. La procédure à suivre est la suivante :

- faites en sorte que la connexion à KDE se fasse sous l'identité **uti1**. Si ce n'est pas le cas, quittez KDE avec l'option **Menu KDE/Quitter l'environnement** puis reconnectez-vous sous l'identité **uti1/azerty**.
- dans le bureau 1, lancez les quatre applications nécessaires aux tests. Vous pouvez le faire par les options de menu ou par des commandes Unix. Par exemple, ouvrez une fenêtre [Konsole] puis tapez les commandes suivantes :

```
[uti1@tahe uti1]$ id  
uid=500(uti1) gid=500(groupe1) groupes=500(groupe1)
```

```
[uti1@tahe uti1]$ kwrite &  
[1] 10079
```

```
[uti1@tahe uti1]$ konqueror &  
[2] 10080
```

```
[uti1@tahe uti1]$ mozilla &  
[3] 10084
```

La première commande permet de vérifier votre identité. Les autres lancent successivement les application *Kwrite*, *Konqueror* et *Mozilla* en tâches de fond (présence du signe &).

- dans le bureau 2, on fait de même après avoir pris l'identité **uti2** :

```
[uti1@tahe uti1]$ su - uti2
Password:
```

```
[uti2@tahe uti2]$ id
uid=502(uti2) gid=501(groupe1) groupes=501(groupe1)
```

```
[uti2@tahe uti2]$ kwrite &
[1] 10186
```

```
[uti2@tahe uti2]$ konqueror &
[2] 10199
```

```
[uti2@tahe uti2]$ mozilla &
[3] 10203
```

- dans le bureau 4, on fait de même après avoir pris l'identité **root** :

```
[uti1@tahe uti1]$ su -
Password:
```

```
[root@tahe root]# id
uid=0(root) gid=0(root) groupes=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

```
[root@tahe root]# kwrite &
[1] 10295
```

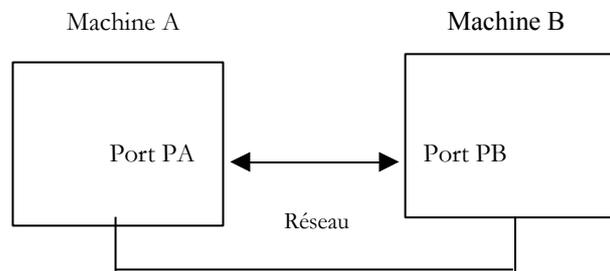
```
[root@tahe root]# konqueror &
[2] 10296
```

```
[root@tahe root]# mozilla &
[3] 10298
```

Dans toute la suite, on suppose que vous travaillez dans la configuration précédente. Si vous utilisez une autre interface que KDE, remplacez les outils KDE précisés ci-dessus, par les outils équivalents de votre interface. L'essentiel des manipulations se fera dans le bureau 4 en tant que **root**. Parfois, il faudra aller dans les bureaux 1 ou 2 pour travailler en mode non super utilisateur.

5.2 Applications client-serveur dans le monde TCP-IP

Considérons la communication entre deux machines distantes A et B :



Lorsque une application *AppA* d'une machine A veut communiquer avec une application *AppB* d'une machine B de l'Internet, elle doit connaître plusieurs choses :

- l'**adresse IP** ou le **nom de la machine B**
- le **numéro du port** avec lequel travaille l'application *AppB*. En effet la machine B peut supporter de nombreuses applications qui travaillent sur l'Internet. Lorsqu'elle reçoit des informations provenant du réseau, elle doit savoir à

quelle application sont destinées ces informations. Les applications de la machine B ont accès au réseau via des guichets appelés également des ports de communication. Cette information est contenue dans le paquet reçu par la machine B afin qu'il soit délivré à la bonne application.

- les **protocoles de communication réseau** compris par la machine B. Ici, il s'agit des protocoles TCP (couche OSI 4) – IP (couche OSI 3).
- le **protocole de dialogue** accepté par l'application *AppB* (couches 5 à 7). En effet, les machines A et B vont se "parler". Ce qu'elles vont dire va être encapsulé dans les protocoles TCP-IP. Néanmoins, lorsqu'au bout de la chaîne, l'application *AppB* va recevoir l'information envoyée par l'application *AppA*, il faut qu'elle soit capable de l'interpréter. Ceci est analogue à la situation où deux personnes A et B communiquent par téléphone : leur dialogue est transporté par le téléphone. La parole va être codée sous forme de signaux par le téléphone A, transportée par des lignes téléphoniques, arriver au téléphone B pour y être décodée. La personne B entend alors des paroles. C'est là qu'intervient la notion de protocole de dialogue : si A parle français et que B ne comprend pas cette langue, A et B ne pourront dialoguer utilement.

Aussi les deux applications communicantes doivent-elles être d'accord sur le type de dialogue qu'elles vont adopter. Par exemple, le dialogue avec un service *ftp* n'est pas le même qu'avec un service *pop* : ces deux services n'acceptent pas les mêmes commandes. Elles ont un protocole de dialogue différent.

Souvent, la communication sur Internet est dissymétrique : la machine A initie une connexion pour demander un service à la machine B : il précise qu'il veut ouvrir une connexion avec le service SB1 de la machine B. Celle-ci accepte ou refuse. Si elle accepte, la machine A peut envoyer ses demandes au service SB1. Celles-ci doivent se conformer au protocole de dialogue compris par le service SB1. Un dialogue demande-réponse s'instaura ainsi entre la machine A qu'on appelle machine **cliente** et la machine B qu'on appelle machine **serveur**. L'un des deux partenaires fermera la connexion.

5.3 Le réseau privé étudiant

Vous installez Linux sur une machine qui appartient au réseau privé étudiant de l'ISTIA. Ce réseau privé a la particularité de ne pouvoir "sortir" directement sur l'Internet. Toute machine de ce réseau doit, pour accéder au réseau public de l'Internet, passer par une machine particulière du réseau appelée "serveur Socks". Ce serveur sert à deux choses :

- il permet à une machine du réseau étudiant d'accéder au réseau public de l'Internet
- il le fait moyennant une authentification de type login/password. Ainsi tout accès à l'Internet est authentifié : les administrateurs du réseau peuvent savoir qui fait quoi. C'est une mesure de sécurité qui permet, lorsqu'une plainte est déposée pour une action délictueuse sur Internet faite à partir d'un poste étudiant de l'université d'Angers de retrouver son auteur.

Pour accéder au serveur Socks, on dispose sous Windows d'un client Socks. C'est lui qui est chargé de "lancer" toutes les applications Internet afin qu'elles passent toutes par le serveur Socks et puissent ainsi fonctionner. Ici, nous n'utiliserons pas un tel client Socks. Nous utiliserons une autre machine passerelle qui relaie les protocoles HTTP (web) et FTP (transfert de fichiers).

Les services réseau que vous installerez seront testés à partir d'un poste se trouvant sur le même réseau privé. Il n'y a donc pas besoin de "sortir" sur l'Internet pour tester les services que vous installerez. Vous pouvez cependant avoir besoin de naviguer sur le web pour consulter des sources d'informations ou télécharger des fichiers pour certaines installations. Pour cela, vous configurerez les applications Konqueror et Mozilla pour qu'elles passent par le proxy HTTP `eproxy.istia.uang`. Cette configuration a déjà été expliquée dans le chapitre précédent et le lecteur est invité à s'y reporter. Avant de continuer, configurez correctement ces deux applications dans les bureaux 1, 2 et 4. A chaque fois, vérifiez que vous pouvez aller sur le web à une URL quelconque.

5.4 Identifier une machine de l'internet

Lors de l'installation de votre serveur vous lui avez donné une adresse IP **172.20.81.X**. Lors des tests vous pourrez désigner votre serveur ou celui de vos voisins par son nom (*XX.istia.uang*) ou son adresse IP (*172.20.81.X*). Voyons comment récupérer ces deux informations. Dans la fenêtre terminal du bureau 4, tapez la commande suivante :

```
[root@localhost root]# ifconfig
....
```

La commande **ifconfig** donne des informations sur la configuration réseau de la machine (faire `man ifconfig`). Notez ci-dessus l'adresse IP de votre poste de la forme `172.20.81.X`. Pour connaître son nom, tapez maintenant la commande suivante :

```
# host 172.20.81.X
```

Il faudra bien sûr remplacer 172.20.81.X par la valeur de votre adresse IP. La commande **host** permet de connaître le nom de machines dont on passe l'adresse IP (faire **man host**). Notez ici, le nom **XX.istia.uang** de votre machine. Dorénavant, pour désigner votre machine, vous pourrez indifféremment utiliser son adresse IP ou son nom.

Vous vous souvenez peut-être qu'à l'installation vous avez désigné un serveur de noms ayant l'adresse IP 172.20.81.5. C'est la machine *vishnou.istia.uang*. Dans les fichiers de configuration de ce serveur de noms, sont associé des noms de machines à des adresses IP. C'est le serveur *vishnou.istia.uang* qui a donné le renseignement que la commande **host** demandait.

Vous pouvez obtenir plus directement le nom de votre machine avec la commande **hostname** :

```
[root@tahe root]# hostname  
tahe.istia.uang
```

Vous utiliserez le nom **localhost** ou l'adresse IP **127.0.0.1** à chaque fois que vous aurez besoin de désigner votre propre machine.

5.5 Les fichiers de configuration réseau du poste Linux

Lors de l'installation de votre machine Linux, vous avez donné quelques indications pour sa configuration réseau. Rappelons-les :

Configuration
du réseau

```
adresse IP : 172.20.81.X  
Masque réseau : 255.255.0.0  
Passerelle par défaut : 172.20.0.254  
Serveur de noms primaire : 172.20.81.5  
-----
```

```
//  
adresse IP : 172.20.81.X
```

Le chapitre des généralités vous a montré que toute machine de l'internet avait une adresse unique appelée adresse IP. Votre machine fera partie du réseau étudiant de l'ISTIA **172.20.81**.

```
Masque réseau : 255.255.0.0
```

Supposons que votre machine veuille communiquer avec la machine d'adresse IP I1.I2.I3.I4. Pour savoir si cette adresse fait partie du même réseau qu'elle, votre machine va faire un "ET logique" de cette adresse avec le masque réseau : I1.I2.I3.I4 & **255.255.0.0**. Sans entrer dans les détails, le résultat de cette opération sera I1.I2.0.0. Votre machine va faire de même avec sa propre adresse IP : 172.20.81.X & **255.255.0.0** ce qui va donner 172.20.0.0. Si on a I1.I2.0.0=172.20.0.0 donc I1=172 et I2=20, alors la machine destinatrice est sur le même réseau que la machine émettrice. Les données lui seront alors directement envoyées. Si on a I1.I2.0.0<>172.20.0.0 alors la machine destinatrice n'est pas sur le même réseau que la machine émettrice. Les données seront alors envoyées à la passerelle (gateway) qui relie le réseau 172.20 à l'internet.

```
Passerelle par défaut : 172.20.0.254
```

L'adresse IP de la passerelle dont on vient de parler.

```
Serveur de noms primaire : 172.20.81.5
```

L'adresse IP du serveur de noms que votre machine pourra interroger. Rappelons que le serveur de noms fait la correspondance adresse IP <--> nom de machine dans les deux sens. Celui-ci est sur le réseau étudiant.

Comment faire si vous vous apercevez que les renseignements que vous avez donnés étaient incorrects ? Ces renseignements ont été enregistrés dans divers fichiers texte où vous pouvez les retrouver et les modifier. Dans une fenêtre terminal du bureau 4, tapez les commandes suivantes :

```
# more /etc/resolv.conf  
// contient l'IP du serveur de noms utilisé par votre machine  
# more /etc/sysconfig/network  
// contient le nom de votre machine
```

```
# more /etc/sysconfig/network-scripts/ifcfg-eth0
// contient l'adresse IP de votre machine
```

Si vous modifiez l'une des valeurs de ces fichiers, il faut redémarrer la machine pour que cette nouvelle configuration réseau soit prise en compte.

5.6 Les services à installer

Pour chaque service seront fournis le nom du service, les fichiers de configuration à créer ou modifier, la mise en route du service et les moyens de le tester. Nous ferons à chaque fois une configurations minimale. Pour des informations complémentaires et pour une optimisation des services, des sources d'informations vous seront conseillées.

Les services sont présentés sous forme de fiches. Il est conseillé de les installer dans l'ordre suivant :

- Les services de connexion à distance (telnet, ssh).
- Les services de courrier (envoi et relève).
- Le service de transfert de fichiers.
- Le service World Wide Web.
- Le service Samba.
- Le service des Quotas.
- La sécurisation des services TCP-IP par le service tcpd.

Rappel : pour l'installation de chaque service il est nécessaire d'être **root** et donc d'opérer dans le **bureau 4**.

5.7 Les tests des services installés

Lorsqu'un service a été installé sur votre serveur, il faudra le tester. Il y a plusieurs méthodes possibles :

- **Tests du service à partir d'une fenêtre terminal locale au serveur**

Soit S le service installé sur votre serveur et C un client local de ce service, local voulant dire présent sur votre serveur. Ainsi le service *telnetd* et son client *telnet*, le service *ftpd* et son client *ftp*. Alors le service S peut en général être testé par la commande : **C localhost**. Ainsi, le service *telnetd* peut être testé par la commande **telnet localhost** et le service *ftpd* par la commande **ftp localhost**.

- **Tests à partir d'un client graphique local au serveur**

Ici on teste le service S avec un client graphique sous KDE et spécifique au service S. Ainsi nous utiliserons l'application Mozilla pour envoyer et lire du courrier.

- **Tests du service à partir d'un terminal Unix distant**

Soit S le service installé sur votre serveur et C un client Unix distant. Ce client atteindra votre service en général par la commande : **C nom_de_votre_serveur** ou **C adresseIP_de_votre_serveur**. On est dans le même cas que le cas 1 où **localhost** est remplacé par le nom de la machine distante abritant le service à tester.

- **Tests à partir d'un client graphique Unix distant**

On retombe dans le cas 2, si ce n'est que le client KDE est sur une machine distante.

- **Tests du service à partir d'un client Windows**

C'est le cas 4, si ce n'est que le client graphique fonctionne sous Windows.

Dans la pratique, on proposera un ou plusieurs de ces tests. Vous pourrez ne faire que certains d'entre-eux dès que les premiers tests montreront clairement que le service que vous avez installé fonctionne correctement.

5.8 Les fichiers de logs

Un fichier de logs est un fichier texte dans lequel un programme écrit des informations permettant à l'administrateur de suivre son bon fonctionnement. Il y a des fichiers de logs d'erreurs. C'est le cas du fichier `/var/log/messages` dans lequel les principaux services Unix écrivent leurs messages d'erreurs. On en trouve également dans le fichier `/var/log/secure`. Après avoir lancé un service et avant même de le tester à partir d'un poste client vous pouvez vérifier ces deux fichiers en faisant :

```
$ tail /var/log/messages ou tail /var/log/secure
```

La commande `tail` (queue en anglais) affiche les 10 dernières lignes d'un fichier. On peut modifier ce point :

```
$ tail -50 /var/log/messages
```

affichera les 50 dernières lignes de `/var/log/messages`. Celles-ci vont défiler sur votre écran de 25 lignes et vous ne verrez que les 25 dernières. Utilisez alors la commande :

```
$ tail -50 /var/log/messages | more
```

qui provoquera un affichage des 50 lignes, page par page. Tapez **[q]** pour stopper la commande **more**. Enfin,

```
$ tail -f /var/log/messages
```

permet de suivre les évolutions du fichier `/var/log/messages` en temps réel. C'est à dire que les lignes qui sont ajoutées au fichier sont visualisées automatiquement par la commande. On a alors intérêt à consacrer une fenêtre terminal rien que pour cette commande. Dans le bureau 4 (root), lancez une nouvelle fenêtre Terminal et tapez la commande suivante :

```
[util@localhost util]$ su -  
Password:
```

```
[root@localhost root]# tail -f /var/log/messages  
Mar 30 11:02:29 localhost syslogd 1.4.1: restart.  
mar 30 11:07:43 localhost su(pam_unix)[1788]: session opened for user root by util(uid=500)
```

Cette fenêtre ne vous servira qu'à cela. Elle vous permettra souvent d'obtenir des informations lorsque l'un de vos services ne fonctionnera pas.

5.9 Les démons

Dans le vocabulaire Unix, on entend souvent parler de démons (*daemons*). Le mot anglais `daemon`, qui est en fait un sigle, a été francisé en démon. Un démon est un programme qui travaille en tâche de fond. Il n'est pas lié à une console, et est donc un peu « invisible » (on le voit quand même avec la commande **ps**). Ce programme fait un certain travail. Les démons sont souvent les programmes qui assurent les services réseau de la machine, mais ils peuvent servir à n'importe quelle autre tâche. Par exemple, **crond** est un démon qui lance automatiquement certains services à certaines heures, la sauvegarde journalière des données utilisateur, par exemple.

5.9.1 Quelques démons

Tout service est fourni par un programme spécialisé (le démon) et est accessible via un programme client. Sous Linux, le programme client du service `telnetd` s'appelle `telnet`. Ne faites pas de confusion entre les deux programmes `telnetd` et `telnet`. Sous Unix, les services s'appellent souvent *Sd* (*d=daemon*) et les clients associés *S*. Voici quelques exemples :

<i>Nom du Service</i>	<i>Programme serveur associé</i>	<i>Programme client associé</i>
telnet	telnetd	telnet
ftp	ftpd	ftp
news	innd	inn
finger	fingerd	finger
Ssh	sshd	ssh
...		

5.9.2 Le super démon réseau xinetd

Une machine Unix peut fournir de nombreux services. (WWW, FTP, DNS, Mail, telnet, etc...). Il en existe des dizaines. Lancer sur un serveur, tous les démons correspondant aux services possibles sur cette machine peut être coûteux en occupation mémoire, surtout lorsque l'on sait que cette machine n'a à offrir ces services que de temps en temps. C'est pour cela qu'existe le super démon **xinetd**. Ce démon prend en charge certains (pas tous) des services que cette machine peut offrir, mais ne fait qu'écouter les ports de ces services. Lorsqu'une requête arrive sur un de ces ports, le super démon active le démon capable de rendre le service demandé. Tous les services réseaux sont décrits dans les fichiers **/etc/services**.

Visualisez le contenu du fichier **/etc/services** par la commande suivante (bureau 4 – terminal - root) :

```
...
echo          7/udp
discard       9/tcp          sink null
discard       9/udp          sink null
systat        11/tcp          users
systat        11/udp          users
daytime       13/tcp
daytime       13/udp
qotd          17/tcp          quote
qotd          17/udp          quote
....
```

La ligne

```
echo          7/udp
```

signifie qu'un service appelé **echo** opère sur le port n° 7 avec le protocole **udp**. Le fichier **/etc/services** n'est qu'un fichier de déclaration des services pouvant être lancés par le super démon réseau. Ce n'est pas pour autant que ces services sont réellement activés.

5.9.3 Configuration du super démon xinetd

Le super démon réseau standard sur les Unix est **inetd**. La configuration de **inetd** (notamment, les démons à lancer en sous main) est dans le fichier **/etc/inetd.conf**. Dans les versions récentes de Linux Redhat, le démon **inetd** a été remplacé par le démon **xinetd** qui amène une meilleure sécurité. Il est configuré à l'aide du fichier **/etc/xinetd.conf** :

```
[root@localhost root]# more /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances          = 60
    log_type            = SYSLOG authpriv
    log_on_success      = HOST PID
    log_on_failure      = HOST
    cps                 = 25 30
}
```

```
includedir /etc/xinetd.d
```

Nous ne nous étendrons pas sur la signification des paramètres ci-dessus. Elle peut être obtenue par **man xinetd.conf**. On remarque que la dernière ligne inclut le dossier **/etc/xinetd.d**. Visualisons le contenu de celui-ci :

```
[root@localhost root]# ls -la /etc/xinetd.d/
-rw-r--r-- 1 root root 297 août 15 2002 chargen
-rw-r--r-- 1 root root 317 août 15 2002 chargen-udp
-rw-r--r-- 1 root root 297 août 15 2002 daytime
....
-rw-r--r-- 1 root root 370 sep 1 2002 imap
-rw-r--r-- 1 root root 365 sep 1 2002 imaps
-rw-r--r-- 1 root root 453 sep 1 2002 ipop2
-rw-r--r-- 1 root root 356 mar 8 14:14 ipop3
-rw-r--r-- 1 root root 335 sep 1 2002 pop3s
-rw-r--r-- 1 root root 361 jun 24 2002 rexec
-rw-r--r-- 1 root root 378 jun 24 2002 rlogin
-rw-r--r-- 1 root root 431 jun 24 2002 rsh
....
```

On y trouve des fichiers qui portent le nom de services déclarés dans **/etc/services**. Examinons le contenu de l'un d'entre eux, le fichier **telnet** :

```
[root@localhost root]# more /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
# unencrypted username/password pairs for authentication.
service telnet
{
    flags          = REUSE
    socket_type    = stream
    wait           = no
    user           = root
    server         = /usr/sbin/in.telnetd
    log_on_failure += USERID
    disable        = yes
}
```

Ce fichier décrit le mode de fonctionnement du démon **telnet**, service qui répond aux demandes de connexion distantes. Notons les points suivants dans la configuration précédente :

- **user=root** : **root** sera le propriétaire du démon **telnet** lorsqu'il aura été lancé
- **server=/usr/sbin/in.telnetd** : le nom du programme qui rend le service **telnet**
- **disable=yes** : le démon **telnet** est inhibé. Si un client demande le service **telnet**, le démon **xinetd** répondra qu'il n'est pas disponible.

Vérifions ce dernier point. Un client pour le démon **telnetd** est le programme **telnet** (faire **man telnet**). Sa syntaxe la plus simple est **telnet machine** pour se connecter à **machine**. Connectons-nous à notre propre machine appelée **localhost** :

```
[root@localhost root]# telnet localhost
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
```

Détaillons ce qui s'est passé. Le démon **telnet** est déclaré dans le fichier **/etc/services** :

```
[root@localhost root]# grep -i telnet /etc/services
telnet      23/tcp
telnet      23/udp
....
```

On voit que le service **telnet** travaille sur le port **23** avec le protocole **udp** ou **tcp**. La commande

```
[root@localhost root]# telnet localhost
```

tente de créer une connexion au port **23** de la machine **localhost**. C'est le super démon **xinetd** qui écoute les demandes arrivant sur tous les ports déclarés dans **/etc/services**. Il répond donc à la demande qui arrive sur le port **23**. Il voit qu'il s'agit d'une demande pour le service **telnet** et consulte alors le fichier **/etc/xinetd.d/telnet**. Là, il voit que le service est désactivé (**disable=yes**). Il refuse donc la connexion.

Avec l'éditeur *knife*, mettez **disable=no** dans le fichier **/etc/xinetd.d/telnet** puis vérifiez :

```
[root@localhost root]# more /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#      unencrypted username/password pairs for authentication.
service telnet
{
    flags          = REUSE
    socket_type    = stream
    wait           = no
    user           = root
    server         = /usr/sbin/in.telnetd
    log_on_failure += USERID
}
disable          = no
}
```

De nouveau, essayez d'utiliser le client **telnet** :

```
[root@localhost root]# telnet localhost
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
```

Rien n'a changé. Pourquoi ? Parce que le démon **xinetd** lit ses fichiers de configuration au démarrage du serveur. Il ne les relit plus ensuite à moins qu'on ne le force à le faire. Ceci peut être obtenu de diverses façons notamment en relançant le service :

```
[root@localhost root]# /etc/rc.d/init.d/xinetd restart
Arrêt de xinetd : [ OK ]
Démarrage de xinetd : [ OK ]
```

Dans le dossier **/etc/rc.d/init.d** (faire **ls -l /etc/rc.d/init.d**), on trouve des scripts permettant de lancer (**start**), arrêter (**stop**) ou relancer (**restart**) des services. **xinetd** ayant été relancé, essayons de nouveau le client **telnet** :

```
[root@localhost root]# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Red Hat Linux release 8.0 (Psyche)
Kernel 2.4.18-14 on an i686
login:
```

Ici, on voit que le service **telnet** est devenu accessible.

5.9.4 Conclusion

On se rappellera donc que si un service est déclaré dans */etc/services* mais que son fichier de configuration est absent de */etc/xinetd.d* ou présent mais désactivé, alors il n'est pas activable par *xinetd*. C'est une amélioration vis à vis du fonctionnement du super-serveur *inetd* où tous les services possibles étaient systématiquement présents dans le fichier */etc/inetd.conf* et donc activables, ceci à l'insu des administrateurs néophytes qui ne maîtrisaient pas bien leur système. Il y avait là un problème de sécurité, car plus il y a de services actifs sur une machine plus il y a une chance de trouver un trou de sécurité sur l'un d'eux.

6 Les services de connexion à distance telnet et ssh

6.1 Fonction

Le service **telnet** est le service qui permet à un utilisateur d'avoir accès à une machine Unix distante en mode terminal. Il est assuré par un démon appelé **telnetd**. Le client associé s'appelle **telnet**. Le service **ssh** a la même fonction que le service **telnet**. La seule différence est qu'il chiffre les communications. Il est assuré par le démon **sshd**. Le client associé s'appelle **ssh**. Il est recommandé d'utiliser le client **ssh** en place et lieu du client **telnet**. Le service **telnet** est désormais inhibé par défaut dans les versions récentes de Linux Redhat.

6.2 Installation du service telnet

Le service *telnet* est déjà installé mais désactivé comme nous l'avons vu dans l'étude du démon *xinetd*. Pour activer le service, il nous a fallu

- modifier le fichier de configuration `/etc/xinetd.d/telnet`
- relancer le démon **xinetd**

Rappelons le contenu du fichier de configuration `/etc/xinetd.d/telnet`

```
[root@localhost util]# more /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#             unencrypted username/password pairs for authentication.
service telnet
{
    flags          = REUSE
    socket_type    = stream
    wait          = no
    user          = root
    server        = /usr/sbin/in.telnetd
    log_on_failure += USERID
    disable       = no
}
```

Le démon **xinetd** a été relancé par la commande :

```
[root@localhost util]# /etc/rc.d/init.d/xinetd restart
Arrêt de xinetd : [ OK ]
Démarrage de xinetd : [ OK ]
```

6.3 Test 1 du service telnet

Nous utilisons le programme client *telnet* pour tester le service *telnetd*. La syntaxe d'appel de *telnet* est la suivante :

telnet machine [port]

avec

machine	nom ou adresse IP de la machine sur laquelle réside le service <i>telnetd</i> à contacter ou un service obéissant au protocole telnet. La machine localhost désigne la machine locale.
port	port sur lequel opère le service demandé (23 pour telnetd - port utilisé par défaut si le port n'est pas précisé)

On teste d'abord le service *telnet* de notre machine. Mettez vous dans le bureau 1(uti1) et dans la fenêtre terminal connectez-vous à localhost sous l'identité **uti1/azerty** :

```
[util@localhost util]$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Red Hat Linux release 8.0 (Psyche)
Kernel 2.4.18-14 on an i686
login: util
Password:
Last login: Sun Mar 30 09:57:57 on :0
```

```
[util@localhost util]$ id
uid=500(util) gid=500(groupe1) groupes=500(groupe1)
```

Vous êtes connecté à votre propre machine sous l'identité *util*. Vous avez établi une nouvelle session de travail sur votre serveur Linux. Vous pouvez taper les commandes habituelles (essayez). Vous quittez la session avec la commande **[exit]** qui vous ramène à la session de travail précédente. Vous obtenez alors l'affichage suivant :

```
Connection closed by foreign host.
[util@localhost util]$
```

6.4 Test 2 du service telnet

Testons maintenant le service *telnetd* d'une autre machine. Demandez à un autre groupe les informations suivantes :

- nom de leur serveur linux (*X.istia.uang*)
- nom d'un de leurs utilisateurs avec le mot de passe associé (*user, passwd*)

Connectez-vous maintenant à leur serveur :

```
# telnet X.istia.uang
```

Si la connexion réussit, cela signifie deux choses :

- le nom de la machine a été reconnu. C'est le serveur de noms qui a fourni son adresse IP.
- le service **telnetd** de la machine contactée est opérationnel.

Donnez le *login/passwd* demandé. Vérifiez que vous pouvez travailler. Quittez la session par la commande **exit**.

6.5 Installation du service ssh

Ssh/sshd sont déjà installés et prêts à fonctionner. SSH ayant été conçu dès le départ avec la sécurité en ligne de mire, il est possible de le configurer de nombreuses façons. Il est malheureusement impossible de détailler ici toutes ses possibilités. Le lecteur, s'il est intéressé aura à coeur de lire la documentation (**man sshd** pour le démon, **man ssh** pour le client). Le service **ssh** est déclaré dans le fichier **/etc/services** (bureau 4 – root – fenêtre terminal) :

```
[root@localhost root]# grep -i ssh /etc/services
ssh          22/tcp      # SSH Remote Login Protocol
ssh          22/udp      # SSH Remote Login Protocol
```

Si on regarde dans le dossier **/etc/xinetd.d**, on ne trouve cependant aucun dossier à son nom :

```
[root@localhost root]# ls -l /etc/xinetd.d
total 92
-rw-r--r--  1 root  root          297 août 15  2002 chargen
-rw-r--r--  1 root  root          317 août 15  2002 chargen-udp
-rw-r--r--  1 root  root          297 août 15  2002 daytime
...
-rw-r--r--  1 root  root          317 août 15  2002 time-udp
-rw-r--r--  1 root  root          275 mar  8 17:17 vsftpd
```

Il se trouve que si *ssh* est bien déclaré dans le fichier **/etc/services**, il n'est cependant pas lancé par le super démon *xinetd*. Il est lancé dès le démarrage du serveur comme le montre la commande suivante qui cherche parmi les processus (**ps**) une ligne qui aurait le texte *ssh* dedans (**grep**) :

```
[root@localhost root]# ps aux | grep -i ssh
root      535  0.0  0.4 3276 1156 ?        S    09:57   0:00 /usr/sbin/sshd
util      790  0.0  0.3 2916  864 ?        S    09:57   0:00 /usr/bin/ssh-agent
```

On voit qu'un programme `/usr/sbin/sshd` est actif. C'est le démon du service `ssh`. On constate ainsi que certains services sont déjà actifs. Comme connaître ceux-ci ? Toujours dans le bureau 4(`root`), lancez l'application **Menu KDE/Paramètres de système/Services**. Seul `root` a accès à cette application. Comme vous travaillez sous l'identité `util`, le mot de passe de `root` (`azerty`) vous sera demandé.

```

 snmptrapd
 spamassassin
 squid
 sshd
 syslog
 telnet

```

Cette application graphique vous montre une liste de services. Il y en a deux sortes :

- ceux lancés par `xinetd`
- les autres

Les services lancés par `xinetd` (comme ci-dessus `telnet`) et qui sont cochés, sont des services actifs, c'est à dire que dans leur fichier de configuration dans `/etc/xinetd.d`, on trouve la ligne `disable=no`. Ceux qui ne sont pas cochés ont dans leur fichier de configuration la ligne `disable=yes`. On peut activer ou désactiver ces démons lancés par `xinetd` en cochant/décochant la case. Cela évite d'aller modifier les fichiers de configuration.

Les services non lancés par `xinetd` (comme ci-dessus `sshd`) et qui sont cochés, sont des services lancés dès le démarrage du serveur. Si on ne souhaitait pas qu'ils soient lancés dès le démarrage, on décocherait la case. En cliquant droit sur le nom du service, on peut lancer, arrêter ou relancer le service. On pourra garder cette fenêtre ouverte dans le bureau 4 pour la suite des installations.

6.6 Test du service ssh

Dans une fenêtre terminal du bureau 1 (`util`), tapez les commandes suivantes :

```
[util@localhost util]$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 77:0d:eb:e4:f9:64:30:13:d6:a9:a4:4d:e3:de:1c:bf.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
util@localhost's password:
```

```
[util@localhost util]$ id
uid=500(util) gid=500(groupe1) groupes=500(groupe1)
```

```
[util@localhost util]$ pwd
/home/util
```

On se connectera en tant que `util/azerty`. Les échanges client-serveur sont cryptés et nécessitent l'installation de clés de cryptage. D'où un premier dialogue sur l'installation de ces clés. Ce dialogue ne se produit qu'une fois. Une fois la connexion établie, on a une session terminal classique dont on sort avec la commande `exit`. Le client `ssh` permet de se connecter sous l'identité que l'on veut. Toujours dans la même fenêtre, quittez la session que vous avez ouverte et reconnectez-vous en tant que `uti2/azerty` :

```
[util@localhost util]$ ssh -l uti2 localhost
uti2@localhost's password:
```

```
[uti2@localhost uti2]$ id
uid=502(uti2) gid=100(users) groupes=100(users),502(groupe2)
```

```
[uti2@localhost uti2]$ pwd
/home/uti2
```

7 Les services de courrier électronique

7.1 Fonction

Le service *SMTP* (Simple Mail Transport Protocol) permet à un utilisateur de votre machine d'envoyer du courrier. Le service *POP* (Post Office Protocol) permet lui la relève de ce courrier.

7.2 Installation et configuration

Le service *SMTP* est déjà installé et pré-configuré. C'est une chance car sa configuration est complexe. Il est lancé dès le démarrage comme le montre la fenêtre des services (bureau 4) :

- rsh
- rsync
- saslauthd
- sendmail
- servers

On peut le constater aussi en regardant la liste des processus actifs (**ps**) et en cherchant parmi ceux-ci le processus **sendmail** :

```
[root@localhost root]# ps aux | grep -i sendmail
root        603  0.0  0.5 5040 1412 ?        S    09:57   0:00 sendmail: accepti
smmsp      613  0.0  0.5 4856 1364 ?        S    09:57   0:00 sendmail: Queue r
```

Le service *POP* est lui aussi installé mais pas forcément activé. Il est lancé par le super démon **xinetd**. Dans la fenêtre des services (bureau 4), cochez la case **ipop3** si ce n'est pas déjà fait :

- imaps
- ipop2
- ipop3
- iptables
- irda

Sauvegardez la nouvelle configuration avec l'option **Fichier/Mettre à jour la liste des services**. Constatez, dans une fenêtre terminal, que le fichier de configuration de **xinetd** pour le service **ipop3** a la ligne **disable=no** :

```
[root@localhost root]# more /etc/xinetd.d/ipop3
# default: off
# description: The POP3 service allows remote users to access their mail \
#               using an POP3 client such as Netscape Communicator, mutt, \
#               or fetchmail.
service pop3
{
    disable = no
    socket_type = stream
    wait = no
    user = root
    server = /usr/sbin/ipop3d
    log_on_success += HOST DURATION
    log_on_failure += HOST
}
```

7.3 Test 1

Pour tester les services *SMTP* et *POP*, nous pouvons dans un premier temps simplement tester leur présence. Pour cela, nous pouvons utiliser le client *telnet* déjà rencontré. La commande **telnet machine port** ouvre une connexion tcp avec le

service qui oeuvre sur le port **port** du serveur **machine**. Si la connexion peut être ouverte, cela montre déjà que le service est présent.

Découvrons sur quels ports officient les services *smtp* et *pop*. Ces renseignements sont donnés dans le fichier `/etc/services`. Dans `bureau4(root)`, fenêtre terminal, tapez les commandes suivantes :

```
[root@tahe root]# grep -i smtp /etc/services
smtp      25/tcp   mail
smtp      25/udp   mail
```

```
[root@tahe root]# grep -i pop /etc/services
...
pop3      110/tcp   pop-3     # POP version 3
...
```

Le service SMTP opère donc sur le port 25 et le service POP sur le port 110. Mettons-nous dans `bureau1(uti1)`, dans une fenêtre terminal et tapons les commandes suivantes :

```
[uti1@tahe uti1]$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tahe.istia.uang ESMTSP Sendmail 8.12.5/8.12.5; Sun, 30 Mar 2003 15:51:05 +0200
```

Le service **smtp** a envoyé un message montrant qu'une connexion avait été ouverte. Le service `smtp` est donc bien présent. A partir de maintenant, nous pourrions dialoguer avec ce service. Il faut pour cela connaître les règles de ce dialogue. Nous n'irons pas jusque là. Nous fermons la connexion en tapant la commande **quit** :

```
quit
221 2.0.0 tahe.istia.uang closing connection
Connection closed by foreign host.
```

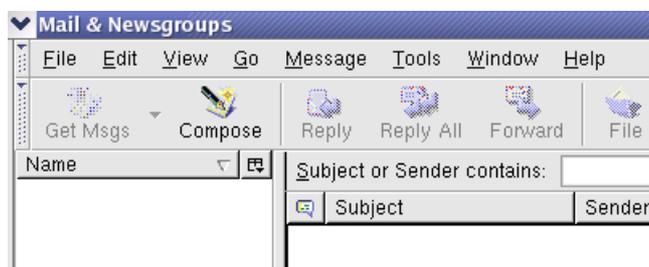
Faisons la même chose pour le service POP, port 110 :

```
[uti1@tahe uti1]$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 tahe.istia.uang v2001.78rh server ready
quit
+OK Sayonara
Connection closed by foreign host.
```

Le service POP est donc bien actif lui aussi.

7.4 Test 2

Nous allons maintenant utiliser l'utilitaire de m^él *Mozilla* pour envoyer et recevoir des courriers. Pour cela nous utiliserons les bureaux 1 et 2 des utilisateurs *uti1* et *uti2*. Allez dans `bureau1(uti1)` et lancez le navigateur Mozilla si ce n'est déjà fait. Utilisez l'option **Window/Mail & Newsgroups** pour avoir accès à l'utilitaire de gestion de courrier de Mozilla.



Prenez ci-dessus, l'option **Edit/Mail & Newsgroups Accounts Settings** pour configurer un compte de courrier. Choisissez l'option [Add Account]. Un assistant démarre. Choisir [Email Account] puis définissez les premières caractéristiques du compte :

Enter the name you would like to appear in the "From" field of your outgoing messages (for example, "John Smith").

Your Name:

Enter your email address. This is the address others will use to send email to you (for example, "user@example.net").

Email Address:

A la place de tahe.istia.uang, mettez le nom de votre machine X.istia.uang. Tout utilisateur U de la machine X.istia.uang a une adresse électronique de la forme U@X.istia.uang. Ensuite définissez les caractéristiques du serveur POP qui va stocker le courrier de l'utilisateur **uti1** :

Select the type of incoming server you are using.

POP IMAP

Enter the name of your incoming server (for example, "mail.example.net").

Incoming Server:

L'utilisateur **uti1** va recevoir son courrier sur votre machine. Le serveur POP est donc votre machine. Utilisez le nom **X.istia.uang** de votre machine.

Enter the user name given to you by your email provider (for example, "jsmith").

User Name:

Ici, il faut mettre le login de l'utilisateur, donc ici **uti1**. L'écran suivant demande un nom pour le compte que vous venez de créer. Vous pouvez laisser celui proposé par défaut ou mettre autre chose.

Enter the name by which you would like to refer to this account (for example, "Work Account", "Home Account" or "News Account").

Account Name:

Le dernier écran de l'assistant donne un récapitulatif des caractéristiques du compte :

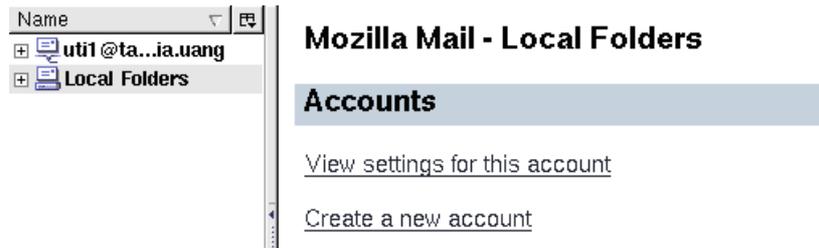
Please verify that the information below is correct.

Account Name:	uti1@tahe.istia.uang
User Name:	uti1
Email Address:	uti1@tahe.istia.uang
Incoming Server Name:	tahe.istia.uang
Incoming Server Type:	POP3
Outgoing Server Name (SMTP):	localhost

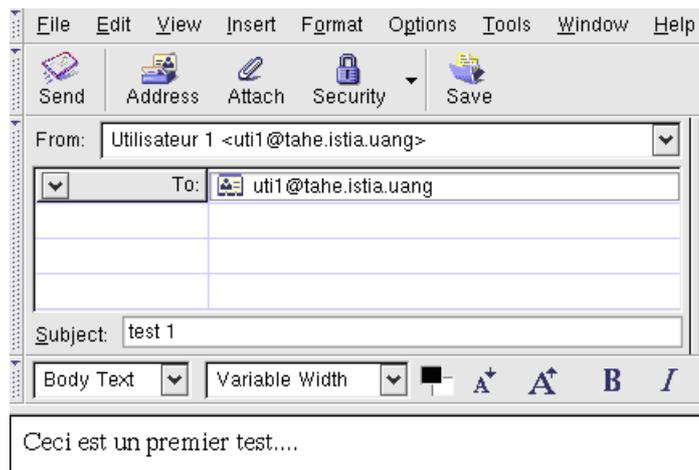
Après validation, l'assistant présente l'écran suivant :



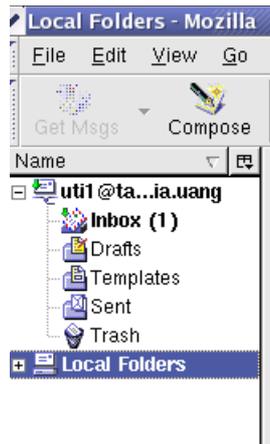
Nous revenons alors dans le gestionnaire de courrier :



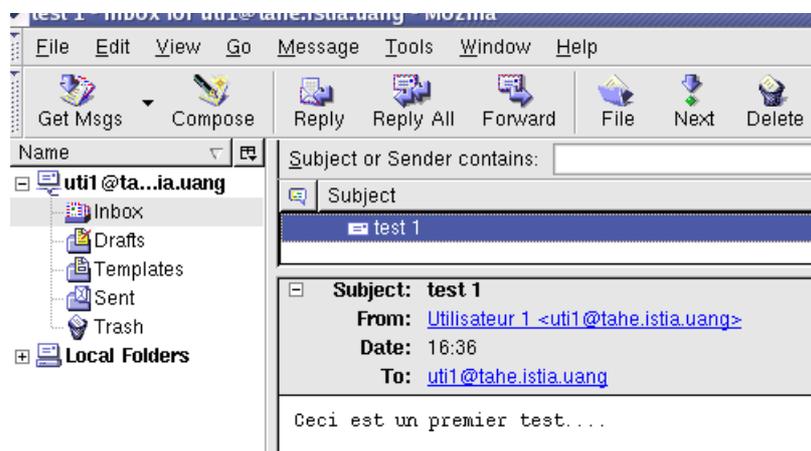
Nous sommes maintenant prêts pour un premier test. L'utilisateur *uti1* va s'envoyer un mél à lui-même (service SMTP) et le lire (service POP). Prenez l'option **Message/New Message** et envoyez un message à uti1@X.istia.uang où X.istia.uang est votre machine.



Une fois le message envoyé, demandez à lire le courrier reçu par l'utilisateur *uti1* en cliquant droit sur le nom du compte uti1@X.istia.uang et prenez l'option **Get messages for account** :



Si tout s'est bien passé, vous devriez avoir un message dans la boîte *Inbox* (cf ci-dessus). Cliquez sur la boîte *Inbox*, le message reçu apparaît alors :



Nous retrouvons bien le message qui a été envoyé, ce qui montre que les services *smtp* (envoi du courrier) et *pop* (relève du courrier) sont fonctionnels.

7.5 Test 3

Dans `bureau2(uti2)`, lancez *mozilla* et configurez un compte courrier pour l'utilisateur *uti2* de la même façon que vous l'avez fait pour l'utilisateur *uti1*. On se rappellera que *mozilla* ne doit pas être lancé directement à partir du menu, mais à partir d'une fenêtre terminal sous l'identité *uti2* :

```
[uti1@tahe uti1]$ su - uti2
Password:
```

```
[uti2@tahe uti2]$ mozilla &
[1] 1688
```

Lorsque le compte de l'utilisateur *uti2* a été configuré :

- revenez dans `bureau1(uti1)` et envoyez un mél à `uti2@X.istia.uang`.
- revenez dans `bureau2(uti2)` et vérifiez qu'il a reçu le mél envoyé

7.6 Utilisation d'alias

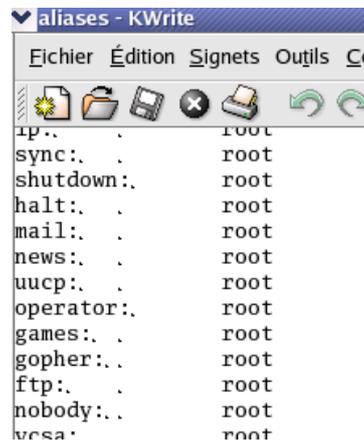
Si l'utilisateur *uti1* de votre machine *X.istia.uang* s'appelle Jean Dupont, il serait agréable qu'on puisse lui écrire à l'adresse `jean.dupont@X.istia.uang` plutôt qu'à `uti1@X.istia.uang`. C'est possible. Tout utilisateur de votre machine peut avoir un ou plusieurs alias, ici `jean.dupont` pour l'utilisateur *uti1*. La définition des alias se fait dans le fichier `/etc/aliases`.

Dans bureau4(root), utilisez l'éditeur *kwrite* pour éditer le contenu du fichier `/etc/aliases`. L'éditeur *kwrite* doit être lancé à partir de la fenêtre terminal sous l'identité *root* :

```
[util@tahe util]$ su - root
Password:
```

```
[root@tahe root]# kwrite &
[1] 1762
```

Le contenu du fichier `/etc/aliases` ressemble à ce qui suit :



Les lignes de ce fichier ont la forme `alias:[tabulation]login`. Créons donc la ligne suivante :

jean.dupont: util

qui associe l'alias *jean.dupont* à l'utilisateur *util*. On n'oubliera pas la tabulation séparant l'alias du login.

```
# Person who should get root's mail
#root:. . . marc
# mailman aliases
mailman: postmaster
mailman-owner: mailman

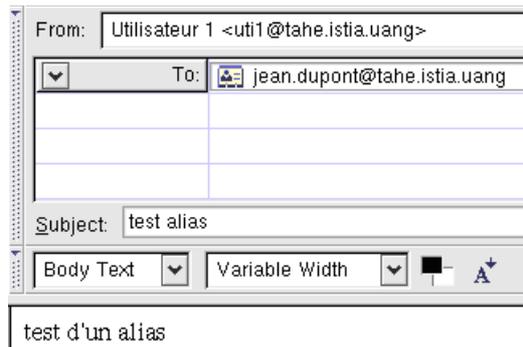
# ajouts
jean.dupont:. . . util
```

On pourrait créer autant d'alias que l'on veut pour l'utilisateur *util* ou pour d'autres utilisateurs en ajoutant d'autres lignes. Sauvegardez le fichier. Le fichier `/etc/aliases` est en fait un fichier texte utilisé pour générer la véritable base des alias utilisés par le service *sendmail*. Cette base d'alias est générée par la commande suivante (bureau4(root) – terminal) :

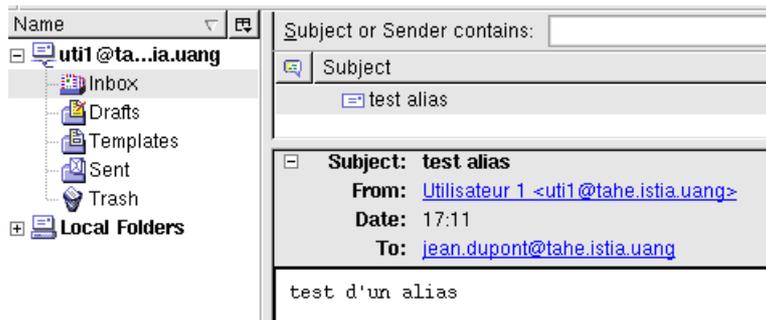
```
[root@tahe root]# newaliases
/etc/aliases: 65 aliases, longest 10 bytes, 651 bytes total
```

Une fois cette commande émise :

- revenez dans bureau1(util) et avec mozilla envoyez un courrier à `jean.dupont@X.istia.uang`



- relevez le courrier de **uti1** et vérifiez qu'il a reçu le courrier envoyé à **jean.dupont**



7.7 Gérer le relayage de courrier

Votre serveur peut désormais servir de serveur d'envoi de courrier (smtp) pour les utilisateurs se trouvant sur votre machine. Il peut également servir de serveur SMTP théoriquement pour n'importe qui sur l'Internet. Par exemple quelqu'un utilisant un utilitaire de courrier à Paris peut indiquer à son logiciel que le serveur SMTP que celui-ci doit utiliser est à Lyon. Lorsque l'utilisateur envoie un message, l'utilitaire de courrier l'envoie en réalité au serveur SMTP qui lui a été désigné et c'est ce serveur SMTP qui sera chargé d'acheminer le courrier à son destinataire. On appelle cela du relayage de courrier : un serveur SMTP sert de relais à des utilisateurs se trouvant ailleurs que sur le serveur lui-même. Cela pose des problèmes de sécurité et les dernières versions du programme *sendmail* inhibent par défaut le relayage de courrier. Ainsi faites le test suivant :

Ouvrez une fenêtre terminal (peu importe le propriétaire du terminal) et tapez les commandes suivantes :

```
[root@tahe util]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tahe.istia.uang ESMTP Sendmail 8.12.8/8.12.8; Thu, 5 Jun 2003 17:20:46 +0200
quit
221 2.0.0 tahe.istia.uang closing connection
Connection closed by foreign host.
```

Le code ci-dessus montre que le service smtp écoute sur le port 25 de la machine localhost. Maintenant refaites la même chose, en remplaçant **localhost** par le nom internet de votre machine :

```
[root@tahe util]# hostname
tahe.istia.uang
[root@tahe util]# telnet tahe.istia.uang 25
Trying 172.19.81.121...
telnet: connect to address 172.19.81.121: Connection refused
```

De façon assez étonnante, le service smtp ne semble plus présent sur le port 25 de la machine si on donne à celle-ci son nom internet plutôt que localhost. Il faut comprendre que les services réseau écoutent en général sur toutes les interfaces réseau présentes sur la machine et de plus sur une interface particulière appelée **loopback** d'adresse IP **127.0.0.1**. On peut obtenir la liste des interfaces réseau actives avec la commande **ifconfig** :

```
[root@tahe util]# ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:01:02:18:C5:43
```

```

    inet adr:172.19.81.121 Bcast:172.19.255.255 Masque:255.255.0.0
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:50876 errors:0 dropped:0 overruns:309 frame:0
    TX packets:2187 errors:0 dropped:0 overruns:0 carrier:0
    collisions:234 lg file transmission:100
    RX bytes:6815391 (6.4 Mb) TX bytes:277712 (271.2 Kb)
    Interruption:10 Adresse de base:0xec00

lo    Lien encap:Boucle locale
    inet adr:127.0.0.1 Masque:255.0.0.0
    UP LOOPBACK RUNNING MTU:16436 Metric:1
    RX packets:119 errors:0 dropped:0 overruns:0 frame:0
    TX packets:119 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 lg file transmission:0
    RX bytes:10954 (10.6 Kb) TX bytes:10954 (10.6 Kb)

```

Les lignes ci-dessus présentent les caractéristiques de deux interfaces réseau :

- lo (loopback) qui est l'interface de la machine vers elle-même. Elle ne correspond pas physiquement à une carte réseau. Grâce à cette interface, on peut utiliser des applications réseau locales sans carte réseau.
- eth0 (carte ethernet n° 0) est l'interface correspondant à une carte réseau ethernet.

Un service réseau écoute habituellement sur toutes les interfaces affichées par la commande **ifconfig**. Ce n'est pas le cas du service smtp livré avec la RH8. Cette option est contrôlé par le fichier **/etc/mail/sendmail.mc**. Ce fichier permet de configurer le service **sendmail**. Le véritable fichier de configuration de sendmail est le fichier **/etc/mail/sendmail.cf**. Ce dernier est très complexe. Le fichier *sendmail.mc* permet lui une configuration plus simple de *sendmail*. Un utilitaire permet de générer le fichier *sendmail.cf* à partir du contenu du fichier *sendmail.mc*. Ceci nous est indiqué par les premières lignes du fichier *sendmail.mc* :

```

[root@tahe util]# more /etc/mail/sendmail.mc
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #

```

On voit ci-dessus qu'on a besoin d'un paquetage appelé **sendmail-cf** pour générer le fichier de configuration de *sendmail*. Le fichier *sendmail.mc* quoique technique reste cependant lisible. Lorsqu'on le parcourt, on découvre la séquence suivante de lignes :

```

dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
dnl #

```

Les lignes de commentaires indiquent clairement que la ligne **DAEMON_OPTIONS** force *sendmail* à n'écouter que l'adresse IP 127.0.0.1 c'est à dire l'interface loopback. Ainsi sendmail n'écoute pas la carte réseau. Comme suggéré par les commentaires, nous mettons en commentaires la ligne **DAEMON_OPTIONS** :

```

dnl #
dnl # DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
dnl #

```

Après avoir sauvegardé le fichier **sendmail.mc**, vérifions la présence du paquetage **sendmail-cf** :

```

[root@tahe util]# rpm -qa | grep -i sendmail
sendmail-8.12.8-5.80
sendmail-cf-8.12.8-5.80

```

Ci-dessus, le paquetage est bien présent mais bien souvent ce n'est pas le cas. Cela dépend du choix de paquetages fait lors de l'installation initiale. Nous décrivons comment installer le paquetage :

- prenez l'option de menu KDE/Paramètres de Système/Paquetages :



- prendre l'option Détails ci-dessus pour le [Serveur de courrier]



- cochez comme ci-dessus le paquetage sendmail-cf et validez
- choisissez l'option [Mise à jour] pour installer le ou les nouveaux paquetages.

Ceci fait, vérifiez la présence du paquetage **sendmail-cf** :

```
[root@tahe util]# rpm -qa | grep -i sendmail-cf
sendmail-cf-8.12.8-5.80
```

Maintenant nous sommes prêts à générer le nouveau fichier de configuration **sendmail.cf**. Nous le faisons comme il était indiqué dans les commentaires du fichier **sendmail.mc** :

```
[root@tahe util]# make -C /etc/mail
make: Entre dans le répertoire `/etc/mail'
make: Quitte le répertoire `/etc/mail'
[root@tahe util]# ls -l /etc/mail/sendmail*
-rw-r--r-- 1 root root 57359 jun 5 17:58 /etc/mail/sendmail.cf
-rw-r--r-- 1 root root 57422 avr 8 09:03 /etc/mail/sendmail.cf.bak
-rw-r--r-- 1 root root 5802 jun 5 17:42 /etc/mail/sendmail.mc
```

Un nouveau fichier **sendmail.cf** a été créé, l'ancien ayant été renommé **.bak**. Maintenant, nous devons demander au service smtp de relire son fichier de configuration. Nous le faisons en relançant le service :

```
[root@tahe util]# /etc/rc.d/init.d/sendmail restart
Arrêt de sendmail : [ OK ]
Arrêt de sm-client : [ OK ]
Démarrage de sendmail : [ OK ]
Démarrage de sm-client : [ OK ]
```

Maintenant essayons l'opération qui avait échoué avec la configuration précédente :

```
[root@tahe util]# telnet tahe.istia.uang 25
Trying 172.19.81.121...
Connected to tahe.istia.uang.
Escape character is '^'.
220 tahe.istia.uang ESMTP Sendmail 8.12.8/8.12.8; Thu, 5 Jun 2003 18:03:54 +0200
quit
221 2.0.0 tahe.istia.uang closing connection
Connection closed by foreign host.
```

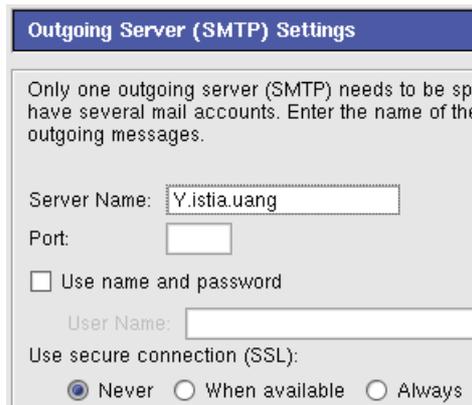
Cette fois-ci, on voit que le service smtp a bien répondu sur le port 25 de l'interface eth0 d'IP 172.19.81.121 ce qu'il ne faisait pas auparavant. Nous sommes maintenant prêts à faire du relayage de courrier. Pour mettre en évidence cette notion de relayage, faites le test suivant :

Dans le bureau 1 de **util**, configurez l'utilitaire de courrier de la façon suivante :

- prenez l'option **Edit/Mail & Newsgroups Accounts Settings**



- prenez ci-dessus l'option **Outgoing Server (SMTP)**



où Y.istia.uang est une autre machine du réseau (demandez à un autre groupe, le nom de sa machine).

- essayez maintenant d'envoyer un courrier à uti1@X.istia.uang comme vous savez le faire. Cela devrait échouer : pour envoyer votre courrier, Mozilla va s'adresser à la machine Y.istia.uang en lui demandant de relayer ce courrier. Par défaut, la machine Y.istia.uang est configurée pour refuser tout relaying de courrier et devrait donc refuser. Mozilla devrait signaler l'erreur.

Le relaying est configuré dans le fichier **/etc/mail/access**.

```
[root@tahe root]# more /etc/mail/access
# Check the /usr/share/doc/sendmail/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/share/doc/sendmail/README.cf is part of the sendmail-doc
# package.
#
# by default we allow relaying from localhost...
tahe.istia.uang      RELAY
localhost            RELAY
127.0.0.1           RELAY
```

On y trouve des lignes de la forme :

```
domaine1    RELAY
domaine2    RELAY
...
```

Chaque ligne indique pour quel domaine de l'internet, votre serveur accepte de faire du relaying de courrier. Dans l'exemple ci-dessus, seule les demandes provenant de la machine locale (localhost) ou de la machine (tahe.istia.uang) sont acceptées.

Pour étendre le relaying à toute demande provenant du domaine **istia.uang**, ajoutez la ligne suivante au fichier **/etc/mail/access** (bureau4(root) – kwrite) :

```
istia.uang    RELAY
```

```
[root@tahe root]# more /etc/mail/access
tahe.istia.uang      RELAY
localhost           RELAY
127.0.0.1           RELAY
istia.uang          RELAY
```

Relancer le service *smtp* afin qu'il prenne en compte sa nouvelle configuration :

```
[root@tahe root]# /etc/rc.d/init.d/sendmail restart
Arrêt de sendmail : [ OK ]
Arrêt de sm-client : [ OK ]
Démarrage de sendmail : [ OK ]
Démarrage de sm-client : [ OK ]
```

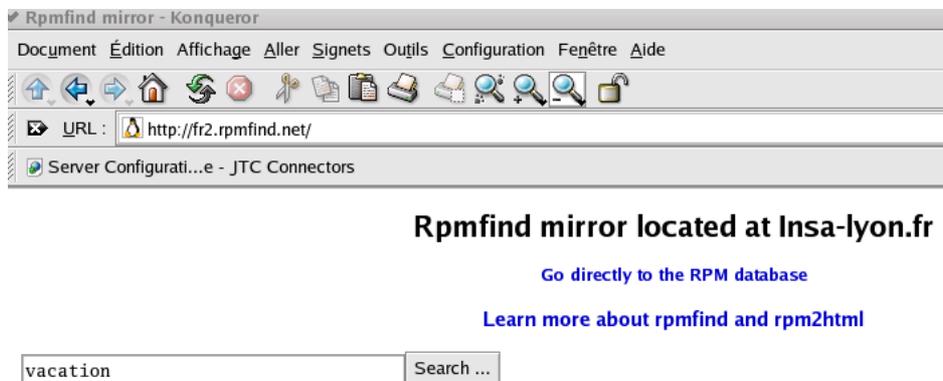
Maintenant demandez à un autre groupe Y de faire le test suivant à partir de sa machine :

- le groupe Y doit configurer le serveur SMTP d'un de ses utilisateurs pour que ce serveur soit votre machine X.istia.uang
- ceci fait, le groupe Y utilise Mozilla ainsi configuré pour envoyer un mél à l'utilisateur *uti1* de votre machine (*uti1@X.istia.uang*)
- vérifiez que votre utilisateur *uti1* a bien reçu le mél envoyé

Si cela a fonctionné, c'est que votre machine X.istia.uang a accepté de relayer une demande d'envoi de courrier d'une autre machine Y.istia.uang. Ainsi à l'ISTIA, la machine istia.univ-angers.fr a été configurée pour relayer toute demande d'envoi de courrier provenant d'une machine X@istia.univ-angers.fr mais n'accepte aucune autre machine.

7.8 Créer un répondeur

Lorsqu'on s'absente de son travail et qu'on n'a plus accès à son mél, on voudrait faire savoir à tous ceux qui vont nous écrire qu'on est absent afin qu'ils ne s'étonnent pas de notre silence. C'est possible en utilisant un programme qui joue le rôle d'un répondeur. Nous utiliserons ici un programme appelé **vacation**. Il n'est pas dans la distribution RH8 aussi allons-nous le chercher sur Internet. Avec **Konqueror**, demandez l'URL fr2.rpmfind.net site où l'on trouve des paquets pour différentes versions de linux :



Dans la zone de saisie, on tape **vacation** et on lance la recherche qui ramène les résultats suivants :

Found 52 RPM for vacation

Package	Summary	Distribution	Download
vacation-1.2.7.rc1-1.2.1.src.html	Automatic mail answering program for Linux	EnGarde	vacation-1.2.7.rc1-1.2.1.src.rpm
vacation-1.2.7.rc1-1.src.html	Automatic mail answering program for Linux	Red Hat Contribs at Bero.Org	vacation-1.2.7.rc1-1.src.rpm
vacation-1.2.7.rc1-1.i386.html	Automatic mail answering program for Linux	Red Hat Contribs at Bero.Org	vacation-1.2.7.rc1-1.i386.rpm
vacation-1.2.6.1-3.ppc.html	Automatic mail answering program for Linux	Polish(ed) Linux Distribution	vacation-1.2.6.1-3.ppc.rpm

Les fichiers contenant le mot clé **src** sont des fichiers source à compiler. Les autres contiennent les binaires des exécutable et les fichiers qui les accompagnent. Cliquons ici sur le lien [vacation-1.2.7.rc1-1.i386.rpm](#) :



Konqueror détecte que le fichier à télécharger est un paquetage rpm. Il demande alors si on veut simplement le télécharger [Enregistrer sur disque] ou bien l'installer [Ouvrir]. Nous choisissons cette dernière option et le gestionnaire de paquets va installer ce nouveau paquetage. On peut vérifier son installation lorsque tout est fini :

```
[root@tahe util]# rpm -qa | grep -i vacation
vacation-1.2.7.rc1-1
```

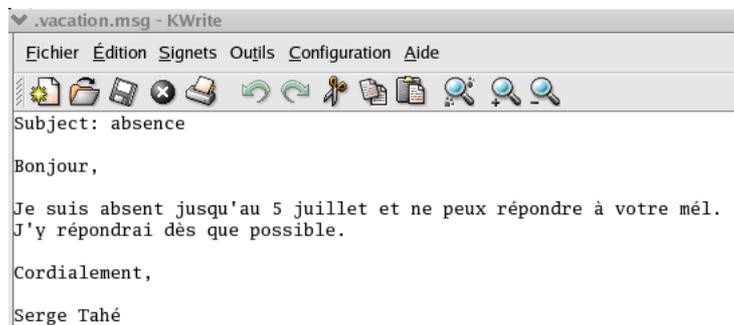
On pourra faire un [man vacation] pour découvrir la façon d'utiliser ce programme :

```
vacation(1)                                User Manuals                                vacation(1)
NAME
    vacation - reply to mail automatically
SYNOPSIS
    vacation [ -I | -i ]
    vacation [ -j ] [ -a alias ] [ -tN ] [ -r ] [ -? ] username
...
```

Mettez-vous dans bureau 1 pour créer un répondeur à **util**. Commençons par créer le texte de la réponse automatique :

```
[util@tahe util]$ export EDITOR=kwrite
[util@tahe util]$ vacation
```

La 1ère ligne fixe un éditeur convivial pour écrire le texte de la réponse automatique. Si ceci n'est pas fait, un éditeur par défaut est utilisé, généralement l'éditeur **vi**. La seconde ligne lance le programme *vacation* qui va automatiquement lancer l'éditeur. On tape alors le texte de la réponse :



Après avoir quitté l'éditeur, le programme *vacation* crée et met à jour divers fichiers. Le texte de la réponse automatique a été placé dans le fichier **.vacation.msg** du répertoire de **util** :

```
[util@tahe util]$ ls -l .vacation.msg
-rw-r--r--  1 util  groupe1  160 jun  7 09:34 .vacation.msg

[util@tahe util]$ more .vacation.msg
Subject: absence

Bonjour,

Je suis absent jusqu'au 5 juillet et ne peux répondre à votre mél.
J'y répondrai dès que possible.

Cordialement,

Serge Tahé
```

Le fichier **.forward** de l'utilisateur **util** a été modifié ou créé :

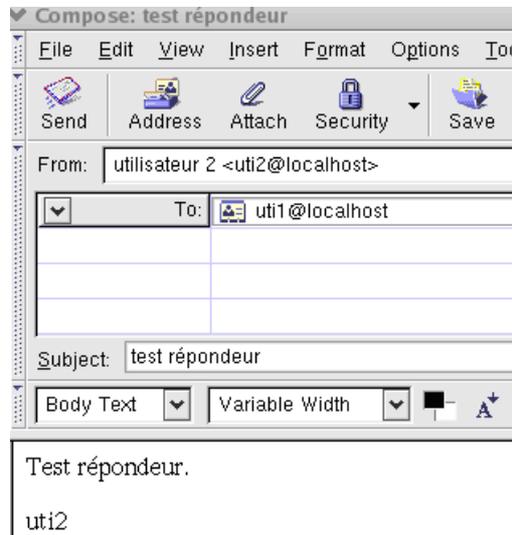
```
[util@tahe util]$ ls -l .forward*
```

```

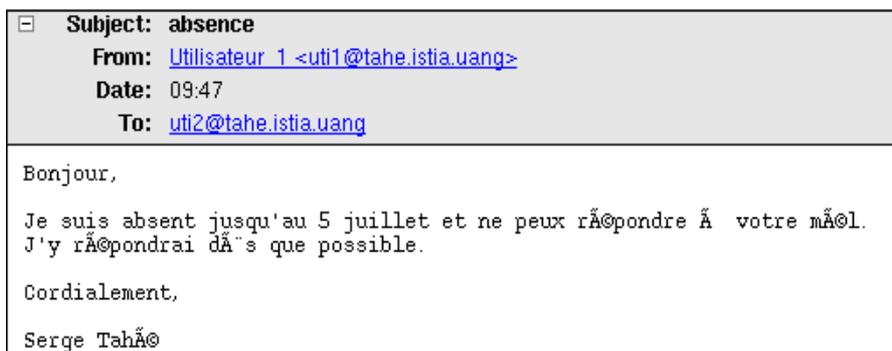
-rw----- 1 uti1  groupe1  33 jun  7 09:35 .forward
-rw----- 1 uti1  groupe1  33 jun  7 09:35 .forward.old
[uti1@tahe uti1]$ more .forward
\uti1, "|/usr/bin/vacation uti1"

```

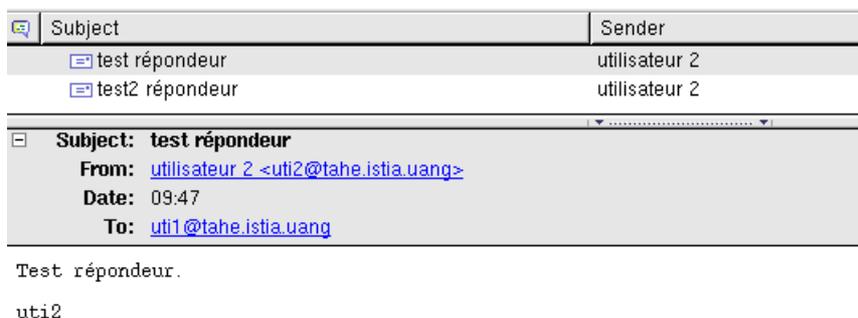
L'ancien fichier `.forward` a été sauvegardé dans `.forward.old`. La ligne placée dans `.forward` redirige tout courrier reçu par `uti1` vers l'utilisateur `uti1` (`\uti1`) et vers le programme `/usr/bin/vacation` (`|/usr/bin/vacation uti1`). Celui-ci va alors générer la réponse automatique. Testons. Nous nous plaçons dans le bureau 2 de `uti2` et celui-ci va envoyer un message à `uti1` :



On envoie le message. Ceci fait, on doit recevoir tout de suite la réponse automatique de `uti1`. Vérifions le courrier reçu :



Si on met de côté le problème des caractères accentués, on peut voir que le répondeur a bien fonctionné. Si maintenant `uti2` envoie un second message à `uti1`, il ne reçoit plus la réponse automatique. Le programme `vacation` n'envoie pas deux fois le message du répondeur à la même personne. Vérifions maintenant ce qu'a reçu l'utilisateur `uti1`. Placez-vous dans le bureau 1 de `uti1` et lancez mozilla pour lire le courrier de `uti1` :



L'utilisateur `uti1` a bien reçu les différents messages de `uti2`.

Pour arrêter le répondeur, l'utilisateur *util* doit supprimer le fichier *forward* ou le remplacer par l'ancien (*forward.old*). Lorsqu'il voudra remettre en route le répondeur, il reprendra la procédure telle qu'elle a été décrite précédemment. On évitera de modifier le fichier *vacation.msg* directement avec un éditeur de texte sans passer par le programme *vacation* car celui-ci ne fait pas que gérer le texte de la réponse. Il gère aussi une base de données **.vacation.db** dans laquelle il met les adresses auxquelles il a déjà envoyé la réponse automatique. Celles-ci ne reçoivent alors pas le message une seconde fois. Il faut donc remettre ce fichier à zéro à chaque nouvelle absence, ce que fait le programme *vacation* à chaque fois qu'il est utilisé.

```
[util@tahe util]$ ls -l .vac*
-rw-r--r--  1 util  groupe1  12312 jun  7 09:47 .vacation.db
-rw-r--r--  1 util  groupe1    160 jun  7 09:34 .vacation.msg
-rw-r--r--  1 util  groupe1    160 jun  7 09:35 .vacation.old
```

8 Le transfert de fichiers

8.1 Fonction

Le service *ftp* de transfert de fichiers permet aux utilisateurs d'autres machines d'avoir accès :

- à leur répertoire s'ils sont des utilisateurs enregistrés sur votre serveur : l'accès est alors contrôlé par mot de passe
- à des fichiers mis à disposition du public dans une arborescence précise sinon : l'utilisateur se déclare **anonymous** et aucun mot de passe n'est demandé. On parle alors de **ftp anonyme**.

Ce service est appelé service **FTP** : **F**ile **T**ransfer **P**rotocol.

8.2 Installation et configuration

Comme le mail, le service ftp est configuré automatiquement à l'installation. Il est lancé par le super démon *xinetd* et est inhibé par défaut. Il faut donc l'activer. Pour cela, dans le bureau 4 (root), on utilise le gestionnaire de services (**Menu KDE/Paramètres de système/Services**). On coche alors si besoin est le service de transfert de fichiers appelé ci-dessous *vsftpd* :

- time-udp
- tux
- vsftpd
- winbind

Ceci fait, on sauvegarde la configuration : **Fichier/Mettre à jour la liste des services**. On peut avoir des informations sur le serveur *vsftpd* par la commande : **man vsftpd**. Le fonctionnement de ce service est contrôlé par le fichier de configuration **/etc/vsftpd.conf** :

```
[root@tahe root]# more /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#.....
```

Nous acceptons pour l'instant tel quel ce fichier de configuration.

8.3 Test 1

Nous utilisons tout d'abord le service *telnet* pour vérifier la présence du service *ftp*. Dans la fenêtre terminal de bureau4 (root) :

```
[root@tahe root]# grep -i ftp /etc/services
ftp          21/tcp
ftp          21/udp      fsp fspd
```

On voit que le service *ftp* opère sur le port 21. Connectons-nous à ce port :

```
[root@tahe root]# telnet localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 ready, dude (vsFTPD 1.1.0: beat me, break me)
quit
```

```
221 Goodbye.  
Connection closed by foreign host.
```

La connexion au port 21 s'est faite correctement. Nous l'avons fermée avec la commande **quit**.

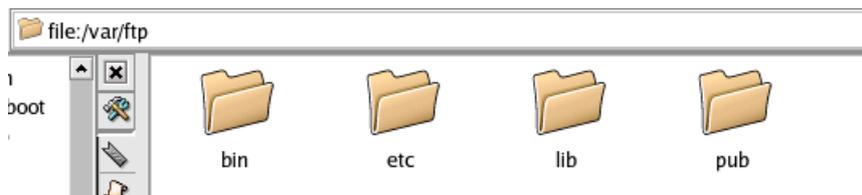
8.4 Test 2

Dans bureau4(root), lancez si besoin est, l'explorateur *Konqueror* :

```
[util@tahe util]$ su - root  
Password:
```

```
[root@tahe root]# konqueror &  
[1] 2552
```

Affichez le contenu du répertoire `/var/ftp` :

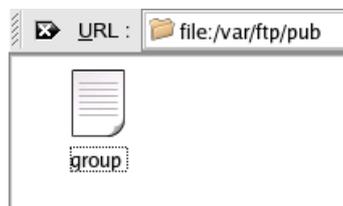


Le dossier **pub** est le dossier dans lequel l'administrateur du service ftp met les fichiers qu'il veut mettre à la disposition de tous. Les dossiers *bin*, *etc* et *lib* sont là pour le bon fonctionnement du serveur ftp.

Placez un fichier dans le répertoire `/var/ftp/pub` (bureau4(root) – terminal)

```
[root@tahe root]# cp /etc/group /var/ftp/pub/  
[root@tahe root]# ls -l /var/ftp/pub/  
total 4  
-rw-r--r--  1 root    ftp          720 mar 30 18:00 group
```

Vérifiez avec *konqueror* :



Maintenant, passez dans le bureau1(uti1) et lancez *Konqueror* si besoin est. Connectez-vous à l'URL <ftp://localhost/pub> :



On a bien accès au fichier `group`. Avec *Konqueror*, vous pouvez alors le copier quelque part dans le dossier de *uti1* (`/home/uti1`). Faites-le.

8.5 Test 3

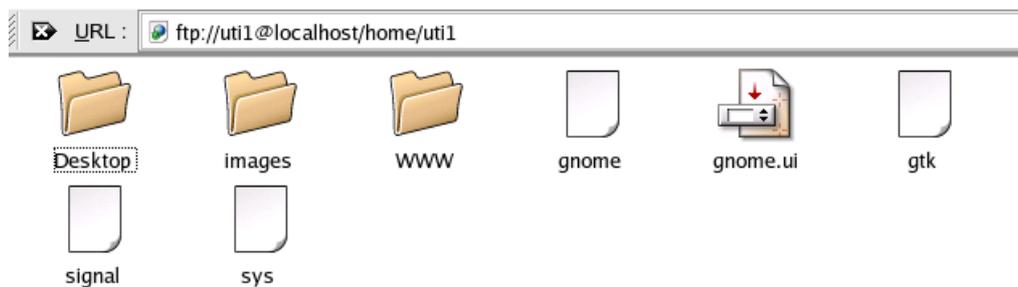
Demandez à un autre groupe (Y) d'atteindre votre machine par ftp. Pour cela, le groupe Y doit avec *Konqueror*, demander l'URL `ftp://X.istia.uang` où `X.istia.uang` est le nom de votre machine.

8.6 Test 4

Dans bureau1(uti1), avec Konqueror demandez l'URL ftp://uti1@localhost. Un mot de passe vous est alors demandé :



Tapez le mot de passe (azerty). Vous avez alors accès au répertoire personnel de l'utilisateur **uti1** :



8.7 Test 5

Faites la même chose que précédemment à partir d'une autre machine linux du réseau.

8.8 Commentaires

Cette section montre plusieurs choses :

- le service FTP de votre machine fonctionnant, vous pouvez mettre des fichiers à disposition du public (ftp anonyme). Il suffit de les placer dans le répertoire `/var/ftp/pub`. En général, on créera différents répertoires sous `pub`. Ces fichiers et répertoires doivent être propriété de l'utilisateur `root` et du groupe `ftp`.
- L'accès `ftp anonyme` place le client dans le répertoire `/var/ftp` qui contient 4 répertoires : `etc`, `bin`, `lib`, `pub`. Aucun de ces répertoires ne doit être en écriture pour tous, sinon n'importe qui pourra y stocker des fichiers.
- Même si un utilisateur de votre machine n'a pas d'accès telnet, il peut quand même stocker des fichiers sur son répertoire Unix en y faisant des transferts par ftp. Cela peut être considéré comme désirable ou indésirable selon les cas :

désirable

Si vous désirez procurer à vos utilisateurs un espace disque de stockage protégé par leur mot de passe. Il vous faut alors contrôler la taille de ces répertoires pour éviter la saturation du disque. Le service des Quotas permet d'assurer ce contrôle.

indésirable

Si vous n'avez pas d'espace disque à offrir à vos utilisateurs. Vous pouvez alors soit supprimer leur répertoire soit le mettre en lecture seulement.

9 Le service World Wide Web (WWW)

9.1 Fonction

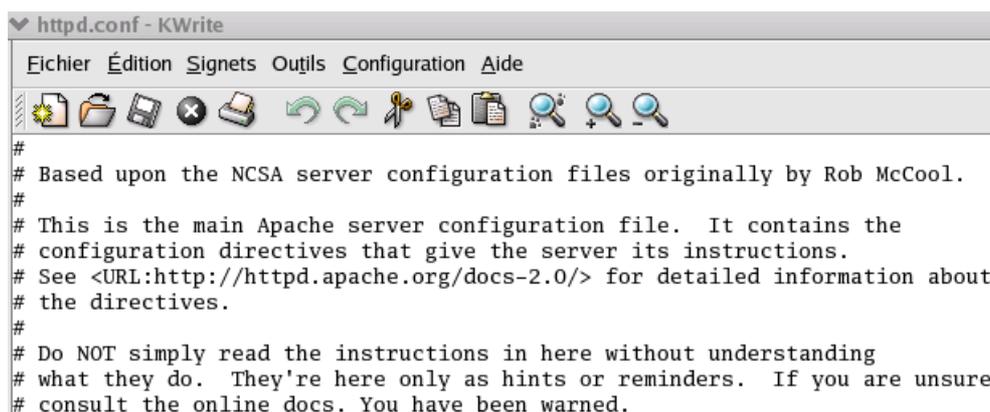
Le service WWW est celui qui permet à un serveur de “publier” des pages Web sur l’Internet. Ces pages sont accessibles via des clients WWW. Ce service est également appelé service **http** (**H**yper**T**ext **T**ransfer **P**rotocol). Sur des postes Windows, on trouve couramment les clients Netscape ou Internet Explorer. Sur les postes Linux Redhat, Netscape est installé par défaut.

9.2 Installation

Pour visualiser des pages Web sur un site, il est nécessaire d’avoir un serveur HTTP. Celui utilisé sur votre serveur Linux est un serveur appelé APACHE (<http://www.apache.org>).

9.3 Le fichier de configuration httpd.conf

Le fichier de configuration du serveur web Apache est le fichier `/etc/httpd/conf/httpd.conf`. Sous l’identité *root* (*bureau4*), prenez un éditeur de texte pour éditer ce fichier :

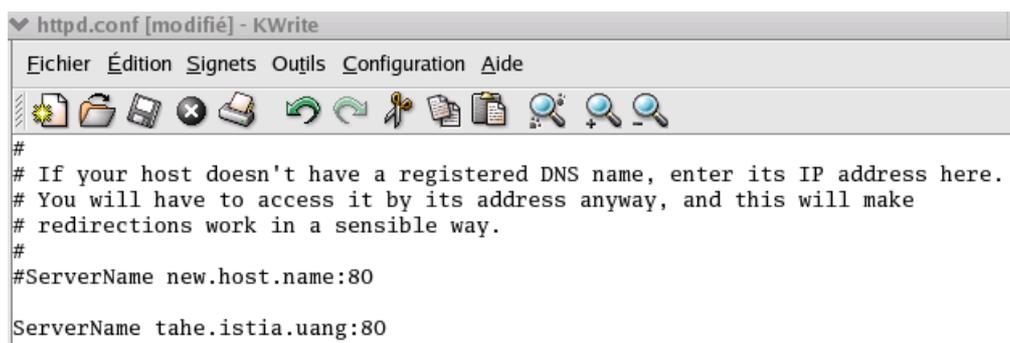


```
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
```

Nous n’allons pas détailler le rôle des différentes sections de ce fichier. Nous allons nous contenter de faire une installation minimale. Si vous administrez un jour un service Web/Apache, vous devrez comprendre les subtilités de ce fichier.

Cherchez une ligne de la forme `ServerName xxxxx`. Avec l’éditeur *kwrite*, vous pouvez procéder ainsi :

- **CTRL-F** pour lancer la 1ère recherche ou *Édition/Chercher*. Cherchez le mot `ServerName` sans distinction majuscules/minuscules et lancez la recherche. Vous trouvez une 1ère occurrence.
- faites **F3** pour trouver les suivantes jusqu’à trouver la suivante :



```
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
#ServerName new.host.name:80
ServerName tahe.istia.uang:80
```

La ligne **ServerName xxx** peut être en commentaires (# en début de ligne). Vous pouvez la laisser en commentaires et ajouter la ligne définissant le nom de votre serveur web :

```
ServerName X.istia.uang:80
```

Mettez le nom de votre serveur. Cette ligne indique également que le service web sera assuré sur le port 80. Trouvez maintenant une ligne ayant le texte **DocumentRoot** :

```
#  
# DocumentRoot: The directory out of which you will serve your  
# documents. By default, all requests are taken from this directory, but  
# symbolic links and aliases may be used to point to other locations.  
#  
DocumentRoot "/var/www/html"
```

La ligne `DocumentRoot /var/www/html` indique que les documents web délivrés par le serveur sont à placer sous `/var/www/html`. On en sait assez pour faire les premiers tests. Sauvegardez **httpd.conf**.

9.4 Démarrage/Arrêt du service HTTP

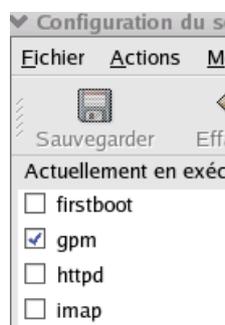
Maintenant, il nous faut lancer le serveur web Apache. On dispose pour lancer/arrêter ce service de plusieurs scripts :

Pour arrêter : `[/etc/rc.d/init.d/httpd stop]`

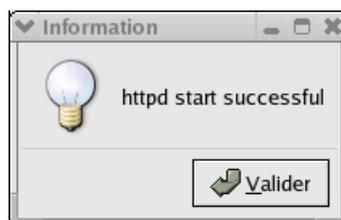
Pour démarrer : `[/etc/rc.d/init.d/httpd start]`

Pour faire les deux successivement : `[/etc/rc.d/init.d/httpd restart]`

Nous avons vu aussi que nous disposons d'un gestionnaire de services permettant de lancer/arrêter n'importe quel service. Dans bureau4(root), lançons ce service : **Menu KDE/Paramètres de système/Services**. Le mot de passe de root est demandé.



On voit que pour l'instant le service HTTP n'est pas lancé au démarrage (case httpd non cochée). On peut choisir de le lancer au démarrage ou non. Dans la pratique c'est souvent le cas. Cochez donc la case et sauvegardons la nouvelle configuration par **Fichier/Mettre à jour la liste des services**. Pour lancer le service, cliquez droit sur le service http et prenez l'option **démarrer**. Le service HTTP va être lancé et vous aurez un message indiquant comment s'est passé ce lancement.



9.5 Test 1

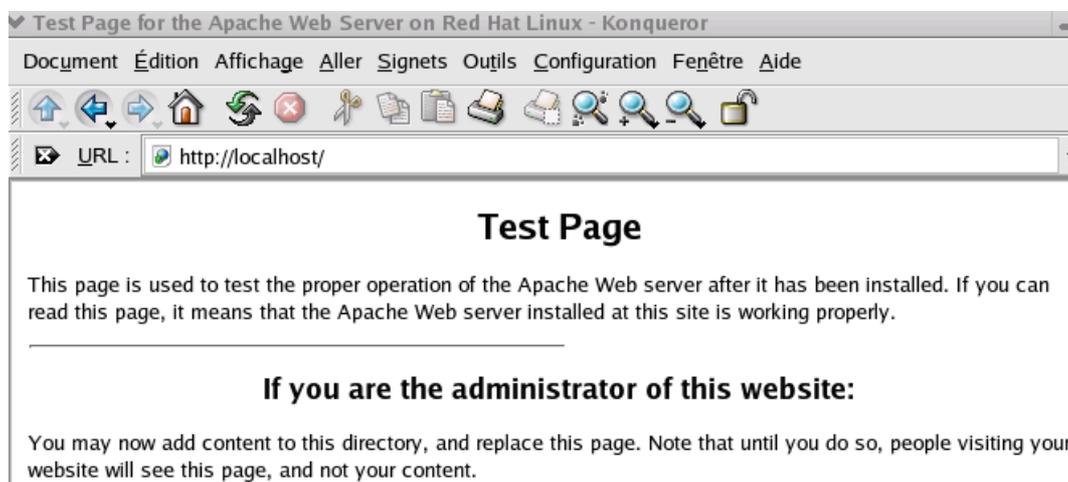
Comme nous l'avons fait pour les précédents services, dans la fenêtre terminal du bureau4(root) nous utilisons le client telnet :

```
[root@tahe root]# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

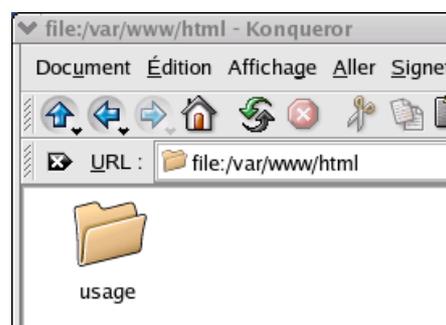
Nous avons eu une réponse. Le service HTTP opère bien sur le port 80. Nous pourrions commencer un dialogue avec ce service. Nous n'allons pas le faire. Pour clôturer la connexion, nous faisons ctrl-C et [Entrée].

9.6 Test 2, navigateur sous linux

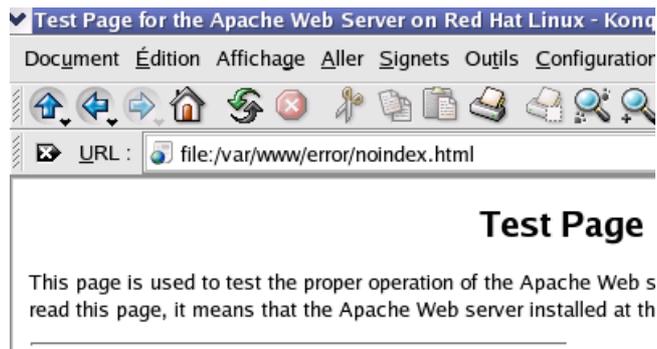
Dans bureau1(uti1), lancez *Konqueror* et demandez l'URL `http://localhost`. On rappelle que *localhost* désigne la machine sur laquelle vous travaillez. Vous devez obtenir une page analogue à la suivante (Inhibez *Affichage/Arborescence* si besoin est).



L'URL ci-dessus `http://localhost` n'est pas l'URL d'un document mais celle d'un dossier. Dans ce cas, le serveur Apache est configuré pour afficher le document **index.html** du dossier. Avec *Konqueror*, affichez le contenu du dossier `/var/www/html` qui est la racine de l'arborescence du serveur web et qui correspond à l'url `http://localhost` précédente :



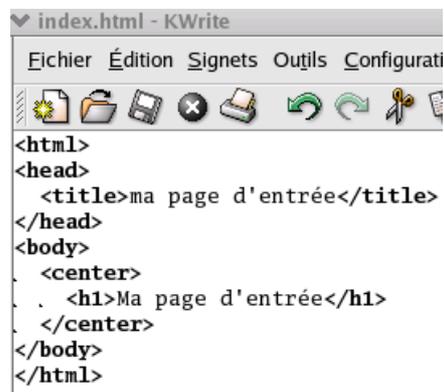
On voit qu'il n'y a pas de fichier **index.html**. Le serveur Apache est configuré, dans ce cas d'erreur précis, pour afficher une autre page, présente dans `/var/www/html/error/noindex.html` :



9.7 Test 3 - modifier la page d'entrée du site

La page d'entrée de votre site web est le fichier texte `/var/www/html/index.html`. Faisons les manipulations suivantes dans `bureau4(root)` pour le créer :

- avec l'éditeur *kwrite*, construisez un fichier *index.html*:



- sauvegardez ce texte dans `/var/www/html/index.html`.
- demandez l'URL `http://localhost` avec *Konqueror* (peu importe le bureau)



Faites ce même test avec le navigateur Mozilla maintenant.

On retiendra que :

- le point d'entrée de votre site Web est le fichier `/var/www/html/index.html`. C'est à partir de cette page que vous développerez votre site Web.
- le gestionnaire de services vous permet d'arrêter/relancer le service HTTP

9.8 Sites web personnels

Votre serveur Web/Apache peut être configuré pour donner à vos utilisateurs la possibilité de créer leurs propres pages Web et de les rendre accessibles à toute personne présente sur l'Internet. Pour accéder aux pages créées par l'utilisateur de login *uti1* de votre serveur *X.istia.uang*, il faut utiliser l'URL :

`http://X.istia.uang/~uti1`

La page réelle associée à cette URL dépend de la configuration du serveur Apache. Ce point est contrôlé par la ligne

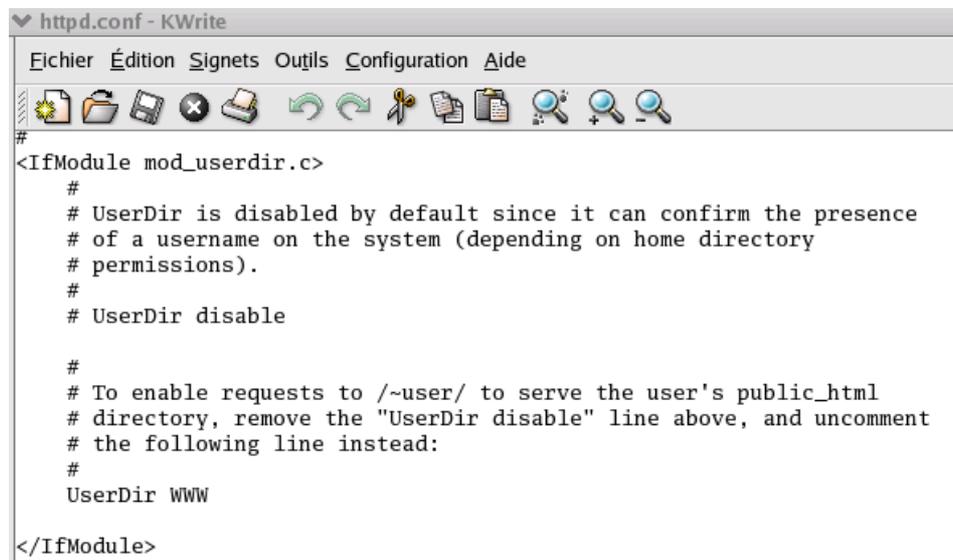
UserDir ...

du fichier `/etc/httpd/conf/httpd.conf`. Si dans ce fichier, on écrit cette ligne de la façon suivante :

UserDir WWW

L'URL `http://X.istia.uang/~uti1` sera associée au fichier `~uti1/WWW/index.html`, où `~uti1` (syntaxe admise sous Unix) désigne le répertoire personnel de l'utilisateur *uti1*. Vous pouvez vérifier les pages personnelles de la façon suivante (bureau4 - root) :

- dans le fichier `/etc/httpd/conf/httpd.conf`, recherchez (CTRL-F, F3) la séquence suivante :



```
#
<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can confirm the presence
# of a username on the system (depending on home directory
# permissions).
#
# UserDir disable
#
#
# To enable requests to /~user/ to serve the user's public_html
# directory, remove the "UserDir disable" line above, and uncomment
# the following line instead:
#
UserDir WWW
</IfModule>
```

Modifiez cette séquence comme indiqué ci-dessus :

- la ligne **UserDir disable** doit être mise en commentaires
- ajoutez la ligne **UserDir WWW** qui indique que chaque utilisateur pourra avoir un site web personnel dans le dossier WWW de son répertoire personnel.
- arrêtez puis relancez le serveur http avec le gestionnaire de services (clic droit/relancer sur service httpd)

Dans le bureau 1 de *uti1*, lancez *Konqueror* et *Kwrite* si besoin est.

- avec *Konqueror*, créez un dossier **WWW** dans `/home/uti1` :



- donnez les droits suivants aux dossiers `/home/uti1` et `/home/uti1/WWW`

Droits d'accès			
Classe	Listage	Écriture	Entrée
Utilisateur	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Groupe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Autres	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Propriété	
Utilisateur :	uti1
Groupe :	groupe1

Quelle est la signification de ces droits ? Le serveur web est un processus qui s'exécute sous l'identité d'un utilisateur appelé **apache**. Ce processus va devoir pénétrer dans le dossier **/home/uti1/WWW**. Il faut donc le droit **x** sur le dossier **/home/uti1** et tous ses sous dossiers pour l'utilisateur appelé **apache**. Une façon de lui donner ces droits est de donner le droit **x** au groupe 'tous'.

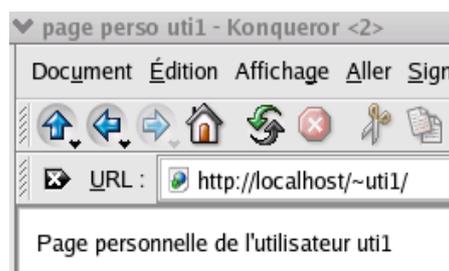
- Avec *kwrite*, construisez le fichier **/home/uti1/WWW/index.html** suivant :

```

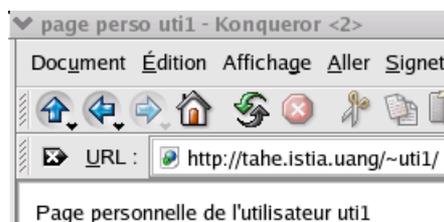
index.html - KWrite
Fichier  Édition  Signets  Outils  Configuration  Aide
<html>
  <head>
    . <title>page perso uti1</title>
  </head>
  <body>
    . Page personnelle de l'utilisateur uti1
  </body>
</html>

```

- Avec *Konqueror* ou *Mozilla* demandez l'URL <http://localhost/~uti1>



- Demandez l'URL <http://X.istia.uang/~uti1> où X.istia.uang est le nom de votre machine



- Répétez les différentes opérations précédentes à partir d'un autre poste de la salle, ou demandez à un autre groupe d'interroger votre service web. De même, vous, interrogez le service web d'un autre groupe

10 Le service Samba

10.1 Fonction

Samba est un service permettant (entre autres choses) à une machine Windows de voir une machine Linux comme un serveur Windows NT. La machine Linux peut alors servir de serveur de fichiers pour les machines Windows : celles-ci voient certains répertoires de Linux comme des disques locaux Dos/Windows. Par ailleurs, Samba permet aux utilisateurs Windows de voir leur répertoire Unix comme un disque réseau Dos/Windows. C'est un moyen pratique de fournir aux utilisateurs un répertoire personnel contrôlé par mot de passe et accessible de façon transparente par une machine Dos/Win.

10.2 Installation et configuration

Samba a été normalement installé en même temps que Linux. Vous pouvez avoir de l'aide par **[man samba]** ou **[man smbld]** pour samba lui-même et par **[man smb.conf]** pour son fichier de configuration.

Samba est piloté par le fichier **/etc/samba/smb.conf**. Celui fourni par défaut ne convient pas toujours. Il est également assez touffu. Pour simplifier, nous ne l'utiliserons pas. Comme il contient de nombreux commentaires instructifs, sauvegardez-le par les commandes suivantes :

```
[cd /etc/samba]
[ls -l smb.conf] // pour vérifier la présence du fichier
[more smb.conf] // pour vérifier le contenu du fichier
[mv smb.conf smb.conf.original] // change le nom du fichier
```

Construire maintenant un nouveau fichier *smb.conf* avec le contenu suivant (en omettant les commentaires //) :

```
[global] // configurations valides pour toutes les ressources
workgroup = groupe.X.istia.uang
// groupe de travail. Seuls les clients appartenant à ce groupe ont accès au service SAMBA.
// Remplacez X.istia.uang par le nom de votre serveur
security=share
// Votre serveur Samba sera un serveur de ressources partagées (security=share)et non un serveur
d'authentification (security=user).
allow bosts = 172.20.81.
// seuls les clients appartenant au réseau 172.20.81.0 seront acceptés
// ne pas oublier le point derrière 81
log file = /var/log/samba.log
// désigne le fichier où samba placera ses logs
max log size = 5000
// taille maxi du fichier de logs (en Kilo octets)
mangle case=no
case sensitive=no
// Lorsqu'on met yes, il arrive que le service Samba ne fonctionne plus.
short preserve case=yes
preserve case=yes
// la "casse" majuscules/minuscules des noms de fichiers sera préservée dans les échanges
DosWin/Unix.
default case = lower
// lorsque Samba est amené à créer des fichiers pour lesquels il n'a pas d'informations sur la "casse" à
utiliser, il écrira en minuscules le nom des fichiers.
lock directory = /home/samba/locks
// répertoire contenant des fichiers de synchronisation de processus. Il faudra créer ce répertoire.
share modes = yes
```

```

create mode = 0755
    // les fichiers & répertoires seront créés avec les droit rwxr-xr-x
browseable=no
    // les ressources offertes par Samba n'apparaîtront pas dans l'explorateur Windows des postes clients
revalidate=yes

    // suivent les ressources partagées offertes par Samba

[private]
comment = Logiciels privés // commentaire
public = no
valid users = uti1 // utilisateurs autorisés à accéder à la ressource -
// accès par mot de passe
force group= admprivate // une fois connecté, l'utilisateur uti1 sera placé dans le groupe admprivate
writeable = yes // ressource disponible en lecture/écriture
path = /home/samba/private // répertoire linux associé à la ressource - il faudra le créer
create mode = 0775 // fichiers & répertoires créés dans la ressource avec les droits rwxrwxr-x

; pour l'utilisation de logiciels
[public]
comment = Logiciels publics
public = yes // accès public sans mot de passe pour un accès en lecture seule
writeable=yes // en lecture/écriture pour les utilisateurs de la write list ci-
dessous // en lecture seule pour les autres
path = /home/samba/public // répertoire linux associé à la ressource - il faudra le créer
write list=@admsamba // utilisateurs autorisés à écrire - ici les membres du groupe
// admsamba

; répertoires personnels
[homes] // ressource permettant aux utilisateurs
enregistrés sur // la machine Linux d'avoir accès, via Samba, à
leur répertoire // personnel
public=no // accès par mot de passe
writeable = yes // en lecture/écriture

```

10.3 Commentaires

Le service SAMBA est très utile si vous avez un réseau de machines Win95/98 ou NT. Avec votre serveur actuel, nous avons vu que chaque étudiant avait un répertoire Unix auquel il pouvait accéder par *ftp*. SAMBA permet de voir ce répertoire Unix distant comme un disque local au PC windows sur lequel travaille l'étudiant. Il peut alors gérer son répertoire Unix avec l'explorateur Windows ou le gestionnaire de fichiers. Cela apporte un confort important vis à vis de *ftp*. Comme la taille de ce répertoire Unix peut être contrôlée par le service des Quotas que nous verrons ultérieurement, l'administrateur n'a pas d'inquiétude à se faire sur la saturation du disque Unix. Par ailleurs, SAMBA permet de faire d'une machine Linux, un serveur de fichiers à l'instar de serveurs tels que Microsoft NT ou Novell NETWARE.

Les 3 services SAMBA à installer sont les suivants :

Private

C'est un dossier d'accès privé (**public=no**), réservé à certains utilisateurs (**valid users=...**). Le document vous propose de mettre *uti1* comme utilisateur. Sachez que vous pouvez mettre autant d'utilisateurs que vous voulez sous la forme *user1*, *user2*, ... ou mieux sous la forme *@groupe1*, *@groupe2*, où *groupei* désigne un groupe défini dans le fichier */etc/group*. Vous donnez ainsi accès au service à tous les membres d'un groupe. Tous les utilisateurs autorisés à utiliser le service devront donner leur mot de passe (Unix) pour passer. Ceci fait, ils pourront écrire dans le dossier *private* (**writeable=yes**). Lorsqu'ils écriront dedans, ils écriront en fait dans le répertoire Unix défini par (**path=/home/samba/private**). Il vous faudra bien sûr créer ce répertoire et vous assurer que les utilisateurs autorisés à écrire dans la ressource Samba [*private*] ont également

les droits d'écriture dans le répertoire Unix associé `/homesamba/private`. Ce sont des droits d'écriture différents qu'il faut positionner de façon séparée.

Le dossier `private` peut être utilisé pour y mettre des logiciels dont on n'a qu'une ou quelques licences. Seuls quelques utilisateurs sont autorisés à utiliser ce dossier. Comme en fait, ce ne sont pas forcément les mêmes utilisateurs pour chacun des logiciels, il faudrait créer un service par logiciel. On peut créer ainsi une liste d'utilisateurs par logiciel. Il faudrait probablement mettre (**writeable=no**) pour éviter que ces utilisateurs n'écrasent le logiciel.

Public

Ce service est d'accès public, c'est à dire sans mot de passe (**public=yes**), en écriture seulement pour les gens du groupe `admsamba` (`writeable=yes`, `write list=@admsamba`). Il est associé au répertoire Unix `/home/samba/public` (**path=/home/samba/public**). Ce répertoire peut être utilisé pour mettre des logiciels accessibles en lecture pour tous. On met dans la `'write list'` les administrateurs qui doivent installer les logiciels dans cette ressource `[public]`.

Homes

C'est ce service qui permet aux utilisateurs de voir leur répertoire Unix comme un disque local Dos/Windows. Ce service n'est pas publié dans *Voisinage Réseau* : il faut donc connecter un lecteur réseau au service. Celui-ci est accédé (d'une façon différente des deux services précédents) par `\\X.istia.uang\login`. SAMBA reconnaît que `login` n'est pas un service déclaré dans son fichier de configuration (comme le sont les services `private` ou `public` précédents). Il le déclare alors à la volée avec la même configuration que le service **homes**. Il demande alors un mot de passe (**public=no**) et associe le service au répertoire de connexion de l'utilisateur défini par **login**. Ainsi, si on demande la connexion au service `\\X.istia.uang\uti1` il faudra donner le mot de passe de l'utilisateur `uti1` de la machine `X.istia.uang`. Ceci fait, un dossier **uti1** apparaît dans le "Voisinage réseau" du poste Win95 ou NT. Ce dossier est en fait le répertoire Unix de l'utilisateur `uti1`. Dans notre exemple, il peut y écrire (**writeable=yes**). Comme dit précédemment, les services SAMBA apparaissent dans le "Voisinage Réseau" de tous les postes clients Microsoft. Il n'est pas utile, qu'à chaque fois qu'un utilisateur se connecte à son répertoire Unix, que celui-ci soit publié dans le "Voisinage Réseau" de tous les postes Win95/98 du réseau. On empêche ceci par l'option **browseable=no**, option qui a été mise ici dans la section `[global]`.

10.4 Préparatifs

Dans `bureau4(root)`. Créez les répertoires nécessaires aux tests :

```
[root@tahe root]# mkdir /home/samba
```

```
[root@tahe root]# mkdir /home/samba/private
```

```
[root@tahe root]# mkdir /home/samba/public
```

```
[root@tahe root]# ls -l /home/samba/
total 8
drwxr-xr-x  2 root  root    4096 mar 31 10:38 private
drwxr-xr-x  2 root  root    4096 mar 31 10:38 public
```

Copiez quelques fichiers dans ces dossiers :

```
[root@tahe root]# cp /etc/passwd /home/samba/private/
```

```
[root@tahe root]# ls -l /home/samba/private/
total 4
-rw-r--r--  1 root  root    1930 mar 31 10:39 passwd
```

```
[root@tahe root]# cp /etc/group /home/samba/public/
```

```
[root@tahe root]# ls -l /home/samba/public/
total 4
-rw-r--r--  1 root  root    731 mar 31 10:40 group
```

Créez le groupe **admsamba** et mettez l'utilisateur **adm1** dedans :

```
[root@tahe root]# groupadd admsamba
```

```
[root@tahe root]# useradd -g admsamba adm1
```

```
[root@tahe root]# id adm1
uid=504(adm1) gid=503(admsamba) groupes=503(admsamba)
```

Fixez-lui un mot de passe (*azerty* par exemple):

```
[root@tahe root]# passwd adm1
Changing password for user adm1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

10.5 Démarrage du service

Le service Samba est arrêté par `[/etc/rc.d/init.d/smb stop]` et relancé par `[/etc/rc.d/init.d/smb start]`. On peut enchaîner un arrêt/démarrage par `[/etc/rc.d/init.d/smb restart]`.

- relancez samba pour qu'il relise son nouveau fichier de configuration *smb.conf*
- vérifiez qu'il a été lancé par `[ps aux | grep smb]`

```
[root@tahe root]# /etc/rc.d/init.d/smb restart
Fermeture des services SMB : [ OK ]
Fermeture des services NMB : [ECHOUE]
Démarrage des services SMB : [ OK ]
Démarrage des services NMB : [ OK ]
[root@tahe root]# ps aux | grep -i smb
root      3443  0.0  0.7 4920 1792 ?        S    10:50   0:00 smbld -D
root      3451  0.0  0.2 3372  676 pts/4    S    10:50   0:00 grep -i smb
```

On peut aussi lancer/arrêter le service samba avec le gestionnaire de services.

10.6 Utilisation du client smbclient

On peut tester le bon fonctionnement du serveur samba avec le client Unix *smbclient*. Pour voir la syntaxe de cette commande, tapez dans une fenêtre terminal :

```
[smbclient]
```

Un résumé de la syntaxe de la commande vous est alors présenté. Elle est de la forme **smbclient service [options]**. Un client samba référence le service **S** d'une machine **M** par la syntaxe `\\M\S`. Il se trouve que sous Unix, le caractère `\` est interprété d'une façon particulière. Pour empêcher cette interprétation particulière, il faut le doubler. C'est pourquoi on écrit `\\\\M\\S` pour référencer le service **S** d'une machine **M**.

10.6.1 Connexion à un service d'accès public

Connectons-nous au service **public** de notre serveur samba **X.istia.uang** (vous remplacerez *X.istia.uang* par le nom réel de votre machine). Dans le fichier *smb.conf* le service **public** est déclaré par la ligne

```
[public]
```

Par défaut le serveur samba prend le nom de la machine sur laquelle il se trouve. Pour se connecter au service `[public]` de la machine *X.istia.uang*, on tapera la commande analogue à la suivante :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\\public -N
Unknown parameter encountered: "revalidate"
Ignoring unknown parameter "revalidate"
Domain=[TAHE] OS=[Unix] Server=[Samba 2.2.5]
smb: \>
```

L'option `-N` indique à *smbclient* de ne pas demander de mot de passe. En effet, ici on se connecte au service `[public]` d'accès pour tous (option `public=yes` dans la définition du service).

Si la commande ci-dessus échoue c'est que le service samba a été probablement mal configuré. Vérifiez alors le fichier `/etc/samba/smb.conf`. Celui-ci déclare que le fichier de logs est `/var/log/samba.log`. Vérifiez les messages enregistrés dans ce fichier par `[tail /var/log/samba.log]`. Ils décrivent en général les erreurs rencontrées.

Si tout s'est bien passé, on est maintenant connecté au service samba en mode interactif. Samba exécute les commandes qu'on tape au clavier. Pour avoir la liste de ces commandes, faites :

```
smb: \> help
?          altname      archive      blocksize    cancel
cd         chmod         chown        del           dir
du         exit          get          help         history
lcd        link          lowercase    ls           mask
md         mget         mkdir        more         mput
newer     open         print       printmode    prompt
put        pwd          q           queue        quit
rd         recurse      rename      rm           rmdir
setmode   symlink      tar         tarmode      translate
```

Utilisons la commande `ls` qui nous permet de voir le contenu du dossier sur lequel on est positionné.

```
smb: \> ls
.          D           0   Mon Mar 31 10:40:34 2003
..         D           0   Mon Mar 31 10:45:39 2003
group     731   Mon Mar 31 10:40:34 2003
```

Quelle est la signification de ce résultat ? Dans la définition du service `[public]`, nous avons écrit `path = /home/samba/public`. En nous connectant à ce service, nous avons été positionnés sur le dossier `/home/samba/public`. La commande `ls` nous montre alors le contenu de ce dossier. Nous y avons mis le fichier `group`.

Essayons de créer un dossier avec la commande `mkdir` :

```
smb: \> mkdir test
NT_STATUS_NETWORK_ACCESS_DENIED making remote directory \test
```

La commande a été refusée, parce que dans la description du service `[public]` on a mis `writeable=no`. Maintenant nous avons mis la ligne `write list=@admsamba` dans la description du service. Cela signifie que les utilisateurs membres du groupe `admsamba` pourront écrire dans le dossier du service `[public]` (même avec l'option `writeable=no`). Essayons :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\public -U adm1
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
```

```
smb: \> ls
.          D           0   Mon Mar 31 17:03:10 2003
..         D           0   Mon Mar 31 17:00:28 2003
group     740   Mon Mar 31 17:03:10 2003

63002 blocks of size 262144. 48618 blocks available
```

```
smb: \> mkdir test
NT_STATUS_NETWORK_ACCESS_DENIED making remote directory \test
```

Nous nous sommes connectés à samba en tant qu'utilisateur `util` (-U util) qui fait partie du groupe `admsamba`. On s'attend donc à ce qu'il ait un droit d'écriture sur le dossier `[public]` et pourtant la création du répertoire `test` échoue. Ici, il faut comprendre que le service samba n'est qu'un intermédiaire entre un client et le système Unix auquel ce client veut accéder. Se connecter à Samba comme utilisateur `util` entraîne que samba va faire toutes les opérations demandées sur le système Unix en tant qu'utilisateur samba. La demande de création du répertoire `test` fait que samba va essayer de créer le répertoire `/home/samba/public/test` sous l'identité `util`. Regardons les droits d'accès de ce dossier :

```
[root@tahe root]# ls -ld /home/samba/public/
drwxr-xr-x  2 root  root  4096 mar 31 17:03 /home/samba/public/
```

On voit que le dossier appartient à l'utilisateur `root` et au groupe `root`. L'utilisateur `util`, avec les droits ci-dessus, a un droit de lecture(`r`) mais pas de droit d'écriture (`w`). Ce qui explique pourquoi notre utilisateur `util` a pu, grâce au client `smbclient`, faire un `ls` (droit de lecture) mais pas un `mkdir` (droit d'écriture). Il nous faut donner des droits d'écriture au groupe `admsamba`. Procédons ainsi :

```
[root@tahe root]# ls -ld /home/samba/public/
drwxr-xr-x  2 root  root  4096 mar 31 17:03 /home/samba/public/
```

```
[root@tahe root]# chown root:admsamba /home/samba/public/
```

```
[root@tahe root]# ls -ld /home/samba/public/
drwxr-xr-x  2 root  admsamba  4096 mar 31 17:03 /home/samba/public/
```

```
[root@tahe root]# chmod g+w /home/samba/public/
```

```
[root@tahe root]# ls -ld /home/samba/public/
drwxrwxr-x  2 root  admsamba  4096 mar 31 17:03 /home/samba/public/
```

Cette fois-ci, tout membre du groupe **admsamba** a un droit d'écriture sur le dossier **/home/samba/public**. Essayons de nouveau de nous connecter à samba en tant qu'utilisateur **uti1** :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\public -U adm1
Password:
Domain=[TAHE] OS=[Unix] Server=[Samba 2.2.5]
```

```
smb: \> ls
.                D            0 Mon Mar 31 10:40:34 2003
..               D            0 Mon Mar 31 10:45:39 2003
group            731 Mon Mar 31 10:40:34 2003

63002 blocks of size 131072. 45697 blocks available
```

```
smb: \> mkdir test
```

```
smb: \> ls
.                D            0 Mon Mar 31 11:23:18 2003
..               D            0 Mon Mar 31 10:45:39 2003
group            731 Mon Mar 31 10:40:34 2003
test             D            0 Mon Mar 31 11:23:18 2003
```

Cette fois-ci, on constate que l'utilisateur **adm1** du groupe **admsamba** a pu écrire dans le dossier **[public]**. Vérifions les propriétaires du dossier **test** :

```
[root@tahe root]# ls -ld /home/samba/public/test/
drwxr-xr-x  2 adm1  admsamba  4096 avr  1 08:36 /home/samba/public/test/
```

Il est, comme attendu, propriété de **uti1(admsamba)**.

10.6.2 Connexion à un service d'accès privé

Connectons-nous maintenant au service **[private]** qui est d'accès privé (option **public=no**). Connectons-nous tout d'abord en tant qu'utilisateur **uti2** :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\private -U uti2
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
tree connect failed: NT_STATUS_WRONG_PASSWORD
```

```
[root@tahe root]# smbclient \\\tahe.istia.uang\public -U uti2
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
```

```
smb: \> ls
.                D            0 Tue Apr  1 08:36:44 2003
..               D            0 Mon Mar 31 17:00:28 2003
group            740 Mon Mar 31 17:03:10 2003
test             D            0 Tue Apr  1 08:36:44 2003
```

Nous nous sommes tout d'abord connectés au service **[private]**. La connexion a échoué avec un message d'erreur qui laisse planer un doute sur la cause de l'échec. Nous nous connectons alors au service **[public]** dans les mêmes conditions. La connexion se fait bien. La connexion au service **[private]** a échoué pour les raisons suivantes :

- le service est privé (**public=no**)

- **uti2** ne fait pas partie des utilisateurs autorisés (**valid users= uti1**)

Essayons donc la connexion avec **uti1** :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\private -U uti1
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
smb: \> ls
.                D            0   Mon Mar 31 17:03:03 2003
..               D            0   Mon Mar 31 17:00:28 2003
passwd           2040        Mon Mar 31 17:03:03 2003
```

Nous constatons que l'utilisateur **uti1** a pu se connecter au service [**private**] et lister son contenu. Cela vient du fait, comme il a été dit, qu'**uti1** est un utilisateur autorisé du service (**valid users= uti1**). On trouve dans le dossier [**private**] le fichier *passwd* qu'on y avait mis. On a le droit d'écrire dans le dossier (option **writable=yes**). Essayons :

```
smb: \> mkdir test
NT_STATUS_ACCESS_DENIED making remote directory \test
```

Pourquoi l'écriture a-t-elle échoué ? Pour la même raison rencontrée un peu plus haut. Une fois qu'**uti1** est connecté, samba va agir sur le système de fichiers d'unix avec l'identité **uti1(groupe1)**. Or regardons les droits du dossier **/home/samba/private** :

```
[root@tahe root]# ls -ld /home/samba/private/
drwxr-xr-x  2 root  root  4096 mar 31 17:03 /home/samba/private/
```

On voit que l'utilisateur **uti1(groupe1)** n'a pas le droit d'écriture sur ce dossier, ce qui explique l'échec précédent. On pourrait être tenté de faire de **groupe1** le groupe propriétaire du dossier et de donner à ce groupe le droit d'écriture. Cela permettrait à *uti1* de pouvoir écrire dans le dossier. Seulement si plus tard, on veut qu'un autre utilisateur à écrire dans le dossier, celui-ci ne fera pas forcément partie de **groupe1**. Il serait préférable de regrouper tous ceux qui sont autorisés à écrire dans ce dossier dans un même groupe. Appelons-le **admprivate**. Nous le créons et en faisons le propriétaire du dossier **private** :

```
[root@tahe root]# groupadd admprivate
[root@tahe root]# chown root:admprivate /home/samba/private/
[root@tahe root]# ls -ld /home/samba/private/
drwxr-xr-x  2 root  admprivate  4096 mar 31 17:03 /home/samba/private/
[root@tahe root]# chmod g+w /home/samba/private/
[root@tahe root]# ls -ld /home/samba/private/
drwxrwxr-x  2 root  admprivate  4096 mar 31 17:03 /home/samba/private/
```

Plutôt que d'ajouter **uti1** au groupe **admprivate**, nous avons ajouté (cf fichier *smb.conf* utilisé) aux propriétés du service [**private**] la ligne **force group=admprivate**. Tout utilisateur qui sera connecté au service se verra automatiquement affecté au groupe **admprivate** pour ses opérations sur le système de fichiers unix. Essayons :

```
[root@tahe root]# smbclient \\\tahe.istia.uang\private -U uti1
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
```

```
smb: \> mkdir test
```

```
smb: \> ls
passwd           2040        Mon Mar 31 17:03:03 2003
test            D            0   Tue Apr  1 09:07:19 2003
```

Vérifions les propriétaires du dossier **test** :

```
[root@tahe root]# ls -ld /home/samba/private/test/
drwxr-xr-x  2 uti1  admprivate  4096 avr  1 09:07 /home/samba/private/test/
```

Il est, comme attendu, propriété de **uti1(admprivate)**.

10.6.3 Connexion au répertoire personnel

Le service [**homes**] permet à un utilisateur distant de se connecter à son répertoire personnel sous son identité. Rappelons les caractéristiques de ce service :

```
[homes]
  public=no
  writeable = yes
```

C'est un service **privé**. Le client est connecté au répertoire personnel de l'utilisateur dont il a fourni les login et mot de passe. Il y aura un droit d'écriture. Essayons :

```
root@tahe root]# smbclient \\\tahe.istia.uang\homes -U util
added interface ip=172.19.81.121 bcast=172.19.255.255 nmask=255.255.0.0
Password:
Domain=[GROUPE1] OS=[Unix] Server=[Samba 2.2.7]
```

```
smb: \> ls
.                D            0 Tue Apr  1 08:52:19 2003
..               D            0 Mon Mar 31 16:49:17 2003
.kde             DH           0 Mon Aug 12 11:26:50 2002
.bash_logout    H            24 Wed Mar  5 15:17:32 2003
...
gnome.ui        342096 Mon Mar 31 15:04:04 2003
...
```

L'utilisateur **util** a été connecté à son dossier. Il y a un droit d'écriture :

```
smb: \> mkdir test
```

```
smb: \> ls test
test                D            0 Tue Apr  1 09:14:11 2003
```

Vérifions les droits avec lesquels a été créé ce dossier :

```
[root@tahe root]# ls -ld /home/util/test
drwxr-xr-x  2 util  groupe1  4096 avr  1 09:14 /home/util/test
```

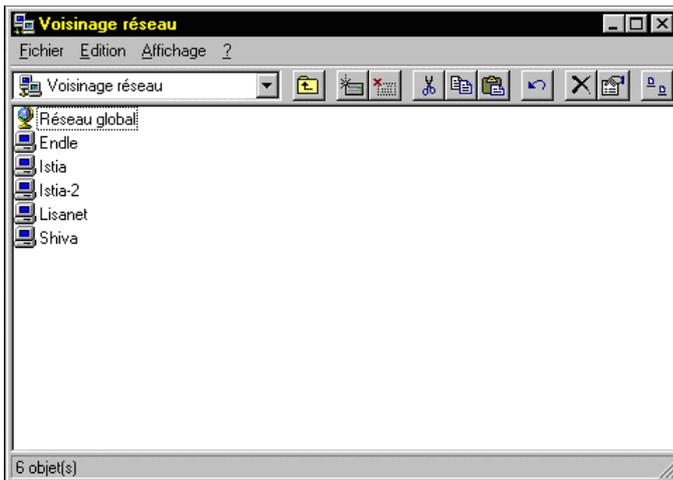
Il est bien propriété de **util(groupe1)**.

10.7 Utilisation d'un client windows

Le véritable intérêt de Samba se trouve dans un environnement windows. Les services que Samba va publier vont pouvoir être vus comme des disques locaux des machines windows. On est là proche du concept unix de montage de système de fichiers. Les tests précédents faits localement sur votre machine unix vous ont permis de déboguer votre configuration. Les tests qui suivent visent à vérifier que vos services sont accessibles d'un poste windows. Pour cela, utilisez les postes windows de la salle où vous vous trouvez.

10.7.1 Voisinage réseau

Vérifiez tout d'abord que votre serveur Samba est vu dans le "Voisinage réseau" d'une machine Windows. Placez vous sur une machine Windows double-cliquez sur l'icône "Voisinage réseau" sur le bureau. Au bout d'un moment, vous devez voir votre serveur (**groupe.X.istia.uang**) dans la liste des machines (vous avez écrit *workgroup=groupe.X.istia.uang*).



Ce n'est cependant pas toujours le cas. On peut alors essayer
[clic droit] Voisinage réseau / Rechercher un ordinateur

Indiquez le nom *X.istia.uang* de votre serveur. Si votre service Samba fonctionne correctement, votre machine samba *X.istia.uang* doit alors être trouvée, sinon votre service Samba n'a pas été lancé ou il est mal configuré.

Double-cliquez sur l'icône de votre serveur : un dossier vide apparaît. Ceci parce qu'on a mis dans les options générales du fichier *smb.conf* : *browseable=no*. Si on avait mis *yes*, on verrait dans le dossier les deux ressources créées : *[private]* et *[public]*.

10.7.2 Accès à la ressource **[public]** en lecture seule

Procédez comme suit :

[clic droit] Voisinage Réseau / Connecter Lecteur



Dans le champ lecteur, sélectionnez le lecteur K. Dans le champ chemin, tapez :

```
\\X.istia.uang\public
```

Le lecteur K est alors associé à la ressource *[public]* de votre serveur Samba. Avec l'explorateur Windows, positionnez-vous sur le lecteur K. Vous devez y trouver les fichiers que vous y aviez placés. Essayez de créer un dossier sur le lecteur K **[clic droit/Nouveau/Dossier]** : cela doit échouer, car la ressource n'est en écriture que pour certains utilisateurs, ceux du groupe *admsamba*.

10.7.3 Accès aux répertoires personnels sur linux, ressource **[homes]**

On va vérifier maintenant que tout utilisateur enregistré sur votre serveur peut voir son répertoire Unix comme un disque local de son PC Windows. Procédez comme suit :

[clic droit] Voisinage Réseau / Connecter Lecteur]



Dans le champ chemin, tapez :

```
\\X.istia.uang\utilisateur
```

avec

X.istia.uang
utilisateur

le nom de votre serveur

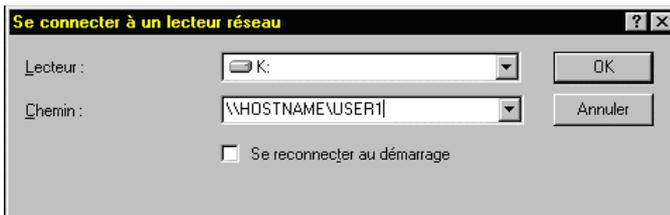
login d'un utilisateur de votre serveur – *util* par exemple

Faire ensuite **[OK]**. Ensuite vous devez entrer le mot de passe de l'utilisateur. Ceci fait, utilisez l'explorateur Windows pour vous positionner sur K. Vous devez voir le répertoire linux de l'utilisateur *util*. Vérifiez que vous pouvez y créer un nouveau dossier ou y copier des fichiers (vous avez mis **writeable=yes** pour le service **homes**).

10.7.4 Accès à la ressource [private]

On essaie d'accéder au service **private**. Procédez comme suit :

[clic droit] Voisinage Réseau / Connecter Lecteur]



Dans le champ [Chemin], mettez cette fois-ci :

```
\\X.istia.uang\private%util
```

avec

X.istia.uang
private
util

le nom de votre serveur

nom du service auquel vous voulez accéder

nom d'utilisateur avec lequel vous voulez accéder au service. On utilise ici **util**, car dans **smb.conf**, pour le service **private**, vous avez écrit **valid users=util**, ce qui signifie que **util** a accès au service **private**.

Faire **[OK]**. Un mot de passe est alors demandé (vous avez écrit *public=no* pour ce service). Donnez le mot de passe de l'utilisateur *util* de votre serveur. Vous devriez voir la ressource *[private]* avec dedans le fichier **passwd** que vous y avez mis. Vérifiez que vous pouvez créer un nouveau dossier ou y copier des fichiers sur K (vous avez écrit **writeable=yes** pour ce service dans **smb.conf**).

La connexion à un service peut parfois être déroutante. Sous Win9x, on utilise l'icône "Voisinage réseau" et son option "Connecter un lecteur réseau" pour se connecter au service `\\X.istia.uang\private` par exemple. Celui-ci répond en demandant un mot de passe. Ce qui est surprenant c'est qu'il ne demande pas qui on est (notre login Unix). On ne voit pas alors comment il fait pour associer le mot de passe qu'on lui donne au bon login. Il faut savoir que dans les réseaux Microsoft, chaque machine, appelée client Microsoft, porte un nom appelé nom Netbios. Celui-ci est indiqué dans l'onglet "Identité" du dossier "Réseau" du "Panneau de configuration". C'est ce nom qu'utilise le service SAMBA pour identifier le client et il associe alors le mot de passe qu'on lui donne à un login Unix identique à ce nom de client Microsoft. Comme ces deux identités n'ont rien à voir l'une avec l'autre, la connexion échoue fatalement.

Pour se connecter à un service SAMBA avec un login Unix précis, il faut demander (comme dans notre exemple) la connexion au service `\\X.istia.uang\private%login`. Le mot de passe demandé est alors relié au login indiqué. Dans

notre exemple, puisque seul *util* a un accès autorisé au service *private*, il faut demander la connexion au service `\\X.istia.uang\private%util` où *X.istia.uang* est le nom de votre serveur.

10.7.5 Accès à la ressource [public] en lecture et écriture

Nous avons vu que la ressource *[public]* avait une *'write list'* c'est à dire une liste de gens autorisés à écrire. Pour tester cette possibilité, on est obligé de passer dans une fenêtre DOS de Win95/98. Dans cette fenêtre tapez :

```
[net /?]
// vous obtenez des informations sur la commande net que l'on va utiliser pour se connecter à notre serveur
Samba
[net config]
// vous donne la configuration de votre poste en tant que client SMB.
[net use k: \\X.istia.uang\public]
// connecte un lecteur K à la ressource [public] en lecture seule.
// vérifiez en essayant de copier des fichiers sur K. Ca devrait échouer.
[net use k: \\X.istia.uang\public%adm1 mdp1]
// où mdp1 est le mot de passe de l'utilisateur adm1 que vous avez créé. Cet utilisateur fait partie du groupe
admsamba qui a un droit d'écriture sur la ressource public
```

Essayez d'écrire dans le dossier k:\ en créant un répertoire par exemple. Cela devrait fonctionner.

10.7.6 Deux utilisateurs win9x particuliers

Dans la fenêtre DOS d'un poste win9x, faites **[net config]** et notez le nom de votre ordinateur (*nomOrdinateur*) et de l'utilisateur (*nomUtilisateur*). Revenez sur votre machine Linux et créez deux nouveaux utilisateurs dans le fameux groupe *admsamba* qui a le droit d'écriture dans la ressource *[public]*. Ces utilisateurs porteront les noms de votre ordinateur et de son utilisateur.

```
[/usr/sbin/useradd -g admsamba nomOrdinateur]
[/usr/sbin/useradd -g admsamba nomUtilisateur]
```

Donnez des mots de passe à chacun de ces utilisateurs (faites en sorte que `motDePasse=azerty` pour simplifier). Maintenant, revenez dans la fenêtre Dos précédente et tapez la commande suivante :

```
[net use k: \\machine\public mdpOrdinateur]
// où mdpOrdinateur est le mot de passe de l'utilisateur nomOrdinateur que vous venez de créer.
// vérifiez que vous avez accès à K en lecture et écriture
```

Refaire la même chose avec la commande :

```
[net use k: \\machine\public mdpUtilisateur]
// où mdpUtilisateur est le mot de passe de l'utilisateur nomUtilisateur que vous venez de créer.
// vérifiez que vous avez accès à K en lecture et écriture
```

Que se passe-t-il exactement ? Lorsqu'on ne précise pas explicitement un utilisateur lors de la connexion à une ressource Samba, Win9x essaie successivement le nom de l'ordinateur et le nom de l'utilisateur de celui-ci avec le mot de passe que vous donnez sur la ligne de commande.

10.8 Conclusion

SAMBA est actuellement utilisé à l'ISTIA pour les raisons suivantes :

- La première : on désirait depuis longtemps fournir aux étudiants un répertoire personnel, sur un serveur dont l'accès soit contrôlé par mot de passe. Novell/Netware nous offrait cette possibilité, mais il était lourd de créer deux séries de compte : une pour Novell/Netware et une autre pour Unix. Netware 4.1 n'offre aucun outil pour créer massivement des comptes. On ne voulait pas, chaque année, créer sur Netware plusieurs centaines de comptes un par un. On voulait utiliser Unix pour lequel on a des outils qui permettent de créer

des dizaines d'utilisateurs d'un coup. SAMBA a été la solution, puisqu'il permet de voir le répertoire Unix de l'étudiant comme un disque Dos/Windows local dont l'accès est contrôlé par le couple *login/mot de passe* de l'étudiant.

- La seconde : avant SAMBA l'utilisation d'Eudora par les étudiants était problématique. En effet, pour qu'Eudora puisse être utilisé sur une machine en libre-service sans que les courriers des différents utilisateurs se mélangent, il existe diverses solutions :
 - Une icône Eudora par utilisateur précisant un répertoire de travail différent à chaque fois – impossible à mettre en œuvre ici.
 - Une seule icône Eudora avec A:\ comme répertoire de travail. C'est une solution qui a le mérite de marcher mais qui est frustrante : les accès à la disquette A sont d'une lenteur désespérante.

Nous n'avions pas trouvé de solution satisfaisante et on préconisait l'utilisation d'autres utilitaires de courrier, moins performants qu'Eudora, mais qui ne nécessitaient qu'une icône sur le poste. Ces outils, contrairement à Eudora, ne ramenaient pas le courrier sur le poste de consultation et donc préservaient la confidentialité du courrier.

Avec SAMBA, on a réglé le problème de la façon suivante : on n'a qu'une icône Eudora avec comme répertoire de travail *K:\courrier*. L'étudiant, avant d'utiliser Eudora, doit assigner la lettre K à son répertoire Unix. Cela se fait avec l'outil "Voisinage réseau" de Win95 ou WinNT et n'est possible que grâce à SAMBA. Le répertoire Unix de l'étudiant est alors vu comme le disque local K. Ainsi lorsque Eudora est lancé, il "descend" le courrier de l'utilisateur dans le répertoire *K:\courrier*, c.a.d. dans le répertoire Unix de l'étudiant. La confidentialité de son courrier est ainsi assurée et il le retrouve stocké dans son répertoire personnel.

Le document vous propose une installation minimale. C'est néanmoins celle qui est à l'œuvre à l'Istia et qui donne satisfaction. Vous pouvez découvrir SAMBA à l'aide du manuel en ligne en faisant : **man smb**.

11 Le service des quotas

11.1 Fonction

Le service des *quotas* est le service qui gère les espaces disque des différents utilisateurs locaux du serveur. Il permet de spécifier en Kilo-octets la taille maximale de l'espace disque de chaque utilisateur. Lorsque la limite de stockage d'un utilisateur est atteinte, l'utilisateur est obligé de libérer de la place s'il souhaite ajouter des données sur son compte.

11.2 Installation

Le service est déjà présent. Il faut simplement le configurer. On fera `[man quota]` pour découvrir le service.

11.3 Les fichiers de configuration

Pour l'installation du service il est nécessaire d'être "root" (bureau 4). Il faut créer un fichier "aquota.user" sous la racine :

```
[root@tahe util]# >/aquota.user
```

```
[root@tahe util]# ls -l /aquota.user
-rw-r--r--  1 root    root          0 avr  1 17:31 /aquota.user
```

Editez le fichier `/etc/fstab` et pour la ligne correspondant à la partition Linux ajouter à la fin `[defaults,usrquota]` :

```
LABEL=/          /          ext3    defaults,usrquota    1 1
none            /dev/pts  devpts  gid=5,mode=620 0 0
```

Les quotas seront ainsi mis en oeuvre sur la partition supportant la racine `/` de l'arborescence (`/dev/hdb1` par exemple) automatiquement à chaque nouveau boot de la machine. **Vérifiez bien que vous n'avez pas fait d'erreur dans la modification de la ligne. Il arrive qu'un mauvais fichier `/etc/fstab` empêche la machine de rebooter.** Pour connaître la partition qui supporte l'arborescence du système, utilisez la commande `df` :

```
[root@tahe util]# df
Système de
fichiers      1K-blocs   Utilisé Disponible U.% Monté sur
/dev/hdb2      16128668   2862712  12446644  19% /
none          127724    0      127724   0% /dev/shm
```

Ci-dessus, on voit que l'arborescence du système est sur la partition `/dev/hdb2`. Rebootez la machine pour que les quotas soient mis en oeuvre.

11.4 Collecte des espaces disques occupés

Les quotas s'appliquent à une partition. Le fichier `/aquota.user` créé précédemment va mémoriser la taille des espaces disques mémorisés par les différents utilisateurs sur la partition `/dev/hdb2`. Nous pouvons forcer le système à vérifier les espaces disques occupés avec la commande `quotacheck` :

```
[root@tahe util]# /sbin/quotacheck -m /dev/hdb2
```

```
[root@tahe util]# ls -l /aquota.user
-rw-r--r--  1 root    root      9216 avr  1 17:53 /aquota.user
```

La commande `quotacheck` ci-dessus a collecté l'information sur les espaces disque occupés sur `/dev/hdb2` et l'a mise dans le fichier `/aquota.user`. Remplacez `/dev/hdb2` par la partition qui supporte l'arborescence de votre système. Une fois cette information collectée, elle est disponible pour certaines commandes. La commande `repquota` permet de connaître les quotas disque des utilisateurs :

```
[root@tahe util]# /usr/sbin/repquota -a
*** Report for user quotas on device /dev/hdb2
Block grace time: 7days; Inode grace time: 7days
```

User	used	Block limits			File limits			
		soft	hard	grace	used	soft	hard	grace
root	-- 2402124	0	0		118277	0	0	
daemon	-- 8	0	0		3	0	0	
lp	-- 256	0	0		23	0	0	
nobody	-- 244336	0	0		1717	0	0	
ntp	-- 12	0	0		4	0	0	
vcsa	-- 0	0	0		128	0	0	

Ci-dessus, la commande **repquota -a** donne les espaces disque (en K octets) occupés par les différents utilisateurs. Pour avoir celui occupé par l'utilisateur *uti1*, on tapera :

```
[root@tahe uti1]# /usr/sbin/repquota -a | grep -i uti1
uti1      --      9664          0          0          949          0          0
```

Maintenant, comment fixer les quotas disque des utilisateurs ? En utilisant la commande **setquota** :

```
[root@tahe uti1]# /usr/sbin/setquota -u uti2 1000 1000 0 0 /dev/hdb2
```

La commande ci-dessus fixe les quotas disque pour l'utilisateur *uti2* (-u uti2). Les quatres valeurs qui suivent sont

1. quota en Koctets (1000) – limite hard
2. limite soft en Koctets (1000). Est inférieure à la limite hard. Lorsque la limite soft est dépassée, l'utilisateur reçoit un avertissement qu'il s'approche de la limite hard.
3. nombre maxi de fichiers (0) – limite hard – 0 veut dire pas de limites
4. nombre maxi de fichiers (0) – limite soft – 0 veut dire pas de limites. Lorsque la limite soft est dépassée, l'utilisateur reçoit un avertissement qu'il s'approche de la limite hard.

On peut vérifier ensuite que les quotas de l'utilisateur *uti2* ont été mis en place :

```
[root@tahe uti1]# /usr/sbin/repquota -a | grep -i uti2
uti2      --       780      1000      1000          50          0          0
```

Faites le test suivant. Dans le bureau 2, ouvrez un terminal sous l'identité *uti2* :

```
[uti1@tahe uti1]$ su - uti2
Password:
```

```
[uti2@tahe uti2]$ id
uid=502(uti2) gid=501(groupe1) groupes=501(groupe1)
```

```
[uti2@tahe uti2]$ pwd
/home/uti2
```

```
[uti2@tahe uti2]$ ls -l
total 0
```

On voit ci-dessus que le répertoire personnel de **uti2** est vide. Copiez-y le fichier **/boot/vmlinuz** qui est le noyau du système :

```
[uti2@tahe uti2]$ ls -l /boot/vmlinuz
lrwxrwxrwx  1 root  root          21 mar 21 09:44 /boot/vmlinuz -> vmlinuz-2.4.18-27.8.0
[uti2@tahe uti2]$ ls -l /boot/vmlinuz-2.4.18-27.8.0
-rw-r--r--  1 root  root    1112185 mar 14 13:04 /boot/vmlinuz-2.4.18-27.8.0
```

Ci-dessus, on voit que **/boot/vmlinuz** est un lien (sous windows on dirait un raccourci) sur un autre fichier qui fait un peu plus de 1 Mo. L'utilisateur *uti2* ne devrait donc pas pouvoir le copier sous son répertoire personnel puisque celui-ci a été limité à 1 Mo. Essayons :

```
[uti2@tahe uti2]$ cp /boot/vmlinuz /home/uti2/vmlinuz1
ide0(3,66): write failed, user block limit reached.
cp: écriture de `/home/uti2/vmlinuz1': Débordement du quota d'espace disque
```

Si cela n'a pas fonctionné, assurez-vous que les quotas sont actifs avec la commande **quotaon** :

```
[root@tahe uti1]# /sbin/quotaon /dev/hdb2
```

La commande **quotaon** est automatiquement activée au boot lorsque des partitions sont montées avec des quotas. Seulement ici, au boot le fichier **/aquota.user** n'était pas encore construit. Il est alors possible que les quotas n'aient pas été activés.

11.5 Commentaires

Les services précédents permettent de construire la configuration suivante :

- Chacun de vos utilisateurs aura un compte lui donnant l'accès au courrier électronique
- Ce compte peut être interdit d'accès *telnet* si vous lui attribuez l'interpréteur */bin/false* (cf commande Unix *chsh*), ce qui sécurisera un peu votre système.
- L'utilisateur pourra néanmoins gérer son courrier
- Le service *ftp* lui permettra de stocker des fichiers de façon sûre dans son répertoire personnel puisque l'accès *ftp* est contrôlé par mot de passe.
- Il pourra également développer des pages Web dans son répertoire personnel, pages qui seront accessibles depuis l'Internet.

La possibilité pour un étudiant d'avoir un espace disque sécurisé sur le réseau est utile :

- Ils ont parfois des gros fichiers qui ne tiennent pas sur des disquettes.
- Avoir un répertoire accessible de n'importe quel poste leur permet de travailler n'importe où, même de l'extérieur si vous avez autorisé l'accès ftp de l'extérieur.
- L'accès par mot de passe sécurise leur répertoire.

Pour l'administrateur, se pose le problème de contrôler la taille de ces répertoires. Le risque que certains y mettent n'importe quoi trouvé sur le réseau et notamment des jeux est grande. Le service des Quotas est là pour contrôler la taille des répertoires.

12 Filtrer l'accès aux services lancés par xinetd

12.1 Fonction

Contrôler l'accès aux services déclarés dans les fichiers `/etc/services` et `/etc/xinetd.conf`.

Nous avons vu qu'un certain nombre de services sont accessibles via le super-serveur `xinetd`. Ces services sont tout d'abord dans `/etc/services`. Parmi ceux-ci seuls ceux présents dans le fichier `/etc/xinetd.conf` sont réellement utilisables. Revenons par exemple sur la déclaration du service **pop** :

Dans le fichier `/etc/services`, nous avons la ligne

```
grep -i pop /etc/services
pop3          110/tcp          pop-3          # POP version 3
```

et dans le fichier `/etc/xinetd.conf`:

```
more /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances                = 60
    log_type                 = SYSLOG authpriv
    log_on_success           = HOST PID
    log_on_failure           = HOST RECORD
}

includedir /etc/xinetd.d
```

C'est dans le répertoire `/etc/xinetd.d` qu'on trouve la déclaration du service `pop3` :

```
# cd /etc/xinetd.d
# more ipop3

# default: off
# description: The POP3 service allows remote users to access their mail \
#              using an POP3 client such as Netscape Communicator, mutt, \
#              or fetchmail.
service pop3
{
    socket_type              = stream
    wait                    = no
    user                     = root
    server                   = /usr/sbin/ipop3d
    log_on_success           += USERID
    log_on_failure           += USERID
    disable                  = no
}
```

Ci-dessus, la propriété `disable=yes/no` permet d'inhiber ou non le service. Parfois, on a besoin d'un service (`disable=no`) mais on voudrait le restreindre à certaines personnes ou certaines machines. Avant de lancer un service, le super-serveur `xinetd` :

- consulte le fichier `/etc/hosts.allow` pour voir si la machine cliente demandant le service est autorisée à le faire. Si oui, le service demandé est lancé. Dans notre exemple ci-dessus, le programme `/usr/sbin/ipop3d` serait lancé.
- Si le client n'est pas dans la liste des machines autorisées à utiliser le service, un second fichier est consulté : `/etc/hosts.deny`. Ce fichier indique explicitement pour chaque service, les machines interdites. Si la machine cliente est trouvée dans `/etc/hosts.deny`, le service lui est refusé.
- Si la machine cliente n'a été trouvée dans aucun des deux fichiers précédents, elle est autorisée à utiliser le service qui est alors lancé.

12.2 Les fichiers de configuration

Vous devez être “**root**” pour modifier les fichiers “**/etc/hosts.allow**” et “**/etc/hosts.deny**”. Le fichier **hosts.allow** ([**more /etc/hosts.allow**]) pourrait ressembler à ceci (cet exemple n'est pas à recopier) :

```
#
# hosts.allow This file describes the names of the hosts which are
#             allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
ipop3d:127.0.0.1 172.20.81. except 172.20.81.8 :spawn (/bin/echo `/bin/date` - service %d - machine
%a %n accepte >>/var/log/%d.log.allowed)
```

Le fichier **/etc/hosts.allow** décrit les services et leurs conditions d'accès. La syntaxe des lignes est la suivante :

service : machines autorisées [EXCEPT machines interdites] : commande Unix

service

nom du programme chargé d'exécuter le service. Les noms de ces programmes se trouvent dans les fichiers de configuration de **xinetd** dans le dossier **/etc/xinetd.d**.

```
[root@tahe root]# ls -l /etc/xinetd.d/ | grep -i pop
-rw-r--r-- 1 root root 453 sep 1 2002 ipop2
-rw-r--r-- 1 root root 358 mar 7 17:22 ipop3
-rw-r--r-- 1 root root 335 sep 1 2002 pop3s
```

```
[root@tahe root]# more /etc/xinetd.d/ipop3
# default: off
# description: The POP3 service allows remote users to access their mail \
#              using an POP3 client such as Netscape Communicator, mutt, \
#              or fetchmail.
service pop3
{
    socket_type      = stream
    wait            = no
    user            = root
    server           = /usr/sbin/ipop3d
    log_on_success  += HOST DURATION
    log_on_failure  += HOST
    disable         = no
}
```

Dans le dossier **xinetd.d**, il y a le fichier de configuration du service **ipop3**. Dans celui-ci, le paramètre **server** donne le chemin de l'exécutable associé au service *ipop3* : **/usr/sbin/ipop3d**. Dans les fichiers **/etc/hosts.allow** et **/etc/hosts.deny**, on ne garde que le nom de l'exécutable (**ipop3d**) et non son chemin complet.

machine

adresse IP ou nom d'une machine. Pour désigner toutes les machines d'un réseau, on utilise l'adresse du réseau. Ci-dessus, le service **ipop3d** est autorisé pour toutes les machines d'adresse IP 172.20.81.X ainsi que la machine d'adresse IP 127.0.0.1 qui est l'adresse de la machine “localhost”, c'est-à-dire le serveur lui-même. On peut soustraire certaines machines du réseau de cette autorisation avec la clause **EXCEPT**. Ainsi dans l'exemple ci-dessus, la machine 172.20.81.8 a été exclue de l'autorisation.

spawn

Si une machine cliente est autorisée, une commande Unix peut alors être exécutée : c'est le champ 3 de la ligne. Il est facultatif. Dans notre exemple, une commande est exécutée à chaque fois qu'une machine est autorisée à utiliser le service **ipop3d** :

spawn (/bin/echo `/bin/date` - Machine %h acceptée >>/var/log/%d.log.allowed)

Cette commande écrit (**/bin/echo**) la date (**/bin/date**) et le nom de la machine acceptée (**%h**) dans le fichier **/var/log/in.telnetd.log.allowed**. Dans la commande, **%d** est remplacé par le nom du service (ici *ipop3d*) et **%h** par l'adresse IP ou le nom de la machine cliente.

Le fichier **hosts.deny** ([**more /etc/hosts.deny**]) a une syntaxe identique :

```
#
# hosts.deny This file describes the names of the hosts which are
#             *not* allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow. In particular
# you should know that NFS uses portmap!

all:all:spawn (/bin/echo `/bin/date` - service %d - machine %a %n refuse >>/var/log/%d.log.denied)
```

La ligne précédente indique que pour tous les services (1^{er} champ **ALL**), toutes les machines (second champ **ALL**) sont interdites. Le 3^{ème} champ fait que le nom de toute machine refusée sera inscrit dans un fichier **/var/log/service.log.denied**, où **service** sera remplacé par le nom réel du service.

Rappelons le fonctionnement du filtre :

- La machine cliente est cherchée dans **hosts.allow**. Si elle y est trouvée le service demandé est lancé.
- Sinon, elle est cherchée dans **hosts.deny**. Si elle y est trouvée, le service lui est refusé.
- Si elle n'est trouvée nulle part, le service lui est accordé.

On voit que la stratégie adoptée dans les deux fichiers exemples **hosts.allow** et **hosts.deny** précédents, est que toute machine non explicitement autorisée à utiliser un service dans **hosts.allow**, sera alors interdite de service par **hosts.deny**. De plus, on pourra retrouver sa trace dans un fichier **/var/log/service.log.denied**.

12.3 Démarrage du service

La modification des fichiers “**hosts.allow**” et “**hosts.deny**” n’entraîne aucune réinitialisation : elle est immédiatement prise en compte.

12.4 Test 1

Le service *vsftpd* est défini dans le fichier **/etc/xinetd.conf** (en fait **/etc/xinetd.d/vsftpd**) de la façon suivante :

```
# default: on
# description: The wu-ftp FTP server serves FTP connections. It uses \
# normal, unencrypted usernames and passwords for authentication.
service ftp
{
    socket_type          = stream
    wait                = no
    user                 = root
    server               = /usr/sbin/vsftpd
    server_args          = -l -a
    log_on_success       +=  DURATION USERID
    log_on_failure       +=  USERID
    nice                 = 10
}
```

Le nom du programme associé au service ftp est donc *vsftpd*. Il faut s'en souvenir car c'est ce nom qu'il faut utiliser dans les fichiers **/etc/hosts.allow** et **/etc/hosts.deny**. Créez le fichier **/etc/hosts.deny** suivant :

```
ALL: ALL :spawn (/bin/echo `/bin/date` - Service %d - Machine %a %n refusee
>>/var/log/%d.log.denied)
```

Dans la ligne ci-dessus, on n'oubliera pas de mettre une marque de fin de ligne sinon elle serait mal interprétée. Ici, tout couple (service/machine) rejeté par les règles du fichier **/etc/hosts.allow** se verra interdire l'accès au service par les règles du **/etc/hosts.deny**. Laissez le fichier **/etc/hosts.allow** vide. Cela signifie que maintenant tout service lancé par *xinetd* est inaccessible. Lancez dans une fenêtre terminal une session *ftp* :

```
[root@tahe root]# ftp localhost
Connected to localhost (127.0.0.1).
421 Service not available, remote server has closed connection
ftp>
```

La connexion a bien été refusée. Vérifions le fichier de logs qui a du être créé :

```
[root@tahe root]# ls -l /var/log/vsftpd.log.denied
-rw-r--r-- 1 root root 97 avr 29 16:01 /var/log/vsftpd.log.denied
```

```
[root@tahe root]# more /var/log/vsftpd.log.denied
Tue Apr 29 16:01:53 CEST 2003 - Service vsftpd - Machine 127.0.0.1 localhost.localdomain refusee
```

Vous devez obtenir quelque chose d'analogue à ci-dessus. Autorisez maintenant votre machine dans */etc/hosts.allow* :

```
vsftpd:127.0.0.1:spawn (/bin/echo `/bin/date` - service %d - machine %a %n acceptee
>>/var/log/%d.log.allowed)
```

Dans la ligne ci-dessus (il n'y en a pas deux), on n'oubliera pas de mettre une marque de fin de ligne sinon elle serait mal interprétée. Refaites une session ftp locale :

```
[root@tahe root]# ftp localhost
Connected to localhost (127.0.0.1).
220 ready, dude (vsFTPD 1.1.0: beat me, break me)
Name (localhost:util): util
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Ici, la connexion est bien acceptée. Vérifiez le fichier de logs qui a du être créé :

```
[root@tahe root]# ls -l /var/log/vsftpd.log.allowed
-rw-r--r-- 1 root root 97 avr 29 16:11 /var/log/vsftpd.log.allowed
```

```
[root@tahe root]# more /var/log/vsftpd.log.allowed
Tue Apr 29 16:11:30 CEST 2003 - service vsftpd - machine 127.0.0.1 localhost.localdomain acceptee
```

12.5 Test 3

Allez sur une autre machine et tentez de faire une session ftp sur votre serveur. Constatez que le service vous est refusé. Allez sur votre serveur pour faire **[more /var/log/vsftpd.log.denied]** et constatez qu'on y trouve une ligne indiquant que telle machine a été refusée à telle heure. Mettez maintenant dans */etc/hosts.allow* la ligne suivante (ou analogue) :

```
vsftpd:127.0.0.1 172.20.81. except 172.20.81.8 :spawn (/bin/echo `/bin/date` - service %d - machine
%a %n acceptee >>/var/log/%d.log.allowed)
```

Cette ligne donne accès au service *vsftpd* à toutes les machines du réseau 172.20.81. (réseau étudiant) sauf à la machine 172.20.81.8. Remplacez cette dernière par une machine de la salle où vous êtes. Faites des tests.

12.6 Commentaires

Une bonne politique de sécurisation est "Tout ce qui n'est pas autorisé explicitement est interdit". Ce qui amène à réduire le fichier */etc/hosts.deny* à la seule ligne :

```
ALL: ALL :spawn (/bin/echo `/bin/date` - Service %d - Machine %a %n refusee
>>/var/log/%d.log.denied)
```

Cette ligne refuse tout accès aux services non acceptés explicitement dans le fichier */etc/hosts.allow*. Par ailleurs, les machines refusées sont enregistrées dans un fichier de logs : */var/log/%d.log.denied* où *%d* représente le nom du service demandé par la machine refusée (in.telnetd, in.ftpd,...). En tant qu'administrateur système, vous devez suivre régulièrement ces fichiers de logs afin de voir si des machines ne reviennent pas régulièrement frapper à votre porte, auquel cas il faudrait les surveiller particulièrement.

Avec cette politique, vous devez explicitement déclarer les services dont vous autorisez l'accès et à qui vous l'autorisez. Ceci se fait dans le fichier */etc/hosts.allow*.

Le service POP doit être autorisé pour tous. En effet, si vous avez des étudiants qui partent en stage, ils voudront en général continuer à lire leur courrier depuis leur lieu de stage. Le service ftp peut également être considéré comme utile aux étudiants en stage : cela leur permet de garder l'accès à leur répertoire Unix et notamment à leurs pages Web personnelles, de leur lieu de stage. On peut donc autoriser ce service pour tous mais ça se discute.

On se rappellera que seuls les services déclarés dans */etc/services* et */etc/xinetd.conf* peuvent être filtrés. Certains services tels SAMBA ne sont pas déclarés dans */etc/services* et ne peuvent donc être filtrés. Samba peut l'être par des filtres déclarés dans son fichier de configuration *smb.conf*.

Compléments système

13 Les systèmes de fichiers

Les exemples qui suivent utilisent une partition de disques et une configuration machine qui ne sont pas nécessairement les vôtres. Adaptez les exemples à votre situation.

13.1 Les supports physiques des fichiers

Sur votre système Linux, il existe différents disques sur lesquels peuvent être stockés des fichiers. Ces périphériques portent un nom qu'il est utile de connaître. Voici les plus courants :

Périphérique	partition	
/dev/hda		Premier disque dur IDE
	/dev/hda1	Partition primaire 1 du disque /dev/hda
	/dev/hda2	Partition primaire 2 du disque /dev/hda
	...	Un disque peut avoir jusqu'à 4 partitions primaires. Une partition primaire peut ensuite être divisée en partitions logiques, la numérotation de celles-ci commençant alors à 5 (/dev/hda5).
/dev/hdb		Second disque dur IDE
	/dev/hdbi	Partition i du disque /dev/hdb
/dev/sda		Premier disque SCSI
	/dev/sdai	Partition i du disque /dev/sda
/dev/fd0		Premier lecteur de disquettes – pas de partitionnement
/dev/fd1		Second lecteur de disquettes – pas de partitionnement
/dev/cdrom		Lecteur de CD – pas de partitionnement

Il n'est pas toujours facile de connaître les disques installés sur un PC. Vous pouvez vous aider du fichier `/var/log/dmesg` qui est un suivi de la phase de démarrage du serveur Linux. Pendant cette phase, linux détecte tous les périphériques présents sur le PC. Affichons le contenu de ce fichier :

```
[root@tahe util]# more /var/log/dmesg
Linux version 2.4.18-14 (bhcompile@stripples.devel.redhat.com) (gcc version 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)) #1 Wed S
ep 4 13:35:50 EDT 2002
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
 BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
 ....

 BIOS-e820: 00000000ffff0000 - 0000000010000000 (ACPI NVS)
 BIOS-e820: 00000000ffff0000 - 0000000010000000 (reserved)
 OMB HIGHMEM available.
 255MB LOWMEM available.
 On node 0 totalpages: 65532
 zone(0): 4096 pages.
 zone(1): 61436 pages.
 zone(2): 0 pages.
 Kernel command line: initrd=initrd.img root=/dev/hdb2 BOOT_IMAGE=vmlinuz
 Initializing CPU#0
 Detected 803.545 MHz processor.
 Speakup v-1.00 CVS: Tue Jun 11 14:22:53 EDT 2002 : initialized
 ...
```

En consultant attentivement ce fichier, on peut apprendre beaucoup de choses sur sa machine :

```
Kernel command line: initrd=initrd.img root=/dev/hdb2 BOOT_IMAGE=vmlinuz
```

```

// cette ligne indique que linux a été installé sur la partition /dev.hdb2
Memory: 253044k/262128k available (1326k kernel code, 6652k reserved, 999k data, 212k init, 0k
highmem)
// la quantité de mémoire disponible
Linux NET4.0 for Linux 2.4
// la version réseau de linux installée
hda: WDC WD200BB-00AUA1, ATA DISK drive
hdb: WDC WD200BB-00AUA1, ATA DISK drive
hdd: MATSHITADVD-ROM SR-8586, ATAPI CD/DVD-ROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
ide1 at 0x170-0x177,0x376 on irq 15
hda: 39102336 sectors (20020 MB) w/2048KiB Cache, CHS=2434/255/63, UDMA(66)
hdb: 39102336 sectors (20020 MB) w/2048KiB Cache, CHS=2434/255/63, UDMA(66)
ide-floppy driver 0.99.newide
Partition check:
 hda: hda1
 hdb: hdb1 < hdb5 > hdb2 hdb3
Floppy drive(s): fd0 is 1.44M

// les disques détectés

```

Dans le contenu du fichier `/var/log/dmesg`, repérez les noms des disques durs, le partitionnement de chacun d'eux, le nom du lecteur Zip s'il y en a un, celui du lecteur de CD s'il y en a un.

13.2 Monter des systèmes de fichiers

Il est possible sous Linux de “ voir ” différents systèmes de fichiers, celui de linux lui-même bien sûr (type ext2, ext3) mais bien d'autres encore dont les systèmes de fichiers des systèmes DOS (type msdos) et NT (type ntfs). Ces systèmes de fichiers peuvent être sur différents types de supports physiques : disques IDE, disques SCSI, CDROM, disques ZIP, disquettes,... L'intégration dans l'arborescence des fichiers du serveur linux d'un système de fichiers présent sur un support physique s'appelle une opération de montage. On peut distinguer deux types de montage :

- Les montages faits automatiquement au démarrage par lecture du fichier `/etc/fstab`
- Les montages faits à la main avec la commande **mount**

13.2.1 Montages faits au démarrage du système

Au démarrage initial du système, l'arborescence des fichiers est construite par de tels montages guidés par le contenu du fichier `/etc/fstab` :

```

[root@tahe util]# more /etc/fstab
LABEL=/          /          ext3          defaults,usrquota          1 1
none            /dev/pts   devpts       gid=5,mode=620 0 0
none            /proc     proc         defaults          0 0
none            /dev/shm  tmpfs       defaults          0 0
/dev/hdb3       swap      swap         defaults          0 0
/dev/cdrom      /mnt/cdrom iso9660      noauto,owner,kudzu,ro 0 0
/dev/fd0        /mnt/floppy auto         noauto,owner,kudzu 0 0

```

Dans cet exemple, on voit que la partition de swap a été installée sur la partition `/dev/hda4`. On ne voit pas sur quelle partition a été montée la racine `/`. Pour le savoir, on peut utiliser la commande **mount** qui indique quels systèmes de fichiers sont actuellement montés :

```

[root@tahe util]# mount
/dev/hdb2 on / type ext3 (rw,usrquota)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)

```

On voit ici que la racine de l'arborescence a été montée sur la partition `/dev/hdb2`.

Afin de connaître le rôle précis du fichier `/etc/fstab`, on pourra faire :

[man /etc/fstab]

Chaque ligne du fichier `/etc/fstab` a 6 champs :

- 1 Support physique contenant le système de fichiers à monter
- 2 Point de montage dans l'arborescence générale du système de fichiers
- 3 Type du système de fichiers à monter
- 4 Options de montage. Elles sont diverses. A retenir :
defaults : options de montage par défaut, notamment le montage automatique au moment du démarrage
noauto : le montage n'est pas fait au démarrage. Néanmoins, les caractéristiques du montage sont enregistrées et faciliteront un montage ultérieur.
- 5 Valeur utilisée par l'utilitaire dump. On mettra 1 pour les systèmes de fichiers linux montés de façon permanente, 0 pour tous les autres.
- 6 Une valeur `<> 0` indique que le système de fichiers doit faire l'objet d'une vérification d'intégrité lors du montage. On mettra 1 pour le système de fichiers principal de linux (`/`), 2 pour les autres qui doivent être vérifiés, 0 pour tout le reste.

Ainsi la ligne

```
/dev/hda3 / ext2 defaults 1 1
```

signifie que la partition `/dev/hda3` contenant un système de fichiers de type **ext2** (linux) sera incluse à la racine `/` de l'arborescence du système de fichiers du serveur. Lors du montage, le système de fichiers sera vérifié (**1** en colonne 6).

La ligne

```
/dev/hdb3 swap defaults 0 0
```

est une ligne particulière pour la partition de swap. Son système de fichiers swap (colonne 3) n'est pas monté dans l'arborescence générale (swap en colonne 2), ne peut faire l'objet d'un dump (0 en colonne 5) et n'est pas vérifié au montage (0 en colonne 6).

La ligne

```
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
```

décrit le montage du périphérique `/dev/fd0` (disquette) contenant un système de fichiers linux (ext2) au point `/mnt/floppy` de l'arborescence. Ce montage n'est pas fait au démarrage (noauto). Il pourra être fait ultérieurement par la commande `[mount /mnt/floppy]`.

La ligne

```
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
```

décrit le montage du périphérique `/dev/cdrom` (lecteur de cd) au point `/mnt/cdrom` de l'arborescence. Ce périphérique contient un système de fichiers **iso9660** qui est un système de fichiers courant sur les CD. Il n'est pas monté au démarrage (noauto) et est en lecture seule (ro=read only). Une fois qu'un CD aura été placé dans le lecteur, on pourra le "monter" par la commande `[mount /mnt/cdrom]`.

A noter que les points de montage doivent exister avant de faire le montage. Dans notre exemple, le point de montage `/` existe toujours. Les autres points de montage `/mnt/floppy` et `/mnt/cdrom` doivent être créés avec la commande `mkdir` s'ils n'existent pas :

```
[mkdir /mnt/floppy]
[mkdir /mnt/cdrom]
```

On peut monter un système de fichiers n'importe où dans l'arborescence. Beaucoup d'administrateurs ont pris l'habitude de les monter sous le répertoire `/mnt`.

13.2.2 Montages manuels

Les montages manuels se font avec la commande **mount**. On fera `[man mount]` pour une description complète de cette commande. Elle n'est utilisable que par l'utilisateur *root* c.a.d. l'administrateur du système. La syntaxe de la commande `mount` suit la syntaxe des lignes du fichier `/etc/fstab` :

```
mount [-t type_fichiers] [périphérique/partition] [point_de_montage]
```

L'option `-t` désigne le type de fichiers à monter. Les plus courants sont

```
ext2, ext3 : système de fichiers linux
msdos : système de fichiers DOS
ntfs : système de fichiers Windows NT
iso9660 : système de fichiers des CD
```

L'option `-t` est facultative. Lorsqu'elle est absente, le type `ext2`, `ext3` est pris par défaut. Lorsqu'on émet la commande `mount`, le répertoire courant ne doit pas être situé sous le point de montage. Il en est de même lorsqu'on démonte le système de fichiers avec la commande `umount`.

Un système de fichiers monté par `mount` pourra être démonté par **umount**. On utilisera la syntaxe :

```
umount périphérique/partition
```

ou

```
umount point_de_montage
```

où les arguments sont ceux utilisés lors de la commande `mount`.

13.2.3 Connaître le partitionnement d'un disque : fdisk

La commande **fdisk** permet de connaître le partitionnement des disques. Elle permet également de partitionner le disque. Ce n'est pas ce qui est cherché ici et il faut donc être prudent en tapant vos commandes. Sur vos machines vous pouvez avoir 1 ou 2 disques durs IDE ainsi qu'un lecteur Zip interne qui est lui aussi considéré comme un disque IDE. Dans ce dernier cas, il faudra mettre une disquette Zip dans le lecteur.

Tapez les commandes suivantes :

```
[fdisk /dev/hda]
```

pour voir le partitionnement du premier disque IDE de votre machine. Faire `[p]` pour avoir la liste des partitions, `[q]` pour quitter.

```
[fdisk /dev/hdb]
```

pour voir le partitionnement du second disque IDE de votre machine. Faire `[p]` pour avoir la liste des partitions, `[q]` pour quitter.

```
[fdisk /dev/hdd] - mettre une disquette ZIP
```

pour voir le partitionnement du troisième disque IDE (lecteur ZIP) de votre machine. Faire `[p]` pour avoir la liste des partitions, `[q]` pour quitter.

13.2.4 Montage d'une partition DOS

Dans l'exemple qui suit, on suppose qu'il existe plusieurs partitions DOS sur le disque IDE `/dev/hda`. La partition primaire `/dev/hda1` représente le disque C. La partition primaire `/dev/hda2` a été divisée en 2 partitions logiques `/dev/hda5` (disque D) et `/dev/hda6` (disque E). Nous allons monter le disque E :

```
[cd /]
```

pour être au plus haut niveau de l'arborescence

```
[mkdir /mnt/E]
```

pour créer le point de montage du disque E

```
[ls -l /mnt/E]
    pour voir le contenu du répertoire – rien normalement
[mount -t msdos /dev/hda6 /mnt/E]
    pour monter le système de fichiers DOS (-t msdos) du disque E (/dev/hda6) dans le répertoire /mnt/E
    de l'arborescence.
[ls -l /mnt/E]
    pour voir le contenu du disque E
[mkdir /mnt/E/rep1]
    pour créer un répertoire rep1 sur le disque E
[ls -l /mnt/E]
    pour vérifier
```

En travaillant maintenant dans le répertoire /mnt/E, vous travaillez en fait sur le disque E du PC donc sur des fichiers de type MS-DOS. C'est un moyen pour échanger des fichiers entre un système Linux et un système DOS.

Lorsqu'on n'a plus besoin du disque E, on pourra le démonter de la façon suivante :

```
[cd /]
    pour être au plus haut niveau de l'arborescence
[umount /mnt/E]
    pour démonter le point de montage du disque E
[ls -l /mnt/E]
    pour voir le contenu du répertoire – rien normalement
```

13.2.5 Montage du CDROM

La ligne du montage du CDROM est normalement présente dans le fichier */etc/fstab*. Vérifiez-le par [more /etc/fstab]. Vous devriez avoir une ligne du genre :

```
/dev/cdrom          /mnt/cdrom          iso9660            noauto,ro          0 0
```

Le CDROM n'est pas monté automatiquement au démarrage (noauto) et est en lecture seule (ro). Comme la ligne de montage est déjà présente dans le fichier */etc/fstab*, la syntaxe de la commande **mount** est simplifiée : [mount point_de_montage].

Mettez un CD dans le lecteur puis tapez les commandes suivantes :

```
[cd /]
    pour être au plus haut niveau de l'arborescence
[ls -l /mnt/cdrom]
    pour voir le contenu du répertoire – le répertoire /mnt/cdrom est normalement créé à l'installation. Si ce
    n'était pas le cas, il faudrait le créer : [mkdir /mnt/cdrom]
[mount /mnt/cdrom]
    pour monter le CDROM. On utilise la syntaxe simplifiée car le montage est déjà décrit dans /etc/fstab.
[ls -l /mnt/cdrom]
    pour voir le contenu du répertoire – vous voyez là le contenu du CD. Pour démonter le CDROM, on
    procédera comme pour le disque E :
```

```
[cd /]
    pour être au plus haut niveau de l'arborescence
[umount /mnt/cdrom]
    pour démonter le point de montage du lecteur de CD
[ls -l /mnt/cdrom]
    pour voir le contenu du répertoire – rien normalement
```

13.2.6 Montage d'une disquette Linux

La ligne du montage du lecteur de disquette est normalement présente dans le fichier `/etc/fstab`. Vérifiez-le par `[more /etc/fstab]`. Vous devriez avoir une ligne du genre :

```
/dev/fd0          /mnt/floppy      ext2             noauto
0 0
```

Il n'est pas monté automatiquement au démarrage (noauto). Comme la ligne de montage est déjà présente dans le fichier `/etc/fstab`, la syntaxe de la commande `mount` est simplifiée : `[mount point_de_montage]`.

Mettez une disquette vierge dans le lecteur de disquettes et tapez les commandes suivantes :

```
[cd /]
    pour être au plus haut niveau de l'arborescence
[mkfs /dev/fd0]
    pour formater la disquette au format linux. Cette opération n'est pas à faire si vous avez mis une disquette
    déjà formatée au format linux.
[mount /mnt/floppy]
    pour monter le lecteur de disquette. On utilise la syntaxe simplifiée car le montage est déjà décrit dans
    /etc/fstab. Si le répertoire /mnt/floppy n'existait pas, il faudrait le créer au préalable.
[ls -l /mnt/floppy]
    demande la liste des fichiers de la disquette – rien normalement
[cp /boot/vmlinuz /mnt/floppy]
    copie un fichier sur la disquette
[ls -l /mnt/floppy]
    pour vérifier
```

Vous pouvez maintenant faire des sauvegardes sur la disquette. On procèdera au démontage de la façon suivante :

```
[cd /]
[umount /mnt/floppy]
```

13.2.7 Montage d'un lecteur ZIP

On s'intéresse ici au montage d'un disque Zip interne IDE pour y placer un système de fichier linux. On suppose ici que le lecteur zip interne s'appelle `/dev/hdd` (4^{ième} disque IDE). Pour le vérifier, consulter le fichier `/var/log/dmesg` par `[more /var/log/dmesg]`. Un lecteur ZIP SCSI interne s'appellera `/dev/sdax` où x est un numéro. Un lecteur ZIP parallèle s'appellera également `/dev/sdax` car il est considéré comme un disque SCSI même s'il n'y a pas de cartes SCSI dans le PC.

Tapez alors les commandes suivantes :

```
[fdisk /dev/hdd]
    // on voit que la partition s'appelle /dev/hdd4.
[mke2fs /dev/hdd4]
    // pour partitionner la partition /dev/hdd4 au format linux.
[mkdir /mnt/zip]
    // pour créer le point de montage s'il n'existait pas
[mount /dev/hdd4 /mnt/zip]
    // pour monter la partition
[ls -l /mnt/zip]
    // pour voir le contenu de la disquette zip.
```

```
total 12
drwxr-xr-x  2 root    root      12288 Apr 19 14:12 lost+found
```

```
[cp /boot/vmlinuz /mnt/zip]
// pour copier un fichier sur la disquette Zip.
```

```
[ls -l /mnt/zip]
// pour voir le contenu du disque zip
```

```
total 619
drwxr-xr-x  2 root    root      12288 Apr 19 14:12 lost+found
-rw-r--r--  1 root    root     617288 Apr 19 14:16 vmlinuz
```

```
[cd /]
[umount /mnt/zip]
// pour démonter le lecteur zip/linux.
```

```
[mkfs -t msdos /dev/hdd4]
// pour formater la disquette zip au format dos.
```

```
[mount -t msdos /dev/hdd4 /mnt/zip]
// pour monter la disquette zip/dos
```

```
[ls -l /mnt/zip]
// pour voir le contenu du disque zip/dos
```

```
total 0
```

```
[cp /boot/vmlinuz /mnt/zip]
// pour copier un fichier sur la disquette Zip.
```

```
[ls -l /mnt/zip]
// pour voir le contenu du disque zip
```

```
total 604
-rwxr-xr-x  1 root    root     617288 Apr 19 14:23 vmlinuz
```

```
[cd /]
[umount /mnt/zip]
// pour démonter le lecteur zip/linux.
```

13.2.8 Les outils mtools

On a montré précédemment comment gérer un disque ZIP au format MSDOS. Tout ce qui a été vu pour le disque ZIP peut être appliqué au lecteur de disquette `/dev/fd0`. Il suffit de remplacer dans l'exemple précédent `/dev/hdd4` par `/dev/fd0`. Mais pour gérer des supports MSDOS, on dispose d'outils plus simples appelés les outils *mtools*. Ceux-ci permettent de gérer les supports MSDOS avec les commandes habituelles du DOS préfixées de la lettre m (mdir au lieu de dir,...). L'aide sur ces outils est obtenue par `[man mtools]`. On trouve là une explication complète sur l'utilisation des mtools. Voici quelques commandes utiles :

```
mformat périphérique
// formate au format MSDOS le périphérique nommé
// mformat a: - formate A
```

```
mmd rep
// crée rep - exemple mmd a:/rep1
```

```
mrd rep
// supprime rep si vide - exemple mrd a:/rep1
```

```
mcd rep
// pour faire de rep le répertoire DOS courant
// exemple mcd z:/rep1
```

```
mdir rep
// affiche le contenu de rep
mdir z:/rep1 - affiche le contenu de z:/rep1
```

mcopy source destination

```
// pour copier un fichier
// mcopy /boot/vmlinuz a:/
// mcopy a:/rep1 z:/rep2
```

mcopy -t source destination

```
// pour copier un fichier texte UNIX vers un fichier texte DOS
// et vice-versa
```

mdel fichier

```
// supprime fichier
// mdel a:/vmlinuz
```

mtype fichier

```
// affiche le contenu du fichier texte fichier
// mtype a:/texte
```

mmove source destination

```
// déplace source dans destination
// mmove a:/rep1/fic1 a:/rep2
```

Le fonctionnement de ces outils est gouverné par le fichier de configuration **/etc/mtools.conf**. Visualisez ce fichier par **[more /etc/mtools.conf]**. Beaucoup de lignes sont commentées. Sur votre PC, plusieurs supports peuvent accueillir des fichiers DOS :

Le disque C, /dev/hda1
 Le disque D, /dev/hda5
 Le disque E, /dev/hda6
 Le lecteur ZIP, /dev/hdd
 Le lecteur de disquette, /dev/fd0

Sauvegardez le fichier de configuration original des *mtools* :

```
[root@tahe util]# cp /etc/mtools.conf /etc/mtools.conf.original
[root@tahe util]# ls -l /etc/mtools.conf*
-rw-r--r-- 1 root root 1913 jun 23 2002 /etc/mtools.conf
-rw-r--r-- 1 root root 1913 avr 4 10:47 /etc/mtools.conf.original
```

Dans les exemples qui vont suivre on va travailler sur une disquette dos. On crée donc le fichier *mtools.conf* suivant :

```
[root@tahe util]# more /etc/mtools.conf
# lecteur A
drive a: file="/dev/fd0" exclusive 1.44m
```

Faites maintenant les tests suivants après avoir placé une disquette vierge dans le lecteur A :

[mformat a:]

```
// formate disquette A
```

[mdir a:]

```
// visualise contenu de A
```

```
[root@tahe util]# mformat a:
```

```
[root@tahe util]# mdir a:
Volume in drive A has no label
Volume Serial Number is 10B7-9DA9
Directory for A:/

No files
1 457 664 bytes free
```

[mmd a:/rep1 a:/rep2]**[mdir a:]**

```
Volume in drive A has no label
Volume Serial Number is 5C24-885F
Directory for A:/
```

```

rep1      <DIR>      04-19-2000  16:51  rep1
rep2      <DIR>      04-19-2000  16:51  rep2
2 files                                0 bytes
1 456 640 bytes free

```

[mcd a:/]

// a:/ sera désormais le répertoire courant DOS

[mmd rep3]

// crée rep3 dans répertoire courant DOS

[mdir]

// contenu du répertoire courant DOS

```

Volume in drive A has no label
Volume Serial Number is 5C24-885F
Directory for A:/

rep1      <DIR>      04-19-2000  16:51  rep1
rep2      <DIR>      04-19-2000  16:51  rep2
rep3      <DIR>      04-19-2000  16:52  rep3
3 files                                0 bytes
1 456 128 bytes free

```

[more /etc/issue]

[mcopy -t /etc/issue a:/issue]

[mdir]

```

Volume in drive A has no label
Volume Serial Number is 5C24-885F
Directory for A:/

rep1      <DIR>      04-19-2000  16:51  rep1
rep2      <DIR>      04-19-2000  16:51  rep2
rep3      <DIR>      04-19-2000  16:52  rep3
issue     68 04-19-2000  16:54  issue
4 files                                68 bytes
1 455 616 bytes free

```

[mtype issue]

```

Red Hat Linux release 6.0 (Hedwig)
Kernel 2.2.5-15 on an i686

```

[mrd a:/rep1 a:/rep2 a:/rep3]

[mdel a:/issue]

[mdir a:/]

```

Volume in drive A has no label
Volume Serial Number is 5C24-885F
Directory for A:/

No files
1 457 664 bytes free

```

13.3 Montage de fichiers exportés par le service samba

Dans le chapitre consacré au service Samba, nous avons vu :

- que le service Samba exportait des services qui sont en fait des arborescences de fichiers
- que ces arborescences de fichiers étaient disponibles à des clients samba. Nous en avons vu deux : le client *smblclient* et le client “Connecter un lecteur réseau” de windows. Ce dernier permet à une machine windows de faire apparaître comme un disque local ce qui est en réalité une ressource distante.

Ce que fait le client windows peut être également réalisé sous Linux avec la commande **smbmount**. Pour découvrir cette commande, faites **smbmount -help** ou man **smbmount** :

```
[root@tahe root]# smbmount --help
Usage: mount.smbfs service mountpoint [-o options,...]
Version 2.2.7-security-rollup-fix

Options:
  username=<arg>          SMB username
  password=<arg>         SMB password
  credentials=<filename> file with username/password
  netbiosname=<arg>     source NetBIOS name
  uid=<arg>              mount uid or username
  gid=<arg>              mount gid or groupname
  port=<arg>             remote SMB port number
  fmask=<arg>           file umask
  dmask=<arg>           directory umask
  debug=<arg>           debug level
  ip=<arg>               destination host or IP address
  workgroup=<arg>       workgroup on destination
  sockopt=<arg>         TCP socket options
  scope=<arg>           NetBIOS scope
  iocharset=<arg>      Linux charset (iso8859-1, utf8)
  codepage=<arg>       server codepage (cp850)
  ttl=<arg>             dircache time to live
  guest                  don't prompt for a password
  ro                     mount read-only
  rw                     mount read-write
```

Nous utiliserons la commande *smbmount* sous la forme suivante :

```
smbmount service point_de_montage -o username=...,password=...
```

Le lecteur est incité à revoir le chapitre sur SAMBA avant de poursuivre. Le service SAMBA étudié dans ce document avait créé un service appelé **private** :

```
; ressources privées
[private]
comment = Logiciels privés
public = no
valid users = util
force group = admprivate
writeable = yes
path = /home/samba/private
create mode = 0775
```

Ce service est d'accès privé (**public=no**) et un seul utilisateur y a accès (**valid users=util**). Le service est associé à l'arborescence */home/samba/private*. C'est cette arborescence qui sera "montée" avec *smbmount*. Pour les tests qui suivent, il faut être *root*. Créons tout d'abord un dossier sous */tmp*.

```
[root@tahe root]# ls -ld /tmp/smbprivate/
drwxr-xr-x  1 root  root  4096 avr 29 17:04 /tmp/smbprivate/
```

```
[root@tahe root]# ls -l /tmp/smbprivate/
total 0
```

Le dossier */tmp/smbprivate* sera le point de montage du service *private* exporté par samba. Vérifions le contenu actuel du dossier */home/samba/private* associé au service *private* :

```
[root@tahe root]# ls -l /home/samba/private/
total 8
-rw-r--r--  1 root  root  2040 mar 31 17:03 passwd
drwxr-xr-x  2 util  admprivate 4096 avr  1 09:07 test
```

Vérifions le nom de notre machine :

```
[root@tahe root]# hostname
tahe.istia.uang
```

Faisons le montage :

```
[root@tahe root]# smbmount \\\tahe.istia.uang\private /tmp/smbprivate/ -o
username=util,password=azerty
[root@tahe root]# ls -l /tmp/smbprivate/
total 6
```

```
-rwxr-xr-x 1 root root 2040 mar 31 17:03 passwd
drwxr-xr-x 1 root root 4096 avr 1 09:07 test
```

On voit ci-dessous que le dossier `/home/samba/private` a bien été monté sous `/tmp/smbprivate`. On remarquera cependant que les propriétaires des fichiers ont changé. Les fichiers sont devenus propriété de celui qui a monté le service. La commande `smbmount` précédente s'est présentée au service samba comme l'utilisateur **uti1**. Celui-ci a un droit d'écriture sur l'arborescence du service `private` (**writeable=yes**). Vérifions-le :

```
[root@tahe root]# echo 1>/tmp/smbprivate/test1
```

La commande ci-dessus crée un fichier dans `/tmp/smbprivate`. Vérifions :

```
[root@tahe root]# ls -l /tmp/smbprivate/
total 7
-rwxr-xr-x 1 root root 2040 mar 31 17:03 passwd
drwxr-xr-x 1 root root 4096 avr 1 09:07 test
-rwxr-xr-x 1 root root 1 avr 29 17:41 test1
```

Le fichier `test1` a bien été créé. Vérifions maintenant le contenu de l'arborescence `/home/samba/private` qui a été montée :

```
[root@tahe root]# ls -l /home/samba/private/
total 12
-rw-r--r-- 1 root root 2040 mar 31 17:03 passwd
drwxr-xr-x 2 uti1 admprivate 4096 avr 1 09:07 test
-rwxrw-r-- 1 uti1 admprivate 1 avr 29 17:41 test1
```

On découvre que le véritable propriétaire du fichier `test1` est **uti1** qui est le nom sous lequel on a monté le service `private`. Le groupe propriétaire est **admprivate** à cause de la ligne **force group=admprivate** du fichier `smb.conf`.

Pour démonter le service `private` on utilise la commande **umount** :

```
[root@tahe root]# umount /tmp/smbprivate/
[root@tahe root]# ls -l /tmp/smbprivate/
total 0
```

Les exemples précédents montrent qu'une machine A pourrait exporter une arborescence vers une machine B de la façon suivante :

- la machine A crée un service Samba associé à l'arborescence à exporter. Ce service serait un service privé auquel on se connecterait avec un login/mot de passe.
- la machine B "monterait" l'arborescence exportée par la machine A avec la commande `smbmount` comme il a été vu précédemment. Ceci fait, l'arborescence exportée par la machine A est vue comme une arborescence locale par la machine B.

Par défaut, seul l'utilisateur `root` peut utiliser la commande comme le montre l'exemple suivant :

```
[uti2@tahe uti2]$ mkdir /tmp/smbuti2
```

```
[uti2@tahe uti2]$ ls -ld /tmp/smbuti2/
drwxr-xr-x 2 uti2 groupe1 4096 avr 30 15:36 /tmp/smbuti2/
```

```
[uti2@tahe uti2]$ ls -l /tmp/smbuti2/
total 0
```

```
[uti2@tahe uti2]$ smbmount \\\tahe.istia.uang\\uti2 /tmp/smbuti2/ -o username=uti2,password=azerty
smbmnt must be installed suid root for direct user mounts (502,502)
smbmnt failed: 1
```

Les commandes précédentes :

- sont exécutées par l'utilisateur **uti2**
- créent un point de montage `/tmp/smbuti2`
- essaient de monter le service `\\tahe.istia.uang\\uti2` sur le point de montage.

L'arborescence associée au service `\\\\tahe.istia.uang\uti2` est le répertoire personnel de **uti2**. On voit que le montage échoue parce que seul *root* peut utiliser *smbmount*. Le message d'erreur nous indique que le fichier **smbmnt** devrait être “**suid root**”. Lorsqu'un fichier exécutable a le droit “suid”, il s'exécute dans un processus qui aura le même propriétaire que celui du fichier exécuté alors que normalement il s'exécute sous le nom de celui qui lance le fichier exécutable. Regardons les droits du programme **smbmnt** :

```
[root@tahe root]# ls -l /usr/bin/smbmnt
-rwxr-xr-x  1 root  root      560774 avr  6 07:47 /usr/bin/smbmnt
```

On voit que **smbmnt** appartient à l'utilisateur *root*. Lorsque l'utilisateur *uti2* l'exécute, un processus est créé et *uti2* en est le propriétaire. Tout ce que fera le processus **smbmnt** sera fait sous le nom *uti2*. Or **smbmnt** doit s'exécuter dans un processus propriété de *root*. C'est pourquoi *uti2* n'a pu utiliser **smbmnt**. Donnons maintenant le droit **suid** au fichier **smbmnt** :

```
[root@tahe root]# chmod u+s /usr/bin/smbmnt
[root@tahe root]# ls -l /usr/bin/smbmnt
-rwSr-xr-x  1 root  root      560774 avr  6 07:47 /usr/bin/smbmnt
```

On voit apparaître un droit S dans les droits du propriétaire du fichier *smbmnt*. Ce droit signifie que lorsque *smbmnt* s'exécutera, le processus dans lequel il s'exécutera sera propriété non pas de l'utilisateur ayant lancé la commande *smbmnt* mais de l'utilisateur propriétaire du fichier lui-même. Ici, il faut faire la différence entre fichier et processus qui n'ont pas forcément les mêmes propriétaires. Le propriétaire du fichier *smbmnt* est *root*. Le droit SUID fait que le processus dans lequel s'exécutera le programme *smbmnt* sera également propriété de *root*. L'utilisateur **uti2** devrait maintenant pouvoir utiliser *smbmount*. Essayons :

```
[uti2@tahe uti2]$ smbmount \\\tahe.istia.uang\uti2 /tmp/smbuti2/ -o username=uti2,password=azerty
```

```
[uti2@tahe uti2]$ ls -l /tmp/smbuti2/
total 1
-rwxr-xr-x  1 uti2  groupe1      1 avr 30 15:36 test
```

Cette fois-ci, le montage a réussi. Créons un fichier dans l'arborescence montée :

```
[uti2@tahe uti2]$ echo 1>/tmp/smbuti2/test1
[uti2@tahe uti2]$ ls -l /tmp/smbuti2/
total 1
-rwxr-xr-x  1 uti2  groupe1      1 avr 30 15:36 test
-rwxr-xr-x  1 uti2  groupe1      1 avr 30 16:04 test1
[uti2@tahe uti2]$ ls -l /home/uti2
total 8
-rw-r--r--  1 uti2  groupe1      1 avr 30 15:36 test
-rwxr--r--  1 uti2  groupe1      1 avr 30 16:04 test1
```

On voit que le fichier *test1* créé dans l'arborescence */tmp/smbuti2* a été en réalité écrit dans */home/uti2*. On a là un mécanisme permettant à un utilisateur de rendre local, un dossier qui lui appartiendrait et qui serait sur une autre machine. L'arborescence montée avec **smbmount** sera démontée avec **smbumount**. Cette commande doit elle aussi avoir le droit *suid* pour pouvoir être utilisée par quelqu'un d'autre que *root* :

```
[root@tahe root]# chmod u+s /usr/bin/smbumount
[root@tahe root]# ls -l /usr/bin/smbumount
-rwSr-xr-x  1 root  root      560448 avr  6 07:47 /usr/bin/smbumount
```

Ceci fait, l'utilisateur *uti2* peut utiliser *smbumount* pour démonter l'arborescence qu'il a montée :

```
[uti2@tahe uti2]$ smbumount /tmp/smbuti2/
[uti2@tahe uti2]$ ls -l /tmp/smbuti2/
total 0
[uti2@tahe uti2]$ ls -l /home/uti2/
total 8
-rw-r--r--  1 uti2  groupe1      1 avr 30 15:36 test
-rwxr--r--  1 uti2  groupe1      1 avr 30 16:04 test1
```

14 La recherche rapide de fichiers

14.1 Introduction

L'outil que nous présentons ici est un peu similaire à l'outil "Rechercher un fichier" qu'on trouve sur les systèmes Windows. Son but est de créer des bases contenant la liste des fichiers présents dans une arborescence afin de faciliter ultérieurement la recherche dans cette arborescence.

14.2 La commande locate

La commande `locate` permet de localiser rapidement un fichier.

La commande

`locate -u`

construit une base de données contenant la liste de tous les fichiers de l'arborescence de fichiers commençant à la racine.

La commande

`locate -U chemin`

fait la même chose mais pour une arborescence définie par *chemin*.

Dans les deux cas, la commande produit une base **`slocate.db`** dans **`/var/lib/slocate`**. Essayez par exemple les commandes suivantes :

```
[root@tahe util]# locate -u

[root@tahe util]# ls -l /var/lib/slocate/
total 1984
-rw-r----- 1 root      slocate  2025485 avr  4 11:00 slocate.db
```

La commande `locate` permet ensuite de faire des recherches dans la base générée.

```
[root@tahe util]# locate kwrite
/var/lib/menu/kde/Applications/Extras/Accessories/kde-kwrite.desktop
/root/.kde/share/config/kwriterc
/usr/bin/kwrite
/usr/bin/kwrited
/usr/lib/kwrite.la
....
```

La commande `locate` permet de faire des interrogations un peu plus sophistiquées que celle ci-dessus. On fera un **[man locate]** pour les découvrir. On peut indexer toute arborescence, par exemple le contenu d'un CD. Installez un CD et faites :

```
[root@tahe util]# locate -U /mnt/cdrom/ -o /var/lib/slocate/slocate.vsnnet
// l'option -U <dir> permet de préciser le répertoire à indexer
// l'option -o <fichier> permet de préciser le nom de la base de données à créer

[root@tahe util]# ls -l /var/lib/slocate/
total 4320
-rw-r----- 1 root      slocate  2025485 avr  4 11:00 slocate.db
-rw-r----- 1 root      slocate  353701 avr  4 11:15 slocate.vsnnet

[root@tahe util]# locate -d /var/lib/slocate/slocate.vsnnet disco.exe
/mnt/cdrom/program files/microsoft.net/frameworksdk/bin/disco.exe
// l'option -d permet de préciser la base de données à explorer
```

14.3 Conclusion

La commande *locate* est souvent utile notamment dans le cas d'installation de nouveaux logiciels. Quelquefois on s'interroge sur l'endroit où ont été placés certains fichiers. La commande *locate* peut nous aider à le savoir. Pour qu'elle soit utile, il faut que la base des fichiers soit régulièrement mise à jour. On pourra la mettre à jour toutes les nuits par exemple. On fera cela à l'aide du service *crond* expliqué dans l'une des fiches suivantes.

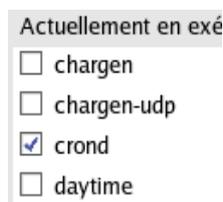
15 Soumettre des travaux à des heures précises

15.1 Introduction

Il est possible d'exécuter des commandes ou des programmes à intervalles réguliers. On utilise souvent cette possibilité pour faire la nuit des opérations longues comme des sauvegardes, mise à jour de la base de fichiers gérée par la commande *locate*, ...

15.2 La commande crontab

Les commandes à exécuter sont définies à l'aide de la commande **crontab**. Elles seront exécutées par le serveur **crond**. Celui-ci est normalement lancé automatiquement au démarrage de linux. Vous pouvez le vérifier avec le gestionnaire de services :



On peut également le lancer à la main par `[/etc/rc.d/init.d/crond start]`. On fera `[man crontab]` et `[man crond]` pour avoir une définition complète du rôle des deux commandes. Commençons par définir un exemple simple. Assurez-vous tout d'abord d'être l'utilisateur *root*.

```
[root@tahe util]# export EDITOR=kwrite
// fixe l'éditeur de texte à utiliser pour modifier le fichier de configuration de crontab
[root@tahe util]# crontab -e &
// lance la modification du fichier de configuration
```

Avec l'éditeur de texte, créez les lignes suivantes :

```
* * * * * date >>/tmp/date
*/2 * * * * (date;who|wc -l) >>/tmp/who
10 23 2 1 * rm -fr /tmp/*
```

Chaque ligne a 6 champs :

Champ 1	minutes * : toutes les minutes */2 : toutes les 2 mn 10 : à la minute 10
Champ 2	heures * : toutes les heures */4 : toutes les 4 h 23 : à 23 h
Champ 3	jour
Champ 4	mois

Champ 1	minutes * : toutes les minutes */2 : toutes les 2 mn 10 : à la minute 10
Champ 5	année
Champ 6	la commande à exécuter

Les lignes précédentes signifient que :

- 1 - la date sera enregistrée dans le fichier `/tmp/date` toutes les minutes
- 2 - la date et le nombre d'utilisateurs seront enregistrés dans le fichier `/tmp/who` toutes les 2 mn
- 3 - chaque année, le 2 janvier à 23 h 10, on détruit toute l'arborescence qui se trouve sous `/tmp`

Sauvegardez votre fichier après l'avoir vérifié. Où ce fichier de commandes est-il sauvegardé ? Dans `/var/spool/cron` comme le montrent les commandes suivantes :

```
[root@tahe util]# ls -l /var/spool/cron/
total 8
-rw----- 1 root root 1230 mar  2 14:07 mailman
-rw----- 1 root root 284  avr  4 09:11 root

[root@tahe util]# more /var/spool/cron/root
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.1342 installed on Fri Apr  4 09:11:45 2003)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
* * * * * date >>/tmp/date
*/2 * * * * (date;(who|wc -l)) >>/tmp/who
10 23 2 1 * rm -fr /tmp/*
```

Pour vérifier ce que font exactement les commandes précédentes, tapez-les au clavier :

```
[root@tahe util]# date
ven avr  4 09:38:35 CEST 2003
[root@tahe util]# date >/tmp/date
[root@tahe util]# more /tmp/date
ven avr  4 09:38:45 CEST 2003
[root@tahe util]# date >>/tmp/date
[root@tahe util]# more /tmp/date
ven avr  4 09:38:45 CEST 2003
ven avr  4 09:38:53 CEST 2003
```

La commande **date** donne la date du jour à l'écran. La commande **date >fic** écrit la date dans le fichier **fic**. Si celui-ci avait déjà un contenu, celui-ci est écrasé. La commande **date >>fic** écrit la date à la fin du fichier **fic**. Le contenu de celui-ci est donc préservé.

```
[root@tahe util]# date
ven avr  4 09:44:18 CEST 2003
[root@tahe util]# who
util      :0          Apr  4 08:21
util     pts/0          Apr  4 08:22
util     pts/2          Apr  4 08:22
util     pts/3          Apr  4 09:20
util     pts/4          Apr  4 09:27
[root@tahe util]# who|wc -l
5
[root@tahe util]# date;(who|wc -l)
ven avr  4 09:44:37 CEST 2003
5
```

La commande **who** affiche la liste des utilisateurs connectés. La commande **date;who** enchaîne les deux commandes **date** et **who** dans cet ordre. La commande **who | wc -l** compte (`wc -l`) le nombre de lignes produites par la commande **who**. Cela donne le nombre d'utilisateurs connectés. La commande **date;(who | wc -l)** enchaîne l'exécution de la commande **date** puis celle de la commande **(who | wc -l)**. Cela permet d'avoir à une heure donnée le nombre d'utilisateurs connectés. Ce type de suivi peut être utile pour détecter les pics de charge dans une journée.

Maintenant, vous pouvez suivre le fonctionnement des commandes précédentes. Elles écrivent dans les fichiers `/tmp/date` et `/tmp/who`. A intervalles réguliers, faites

```
[root@tahe util]# ls -l /tmp/date
-rw-r--r-- 1 root root 180 avr 4 09:17 /tmp/date
```

```
[root@tahe util]# more /tmp/date
Fri Apr 4 09:12:01 CEST 2003
Fri Apr 4 09:13:00 CEST 2003
Fri Apr 4 09:14:00 CEST 2003
Fri Apr 4 09:15:01 CEST 2003
Fri Apr 4 09:16:00 CEST 2003
Fri Apr 4 09:17:00 CEST 2003
```

```
[root@tahe util]# ls -l /tmp/who
-rw-r--r-- 1 root root 152 avr 4 09:18 /tmp/who
```

```
[root@tahe util]# more /tmp/who
Fri Apr 4 09:12:01 CEST 2003
 3
Fri Apr 4 09:14:01 CEST 2003
 3
Fri Apr 4 09:16:00 CEST 2003
 3
Fri Apr 4 09:18:00 CEST 2003
 3
```

On voit clairement que la commande 1 est exécutée toutes les minutes et la commande 2 toutes les 2 minutes. Pour modifier un fichier crontab, il y a diverses manières : refaire l'opération précédente ou **[crontab -r]** pour supprimer définitivement le fichier. L'accès à la commande *crontab* est défini par les fichiers */etc/cron.allow* et */etc/cron.deny*. Si ces fichiers n'existent pas, tout le monde peut créer un fichier crontab. Faisons-en l'expérience.

[rm /etc/cron.allow]

```
// pour effacer cron.allow s'il existait
```

[rm /etc/cron.deny]

```
// pour effacer cron.deny s'il existait
```

Mettez-vous sur une console en tant qu'utilisateur *util* par exemple et émettez les deux commandes suivantes :

```
[util@tahe util]$ export EDITOR=kwrite
[util@tahe util]$ crontab -e &
[1] 1551
[util@tahe util]$ no crontab for util - using an empty one
```

Mettez alors la commande suivante :

```
* * * * * date>>~util/date
```

qui toutes les minutes enregistrera la date dans le fichier *date* du répertoire de connexion de l'utilisateur *util* (*~util*). Alors qu'avec l'utilisateur *root*, on n'a pas à se préoccuper des droits, avec un autre utilisateur il faut le faire. En effet, le serveur *crond* exécute les commandes avec les droits de l'utilisateur propriétaire du fichier *crontab*. Ici ce sera *util*. Dans notre exemple on écrit donc dans un fichier du répertoire de l'utilisateur *util* afin d'être sûr d'avoir les droits d'écriture. Sauvegardez le fichier précédent. Sur une console où vous êtes *root*, faites :

```
[root@tahe util]# ls -l /var/spool/cron/
total 12
-rw----- 1 root root 1230 mar 2 14:07 mailman
-rw----- 1 root root 284 avr 4 09:11 root
-rw----- 1 root groupe1 218 avr 4 09:21 util
```

```
[root@tahe util]# more /var/spool/cron/util
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.1551 installed on Fri Apr 4 09:21:44 2003)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
* * * * * date>>~util/date
```

Revenez maintenant sur la console où vous êtes *util* :

```
[util@tahe util]$ more date
Fri Apr 4 09:22:00 CEST 2003
Fri Apr 4 09:23:00 CEST 2003
[util@tahe util]$ ls -l /home/util/date
```

```
-rw-r--r-- 1 util groupe1 60 avr 4 09:23 /home/util/date
[util@tahe util]$ more /home/util/date
```

```
Fri Apr 4 09:22:00 CEST 2003
Fri Apr 4 09:23:00 CEST 2003
Fri Apr 4 09:24:01 CEST 2003
```

On a bien une exécution toutes les minutes. Il peut ne pas être souhaitable d'autoriser tout le monde à utiliser les possibilités du serveur **crond**. On mettra alors dans le fichier */etc/cron.allow* les logins des utilisateurs autorisés, à raison d'un par ligne.

```
[root@tahe util]# more /etc/cron.allow
root
util
```

Mettez-vous sur une console avec l'identité de *uti1* et faites :

```
# export EDITOR=kwrite
# crontab -e
```

On constate que l'utilisateur **uti1** peut modifier son fichier crontab. Mettez-vous sur une console avec l'identité de *uti2* et faites de même :

```
[uti1@tahe util]$ su - uti2
Password:
```

```
[uti2@tahe uti2]$ export EDITOR=kwrite
```

```
[uti2@tahe uti2]$ crontab -e &
[1] 2019
You (uti2) are not allowed to use this program (crontab)
See crontab(1) for more information
[1]+  Exit 1                  crontab -e
```

Le fichier */etc/cron.deny* n'est consulté que si le fichier */etc/cron.allow* n'existe pas. Dans ce cas, un utilisateur est autorisé à utiliser la commande **crontab** s'il n'est pas présent dans le fichier */etc/cron.deny*. Une règle de sécurité pourrait être qu'on n'autorise que l'utilisateur **root** à utiliser le service **crontab**. Pour cela, on ne met que son nom dans le fichier */etc/cron.allow*.

15.3 Les répertoires hourly, daily, monthly

Il existe trois répertoires appelés */etc/cron.hourly*, */etc/cron.daily*, */etc/cron.weekly*. Les scripts placés dans ces répertoires seront exécutés :

- Toutes les heures s'ils sont dans */etc/cron.hourly*, au changement d'heure
- Tous les jours s'ils sont dans */etc/cron.daily*, au changement de jour
- Toutes les semaines s'ils sont dans */etc/cron.weekly*, au changement de semaine

```
[root@tahe util]# ls -ld /etc/cron*
-rw-r--r-- 1 root root          5 avr 4 09:30 /etc/cron.allow
drwxr-xr-x 2 root root      4096 jui 20  2002 /etc/cron.d
drwxr-xr-x 2 root root      4096 mar  2 14:05 /etc/cron.daily
drwxr-xr-x 2 root root      4096 jun 23  2002 /etc/cron.hourly
drwxr-xr-x 2 root root      4096 mar  2 13:52 /etc/cron.monthly
-rw-r--r-- 1 root root        309 mar  2 14:07 /etc/crontab
drwxr-xr-x 2 root root      4096 mar  2 13:52 /etc/cron.weekly
```

Ainsi si nous plaçons le script **affichdate** suivant dans */etc/cron.hourly* sous l'identité **root** :

```
[root@tahe util]# ls -l /etc/cron.hourly
total 4
-rw-r--r-- 1 root root          18 avr 4 09:34 affichdate
```

```
[root@tahe util]# more /etc/cron.hourly/affichdate
date >>/tmp/date2
```

nous aurons un affichage de la date toutes les heures dans le fichier */tmp/date2*.

15.4 Affichages écran des scripts soumis à crond

Un script soumis à *crond* ne lira pas de données tapées au clavier (il n'y a personne pour les taper la nuit) mais peut faire des affichages écran. Le serveur *crond* les interceptera et les enverra en courrier au propriétaire du script exécuté. Il peut donc être utile de faire des affichages écran dans un script soumis à *crond* afin d'être averti de certains événements, par exemple du bon déroulement ou non du script.

Faites par exemple le test suivant. Sous *root*, créez le fichier *crontab* suivant :

```
[root@tahe util]# more /var/spool/cron/root
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.2375 installed on Fri Apr 4 09:53:44 2003)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
*/5 * * * * date
```

Le script précédent écrit la date et l'heure à l'écran toutes les 5 mn. Soumis à *crond*, cette date/heure sera en fait envoyée à l'utilisateur *root* par mail :

```
☐ Subject: Cron <root@tahe> date
From: Cron Daemon <root@tahe.istia.uang>
Date: 10:00
To: root@tahe.istia.uang

Fri Apr 4 10:00:00 CEST 2003
```

On se rappellera donc que tous les affichages qu'une commande ferait normalement à l'écran sont envoyés par mél à celui qui a soumis les commandes au service **crond**. Il est donc assez facile de créer des systèmes d'alerte :

- on écrit un script dans un langage quelconque qui surveille tel ou tel fonctionnement du système et écrit à l'écran ses messages d'alerte
- on met ce script dans les commandes soumises au service *crond*. Les messages d'alerte seront alors reçus par courrier.

BASES DE DONNEES

- . MySQL
- . Postgres

Nous présentons maintenant l'installation de deux systèmes de gestion de bases de données relationnels (SGBDR) :

- MySQL est un système de gestion de base de données apparu récemment et qui a immédiatement été très utilisé dans le développement Web sous Linux en conjonction notamment avec PHP. Il ne respecte pas l'intégralité de la norme SQL.
- Postgres est le premier SGBD qui ait été distribué avec les distributions Linux. Il respecte la norme SQL et peut être un bon outil de formation aux SGBD en général et au langage SQL en particulier.

16 Le SGBDR MySQL

16.1 Introduction

MySQL est un système de gestion de bases de données relationnelles gratuit distribué par la société TCX. Performant, il permet de s'initier à bon compte à la gestion de bases de données relationnelles et à leur interrogation avec le langage SQL (Structured Query Language). MySQL est couramment utilisé sur les systèmes linux en conjonction avec le langage PHP et le serveur Web Apache pour créer des sites Web dont le contenu est tiré de bases de données.

Il existe des versions de MySQL aussi bien pour Linux que pour Windows. On trouve également pour cette base des pilotes ODBC et JDBC qui permettent à des applications écrites dans des langages divers (C, C++, Java, VB, Perl, ...) d'avoir accès au contenu d'une base MySQL.

Où trouver MySQL ? L'URL du site officiel de MySQL est <http://www.mysql.com>. Pour Linux RedHat, il existe des paquetages RPM pour MySQL qu'on trouvera sur les CD. Sur le site officiel de MySQL, on trouvera de la documentation au format PDF. Il faut la télécharger car elle s'avère indispensable dès qu'on veut comprendre comment gérer les bases de données MySQL.

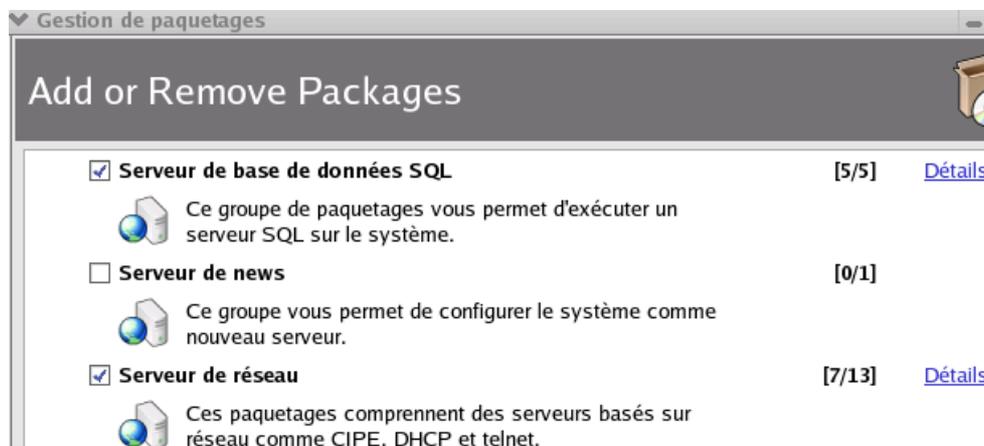
16.2 Installation de MySQL

Vous avez pu déjà installer les paquetages MySQL à l'installation initiale de Linux. Vous pouvez le vérifier avec la commande **rpm** (faire **man rpm**). Cette commande gère les paquetages RPM (Redhat Package Manager).

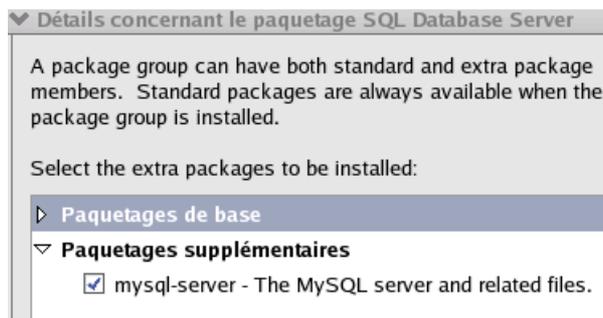
```
[root@tahe util1]# rpm -qa | grep -i mysql
perl-DBD-MySQL-2.1017-3
mysql-server-3.23.52-3
mysql-3.23.52-3
```

La commande **rpm -qa** permet d'avoir la liste des paquetages déjà installés. La commande **rpm -qa | grep -i mysql** liste ceux qui contiennent la chaîne mysql. Dans l'exemple ci-dessus, Le serveur MySQL est installé (mysql-server-3.23.52-3), le client pour ce serveur également (mysql-3.23.52-3) ainsi qu'un module Perl d'accès à la base MySQL.

Si des RPM analogues ne sont pas présents, c'est que MySQL n'est pas installé. Lancez alors, en mode *root*, l'application de gestion des paquetages : **Menu KDE/paramètres de système/Paquetages** :



Cliquez sur le lien **Détails** :



Vérifiez que le paquetage MySQL est coché. Par défaut il ne l'est pas dans la distribution RH8. Ceci fait, cliquez OK autant de fois que nécessaire pour lancer l'installation. Les CD vous seront réclamés.

16.3 Lancement du serveur mysqld

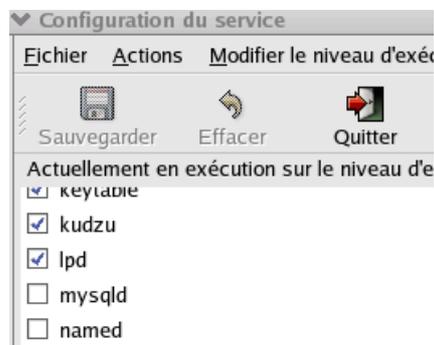
Comme beaucoup de services sous RedHat, il existe des scripts de lancement et d'arrêt du service *mysqld* :

`[/etc/rc.d/init.d/mysqld start]` pour lancer le service
`[/etc/rc.d/init.d/mysqld stop]` pour arrêter le service
`[/etc/rc.d/init.d/mysqld restart]` pour enchaîner arrêt/démarrage

Lançons donc le service *mysqld* :

```
[root@tahe util]# /etc/rc.d/init.d/mysqld start
Démarrage de MySQL : [ OK ]
```

On aurait pu également utiliser le gestionnaire de services : **Menu KDE/Paramètres du serveur/Services** :

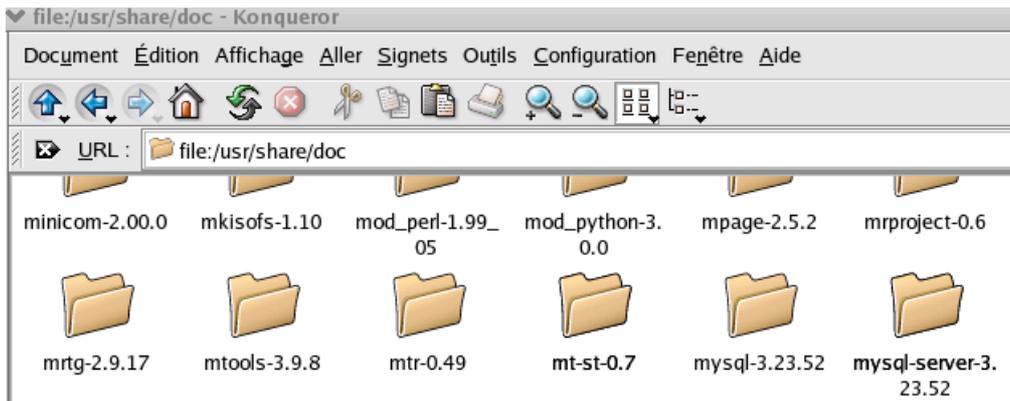


16.4 La documentation de MySQL

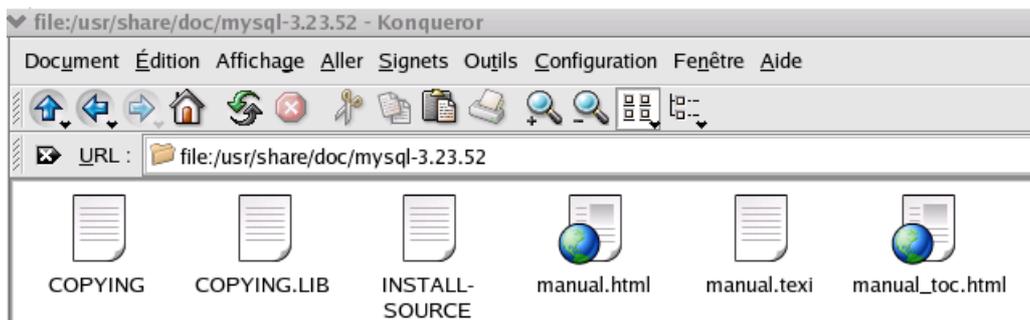
La documentation de mysql est disponibles sous plusieurs formes :

- **man mysqld** donnera des informations sur **mysqld**
- **man mysql** donnera des informations sur **mysql**

Une documentation au format HTML est également disponible dans le dossier `/usr/share/doc` :



Allez dans le dossier du client **mysql** par exemple :



Le document **manual_toc.html** ci-dessus est le point d'entrée d'une documentation HTML bien faite. Utilisez-la pour découvrir comment utiliser les services de MySQL.

16.5 Utilisation du client MySQL

Nous présentons ici un court tutoriel d'utilisation de la base MySQL. L'utilisateur est invité à lire la documentation PDF du produit. Au besoins, lancer le service *mysqld* si ce n'est déjà fait puis vérifier que le démon *mysqld* est actif.

```
[root@tahe util]# ps aux | grep -i mysql
mysql      1275  0.0  1.8 28800 4656 pts/2    S    08:23   0:00 /usr/libexec/mysq
```

Il n'y a nul besoin d'être *root* pour utiliser MySQL. Dans la suite, nous ferons les manipulations sous l'identité Unix **util**. Vous ferez attention au fait que le SGBD MySQL définit ses propres utilisateurs et qu'il n'y a aucun lien entre utilisateur Unix et utilisateur MySQL.

// utiliser le client mysql

```
[util@tahe util]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.52

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

// les commandes tapées au clavier seront maintenant exécutées par le
// programme client mysql. Les commandes sont archivées et récupérables avec les
// flèches [HAUT] et [BAS] du clavier.
// les commandes se terminent par le caractère ;
// lorsque mysql affiche le caractère ->, c'est que le ; n'a pas été tapé
// il faut alors continuer la commande - la saisie se terminera par
// l'omission du caractère ; de fin de commande.
```

```
mysql> help
MySQL commands:
Note that all text commands must be first on line and end with ';'
help      (\h)    Display this help.
```

```

?      (\?)      Synonym for `help`.
clear  (\c)      Clear command.
connect (\r)     Reconnect to the server. Optional arguments are db and host.
edit   (\e)     Edit command with $EDITOR.
ego    (\g)     Send command to mysql server, display result vertically.
exit   (\q)     Exit mysql. Same as quit.
go     (\g)     Send command to mysql server.
nopager (\n)    Disable pager, print to stdout.
notee  (\t)     Don't write into outfile.
pager  (\P)     Set PAGER [to_pager]. Print the query results via PAGER.
print  (\p)     Print current command.
quit   (\q)     Quit mysql.
rehash (\#)     Rebuild completion hash.
source (\.)     Execute a SQL script file. Takes a file name as an argument.
status (\s)     Get status information from the server.
tee    (\T)     Set outfile [to_outfile]. Append everything into given outfile.
use    (\u)     Use another database. Takes database name as argument.

```

```

mysql> show databases;
// affiche les bases de données déjà créées
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.00 sec)

```

```

mysql> create database commerce;
ERROR 1044: Access denied for user: '@localhost' to database 'commerce'

```

Nous avons voulu créer une base de données (un ensemble de tables) appelée **commerce** et le droit nous en a été refusé. vérifions notre identité MySQL :

```

mysql> select user();
+-----+
| user() |
+-----+
| util@localhost |
+-----+

```

Nous sommes identifiés comme l'utilisateur **util** et probablement que la base MySQL n'a pas dans ses tables d'utilisateurs quelqu'un qui s'appelle **util**. Déconnectons-nous et reconnectons-nous de nouveau, cette fois sous l'identité MySQL **root**. Le serveur MySQL a en effet par défaut un utilisateur portant ce nom. C'est le super-utilisateur du SGBD MySQL :

```

mysql> exit
Bye
[util@tahe util]$ mysql --user=root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.23.52

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select user();
+-----+
| user() |
+-----+
| root@localhost |
+-----+

```

Nous sommes le super-utilisateur. Créons la base **commerce** :

```

mysql> create database commerce;
Query OK, 1 row affected (0.10 sec)

```

Créons un utilisateur **mysqlutil** qui aurait tous les droits sur la base **commerce** :

```

mysql> grant all privileges on commerce.* to 'mysqlutil'@'%' identified by 'azerty';
Query OK, 0 rows affected (0.21 sec)

mysql> grant all privileges on commerce.* to 'mysqlutil'@'localhost' identified by 'azerty';
Query OK, 0 rows affected (0.01 sec)

```

Nous avons accordé (**grant**) tous les privilèges sur la base commerce et son contenu (**commerce.***) à un utilisateur appelé **mysqlutil** qui pourra se connecter localement (**'mysqlutil'@'localhost'**) ou à distance de n'importe quelle machine

('mysqlutil'@'localhost'). Il sera identifié par le mot de passe **azerty**. On voit que les droits s'appliquent à un utilisateur qui se connecte à partir d'une machine donnée (syntaxe 'user'@'machine'). Déconnectons-nous pour nous connecter à la base **commerce** sous le nom **mysqlutil** :

```
mysql> exit
Bye
[util@tahe util]$ mysql --user=mysqlutil --database=commerce -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 3.23.52

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

L'option **-p** de la commande `mysql` indique qu'un mot de passe doit être demandé (ici **azerty**). Vérifions notre identité :

```
mysql> select user();
+-----+
| user() |
+-----+
| mysqlutil@localhost |
+-----+
```

Vérifions nos droits :

```
mysql> show databases;
+-----+
| Database |
+-----+
| commerce |
| mysql    |
| test     |
+-----+
```

La commande ci-dessus montre la nouvelle base **commerce**. Essayons de travailler (use) avec la base `mysql` qui est la base réservée à **root** :

```
mysql> use mysql;
ERROR 1044: Access denied for user: 'mysqlutil@localhost' to database 'mysql'
```

L'accès est interdit. C'est la moindre des choses. Aucun utilisateur ne devrait avoir accès à la base **mysql**. C'est l'utilisateur **root** qui a fixé le mot de passe de l'utilisateur **mysqlutil**. Comment celui-ci peut-il changer son mot de passe ?

```
mysql> set password=password('abcd');
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

[util@tahe util]$ mysql --user=mysqlutil --database=commerce --password=abcd
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 3.23.52

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

La commande **set password=password('abcd')** a permis à l'utilisateur **mysqlutil** de fixer son mot de passe à **'abcd'**. On voit qu'il arrive à se reconnecter avec ce nouveau mot de passe. Ici on a utilisé une variante en passant le mot de passe sur la ligne de commande. C'est déconseillé car il peut alors être intercepté.

Par défaut, le super-utilisateur *root* de MySQL n'a pas de mot de passe. L'une des premières chose à faire est donc de lui en fixer un :

```
[util@tahe util]$ mysql --user=root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 3.23.52

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> set password=password('azerty');
Query OK, 0 rows affected (0.00 sec)

mysql> exit
```

```
Bye
```

```
[util@tahe util]$ mysql --user=root --password='azerty'  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 9 to server version: 3.23.52  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql>
```

Maintenant les préparatifs de base sont terminés. On se reconnecte de nouveau sous l'identité **mysqlutil** après avoir remis son mot de passe à **azerty** pour s'en souvenir plus facilement :

```
[util@tahe util]$ mysql --user=mysqlutil --database=commerce --password=azerty  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11 to server version: 3.23.52  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql>
```

Vérifions sur quelle base nous sommes positionnés (notion de base courante) :

```
mysql> select database();  
+-----+  
| database() |  
+-----+  
| commerce  |  
+-----+
```

```
mysql> show tables;  
// affiche les tables de la base courante  
Empty set (0.00 sec)
```

```
mysql> create table articles (code char(4), nom char(20), prix decimal(10,2), stock integer);  
// crée une table articles  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show tables;  
// vérification  
+-----+  
| Tables in commerce |  
+-----+  
| articles            |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> create table clients (code char(4), nom char(20));  
// crée une table clients  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show tables;  
+-----+  
| Tables in commerce |  
+-----+  
| articles            |  
| clients            |  
+-----+  
2 rows in set (0.00 sec)
```

```
mysql> create table ventes (codeClient char(4), codeArticle char(4), quantite integer);  
// table des ventes  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show tables;  
+-----+  
| Tables in commerce |  
+-----+  
| articles            |  
| clients            |  
| ventes             |  
+-----+  
3 rows in set (0.00 sec)
```

```
mysql> describe articles;  
// affiche la structure de la table articles
```

```

+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| code  | char(4)   | YES  |     | NULL    |      |
| nom   | char(20)  | YES  |     | NULL    |      |
| prix  | decimal(10,2) | YES  |     | NULL    |      |
| stock | int(11)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

```

mysql> describe clients;
// structure de la table clients
+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| code  | char(4)   | YES  |     | NULL    |      |
| nom   | char(20)  | YES  |     | NULL    |      |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> describe ventes;
// structure de la table ventes
+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| codeClient | char(4)   | YES  |     | NULL    |      |
| codeArticle | char(4)   | YES  |     | NULL    |      |
| quantite   | int(11)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

```

mysql> insert into articles (code,nom,prix,stock) values ('a001','article1',1000,100);
// insertion d'une ligne dans la table articles
Query OK, 1 row affected (0.00 sec)

```

```

mysql> insert into articles (code,nom,prix,stock) values ('a002','article2',2000,200);
mysql> insert into articles (code,nom,prix,stock) values ('a003','article3',3000,300);
mysql> insert into articles (code,nom,prix,stock) values ('a004','article4',4000,400);
mysql> insert into articles (code,nom,prix,stock) values ('a005','article5',5000,500);

```

```

mysql> select * from articles;
// affiche le contenu de la table articles
+-----+-----+-----+-----+
| code | nom      | prix      | stock |
+-----+-----+-----+-----+
| a001 | article1 | 1000.00   | 100   |
| a002 | article2 | 2000.00   | 200   |
| a003 | article3 | 3000.00   | 300   |
| a004 | article4 | 4000.00   | 400   |
| a005 | article5 | 5000.00   | 500   |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> insert into clients (code,nom) values ('c001','client1');
// ajoute une ligne dans la table clients
Query OK, 1 row affected (0.00 sec)

```

```

mysql> insert into clients (code,nom) values ('c002','client2');
mysql> insert into clients (code,nom) values ('c003','client3');

```

```

mysql> select * from clients;
// contenu de la table clients
+-----+-----+
| code | nom      |
+-----+-----+
| c001 | client1  |
| c002 | client2  |
| c003 | client3  |
+-----+-----+
3 rows in set (0.01 sec)

```

```

mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c001','a003',10);
// le client c001 a acheté 10 articles a003
Query OK, 1 row affected (0.00 sec)

```

```

// on fait d'autres insertions dans la table ventes
mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c001','a002',5);
mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c002','a002',8);
mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c002','a001',2);
mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c003','a001',20);
mysql> insert into ventes (codeClient, codeArticle, quantite) values ('c003','a005',12);

```

```
mysql> select * from ventes;
// les ventes effectuées
+-----+-----+-----+
| codeClient | codeArticle | quantite |
+-----+-----+-----+
| c001       | a003       | 10      |
| c001       | a002       | 5       |
| c002       | a002       | 8       |
| c002       | a001       | 2       |
| c003       | a001       | 20      |
| c003       | a005       | 12      |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

// quelques interrogations

```
mysql> select nom, prix from articles;
// seulement certaines colonnes
+-----+-----+
| nom      | prix    |
+-----+-----+
| article1 | 1000.00 |
| article2 | 2000.00 |
| article3 | 3000.00 |
| article4 | 4000.00 |
| article5 | 5000.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select nom, prix from articles order by prix desc;
// idem dans l'ordre décroissant des prix
+-----+-----+
| nom      | prix    |
+-----+-----+
| article5 | 5000.00 |
| article4 | 4000.00 |
| article3 | 3000.00 |
| article2 | 2000.00 |
| article1 | 1000.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select nom, prix, stock from articles where stock<300 order by prix desc;
// idem mais seulement avec les lignes où stock<300
+-----+-----+-----+
| nom      | prix    | stock |
+-----+-----+-----+
| article2 | 2000.00 | 200   |
| article1 | 1000.00 | 100   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select nom, prix, stock from articles where stock<300 and prix<2000 order by prix desc;
// idem mais on a rajouté la condition prix<2000
+-----+-----+-----+
| nom      | prix    | stock |
+-----+-----+-----+
| article1 | 1000.00 | 100   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from ventes order by quantite;
// les ventes par ordre croissant des quantités
+-----+-----+-----+
| codeClient | codeArticle | quantite |
+-----+-----+-----+
| c002       | a001       | 2       |
| c001       | a002       | 5       |
| c002       | a002       | 8       |
| c001       | a003       | 10      |
| c003       | a005       | 12      |
| c003       | a001       | 20      |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select articles.nom,ventes.codeClient,ventes.quantite from ventes,articles
-> where articles.code=ventes.codeArticle
-> order by quantite;
```

```
// idem mais avec les noms des clients - jointure entre 2 tables
+-----+-----+-----+
| nom      | codeClient | quantite |
+-----+-----+-----+
| article1 | c002       | 2        |
| article2 | c001       | 5        |
| article2 | c002       | 8        |
| article3 | c001       | 10       |
| article5 | c003       | 12       |
| article1 | c003       | 20       |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select articles.nom article ,ventes.codeClient client,ventes.quantite from ventes, articles
-> where articles.code=ventes.codeArticle
-> order by quantite;
```

```
// idem mais avec des alias pour les colonnes (article, client)

+-----+-----+-----+
| article  | client  | quantite |
+-----+-----+-----+
| article1 | c002    | 2        |
| article2 | c001    | 5        |
| article2 | c002    | 8        |
| article3 | c001    | 10       |
| article5 | c003    | 12       |
| article1 | c003    | 20       |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select articles.nom article, clients.nom client, ventes.quantite
-> from ventes, articles, clients
-> where ventes.codeArticle=articles.code
-> and ventes.codeClient=clients.code
-> order by quantite;
```

```
// on rajoute le nom du client
```

```
+-----+-----+-----+
| article | client | quantite |
+-----+-----+-----+
| article1 | client2 | 2        |
| article2 | client1 | 5        |
| article2 | client2 | 8        |
| article3 | client1 | 10       |
| article5 | client3 | 12       |
| article1 | client3 | 20       |
+-----+-----+-----+
6 rows in set (0.01 sec)
```

```
// quelques mises à jour
```

```
mysql> select nom, prix from articles;
```

```
// état actuel des prix
```

```
+-----+-----+
| nom      | prix      |
+-----+-----+
| article1 | 1000.00   |
| article2 | 2000.00   |
| article3 | 3000.00   |
| article4 | 4000.00   |
| article5 | 5000.00   |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> update articles set prix=prix*1.1;
```

```
// augmente les prix de 10%
```

```
Query OK, 5 rows affected (0.00 sec)
```

```
Rows matched: 5 Changed: 5 Warnings: 0
```

```
mysql> select nom, prix from articles;
```

```
// vérification
```

```
+-----+-----+
| nom      | prix      |
+-----+-----+
| article1 | 1100.00   |
| article2 | 2200.00   |
| article3 | 3300.00   |
| article4 | 4400.00   |
+-----+-----+
```

```
| article5 | 5500.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> update articles set prix=prix+1000 where prix>3000;
// une hausse pour les articles dont le prix est >3000
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3 Changed: 3 Warnings: 0
```

```
mysql> select nom, prix from articles;
// vérification
+-----+-----+
| nom      | prix    |
+-----+-----+
| article1 | 1100.00 |
| article2 | 2200.00 |
| article3 | 4300.00 |
| article4 | 5400.00 |
| article5 | 6500.00 |
+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> delete from articles where nom='article4';
// suppression article 4
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select nom, prix from articles;
// vérification
+-----+-----+
| nom      | prix    |
+-----+-----+
| article1 | 1100.00 |
| article2 | 2200.00 |
| article3 | 4300.00 |
| article5 | 6500.00 |
+-----+-----+
4 rows in set (0.00 sec)
```

```
// on quitte
mysql> quit
Bye
```

16.6 Poursuivre

Il est maintenant temps de lire la documentation de MySQL au format PDF que vous trouverez sur les sites tels que <http://www.mysql.com> et ses sites miroirs.

17 Le SGBDR POSTGRES

17.1 Introduction

Postgres est un système de gestion de bases de données relationnelles gratuit livré en standard avec les distributions linux Redhat. Vis à vis de MySQL, ce SGBD offre l'avantage de respecter la norme SQL dans son intégralité. Par ailleurs, elle offre la notion de transaction que n'offre pas MySQL (mai 2003). Il existe des versions de Postgres aussi bien pour Linux que pour Windows. On trouve également pour cette base des pilotes ODBC et JDBC qui permettent à des applications écrites dans des langages divers (C, C++, Java, VB, Perl, ...) d'avoir accès au contenu d'une base Postgres.

Où trouver Postgres ? L'URL du site officiel de Postgres est <http://www.postgresql.org>. Pour Linux RedHat, il existe des paquetages RPM pour Postgres qu'on trouvera sur les CD.

17.2 Installation de Postgres

Vous avez pu déjà installer les paquetages Postgres à l'installation initiale de Linux. Vous pouvez le vérifier avec la commande **rpm** (faire **man rpm**). Cette commande gère les paquetages RPM (Redhat Package Manager).

```
[root@tahe util]# rpm -qa | grep -i postgres
```

```
postgresql-7.2.2-1
postgresql-server-7.2.2-1
postgresql-libs-7.2.2-1
```

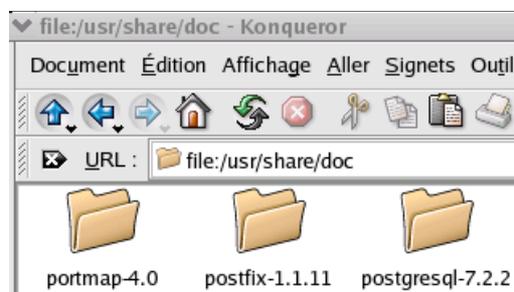
Dans l'exemple ci-dessus, Le serveur Postgres est installé (postgresql-server-7.2.2-1), le client pour ce serveur également 'postgresql-7.2.2-1) ainsi que les bibliothèques nécessaires à Postgres (postgresql-libs-7.2.2-1). Si des RPM analogues ne sont pas présents, c'est que Postgres n'est pas installé. Lancez alors, en mode root, l'application de gestion des paquets : **Menu KDE/paramètres de système/Paquetages** et suivez la méthode déjà présentée pour installer les paquets de MySQL.

17.3 La documentation de Postgres

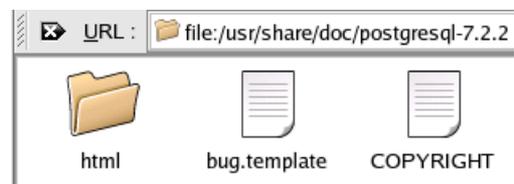
La documentation de Postgres est disponibles sous plusieurs formes :

- **man postgres** donnera des informations sur le SGBDR Postgres
- **man psql** donnera des informations sur le client sql livré avec Postgres

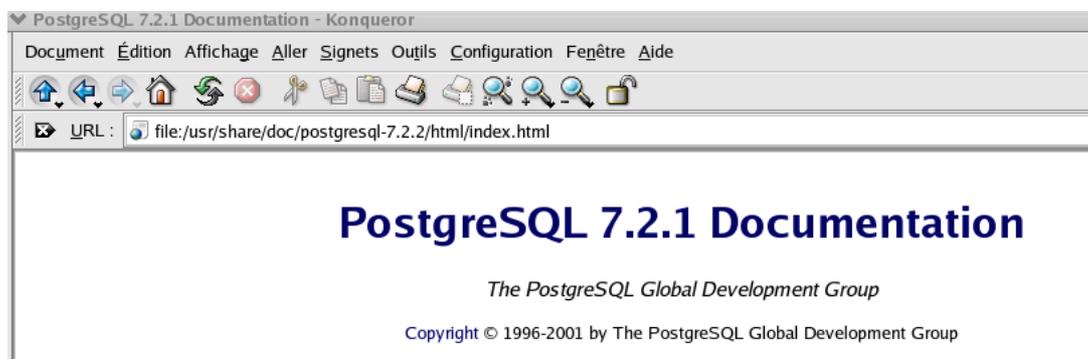
Une documentation au format HTML est également disponible dans le dossier **/usr/share/doc** :



Allez dans le dossier **postgresql** :



Allez dans le dossier **html** et ouvrez le fichier **index.html** :



Le document ci-dessus est le point d'entrée de la documentation HTML qui vous permettra de maîtriser POSTGRES.

17.4 Lancement du serveur postgres

Comme beaucoup de services sous RedHat, il existe des scripts de lancement et d'arrêt du service *postgres* :

```
[/etc/rc.d/init.d/postgres start] pour lancer le service  
[/etc/rc.d/init.d/postgres stop] pour arrêter le service  
[/etc/rc.d/init.d/postgres restart] pour enchaîner arrêt/démarrage
```

Lançons le service *postgres* :

```
[root@tahe util]# /etc/rc.d/init.d/postgresql start  
Starting postgresql service: [ OK ]
```

On aurait pu également utiliser le gestionnaire de services : **Menu KDE/Paramètres du serveur/Services**.

17.5 L'utilisateur Unix postgres

A l'installation de Postgres, est créé un utilisateur unix appelé **postgres** et appartenant au groupe **postgres** :

```
[root@tahe util]# id postgres  
uid=26(postgres) gid=26(postgres) groupes=26(postgres)  
  
[root@tahe util]# grep -i postgres /etc/passwd  
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash  
  
[root@tahe util]# grep -i postgres /etc/group  
postgres:x:26:
```

Cet utilisateur particulier est le super-utilisateur du SGBDR POSTGRES. Il aura tous les droits sur tous les objets du SGBDR. Commençons par lui donner un mot de passe (**azerty**) :

```
[root@tahe util]# passwd postgres  
Changing password for user postgres.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

17.6 Première utilisation du client psql

Créons une session client-serveur avec le serveur POSTGRES à l'aide du client **psql**. Tout d'abord, prenons l'identité de l'utilisateur **postgres** :

```
[util@tahe util]$ su - postgres  
Password:  
  
-bash-2.05b$ whoami  
postgres
```

Le client **psql** est l'outil standard de travail en mode texte avec le sgbdr postgres. On fera **man psql** pour avoir des informations sur cet outil. L'option **-l** permet d'avoir la liste des bases de données :

```
-bash-2.05b$ psql -l  
Liste des bases de données  
Nom | Propriétaire | Encodage  
-----+-----+-----  
template0 | postgres | SQL_ASCII  
template1 | postgres | SQL_ASCII
```

On voit qu'il existe déjà deux bases de données propriété de l'utilisateur **postgres**. On remarquera ici que l'utilisateur **postgres** en question, est un utilisateur du sgbdr et non celui du système unix. L'installation des RPM de Postgres a donc créé deux bases de données qui serviront de modèles lors de la création de nouvelles bases. Où sont-elles stockées ? Dans l'installation faite avec des paquetages RPM sous RH8, le dossier de postgres est **/var/lib/pgsql/data** :

```
[root@tahe util]# ls -l /var/lib/pgsql/data/
```

```
total 60
drwx-----  6 postgres postgres    4096 avr  6 14:32 base
drwx-----  2 postgres postgres    4096 avr  7 09:09 global
-rw-r--r--  1 postgres postgres     38 avr  6 14:20 password
-rw-r--r--  1 postgres postgres     19 avr  6 14:19 password.bk
drwx-----  2 postgres postgres    4096 avr  6 10:38 pg_clog
-rw-----  1 postgres postgres   10219 avr  6 14:40 pg_hba.conf
-rw-----  1 postgres postgres    1250 avr  6 10:38 pg_ident.conf
-rw-----  1 postgres postgres     4 avr  6 10:38 PG_VERSION
drwx-----  2 postgres postgres    4096 avr  6 10:38 pg_xlog
-rw-----  1 postgres postgres   3848 avr  6 10:38 postgresql.conf
-rw-----  1 postgres postgres     20 avr  7 08:56 postmaster.opts
```

Le script `/etc/rc.d/init.d/postgres` travaille avec les bases de données situées sous ce dossier. Pour arrêter le sgbd, on exécutera la commande :

```
[root@tahe util]# /etc/rc.d/init.d/postgresql stop
Arrêt du service postgresql : [ OK ]
```

Si l'installation ne se fait pas avec des RPM, ces bases de données initiales ne sont en général pas créées. La procédure de création de ces bases initiales est alors la suivante (à ne pas faire dans notre exemple).

17.7 Création des premières bases de données (installation faite sans RPM)

Lorsque Postgres a été installé sans base de données pré-configurée, il faut en créer une avant de lancer le SGBDR. Cela se fait avec la commande `initdb` (faire `man initdb`). Cette première base de données servira ensuite de modèle aux suivantes. Une base de données peut être créée avec la commande unix `createdb` (faire `man createdb`). Ainsi construite la base de données appartiendra à un utilisateur POSTGRES portant le même nom que l'utilisateur unix ayant émis la commande. Il en est de même avec `initdb` qui crée la toute première base de données. Celle-ci appartiendra à l'utilisateur qui émettra la commande. Celui-ci peut être n'importe qui. On utilise habituellement l'utilisateur unix `postgres`. La syntaxe de `initdb` pour créer la première base est :

`initdb -D <dir>`

où `<dir>` est le répertoire sous lequel sera créée la base. Les dossiers et fichiers qui y seront stockés le seront sous le nom de celui qui exécute la commande. `<dir>` sera ici `/usr/local/postgres`. Nous le créons et le donnons à l'utilisateur `postgres` :

```
[root@tahe util]# mkdir /usr/local/postgres
[root@tahe util]# chown postgres:postgres /usr/local/postgres/
[root@tahe util]# ls -ld /usr/local/postgres/
drwx-----  6 postgres postgres    4096 avr  6 10:45 /usr/local/postgres/
```

Ceci fait, nous prenons l'identité de l'utilisateur `postgres` puis utilisons la commande `initdb` pour créer la première base :

```
[root@tahe util]# su - postgres
-bash-2.05b$ initdb -D /usr/local/postgres/
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

Fixing permissions on existing directory /usr/local/postgres/... ok
creating directory /usr/local/postgres//base... ok
creating directory /usr/local/postgres//global... ok
creating directory /usr/local/postgres//pg_xlog... ok
creating directory /usr/local/postgres//pg_clog... ok
creating template1 database in /usr/local/postgres//base/1... ok
creating configuration files... ok
initializing pg_shadow... ok
enabling unlimited row size for system tables... ok
creating system views... ok
loading pg_description... ok
vacuuming database template1... ok
copying template1 to template0... ok

Success. You can now start the database server using:
```

```
/usr/bin/postmaster -D /usr/local/postgres/  
or  
/usr/bin/pg_ctl -D /usr/local/postgres/ -l logfile start
```

Les lignes ci-dessus, montrent que la création de la base de données a réussi. Le contenu du dossier racine des données de **postgres** est le suivant :

```
[root@tahe util]# ls -l /usr/local/postgres/  
total 40  
drwx----- 4 postgres postgres 4096 avr 6 10:45 base  
drwx----- 2 postgres postgres 4096 avr 6 10:45 global  
drwx----- 2 postgres postgres 4096 avr 6 10:45 pg_clog  
-rw----- 1 postgres postgres 10168 avr 6 10:45 pg_hba.conf  
-rw----- 1 postgres postgres 1250 avr 6 10:45 pg_ident.conf  
-rw----- 1 postgres postgres 4 avr 6 10:45 PG_VERSION  
drwx----- 2 postgres postgres 4096 avr 6 10:45 pg_xlog  
-rw----- 1 postgres postgres 3848 avr 6 10:45 postgresql.conf
```

Comme indiqué dans les affichages de la commande *initdb*, le service **postgres** est lancé sous l'identité **postgres** avec la commande **pg_ctl** (faire *man pg_ctl* et *man postmaster*) :

```
-bash-2.05b$ id  
uid=26(postgres) gid=26(postgres) groupes=26(postgres)  
  
-bash-2.05b$ pg_ctl -D /usr/local/postgres/ start  
postmaster successfully started  
DEBUG: database system was shut down at 2003-04-07 09:16:05 CEST  
DEBUG: checkpoint record is at 0/109724  
DEBUG: redo record is at 0/109724; undo record is at 0/0; shutdown TRUE  
DEBUG: next transaction id: 92; next oid: 16556  
DEBUG: database system is ready
```

Nous pouvons vérifier les bases de données disponibles :

```
-bash-2.05b$ psql -l  
Liste des bases de données  
Nom | Propriétaire | Encodage  
-----+-----+-----  
template0 | postgres | SQL_ASCII  
template1 | postgres | SQL_ASCII
```

Pour arrêter la base, nous utilisons de nouveau la commande **pg_ctl** :

```
-bash-2.05b$ pg_ctl -D /usr/local/postgres/ stop  
DEBUG: smart shutdown request  
DEBUG: shutting down  
waiting for postmaster to shut down....DEBUG: database system is shut down  
done  
postmaster successfully shut down
```

17.8 Le client psql

Découvrons maintenant le client **psql**. Il faut se connecter sous l'identité d'un utilisateur déclaré dans postgres. Au début, nous n'avons guère le choix. Seul l'utilisateur **postgres** existe. Il y a plusieurs façons de lancer *psql* (faire *man psql*) :

<pre>psql</pre>	Si l'utilisateur unix qui lance la commande est X, cela revient à se connecter à une base X sous le nom X
<pre>psql -U <Y></pre>	Si l'utilisateur unix qui lance la commande est X, cela revient à se connecter à une base X sous le nom <Y>
<pre>psql -U <Y> -d <base></pre>	Connecte à une base <base> sous le nom <Y>

```
-bash-2.05b$ psql  
psql: FATAL 1: Database "postgres" does not exist in the system catalog.
```

Nous sommes sous l'identité unix **postgres**. Ci-dessus, *psql* cherche donc à se connecter à une base appelée **postgres** qui n'existe pas. D'où l'erreur.

```
-bash-2.05b$ psql -d template1
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
        \h pour l'aide-mémoire sur les commandes SQL
        \? pour l'aide-mémoire sur les commandes internes
        \g ou point-virgule pour exécuter une requête
        \q pour quitter

template1=#
```

Ci-dessus on a précisé une base existante mais pas d'utilisateur. *psql* va donc chercher un utilisateur portant le nom de l'utilisateur unix qui lance la commande (ici **postgres**). Il le trouve. On remarquera qu'il ne demande pas de mot de passe. Faites **\q** pour quitter. Ci-dessous, on fait la même chose en précisant explicitement le nom de l'utilisateur **postgres** :

```
-bash-2.05b$ psql -d template1 -U postgres
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
        \h pour l'aide-mémoire sur les commandes SQL
        \? pour l'aide-mémoire sur les commandes internes
        \g ou point-virgule pour exécuter une requête
        \q pour quitter

template1=#
```

Faisons **\q** pour quitter et faisons la même chose avec l'identité unix **util** par exemple :

```
[util@tahe util]$ psql -U postgres -d template1
psql: FATAL 1: IDENT authentication failed for user "postgres"
```

La connexion a échoué. Nous verrons pourquoi ultérieurement. Reconnectons-nous à partir du compte unix **postgres** :

```
[util@tahe util]$ su - postgres
Password:
-bash-2.05b$ psql -d template1
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
        \h pour l'aide-mémoire sur les commandes SQL
        \? pour l'aide-mémoire sur les commandes internes
        \g ou point-virgule pour exécuter une requête
        \q pour quitter

template1=#
```

A l'aide du client *psql*, nous nous sommes connectés à la base de données initiale qui s'appelle **template1**. L'invite de *psql* indique le nom de la base courante. Pour

- connaître la liste des commandes internes, tapez **\?**
- connaître la liste des commandes SQL, tapez **\h**
- avoir de l'aide sur une commande SQL précise, tapez **\h commande**
- quitter, tapez **\q**

Listons les commandes internes :

```
\a          toggle between unaligned and aligned output mode
\c[onnect] [BASE|- [UTILISATEUR]]
             connecte à une base de données nouveau (à présent «template1»)
\C TITRE    fixe le titre des tables
\cd [REPERT] remplace le répertoire courant
\copy ...   perform SQL COPY with data stream to the client host
\copyright  show PostgreSQL usage and distribution terms
\d TABLE   décris table (ou vue, index, séquence)
\d{t|i|s|v}... affiche des tables/index/séquences/vues
\d{p|S|l}   affiche des privilèges d'accès, tables de système ou
             objets grands
\da        affiche des fonctions d'agrégation
\dd NOM     affiche les commentaires sur table, type, fonction ou
             operateur
\df        affiche des fonctions
\do        affiche des operateurs
\dT        affiche des types de donnée
```

```

\e FICHIER      édite le tampon de requête ou le fichier par un éditeur externe
\echo TEXTE     écris texte sur la sortie standard
\encoding ENCODAGE fixe l'encodage du client
\f CHAINE       fixe le séparateur de champs
\g FICHIER      exécute commande SQL (et écris les résultats dans une
                fichier ou |tube)
\h NOM          aide-mémoire pour les commandes SQL, * pour toutes les commandes
\H             toggle HTML output mode (currently désactivé)
\i FICHIER      exécute les commandes du fichier
\l             affiche une liste des bases de données disponibles
\lo_export, \lo_import, \lo_list, \lo_unlink
                Large object operations
\o FICHIER      écris les résultats des requêtes dans un fichier (ou |tube)
\p             affiche le contenu actuel du tampon de requête
\pset VAR       set table output option (VAR := {format|border|expanded|
                fieldsep|null|recordsep|tuples_only|title|tableattr|pager})
\q             quitte psql
\qecho TEXTE    écris texte sur la sortie pour les résultats des requêtes
                (vois \o)
\r             efface le tampon de requête
\s FILENAME     print history or save it to file
\set NAME VALUE set internal variable
\t             show only rows (currently désactivé)
\T TEXT        set HTML table tag attributes
\unset NAME     unset (delete) internal variable
\w FICHIER      écris le contenu du tampon de requête dans un fichier
\x             toggle expanded output (currently désactivé)
\z             affiche des privilèges d'accès des tables
\! [COMMAND]    execute command in shell or start interactive shell

```

On lira attentivement cette liste de commandes. Pour avoir la liste des bases, tapons **\l** :

```

templatel=# \l
        Liste des bases de données
        Nom      | Propriétaire | Encodage
        -----+-----+-----
template0 | postgres     | SQL_ASCII
templatel | postgres     | SQL_ASCII

```

Pour avoir la liste des commandes SQL, tapons **\h** :

```

templatel=# \h
Aide-mémoire disponible:
ABORT          CREATE TABLE AS      FETCH
ALTER GROUP    CREATE TRIGGER        GRANT
ALTER TABLE   CREATE TYPE           INSERT
ALTER USER     CREATE USER           LISTEN
ANALYZE        CREATE VIEW           LOAD
BEGIN          DECLARE              LOCK
CHECKPOINT     DELETE                MOVE
CLOSE          DROP AGGREGATE        NOTIFY
CLUSTER        DROP DATABASE         REINDEX
COMMENT        DROP FUNCTION         RESET
COMMIT         DROP GROUP            REVOKE
COPY           DROP INDEX            ROLLBACK
CREATE AGGREGATE DROP LANGUAGE         SELECT
CREATE CONSTRAINT TRIGGER DROP OPERATOR         SELECT INTO
CREATE DATABASE DROP RULE              SET
CREATE FUNCTION DROP SEQUENCE         SET CONSTRAINTS
CREATE GROUP    DROP TABLE            SET SESSION AUTHORIZATION
CREATE INDEX    DROP TRIGGER           SET TRANSACTION
CREATE LANGUAGE DROP TYPE              SHOW
CREATE OPERATOR DROP USER             TRUNCATE
CREATE RULE     DROP VIEW              UNLISTEN
CREATE SEQUENCE END                    UPDATE
CREATE TABLE  EXPLAIN                VACUUM

```

Pour avoir de l'aide sur la commande **create user** :

```

templatel=# \h create user
Commande: CREATE USER
Description: define a new database user account
Syntaxe:
CREATE USER username [ [ WITH ] option [ ... ] ]
where option can be:
        SYSID uid
        | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'

```

```

| CREATEDB | NOCREATEDB
| CREATEUSER | NOCREATEUSER
| IN GROUP groupname [, ...]
| VALID UNTIL 'abstime'

```

Pour avoir de l'aide sur la commande **create database** :

```

template1=# \h create database
Commande: CREATE DATABASE
Description: create a new database
Syntaxe:
CREATE DATABASE name
  [ WITH [ LOCATION = 'dbpath' ]
        [ TEMPLATE = template ]
        [ ENCODING = encoding ] ]

```

17.9 Création de bases de données

Comme indiqué-dessus, créons deux bases de données appelée **db1** et **db2** :

```

template1=# create database db1;
CREATE DATABASE
template1=# create database db2;
CREATE DATABASE
template1=# \l
      Liste des bases de données
  Nom      | Propriétaire | Encodage
-----+-----+-----
db1       | postgres     | SQL_ASCII
db2       | postgres     | SQL_ASCII
template0 | postgres     | SQL_ASCII
template1 | postgres     | SQL_ASCII

```

On voit que toutes les bases sont propriété de l'utilisateur **postgres**. Créons une nouvelle base **db3** puis supprimons-la :

```

template1=# create database db3;
CREATE DATABASE
template1=# \l
      Liste des bases de données
  Nom      | Propriétaire | Encodage
-----+-----+-----
db1       | postgres     | SQL_ASCII
db2       | postgres     | SQL_ASCII
db3       | postgres     | SQL_ASCII
template0 | postgres     | SQL_ASCII
template1 | postgres     | SQL_ASCII
(5 lignes)

template1=# drop database db3;
DROP DATABASE
template1=# \l
      Liste des bases de données
  Nom      | Propriétaire | Encodage
-----+-----+-----
db1       | postgres     | SQL_ASCII
db2       | postgres     | SQL_ASCII
template0 | postgres     | SQL_ASCII
template1 | postgres     | SQL_ASCII

```

17.10 Création d'utilisateurs

Créons trois utilisateurs **pguti1**, **pguti2**, **pguti3** qui n'auraient pas le droit de créer de bases (**nocreatedb**) ni d'utilisateurs (**nocreateuser**) :

```

template1=# create user pguti1 password 'pguti1' nocreatedb nocreateuser;
CREATE USER
template1=# create user pguti2 password 'pguti2' nocreatedb nocreateuser;
CREATE USER
template1=# create user pguti3 password 'pguti3' nocreatedb nocreateuser;
CREATE USER

```

Pour supprimer un utilisateur, on utilise la commande SQL **drop user** :

```
template1=# drop user pguti3
```

17.11 Authentification des utilisateurs

Déconnectons-nous de **psql**, et sous l'identité unix **util**, essayons de nous connecter à la base **db1** sous l'identité postgres **pgutil** :

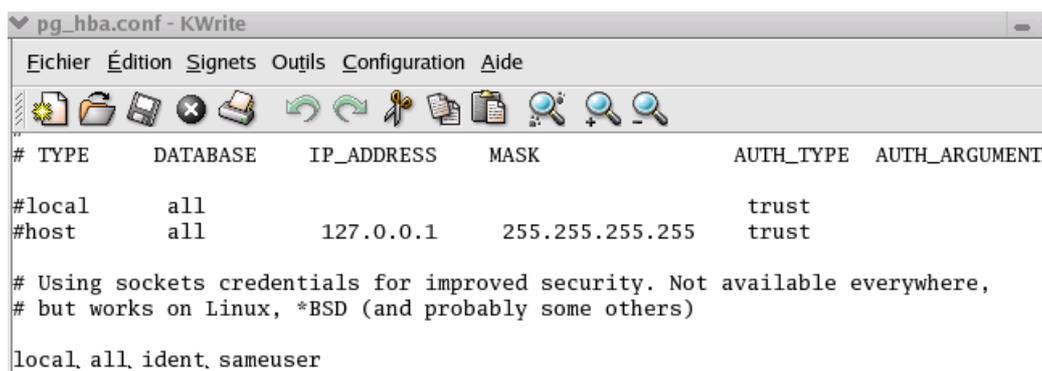
```
[util@tahe util]$ id
uid=500(util) gid=500(groupe1) groupes=500(groupe1)

[util@tahe util]$ psql -U pgutil -d db1
psql: FATAL 1: IDENT authentication failed for user "pgutil"
```

On voit qu'il se produit une erreur. Il est temps maintenant de parler d'authentification des utilisateurs *postgres*. Celle-ci est configurée par le fichier **pg_hba.conf** situé dans le dossier des données de postgres :

```
[root@tahe util]# ls -l /var/lib/pgsql/data/pg_hba.conf
-rw----- 1 postgres postgres 10219 avr 6 14:40 /var/lib/pgsql/data/pg_hba.conf
```

On voit que ce fichier est propriété de l'utilisateur **postgres**. Prenez l'identité unix **postgres** et consultez le fichier **pg_hba.conf** avec *knife* par exemple. Lisez son contenu et positionnez-vous en fin de fichier :



```
pg_hba.conf - KWrite
Fichier Édition Signets Outils Configuration Aide
# TYPE      DATABASE    IP_ADDRESS  MASK          AUTH_TYPE  AUTH_ARGUMENT
#local      all
#host       all         127.0.0.1   255.255.255.255 trust
# Using sockets credentials for improved security. Not available everywhere,
# but works on Linux, *BSD (and probably some others)
local, all, ident, sameuser
```

La ligne non commentée ci-dessus fixe les règles d'authentification :

champ 1 machine à partir de laquelle l'utilisateur se connecte :

local : machine locale

host : machine de l'internet – doit suivre alors l'adresse IP des machines autorisées avec leurs masques de réseau

champ 2 bases de données autorisées :

all : toutes

sameuser : base portant le nom de l'utilisateur

<dbname> : nom de la base

champ 3 type d'authentification :

trust : pas d'authentification

password, md5 : authentification par mot de passe

ident sameuser : l'utilisateur unix doit porter le même nom que l'utilisateur postgres

La ligne **local all ident sameuser** autorise tout utilisateur de la machine locale à se connecter à toute base de données définie dans postgres à condition qu'un utilisateur de même nom existe dans la base. Rappelons les conditions de notre échec de connexion :

```
[util@tahe util]$ id
uid=500(util) gid=500(groupe1) groupes=500(groupe1)

[util@tahe util]$ psql -U pgutil -d db1
psql: FATAL 1: IDENT authentication failed for user "pgutil"
```

L'utilisateur unix **util** essayait de se connecter. Or il n'existe pas d'utilisateur postgres appelé **util**. D'où l'échec. Vérifions notre hypothèse. Créons un utilisateur postgres *util*. Pour cela, prenons l'identité unix *postgres* et utilisons *psql* :

```
-bash-2.05b$ psql -d template1
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
       \h pour l'aide-mémoire sur les commandes SQL
       \? pour l'aide-mémoire sur les commandes internes
       \g ou point-virgule pour exécuter une requête
       \q pour quitter

template1=# create user util ncreatedb ncreateuser password 'util';
CREATE USER

template1=# \q
```

Maintenons, prenons l'identité unix **util** et essayons de nous connecter à la base **db1** de postgres :

```
[util@tahe util]$ psql -d db1
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
       \h pour l'aide-mémoire sur les commandes SQL
       \? pour l'aide-mémoire sur les commandes internes
       \g ou point-virgule pour exécuter une requête
       \q pour quitter

db1=> \l
      Liste des bases de données
  Nom      | Propriétaire | Encodage
-----+-----+-----
 db1      | postgres    | SQL_ASCII
 db2      | postgres    | SQL_ASCII
 template0| postgres    | SQL_ASCII
 template1| postgres    | SQL_ASCII
(4 lignes)

db1=>
```

Cette fois-ci, nous avons réussi. Nous remarquerons que le mot de passe n'a pas été demandé. Renforçons la sécurité en adoptant une nouvelle politique d'authentification :

```
#local      all      ident      sameuser
local      all      md5
```

La nouvelle ligne signifie qu'un utilisateur peut se connecter de la machine locale (local) à toute base de données (all) et qu'il devra fournir un mot de passe (md5). A la place de *md5*, on aurait pu choisir *password*. La technique *md5* est plus sécurisée que a technique *password* pour les machines distantes car elle crypte le mot de passe avant de l'envoyer sur le réseau. Sauvegardons ce nouveau fichier de configuration et relançons le serveur postgres (à faire sous l'identité unix *postgres* par exemple) :

```
-bash-2.05b$ pg_ctl -D /usr/local/postgres/ restart
DEBUG: smart shutdown request
DEBUG: shutting down
waiting for postmaster to shut down.....DEBUG: database system is shut down
done
postmaster successfully shut down
postmaster successfully started
-bash-2.05b$ DEBUG: database system was shut down at 2003-04-07 11:32:21 CEST
DEBUG: checkpoint record is at 0/12E7D4
DEBUG: redo record is at 0/12E7D4; undo record is at 0/0; shutdown TRUE
DEBUG: next transaction id: 163; next oid: 16561
DEBUG: database system is ready
```

Maintenant, refaisons la connexion qui avait échoué précédemment :

```
[uti2@tahe uti2]$ psql -U pgutil -d db1
Mot de passe:
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
        \h pour l'aide-mémoire sur les commandes SQL
        \? pour l'aide-mémoire sur les commandes internes
        \g ou point-virgule pour exécuter une requête
        \q pour quitter

db1=>
```

Ici, c'est l'utilisateur unix **uti2** qui tente de se connecter à la base postgres **db1** sous l'identité postgres **pgutil**. On voit qu'un mot de passe est demandé (**pgutil**). La connexion réussit. On peut aussi se connecter à la base **db2** :

```
db2=> \l
      Liste des bases de données
  Nom      | Propriétaire | Encodage
-----+-----+-----
 db1      | postgres     | SQL_ASCII
 db2      | postgres     | SQL_ASCII
 template0| postgres     | SQL_ASCII
 template1| postgres     | SQL_ASCII

db1=> \c db2
Vous êtes maintenant connectés à la base de données db2.
```

Il serait sans doute pratique que les utilisateurs ne puissent se connecter qu'à certaines bases. Faisons en sorte par exemple que l'utilisateur postgres **pgutil** ne puisse se connecter qu'à la base **db1** et l'utilisateur **pgutil2** qu'à la base **db2**. Pour ce faire, modifions de nouveau notre fichier de configuration **pg_hba.conf** :

```
#local         all             ident         sameuser
#local         all             md5
local db1 md5 utidb1
local db2 md5 utidb2
```

Analysons la ligne **local db1 md5 utidb1** :

local	connexion à partir de la machine locale
db1	connexion à la base db1 uniquement
md5	mot de passe exigé
utidb1	nom d'un fichier contenant le nom des utilisateurs postgres autorisés à se connecter à la base db1, à raison d'un nom par ligne. Le fichier doit être dans le dossier des données de postgres.

```
[root@tahe uti1]# ls -l /var/lib/pgsql/data
....
-rw-----  1 postgres postgres  10225 avr  7 11:39 pg_hba.conf
...
-rw-----  1 postgres postgres    7 avr  7 11:39 utidb1
-rw-----  1 postgres postgres    7 avr  7 11:39 utidb2
```

```
[root@tahe uti1]# more /var/lib/pgsql/data/utidb1
pgutil
```

On voit ici, que seul l'utilisateur postgres **pgutil** est autorisé à se connecter à la base **db1**. Vérifions :

```
[uti2@tahe uti2]$ psql -U pgutil -d db1
Mot de passe:
Bienvenu à psql, l'interface interactif de PostgreSQL.

Tapez: \copyright pour l'information de copyright
        \h pour l'aide-mémoire sur les commandes SQL
        \? pour l'aide-mémoire sur les commandes internes
        \g ou point-virgule pour exécuter une requête
        \q pour quitter

db1-> \c db2
FATAL 1: Password authentication failed for user "pgutil"

La connexion précédente était gardée.
db1->
```

On voit ci-dessus que l'utilisateur unix **uti2** a pu se connecter à la base **db1** sous l'identité postgres **pgutil**. Normal. Il a ensuite essayé de se connecter à la base **db2** (syntaxe `\c <database>`). Il a échoué. Nous avons donc créé un mécanisme permettant la connexion d'utilisateurs qu'à certaines bases et ce moyennant un mot de passe.

17.12 Commandes SQL

Maintenant, nous abordons une partie plus classique, celle du langage SQL. Nous ne nous apesantirons pas, renvoyant le lecteur à l'exemple développé pour le SGBD MySQL. Montrons cependant quelques exemples traités avec la base *db1* :

```
// création d'une table
db1=> create table t1 (id int primary key, nom varchar(20));
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index 't1_pkey' for table 't1'
CREATE

// liste des tables
db1=> \dt
      Liste des relations
  Nom | Type | Propriétaire
-----+-----
  t1  | table | pgutil
(1 ligne)

// structure de la table t1
db1=> \d t1
      Table «t1»
  Colonne |          Type          | Modifications
-----+-----
  id      | integer                | not null
  nom     | character varying(20) |
Primary key: t1_pkey
```

// quelques insertions

```
db1=> insert into t1 values(1,'nom1');
INSERT 16564 1
db1=> insert into t1 values(2,'nom2');
INSERT 16565 1
db1=> insert into t1 values(3,'nom3');
INSERT 16566 1
db1=> select * from t1;
 id | nom
----+----
  1 | nom1
  2 | nom2
  3 | nom3
```

// gestion d'une transaction (commit/rollback)

```
db1=> begin transaction;
BEGIN
db1=> select * from t1;
 id | nom
----+----
  1 | nom1
  2 | nom2
  3 | nom3
(3 lignes)

db1=> delete from t1;
DELETE 3
db1=> select * from t1;
 id | nom
----+----
(0 lignes)

db1=> rollback;
ROLLBACK
db1=> select * from t1;
 id | nom
----+----
  1 | nom1
  2 | nom2
  3 | nom3
```

On commence une transaction par **begin transaction**. Elle se termine soit par **commit** (la transaction est validée), soit par **rollback** (la transaction est annulée). Ici, **rollback** nous a permis d'annuler la suppression de lignes qui avait été opérée au cours de la transaction.

Le lecteur est maintenant invité à créer ses propres exemples.

DEVELOPPER SOUS LINUX

18 Outils de développement

Nous présentons ici quelques outils de développement disponibles sous Linux. Nous présentons aussi bien les langages de scripts utiles aux administrateurs que des langages compilés tels C et C++. Nous terminons par le langage Java particulièrement intéressant pour sa portabilité. A chaque fois, nous ne présentons qu'un test basique. Pour la plupart des tests, il y a nul besoin d'être **root**. Prenez l'identité d'un utilisateur normal, **util** par exemple.

18.1 Perl

Perl est installé par défaut avec Linux si vous avez pris l'option "Développement". Avec un éditeur de texte, construisez le programme suivant :

```
#!/usr/bin/perl
print "coucou\n";
exit 0;
```

Sauvegardez-le sous le nom *essai1.pl*. On suppose par la suite qu'on est dans le répertoire du programme :

```
[ls -l]
total 16
-rw-r--r--  1 root    root          44 mai  9 07:28 essai1.pl
```

On donne le droit d'exécution (x) à notre programme :

```
[chmod u+x essai1.pl]
[ls -l]
total 16
-rwxr--r--  1 root    root          44 mai  9 07:28 essai1.pl
```

Nous l'exécutons :

```
[./essai1.pl]
coucou
```

18.2 Python

Python est installé par défaut avec Linux si vous avez pris l'option "Développement". Avec un éditeur de texte, construisez le programme suivant :

```
#!/usr/bin/python
print 'coucou\n';
exit;
```

Sauvegardez-le sous le nom *essai1.py*. On suppose par la suite qu'on est dans le répertoire du programme :

```
[ls -l essai1.py]
-rw-r--r--  1 root    root          44 mai  9 07:43 essai1.py
```

On donne le droit d'exécution (x) à notre programme :

```
[chmod u+x essai1.py]
[ls -l essai1.py]
-rwxr--r--  1 root    root          44 mai  9 07:43 essai1.py
```

Nous l'exécutons :

```
[./essai1.py]
coucou
```

18.3 C

Plusieurs compilateurs C sont installés par défaut avec Linux si vous avez pris l'option "Développement". Nous utiliserons le compilateur **gcc**. Avec un éditeur de texte, construisez le programme suivant :

```
#include <stdio.h>

int main(void){
    puts("coucou");
    return(0);
}
```

Sauvegardez-le sous le nom *essai1.c*. On suppose par la suite qu'on est dans le répertoire du programme :

```
[ls -l *.c]
-rwxr-xr-x  1 root    root          70 mai  9 07:39 essai1.c
```

On compile le programme :

```
[gcc -o essai1 essai1.c]
// compile le source essai1.c pour produire l'excutable essai1 (-o)
```

On vérifie la présence de l'exécutable :

```
[ls -l essai1]
-rwxr-xr-x  1 root    root        13480 mai  9 07:39 essai1
```

On l'exécute :

```
[./essai1]
coucou
```

18.4 C++

Le compilateur g++ est installé par défaut avec Linux si vous avez pris l'option "Développement". Avec un éditeur de texte, construisez le programme suivant :

```
#include <iostream.h>
void main(void){
    cout << "coucou\n";
}
```

Sauvegardez-le sous le nom *essai1.cpp*. On suppose par la suite qu'on est dans le répertoire du programme. On compile puis exécute le programme :

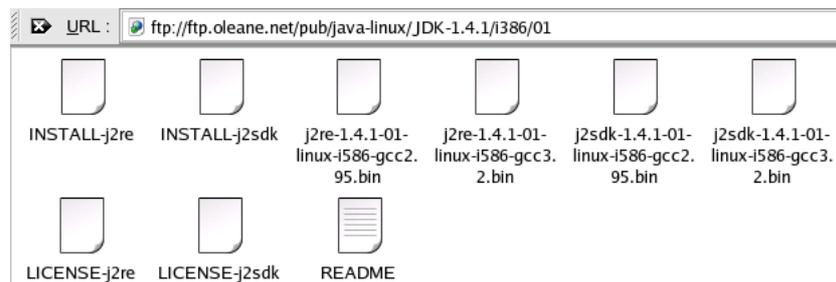
```
[util@tahe c++]$ g++ -o essai1 -Wno-deprecated essai1.cpp
[util@tahe c++]$ ls -l
total 16
-rwxr-xr-x  1 util    groupe1    11346 avr  7 17:28 essai1
-rw-r--r--  1 util    groupe1         61 avr  7 17:28 essai1.cpp
[util@tahe c++]$ ./essai1
coucou
```

18.5 Java

Le langage java n'est pas installé par défaut lors de l'installation de Linux RH. On en trouve une version du domaine public à l'url <http://www.blackdown.org>.



Suivez le lien **download** ci-dessus. une liste de sites miroir est offerte. Choisissez l'un d'eux et la version de Java désirée :



Ici, nous avons choisi la version 1.4.1 du JDK. Il y a plusieurs versions du JDK. Nous choisissons le fichier `j2sdk-1.4.1-01-linux-i586-gcc3.2.bin`. Une fois téléchargé, nous exécutons ce fichier :

```
[root@tahe java]# ls -l
-rw-r--r--  1 root      root      40813906 avr  7 18:10 j2sdk-1.4.1-01-linux-i586-gcc3.2.bin
[root@tahe java]# ./j2sdk-1.4.1-01-linux-i586-gcc3.2.bin
```

L'arborescence java, dans l'exemple ci-dessus a été placée dans le dossier `/usr/local/j2sdk1.4.1` :

```
[root@tahe java]# ls -l /usr/local/j2sdk1.4.1/
total 10424
drwxr-xr-x  2 1000    1000      4096 fév 11 19:11 bin
-rw-r--r--  1 1000    1000      4431 fév 11 19:11 COPYRIGHT
drwxr-xr-x  6 1000    1000      4096 fév 11 19:11 demo
drwxr-xr-x  3 1000    1000      4096 fév 11 19:11 include
-rw-r--r--  1 1000    1000      4360 fév 11 19:11 INSTALL
drwxr-xr-x  5 1000    1000      4096 fév 11 19:10 jre
drwxr-xr-x  2 1000    1000      4096 fév 11 19:11 lib
-rw-r--r--  1 1000    1000     16955 fév 11 19:11 LICENSE
drwxr-xr-x  4 1000    1000      4096 fév 11 19:11 man
-rw-r--r--  1 1000    1000      9510 fév 11 19:11 README
-rw-r--r--  1 1000    1000     16218 fév 11 19:11 README.html
-rw-r--r--  1 1000    1000    10567358 fév 11 19:11 src.zip
```

Pour tester votre installation, écrivez le programme Java suivant qui écrit *coucou* à l'écran et sauvegardez-le sous le nom `coucou.java` (le fichier doit porter le nom de la classe, ici `coucou`) :

```
import java.io.*;

public class coucou{
    public static void main(String args[]){
        System.out.println("coucou");
        System.exit(0);
    }
}
```

Compilez le programme `coucou.java` :

```
[uti1@tahe java]$ /usr/local/j2sdk1.4.1/bin/javac coucou.java
```

puis exécutez-le :

```
[util@tahe java]$ /usr/local/j2sdk1.4.1/bin/java coucou  
coucou
```

Cette version du jdk est suffisante pour construire n'importe quelle application. Cependant, si vous êtes habitué à développer dans un environnement de développement graphique (Delphi, JBuilder, C++ Builder, Visual Studio,...) vous trouverez peut-être "rustique" l'environnement de développement du jdk. La société Inprise a porté sur Linux son environnement de développement Java appelé Jbuilder. Nous présentons maintenant l'installation de cet environnement.

18.6 Jbuilder

Recherchons Jbuilder avec Google :



Nous allons sur le site de Borland :



Suivons les liens Jbuilder Personal jusqu'à la page de téléchargement :

URL : http://www.borland.com/products/downloads/download_jbuilder.html

► JBuilder Downloads

Downloads (keys where required)					
Name	Platform	Version	Release Date	Size	Notes
Enterprise Trial	Windows	8	12/02/2002	133 Mb	English, French, German, Japanese
Personal	Windows, Solaris, Linux	8	12/26/2002	47.1 - 71 Mb	English, French, German, Japanese, Spanish

Suivons le lien **Personal**. Une demande d'authentification est faite. Si on est nouveau, il faut remplir un formulaire. Si on est un utilisateur enregistré, on donne son adresse mél et son mot de passe. Cette étape est obligatoire. Le produit Jbuilder nécessite une clé d'activation qui est envoyée par mél.

URL : http://bdn.borland.com/cgi-bin/surveys/web_download_sur

Borland®

Please log in. Or if you are a new user, click the New User button.

Cookies Required.

Please enter your

Login Name
 or Email Address
 and Password

Après avoir donné moult renseignements, on finit par arriver à la page de téléchargement :

URL : http://bdn.borland.com/cgi-bin/surveys/thanx.cgi?jb8_per_kdown

We are currently experiencing heavy volume on this download.
If you have trouble downloading a file, please try the link again at a later time.

The documentation and sample files for JBuilder 8 Personal are available as a [separate download](#)

Language	Platform	Download ftp	Download http
English	Windows	jb8_windows.zip (59.2 Mb)	jb8_windows.zip (59.2 Mb)
	Solaris	jb8_solaris.tar.gz (75.1 Mb)	jb8_solaris.tar.gz (75.1 Mb)
	Linux	jb8_linux.tar.gz (71 Mb)	jb8_linux.tar.gz (71 Mb)
French	Windows	jb8_windows.zip (59.5 Mb)	jb8_windows.zip (59.5 Mb)
	Solaris	jb8_solaris.tar.gz (75.8 Mb)	jb8_solaris.tar.gz (75.8 Mb)
	Linux	jb8_linux.tar.gz (71.7 Mb)	jb8_linux.tar.gz (71.7 Mb)

Ici, nous téléchargeons la version pour linux. Par ailleurs, pour obtenir la documentation et des exemples, il faut suivre le lien **separate download** ci-dessus. Au final, nous obtenons les trois fichiers suivants :



Chaque fichier est décompressé :

```
[root@tahe downloads]# tar xvzf jb8_linux.tar.gz
[root@tahe downloads]# tar xvzf jb8samples.tar.gz
[root@tahe downloads]# tar xvzf jb8_linux.tar.gz
```

Cela donne naissance à trois dossiers :

```
[root@tahe downloads]# ls -ld jb8*
drwxr-xr-x  2 nobody  nobody      4096 avr  8 13:48 jb8docs
drwxr-xr-x  2 nobody  nobody      4096 avr  8 13:48 jb8_linux
drwxr-xr-x  2 nobody  nobody      4096 avr  8 13:48 jb8samples
```

Le dossier **jb8_linux** permet l'installation de Jbuilder :

```
[root@tahe downloads]# cd jb8_linux
[root@tahe jb8_linux]# ls -l
total 85612
...
-rwxr-xr-x  1 nobody  nobody    87262327 déc  9 17:00 per_install.bin
...
```

Le script **per_install.bin** permet l'installation du produit :

```
[root@tahe downloads]# ./per_install.bin
```

Une installation à l'aide d'un assistant graphique s'ensuit. On pourra accepter les valeurs proposées par défaut notamment le dossier d'installation (ici /opt/JBuilder8). On procédera de même pour installer l'aide de jbuilder et les exemples. A chaque fois, il y a un script **per_install.bin** à exécuter. Le contenu du dossier de Jbuilder ressemble à ce qui suit :

```
[root@tahe jb8_linux]# ls -l /opt/JBuilder8/
total 364
drwxrwxrwx  2 root  root      4096 avr  8 13:57 bin
-rw-rw-r--  1 root  root    22541 avr  8 13:58 Borland_JBuilder_8_Personnel_InstallLog.log
-rwxrwxrwx  1 root  root      20 jan 27 21:52 buildnum.txt
-rw-rw-r--  1 root  root     194 avr  8 13:58 ca.list
-rw-rw-r--  1 root  root    31602 avr  8 13:58 copyrights.html
-rw-rw-r--  1 root  root    3381 avr  8 13:58 country.list
drwxrwxrwx  3 root  root     4096 avr  8 13:58 defaults
drwxrwxrwx  2 root  root     4096 avr  8 13:57 doc
drwxrwxr-x  3 root  root     4096 avr  8 13:57 ia_jre_extraction_dir_2596.tmp
drwxrwxr-x  8 root  root     4096 avr  8 13:57 jdk1.4
drwxrwxrwx  5 root  root     4096 avr  8 13:58 lib
-rw-rw-r--  1 root  root   41524 avr  8 13:58 license.html
drwxrwxrwx  2 root  root     4096 avr  8 13:58 patch
-rw-rw-r--  1 root  root     782 avr  8 13:58 phone.list
-rw-rw-r--  1 root  root    6713 avr  8 13:58 privacy.html
drwxrwxrwx  2 root  root     4096 avr  8 13:58 redist
-rw-rw-r--  1 root  root    12746 avr  8 13:58 register_jb.html
-rw-rw-r--  1 root  root    94709 avr  8 13:58 release_notes.html
drwxrwxrwx  3 root  root     4096 avr  8 13:58 samples
-rw-rw-r--  1 root  root    6582 avr  8 13:58 setup_linux.html
drwxrwxrwx  2 root  root     4096 avr  8 13:58 UninstallPersonal
-rw-rw-r--  1 root  root     705 avr  8 13:58 us.list
-rw-rw-r--  1 root  root   69332 avr  8 13:58 whatsnew.html
```

L'exécutable Jbuilder est dans le dossier **bin**. Jbuilder peut être lancé de la façon suivante :

```
[root@tahe bin]# jbuilder &
JBuilder 8 Personal (Non enregistré)
Usage NON commercial
Copyright (c) 1996-2002 Borland Software Corporation.  TOUS DROITS RESERVES.
La licence a expiré
```

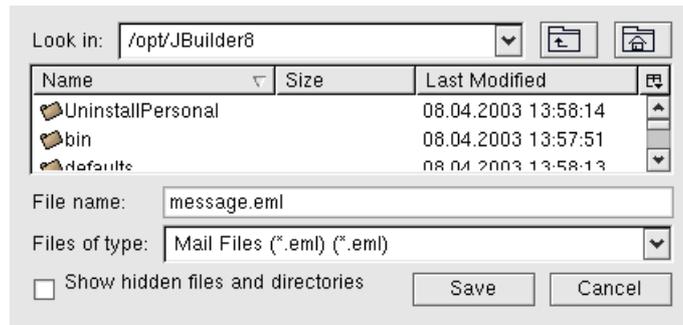
Lors de son premier lancement, Jbuilder cherche une clé d'activation qu'il ne trouve pas :



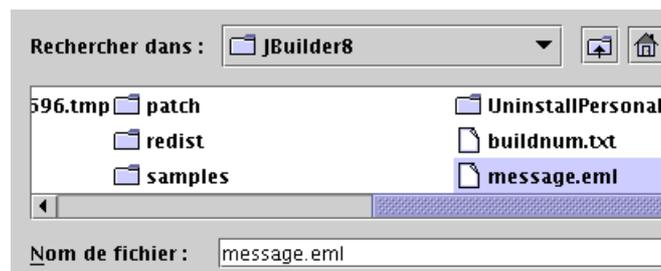
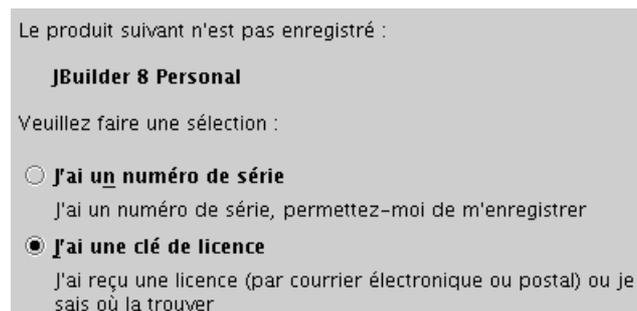
Si on a suivi la procédure de téléchargement, une clé d'activation a été envoyée par mél.

Subject	Sender	Date
Enregistrement de produit Borland	Borland	09:20

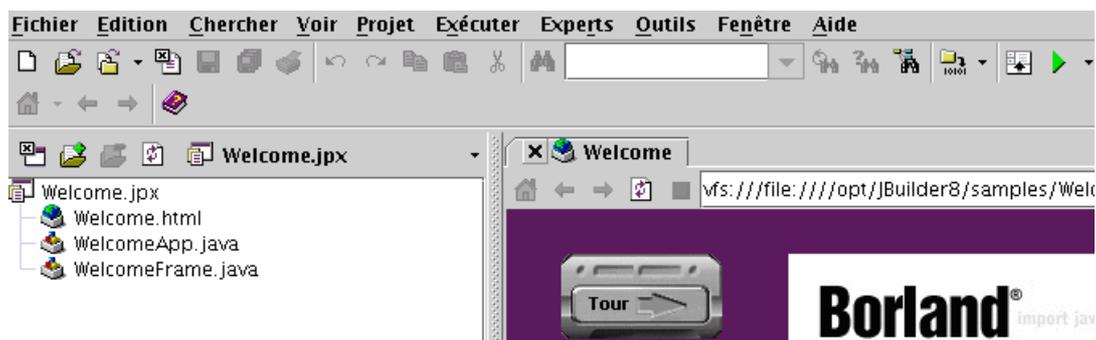
Il suffit de sauvegarder ce mél dans le répertoire d'installation de Jbuilder :



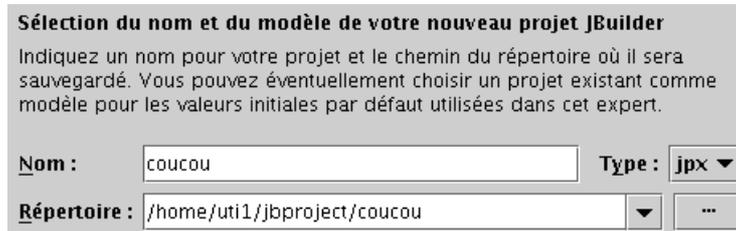
Une fois sauvegardé le fichier de licence, on peut relancer Jbuilder et lui indiquer qu'on a une clé de licence :



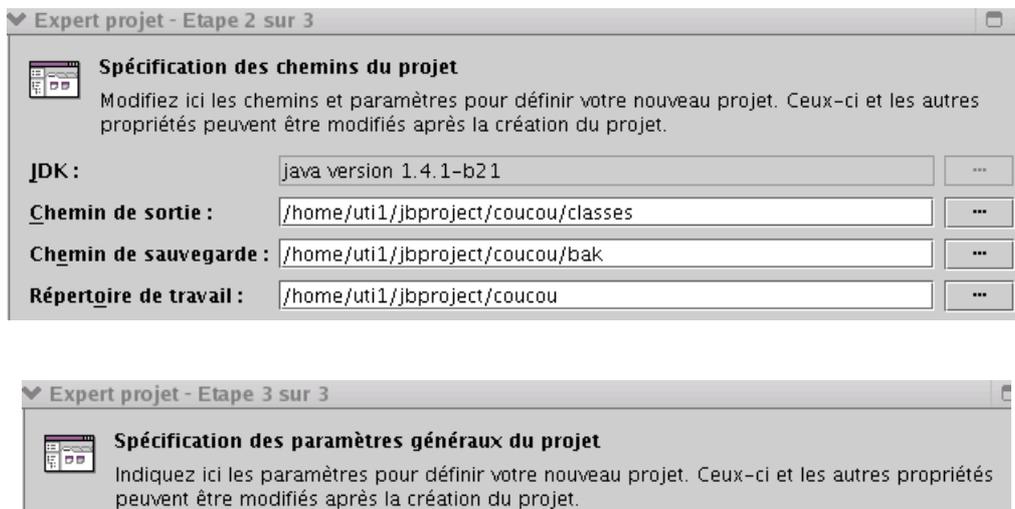
Ceci fait, Jbuilder se lance :



Nous pouvons maintenant écrire notre premier programme. Il est préférable de ne pas être **root**. Nous nous mettons sous l'identité unix **uti1** et lançons Jbuilder. Nous prenons l'option **Fichier/Nouveau projet**. Un assistant à trois écrans nous aide à définir le projet.



Ici, nous définissons le nom du projet : **coucou**. Nous acceptons le répertoire par défaut qui est proposé. Pour les deux écrans suivants, nous acceptons les valeurs par défaut.



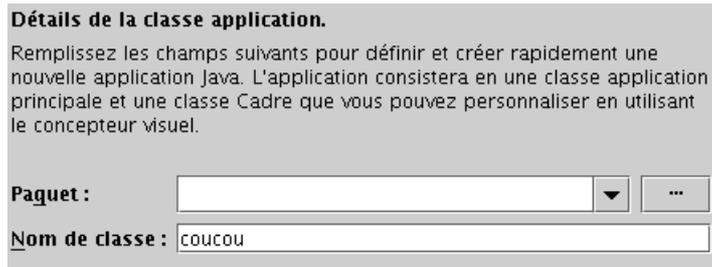
Dans l'IDE de Jbuilder, le projet apparaît alors :



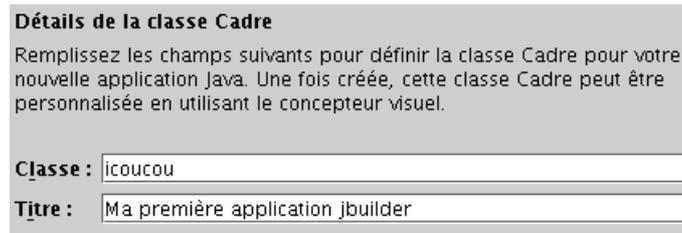
Un projet est constitué d'un ensemble de fichiers de classe, HTML, ... Indiquons que nous voulons créer une application avec interface visuelle. Pour cela, nous prenons l'option **Fichier/Nouveau** :



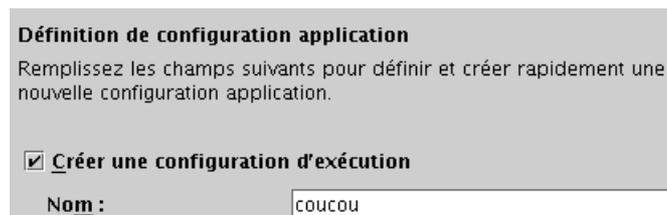
Ici, nous choisissons **Application** qui correspond à une application avec interface visuelle. Un nouvel assistant apparaît :



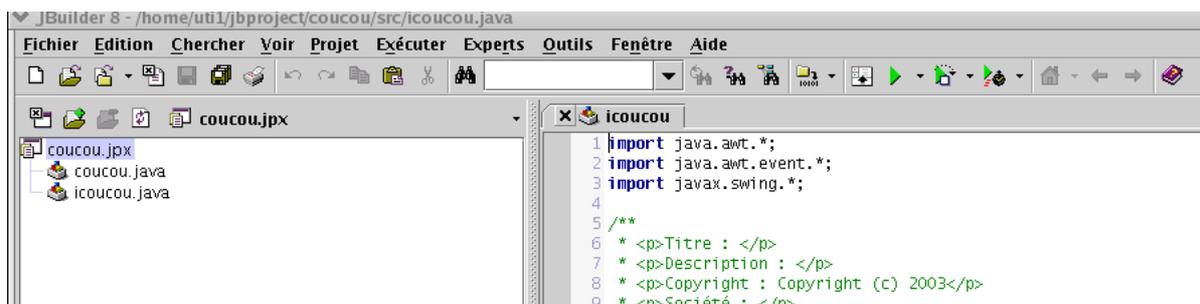
Nous ne prenons pas les valeurs par défaut proposées mais celles affichées ci-dessus. Nous créons ici une classe principale qui sera appelée **coucou.java**. Cette classe principale utilisera une classe pour gérer l'interface visuelle. C'est l'objet de l'écran suivant :



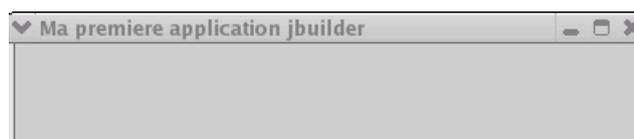
Ici, nous définissons la classe java qui représentera l'interface visuelle. Elle s'appellera **icoucou.java**. Nous définissons également le titre de la fenêtre qui sera affichée. Nous passons ensuite à l'écran suivant où nous acceptons la valeur par défaut proposée :



Jbuilder génère les deux classe **coucou.java** et **icoucou.java** :



Nous pouvons exécuter cette application de base par **Exécuter/Exécuter le projet**. La fenêtre est alors affichée :



On retrouve bien le titre donné à celle-ci lors de la configuration de l'application. Maintenant, il faudrait entrer dans les détails de Jbuilder pour développer une véritable application. Ce travail est laissé au lecteur. Notre objectif était ici de montrer qu'il existe un outil professionnel pour développer en java sous linux. Du fait de la portabilité de java, les applications développées sous linux seront utilisables sous windows.

Sun propose un produit appelé **Sun One Studio** qui semble supérieur à Jbuilder. Il est cependant d'abord plus complexe parce qu'offrant davantage de possibilités. Il est lui aussi gratuit et est disponible pour différents OS dont Linux et Windows.

Développement Web

autour du serveur Apache

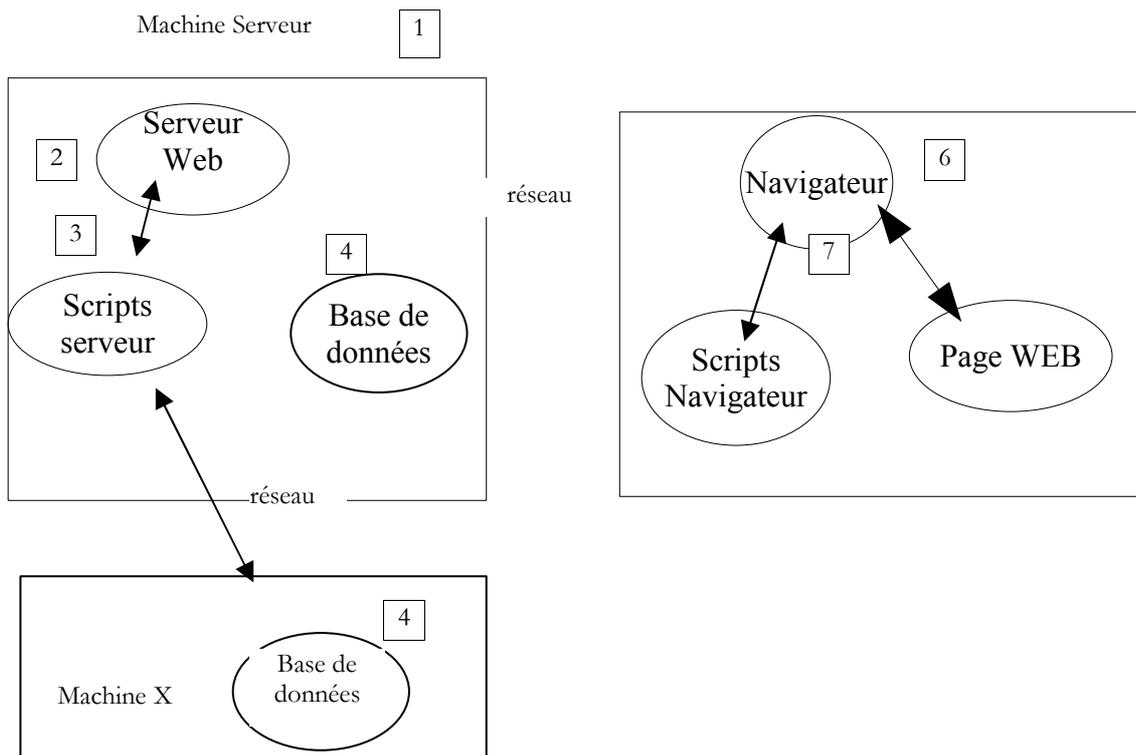
19 Développement Web

Nous donnons ci-dessous les échanges d'informations que l'on trouve dans une application Web. Il est important de savoir ce que peut faire un serveur web et quel est le rôle d'un client web.

19.1 Les composantes d'une application Web

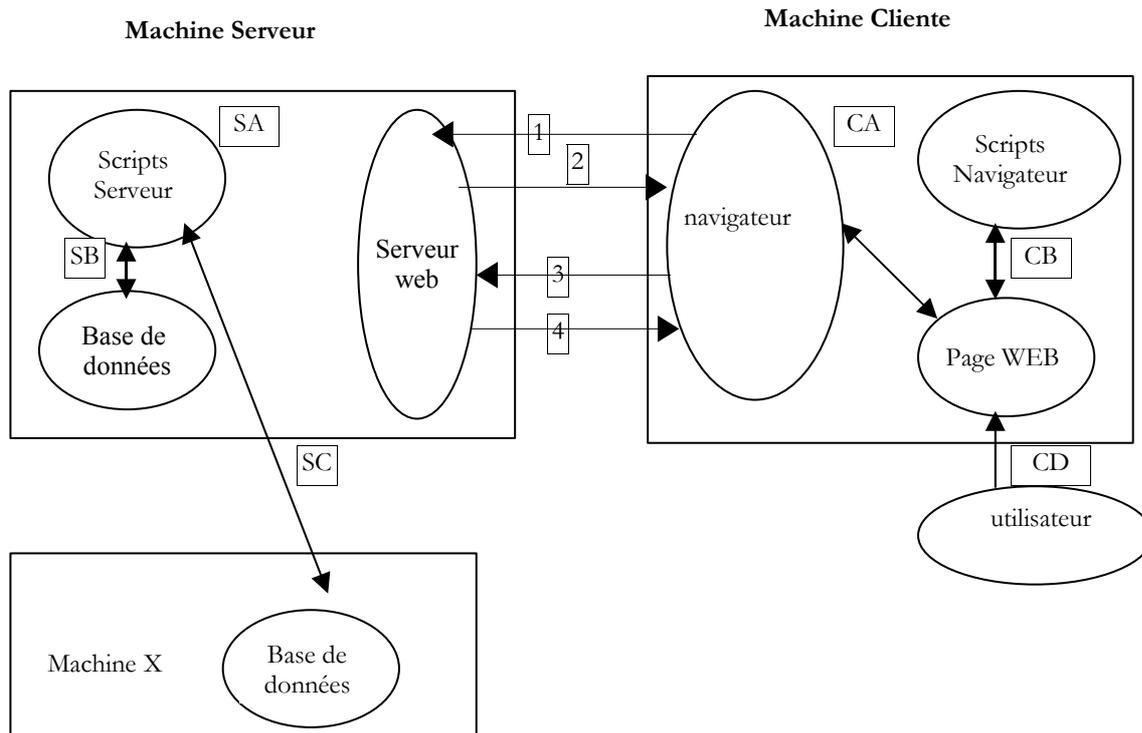
Machine Cliente

5



n°	Rôle	Exemples courants
1	OS Serveur	Linux, Windows
2	Serveur Web	Apache (Linux, Windows) IIS (NT), PWS(Win9x)
3	Scripts exécutés côté serveur. Ils peuvent l'être par des modules du serveur ou par des programmes externes au serveur (CGI).	PERL (Apache, IIS, PWS) VBSCRIPT (IIS,PWS) JAVASCRIPT (IIS,PWS) PHP (Apache, IIS, PWS) JAVA (Apache, IIS, PWS)
4	Base de données - Celle-ci peut être sur la même machine que le programme qui l'exploite ou sur une autre via Internet.	Oracle (Linux, Windows) MySQL (Linux, Windows) Access (Windows) SQL Server (Windows)
5	OS Client	Linux, Windows
6	Navigateur Web	Netscape, Internet Explorer
7	Scripts exécutés côté client au sein du navigateur. Ces scripts n'ont aucun accès aux disques du poste client.	VBscript (IE) Javascript (IE, Netscape) Perlscript (IE) Applets JAVA

19.2 Les échanges dans une application web comportant un formulaire



n°	Rôle
----	------

- 1 Le navigateur demande une URL pour la 1ère fois (*http://machine/url*). Aucun paramètre n'est passé.
- 2 Le serveur Web lui envoie la page Web de cette URL. Elle peut être statique ou bien dynamique, générée par un script serveur (SA) qui a pu utiliser le contenu de bases de données (SB, SC). Ici, le programme détectera que l'URL a été demandée sans passage de paramètres et générera la page WEB initiale. Le navigateur reçoit la page. Il l'affiche (CA). Des scripts côté navigateur (CB) ont pu modifier la page initiale envoyée par le serveur. Ensuite par des interactions entre l'utilisateur (CD) et les scripts (CB) et éventuellement d'autres machines (CE) la page Web va être modifiée. Les formulaires vont notamment être remplis.
- 3 Le navigateur redemande la même URL ou une autre mais cette fois-ci avec des paramètres : méthode **GET** `url?param1=val1¶m2=val2&...` ou méthode **POST** `url` et les paramètres sont envoyés en fin de dialogue navigateur/serveur sous la forme `param1=val1¶m2=val2&...`. Le serveur déclenche le programme (SA) associé à l'URL qui va détecter qu'il y a maintenant des paramètres et va donc les traiter.
- 4 Le serveur délivre la page WEB construite par programme (SA, SB, SC). Cette étape est identique à l'étape 2 précédente. Les échanges se font désormais selon les étapes 2 et 3.

19.3 Quelques ressources

Serveur Apache	http://www.apache.org
Serveur IIS, PWS	http://www.microsoft.com
PERL	http://www.activestate.com
PHP	http://www.php.net
VBSSCRIPT, ASP	http://msdn.microsoft.com/scripting/vbscript/download/vbsdoc.exe http://msdn.microsoft.com/scripting/default.htm?scripting/vbscript/

JAVASCRIPT	http://msdn.microsoft.com/scripting/jscript/download/jsdoc.exe http://developer.netscape.com/docs/manuals/index.html
HTML	http://developer.netscape.com/docs/manuals/index.html
JAVA	http://www.sun.com
Base de données	http://www.mysql.com http://www.oracle.com http://www.postgresql.org/

Nous présentons ici plusieurs outils permettant de faire du développement web. Ces outils visent tous à générer des pages HTML dynamiquement. Ils sont dans la pratique souvent liés à une base de données. Nous utilisons Apache comme serveur Web. Il est très répandu et est livré en standard avec Linux. Nous présentons trois outils de développement pour Apache : Perl, Php, Java.

19.4 Apache-Perl

Historiquement, Perl a été le premier langage à être largement utilisé pour créer des pages Web dynamiques. Si vous avez suivi l'installation préconisée au début de ce document, Perl et le module Perl pour Apache sont déjà installés. Vous pouvez le vérifier avec la commande rpm :

```
[root@tahe util]# rpm -qa | grep -i perl
```

```
perl-Parse-Yapp-1.05-26
perl-libxml-perl-0.07-25
perl-XML-Grove-0.46alpha-21
perl-DBI-1.30-1
perl-CGI-2.81-55
perl-DBD-MySQL-2.1017-3
perl-Digest-HMAC-1.01-8
perl-5.8.0-55
perl-DateManip-5.40-27
perl-HTML-Parser-3.26-14
perl-URI-1.21-3
perl-XML-Parser-2.31-12
perl-XML-Dumper-0.4-22
perl-libxml-ennio-1.02-25
perl-XML-Twig-3.05-3
perl-DB_File-1.804-55
mod_perl-1.99_05-3
perl-Digest-SHA1-2.01-6
perl-Net-DNS-0.26-2
perl-SGMLSpm-1.03iii-6
perl-Filter-1.28-9
perl-CPAN-1.61-55
perl-HTML-Tagset-3.03-25
perl-libwww-perl-5.65-2
perl-XML-Encoding-1.01-20
perl-DBD-Pg-1.13-5
```

La bibliothèque Perl est particulièrement riche. Des centaines de modules Perl sont par ailleurs disponibles sur le réseau CPAN (<http://www.cpan.org/>). Le module nécessaire à Apache est ci-dessus :

```
mod_perl-1.99_05-3
```

Le lien Apache-Perl est défini dans le fichier `/etc/httpd.conf` d'Apache. Editez-le et cherchez (ctrl-F) la chaîne `cgi-bin`. Vous trouverez alors la séquence de lignes suivantes :

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
.....
#
# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
```

```
<Directory "/var/www/cgi-bin">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
</Directory>
```

La ligne **ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"** donne un alias (*/cgi-bin/*) au répertoire */var/www/cgi-bin*. Un fichier *F* placé dans le répertoire */var/www/cgi-bin* sera demandé avec l'url **http://machine/cgi-bin/F**. Essayons. Créez le fichier *heure.pl* suivant et sauvegardez-le dans */var/www/cgi-bin* :

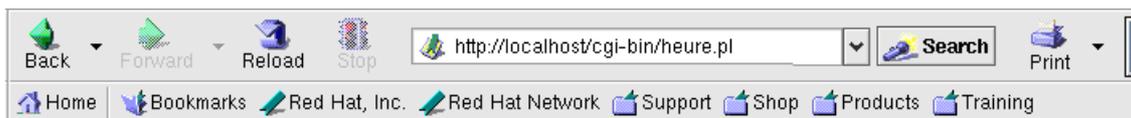
```
#!/usr/bin/perl
use strict;

# on récupère l'heure
my ($secondes,$minutes,$heure)=localtime(time);
# entête HTTP
print "Content-type: text/html\n\n";
# corps HTML
print <<HTML
    <html>
        <head>
            <title>Script CGI Perl</title>
        </head>
        <body>
            <center>
                <h1>Une page générée dynamiquement par un script PERL</h1>
                <h2>Il est $heure:$minutes:$secondes</h2>
                <br>
                A chaque fois que vous rafraîchissez cette page, l'heure change.
            </center>
        </body>
    </html>
HTML
;
# fin du script
exit 0;
```

Ce programme Perl affiche l'heure dans une page HTML. Il nous faut rendre exécutable ce fichier :

```
[root@tahe util]# ls -l /var/www/cgi-bin/heure.pl
-rw-r--r-- 1 root root 506 avr 11 18:31 /var/www/cgi-bin/heure2.pl
[root@tahe util]# chmod o+x /var/www/cgi-bin/heure.pl
[root@tahe util]# ls -l /var/www/cgi-bin/heure.pl
-rw-r--r-x 1 root root 506 avr 11 18:26 /var/www/cgi-bin/heure.pl
```

Maintenant, prenez un navigateur de votre machine et demandez l'url *http://localhost/cgi-bin/heure.pl* :



Une page générée dynamiquement par un script PERL

Il est 18:28:35

A chaque fois que vous rafraîchissez cette page, l'heure change.

Lorsque Apache reçoit une demande pour un script CGI (**C**ommon **G**ateway **I**nterface), il lance à chaque fois l'interpréteur nécessaire à l'exécution du script. Ainsi, ci-dessus, si nous demandons la page *heure.pl* de façon répétée, l'interpréteur Perl sera chargé autant de fois. Cela peut devenir coûteux en ressources et performances si le site abrite des scripts CGI appelés fréquemment. *Pour Perl*, il existe un module Apache appelé **mod_perl** qui contient un interpréteur Perl. Les modules d'Apache sont chargés lors du démarrage du serveur. Leur chargement est explicitement demandé dans le fichier *httpd.conf*. Une fois le module *mod_perl* chargé, il reste en mémoire. Il exécute les scripts Perl que lui donne Apache sans avoir à être systématiquement rechargé, d'où un gain de temps.

Editez de nouveau le fichier *httpd.conf* et recherchez le mot clé *Include* :

```
#
# Load config files from the config directory "/etc/httpd/conf.d".
#
Include conf.d/*.conf
```

Comme l'indique le commentaire, des fichiers de configuration supplémentaires sont inclus dans le dossier **/etc/httpd/conf.d**

```
[root@tahe util]# ls -l /etc/httpd/conf.d/
total 36
-rw-r--r-- 1 root root 3424 sep 2 2002 auth_pgsql.conf
-rw-r--r-- 1 root root 954 mar 21 11:02 perl.conf
-rw-r--r-- 1 root root 459 jan 22 17:40 php.conf
-rw-r--r-- 1 root root 988 sep 2 2002 python.conf
-rw-r--r-- 1 root root 180 oct 9 2002 README
-rw-r--r-- 1 root root 96 jan 6 12:28 squirrelmail.conf
-rw-r--r-- 1 root root 11140 oct 9 2002 ssl.conf
```

On y trouve le fichier **perl.conf** qui configure le module Perl :

```
#
# Mod_perl incorporates a Perl interpreter into the Apache web server,
# so that the Apache web server can directly execute Perl code.
# Mod_perl links the Perl runtime library into the Apache web server
# and provides an object-oriented Perl interface for Apache's C
# language API. The end result is a quicker CGI script turnaround
# process, since no external Perl interpreter has to be started.
#
LoadModule perl_module modules/mod_perl.so

# This will allow execution of mod_perl to compile your scripts to
# subroutines which it will execute directly, avoiding the costly
# compile process for most requests.
#
Alias /perl /var/www/perl
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler ModPerl::Registry::handler
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>
```

La ligne

```
LoadModule perl_module modules/mod_perl.so
```

demande le chargement du module *mod_perl*. Apache chargera le fichier */etc/httpd/modules/mod_perl.so* au démarrage. Un fichier *.so* (shared object) est une bibliothèque de fonctions qu'on peut charger dynamiquement (comme une dll sous windows).

```
[root@tahe util]# ls -l /etc/httpd/modules/mod_perl.so
-rwxr-xr-x 1 root root 136226 sep 3 2002 /etc/httpd/modules/mod_perl.so
```

Le paragraphe

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler ModPerl::Registry::handler
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>
```

définit des propriétés si le module *mod_perl* est chargé. Nous en noterons deux :

```
Alias /perl/ /var/www/perl/
```

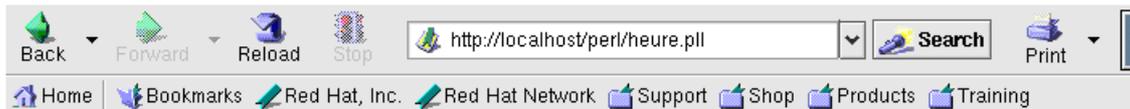
qui associe l'alias */perl* au répertoire */var/www/perl*. La section

```
<Directory /var/www/perl> ... </Directory>
```

définit les propriétés liées à l'alias */perl* donc au répertoire */var/www/perl*. Elles indiquent que ce qui se trouve dans ce répertoire devra être considéré comme un script perl à exécuter. Essayons. Commençons par construire le répertoire */var/www/perl* qui n'existe pas. Puis mettons dedans le script *heure.pl* construit précédemment sous le nom *heure.pll* (pas *heure.pl*).

```
[root@tahe util]# mkdir /var/www/perl
[root@tahe util]# cp /var/www/cgi-bin/heure.pl /var/www/perl/heure.pll
[root@tahe util]# ls -l /var/www/perl/
-rw-r--r-x 1 root root 482 mar 21 10:39 heure.pll
```

Vérifiez bien que le script *heure.pll* a les droits r-x pour tous. Vérifiez que le serveur Apache est lancé. Prenez un navigateur local et demandez l'url *http://localhost/perl/heure.pll* :



Une page générée dynamiquement par un script PERL

Il est 10:12:23

A chaque fois que vous rafraîchissez cette page, l'heure change.

Seul l'administrateur système pouvait créer les deux scripts précédents. En effet, ils devaient être placés dans des répertoires appartenant à *root*. Vous pouvez configurer Apache pour permettre à tout utilisateur d'inclure des scripts Perl dans son site personnel. C'est à vous de voir si c'est souhaitable ou non. Pour le faire, il nous faut modifier *perl.conf* :

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler ModPerl::Registry::handler
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>
```

Faites un copier/coller de la partie **<Directory ...> ... </Directory>** et modifiez-la comme suit :

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler ModPerl::Registry::handler
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>
<Files ~ *\*.pll>
    SetHandler perl-script
    PerlHandler ModPerl::Registry::handler
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Files>
```

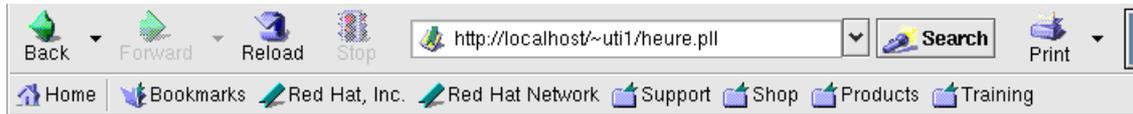
La section **<Files ~ "*"*.pll">...</Files>** fixe les propriétés des fichiers ayant le suffixe *.pll*. On donne à ces fichiers les mêmes droits qu'aux fichiers du répertoire */var/www/perl*. Ils seront donnés à *mod_perl* pour exécution. Sauvegardez ces modifications et relancez Apache pour qu'il les prenne en compte :

```
[/etc/rc.d/init.d/httpd restart]
Shutting down http: [ OK ]
Starting httpd: [ OK ]
```

Maintenant copions le script *heure.pll* déjà étudié, dans le site personnel de l'utilisateur *uti1*, *~uti1/WWW* pour nous, où *~uti1* désigne le répertoire de connexion de *uti1*.

```
[root@tahe util]# cp /var/www/perl/heure.pll /home/uti1/WWW/heure.pll
[root@tahe util]# ls -l /home/uti1/WWW/heure.pll
-rw-r--r-x  1 root    root      506 avr 12 10:21 /home/uti1/WWW/heure.pll
[root@tahe util]# chown uti1:groupe1 /home/uti1/WWW/heure.pll
[root@tahe util]# ls -l /home/uti1/WWW/heure.pll
-rw-r--r-x  1 uti1    groupe1  506 avr 12 10:21 /home/uti1/WWW/heure.pll
```

Le script *heure.pll* est maintenant dans le répertoire personnel de *uti1* et lui appartient. Vérifiez qu'il a bien les droits r-x pour tous. Maintenant demandez l'url *http://localhost/~uti1/heure.pll* qui est l'url de notre script ci-dessus :



Une page générée dynamiquement par un script PERL

Il est 10:28:28

A chaque fois que vous rafraîchissez cette page, l'heure change.

Maintenant vos utilisateurs peuvent inclure des scripts perl dans leur site web personnel. Encore une fois, c'est à l'administrateur de voir si c'est souhaitable ou non. Plus on accorde de facilité aux utilisateurs, plus la sécurité du système est fragilisée.

19.5 Apache-PHP

De la même façon qu'on utilise Perl pour générer dynamiquement des pages HTML, on peut utiliser un autre langage : PHP (**P**ersonal **H**ome **P**age, **H**ypertext **P**reprocessor). C'est un langage plus récent que Perl et qui en a repris beaucoup d'attributs. Il n'y a pas de difficulté particulière à passer de Perl à PHP et vice-versa. Si vous avez suivi le mode d'installation préconisé au début de ce document, PHP est déjà présent sur votre système. Utilisez la commande **rpm** pour le vérifier :

```
[root@tahe util]# rpm -qa | grep -i php
php-ldap-4.2.2-8.0.7
php-imap-4.2.2-8.0.7
php-pgsql-4.2.2-8.0.7
php-4.2.2-8.0.7
```

L'interpréteur PHP est fourni par le module *php-4.2.2* ci-dessus. S'il n'était pas présent, utilisez l'application **Menu démarrer/Paramètres de système/Paquetages** pour l'installer. Il nous faut voir maintenant les liens entre PHP et le serveur Web Apache. Nous avons déjà vu qu'un certain nombre de fichiers de configuration étaient présents dans */etc/httpd/conf.d* :

```
[root@tahe util]# ls -l /etc/httpd/conf.d/
total 36
-rw-r--r--  1 root    root      3424 sep  2  2002 auth_pgsq1.conf
-rw-r--r--  1 root    root      954 mar 21 11:02 perl.conf
-rw-r--r--  1 root    root      459 jan 22 17:40 php.conf
-rw-r--r--  1 root    root      988 sep  2  2002 python.conf
-rw-r--r--  1 root    root      180 oct  9  2002 README
-rw-r--r--  1 root    root       96 jan  6 12:28 squirrelmail.conf
-rw-r--r--  1 root    root     11140 oct  9  2002 ssl.conf
```

On y trouve le fichier **php.conf** qui configure l'utilisation de PHP avec Apache :

```
#
# PHP is an HTML-embedded scripting language which attempts to make it
# easy for developers to write dynamically generated webpages.
#
```

```
LoadModule php4_module modules/libphp4.so

#
# Cause the PHP interpreter handle files with a .php extension.
#
<Files *.php>
    SetOutputFilter PHP
    SetInputFilter PHP
    LimitRequestBody 524288
</Files>

#
# Add index.php to the list of files that will be served as directory
# indexes.
#
DirectoryIndex index.php
```

Le module nécessaire au module Apache/PHP est **libphp4.so** :

```
[root@tahe util]# ls -l /etc/httpd/modules/libphp4.so
-rwxr-xr-x 1 root root 1442301 jan 22 17:40 /etc/httpd/modules/libphp4.so
```

Le fichier de configuration ci-dessus dit essentiellement que tout fichier ayant le suffixe `.php` doit être traité par l'interpréteur PHP. Essayons. Construisons le fichier texte suivant sous l'identité **util** :

```
<?php
    phpinfo();
?>
```

Il contient du code PHP entre les balises `<?php ... ?>`. Il fait appel à une fonction **phpinfo** qui affiche des informations très utiles sur la configuration de PHP. Sauvegardez ce fichier dans **/home/uti1/WWW** sous le nom **phpinfo.php** :

```
[root@tahe util]# ls -l /home/uti1/WWW/phpinfo.php
-rw----r-x 1 uti1 groupe1 22 mar 21 11:10 /home/uti1/WWW/phpinfo.php
```

Assurez-vous que les droits `r` et `x` sont donnés à tous les utilisateurs. Demandez maintenant avec un navigateur local l'url `http://localhost/~uti1/phpinfo.php` :

PHP Version 4.2.2	
System	Linux stripples.devel.redhat.com 2.4.20-2.11smp #1 SMP Fri Jan 10 06:18:38 EST 2003 i686 i686 i386 GNU/Linux
Build Date	Jan 22 2003 11:39:47
Configure Command	./configure '--host=i386-redhat-linux' '--build=i386-redhat-linux' '--target=i386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--prefix=/usr' '--with-config-file-path=/etc' '--enable-force-cgi-redirect' '--disable-debug' '--enable-pic' '--disable-rpath' '--enable-inline-optimization' '--with-bz2' '--with-db3' '--with-curl' '--with-dom=/usr' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--with-gd' '--enable-gd-native-ttf' '--with-ttf' '--with-gdbm' '--with-gettext' '--with-ncurses' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pspell' '--with-regex=system' '--with-xml' '--with-xml2' '--with-xml2-dir=/usr' '--with-zlib' '--with-layout=GNU' '--enable-bcmath' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-safe-mode' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-discard-path' '--enable-track-vars' '--enable-trans-sid' '--enable-yp' '--enable-wddx' '--without-oci8' '--with-pear=/usr/share/pear' '--with-imap=shared' '--with-imap-ssl' '--with-kerberos=/usr/kerberos' '--with-ldap=shared' '--with-mysql=shared,/usr' '--with-pgsql=shared' '--with-snmp=shared,/usr' '--with-snmp=shared' '--enable-ucd-snmp-hack' '--with-unixODBC=shared' '--enable-memory-limit' '--enable-bcmath' '--enable-shmop' '--enable-versioning' '--enable-calendar' '--enable-dbx' '--enable-dio' '--enable-mcal' '--with-apxs2=/usr/sbin/apxs'
Server API	Apache 2.0 Filter
Virtual Directory Support	disabled

Si vous vous intéressez à la programmation PHP, il est du plus grand intérêt de lire cette page. Reprenons maintenant le script de l'heure vu avec Perl et adaptons-le à PHP :

```
<html>
<head>
<title>Script PHP</title>
</head>
```

```

<body>
  <center>
    <h2>Une page générée par un script PHP</h2>
    <h3>Il est <?php echo date('j/m/y, h:i:s',time()) ?></h3>
    <br>
    A chaque fois que vous rafraîchissez cette page, l'heure change.
  </center>
</body>
</html>

```

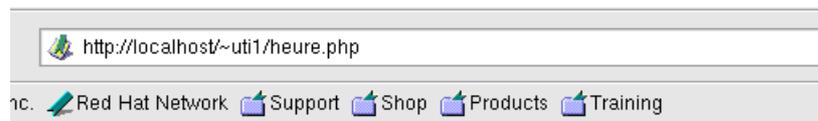
Sauvegardez ce texte dans le fichier `/home/uti1/WWW/heure.php` :

```

[root@tahe uti1]# ls -l /home/uti1/WWW/heure.php
-rwx---r-x  1 uti1  groupe1  304 mar 21 12:56 /home/uti1/WWW/heure.php

```

Vérifiez bien les droits du fichier. Ils doivent être ceux indiqués ci-dessus. Demandez maintenant l'URL <http://localhost/~uti1/heure.php> dans un navigateur local :



Une page générée par un script PHP

Il est 12/04/03, 11:04:20

A chaque fois que vous rafraîchissez cette page, l'heure change.

19.6 Apache - Java

De la même façon qu'on utilise Perl et PHP pour générer des pages dynamiquement HTML, on peut utiliser Java. L'URL demandée par un navigateur client peut être directement une classe Java : on parle alors de **servlets Java**. L'URL peut aussi désigner un fichier texte mélangeant code HTML et code Java de façon analogue aux pages PHP où sont mélangés code HTML et code PHP : on parle alors de **pages JSP** (Java Server Pages). Apache n'est pas pour l'instant un serveur pouvant gérer ces technologies. Mais il peut le faire à l'aide d'outils tiers. Nous allons présenter ici l'un d'entre-eux : **TOMCAT**. Tomcat est un serveur web à l'instar d'Apache. Il est capable de gérer des servlets et des pages JSP. Il y a moyen de configurer le serveur Apache pour qu'il travaille avec un serveur Tomcat. Lorsqu'un navigateur demande une servlet Java ou une page Jsp, Apache demande la page au serveur Tomcat qui la génère. Apache se contente de renvoyer au client ce que Tomcat lui a donné.

19.6.1 Installation du serveur web Tomcat

Tomcat est disponible à l'URL : <http://jakarta.apache.org/downloads/binindex.html>. Il existe des sites miroirs, par exemple celui-ci : <http://apache.crihan.fr/dist/jakarta/tomcat-4/binaries/>. A cette adresse, on a récupéré la version Tomcat tomcat-4.1.24.tar.gz. L'installation consiste simplement à décompresser cette archive dans un répertoire sous l'identité `root` :

```

[root@tahe downloads]# tar xvzf tomcat-4.1.24.tar.gz

```

On récupère un répertoire `jakarta-tomcat-4.1.24`

```

[root@tahe downloads]# ls -ld jakarta-tomcat-4.1.24/
drwxr-xr-x  11 root  root  4096 avr 12 11:13 jakarta-tomcat-4.1.24/

```

Nous déplaçons ce répertoire à un endroit adéquat de notre système, par exemple `/usr/local` :

```

[root@tahe downloads]# mv jakarta-tomcat-4.1.24/ /usr/local/
[root@tahe downloads]# ls -ld /usr/local/jakarta-tomcat-4.1.24/

```

```

drwxr-xr-x  11 root    root      4096 avr 12 11:13 /usr/local/jakarta-tomcat-4.1.24/
[root@tahe downloads]# ls -l /usr/local/jakarta-tomcat-4.1.24/
total 108
drwxr-xr-x  2 root    root      4096 avr 12 11:13 bin
drwxr-xr-x  5 root    root      4096 mar 19 10:20 common
drwxr-xr-x  2 root    root      4096 avr 12 11:13 conf
-rw-r--r--  1 root    root     4568 mar 19 10:20 LICENSE
drwxr-xr-x  2 root    root      4096 mar 19 10:20 logs
-rw-r--r--  1 root    root     3150 mar 19 10:20 README.txt
-rw-r--r--  1 root    root    43343 mar 19 10:20 RELEASE-NOTES-4.1.txt
-rw-r--r--  1 root    root     5149 mar 19 10:20 RELEASE-PLAN-4.1.txt
-rw-r--r--  1 root    root     6701 mar 19 10:20 RUNNING.txt
drwxr-xr-x  5 root    root      4096 mar 19 10:20 server
drwxr-xr-x  4 root    root      4096 mar 19 10:20 shared
drwxr-xr-x  2 root    root      4096 avr 12 11:13 temp
drwxr-xr-x  6 root    root      4096 avr 12 11:13 webapps
drwxr-xr-x  2 root    root      4096 mar 19 10:20 work

```

Des fichiers d'aide au format HTML sont disponibles dans le dossier `/usr/local/jakarta-tomcat-4.1.24/webapps/tomcat-docs/`. Voici le fichier `index.html` :



Le script de lancement du serveur Tomcat se trouve dans le dossier `bin` de son dossier d'installation :

```

[root@tahe downloads]# ls -l /usr/local/jakarta-tomcat-4.1.24/bin/startup.sh
-rwxr-xr-x  1 root    root      788 mar 19 10:20 /usr/local/jakarta-tomcat-4.1.24/bin/startup.sh

```

Lançons Tomcat :

```

[root@tahe downloads]# /usr/local/jakarta-tomcat-4.1.24/bin/startup.sh
The JAVA_HOME environment variable is not defined
This environment variable is needed to run this program

```

Tomcat a besoin d'une machine virtuelle Java et ne sait où la trouver. Il faut lui fixer une variable d'environnement `JAVA_HOME`. Dans notre exemple, Java a été installé dans `/usr/local/j2sdk1.4.1/` :

```

[root@tahe downloads]# ls -ld /usr/local/j2sdk1.4.1/
drwxr-xr-x  8 root    root      4096 fév 11 19:11 /usr/local/j2sdk1.4.1/

```

Nous créons la variable d'environnement `JAVA_HOME` :

```

[root@tahe downloads]# export JAVA_HOME=/usr/local/j2sdk1.4.1/

```

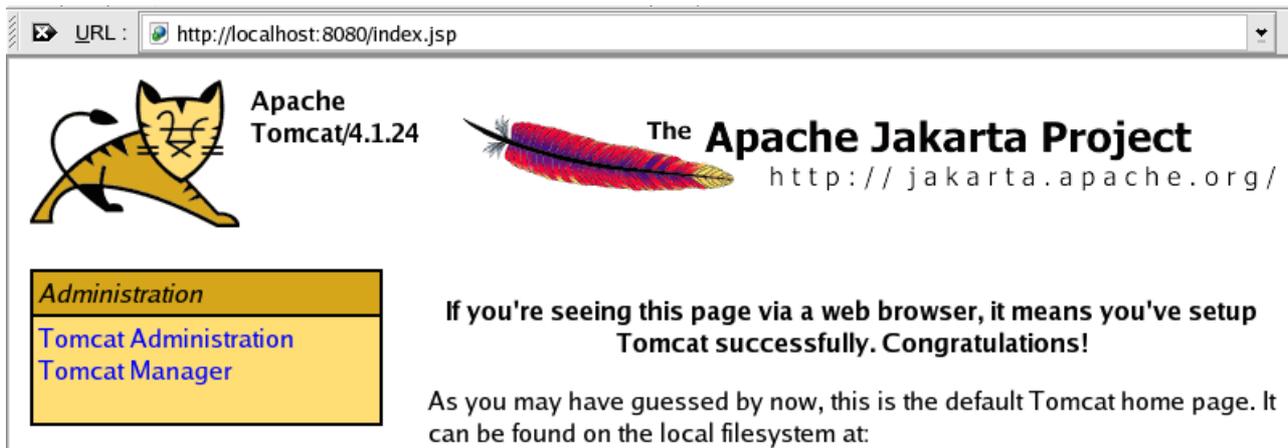
et essayons de relancer Tomcat :

```

[root@tahe downloads]# /usr/local/jakarta-tomcat-4.1.24/bin/startup.sh
Using CATALINA_BASE:  /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_HOME:  /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:     /usr/local/j2sdk1.4.1/

```

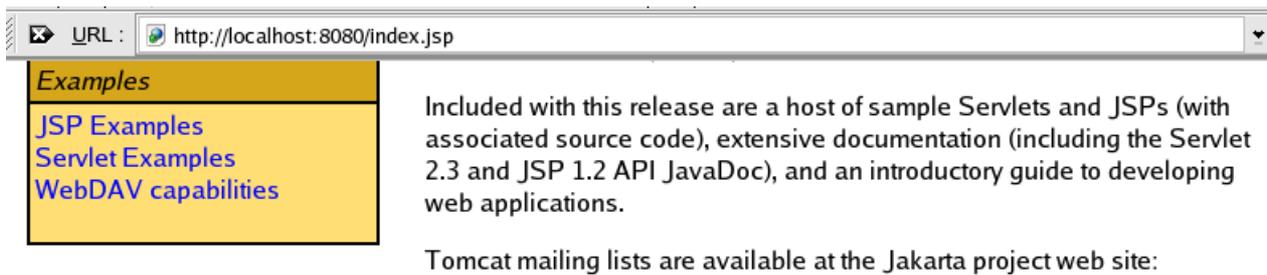
Cette fois-ci, nous n'avons pas d'erreurs. Le serveur Tomcat est un serveur Web qui opère sur le port 8080. Essayons l'URL `http://localhost:8080` :



Tomcat a bien répondu. On voit même que sa page d'entrée a été générée par une page JSP (*index.jsp*). Le serveur est arrêté avec le script **shutdown.sh** :

```
[root@tahe downloads]# /usr/local/jakarta-tomcat-4.1.24/bin/shutdown.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:       /usr/local/j2sdk1.4.1/
```

Relançons Tomcat et revenons sur sa page d'entrée d'URL <http://localhost:8080> :



On lira attentivement les informations données sur cette page d'entrée. Pour l'heure, suivons le lien **JSP Examples** ci-dessus :



Suivons maintenant le lien *snoop* (Execute) :



Vous pouvez tester les différents exemples de servlets Java et de pages JSP qui vous sont proposés.

19.6.2 Un script JSP

Le script suivant `heure.jsp` est une page JSP qui affiche l'heure :

```
<%
  // programme Java affichant l'heure
%>
<% // classes importées %>
<%@ page import="java.util.*" %>

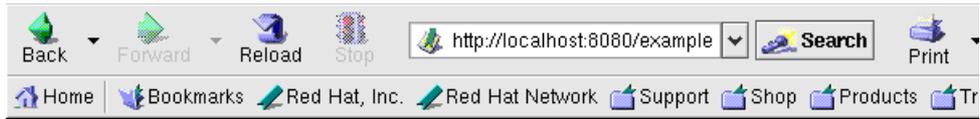
<% // code JAVA pour calculer l'heure
  Date maintenant=new Date();
  String heures="" +maintenant.getHours();
  String minutes="" +maintenant.getMinutes();
  String secondes="" +maintenant.getSeconds();
  // heures, minutes, secondes sont des variables globales
  // qui pourront être utilisées dans le code HTML
%>

<% // code HTML %>
<html>
  <head>
    <title>Page JSP affichant l'heure</title>
  </head>
  <body>
    <center>
      <b1>Une page JSP générée dynamiquement</b1>
      <b2>Il est <%=heures%>:<%=minutes%>:<%=secondes%></b2>
      <br>
      <b3>A chaque fois que vous rechargez la page, l'heure change</b3>
    </body>
  </html>
```

- mettre le script `heure.jsp` dans `/usr/local/jakarta-tomcat-4.1.24/webapps/examples/jsp`

```
[root@tahe root]# ls -l /usr/local/jakarta-tomcat-4.1.24/webapps/examples/jsp/heure.jsp
-rw-r--r-- 1 root root 741 avr 15 11:35 /usr/local/jakarta-tomcat-4.1.24/webapps/examples/jsp/heure.jsp
```

- lancer le serveur Tomcat si besoin est
- demander l'URL `http://localhost:8080/examples/jsp/heure.jsp` avec un navigateur



Une page JSP générée dynamiquement

Il est 11:39:19

A chaque fois que vous rechargez la page, l'heure change

Compléments

20 Configuration d'une imprimante HP Jet Direct

20.1 Configurer l'imprimante avec Menu KDE

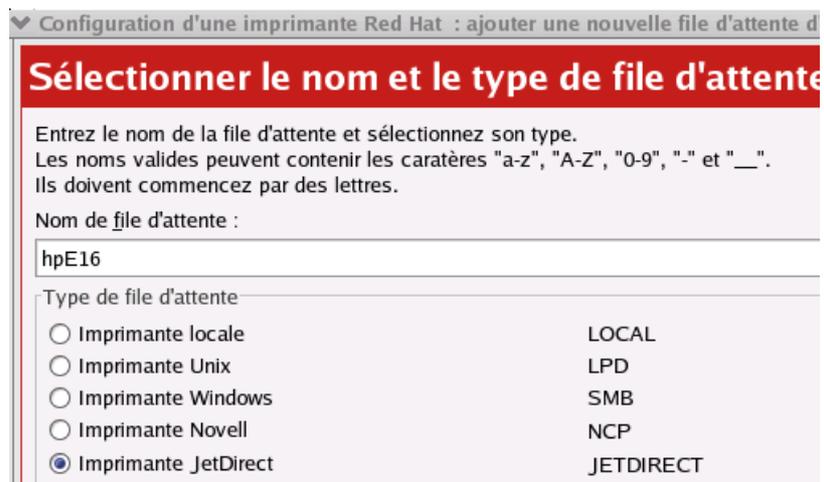
Prendre l'option **Menu/Paramètres de système/Printing** :



Cliquer sur le bouton **[Nouveau]** pour ajouter une nouvelle imprimante. Un assistant de configuration démarre :



Faites **[Suivant]** :



Sélectionnez l'imprimante JetDirect et donnez-lui un nom. Faites **[Suivant]** :



Indiquez l'adresse IP de l'imprimante. Le port proposé par défaut est correct. Si vous ne connaissez pas l'adresse IP de l'imprimante, demandez la à votre instructeur. Faites **[Suivant]** :



Précisez le type d'imprimante HP utilisée puis faites **[Suivant]** :



Le dernier écran récapitule les choix faits. Faites **[Terminer]** pour créer la nouvelle file d'attente.



L'écran ci-dessus montre la nouvelle file d'attente. Utilisez le bouton **[Test]** pour envoyer une page de test à l'imprimante.

20.2 Faire un test avec OpenOffice

Ouvrez OpenOffice Writer par exemple et imprimez un document.

21 Gestion des paquetages

21.1 Mise à jour des paquetages installés

Linux RedHat offre la possibilité de mettre à jour simplement les paquetages installés sur le poste. Après une installation de Linux RedHat, il est intéressant de faire immédiatement une telle mise à jour pour avoir la dernière version des paquetages qu'on a installés.

L'outil de mise à jour est disponible dans la barre des tâches à droite :



Double-cliquez sur l'icône pour lancer l'utilitaire :



Avant de pouvoir procéder à des mises à jour, il faut procéder à la configuration de l'utilitaire. On clique droit sur l'icône et on prend l'option [**Configuration**]. Cliquez sur [**Suivant**]. Apparaît une page d'informations.



Cliquez sur [**Suivant**].



Ici, nous devons configurer le proxy HTTP s'il y en a pour accéder à l'Internet. C'est le cas à l'ISTIA pour le réseau étudiant. Indiquez comme ci-dessus les coordonnées du proxy HTTP puis faites [**Suivant**].



La configuration est terminée. Si plus tard, vous souhaitez la refaire, cliquez droit sur l'icône de mise à jour et prenez l'option [Configuration]. Pour l'instant, utilisez le bouton [Appliquer] puis double-cliquez sur icône **Udpate** et prenez l'option [Démarrage de up2date].



Les paquetages sont signés numériquement. Pour vérifier la validité de la signature numérique, il est nécessaire d'avoir la clé publique de la compagnie RedHat. Il faut répondre [Oui] à l'écran précédent pour installer cette clé sur le poste.



La mise à jour commence par un message de bienvenue. Faites [Suivant].



Encore un message d'informations. Lisez-le et faites [Suivant].

Agent de mise à jour de Red Hat

Step 2: Register or Update a User Account

Required Information

Are you already registered with redhat.com?
 Yes: Enter your current user name and password below.
 No: Choose a new user name and password and enter it below.

Nom d'utilisateur :

Mot de passe :

Password again, for verification:

Adresse e-mail :

La mise à jour nécessite la création d'un compte. Indiquez un login, un mot de passe et votre adresse électronique. A celle-ci, vous recevrez les alertes de RedHat pour des mises à jour critiques du système. Cette inscription vous donne droit à abonnement de base : vous pouvez télécharger les mises à jour mais avec une faible priorité qui fait que si le serveur de mises à jour est chargé, l'accès vous est interdit. Pour une somme modique, vous pouvez obtenir un abonnement débarrassé de cette contrainte. Faites **[Suivant]**.

Agent de mise à jour de Red Hat

Etape 2 : enregistrer un compte utilisateur

All information is optional.

Titre :

First name: Last name:

Société : Emplacement :

Adresse :

Adresse 2 :

Ville : State / Province:

ZIP / Postal code: Pays :

Toujours des renseignements qui sont facultatifs. Faites **[Suivant]**.

Agent de mise à jour de Red Hat

Etape 3 : enregistrer un profil de système - Matériel

A Profile Name is a descriptive name that you choose to identify this System Profile on Red Hat Network web pages. Optionally, include a computer serial or identification number.

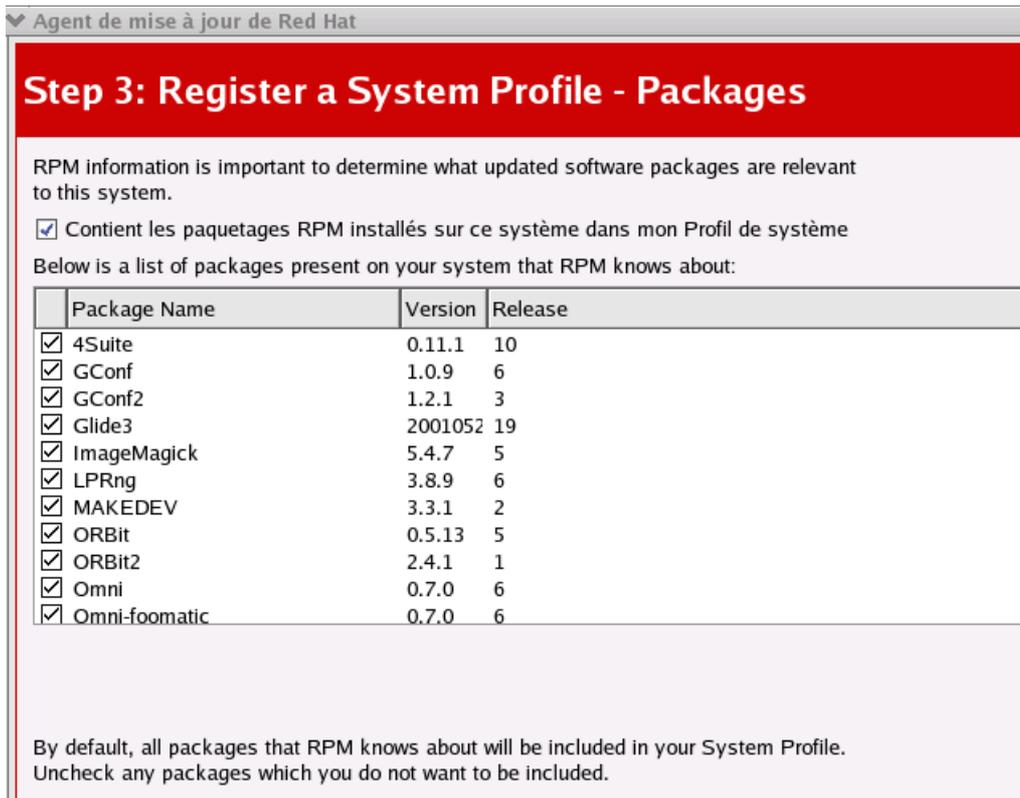
Nom de profil :

Hardware information is important to determine what updated software and drivers are relevant to this system. The minimum set of information you can include will contain your system's architecture and Red Hat Linux version.

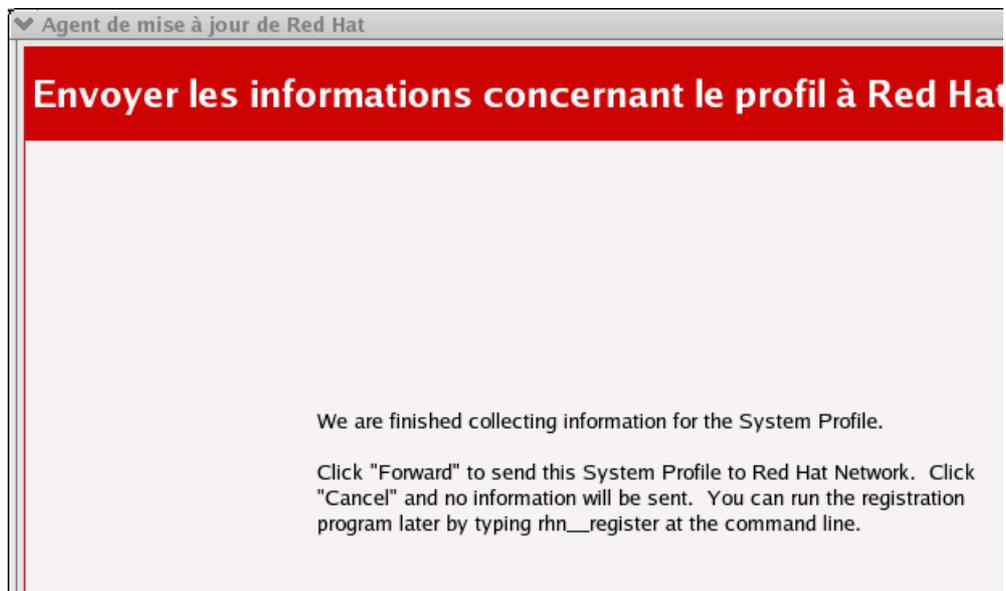
Include information about hardware and network

Included information

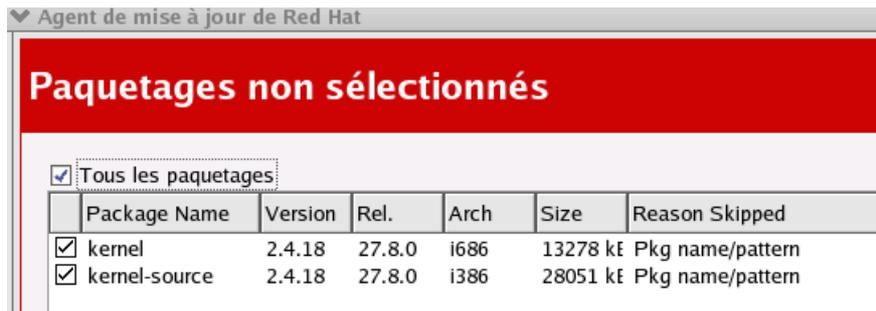
Sur l'écran ci-dessus, des informations par défaut sont proposées. Acceptez-les en faisant **[Suivant]**.



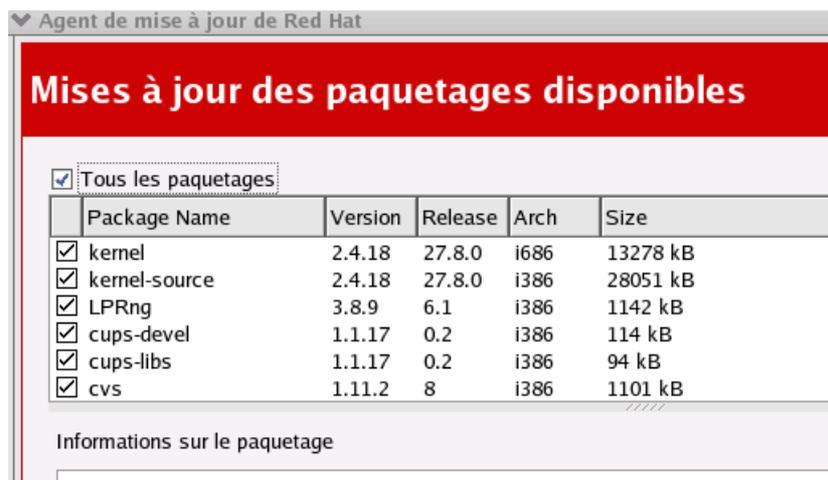
L'écran ci-dessus fait une liste des paquetages installés sur votre machine et pour lesquels l'utilitaire va chercher des mises à jour. Faites **[Suivant]**.



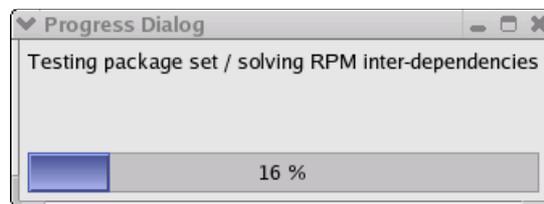
L'écran ci-dessus précède la connexion au serveur de mise à jour. Faites **[Suivant]**.



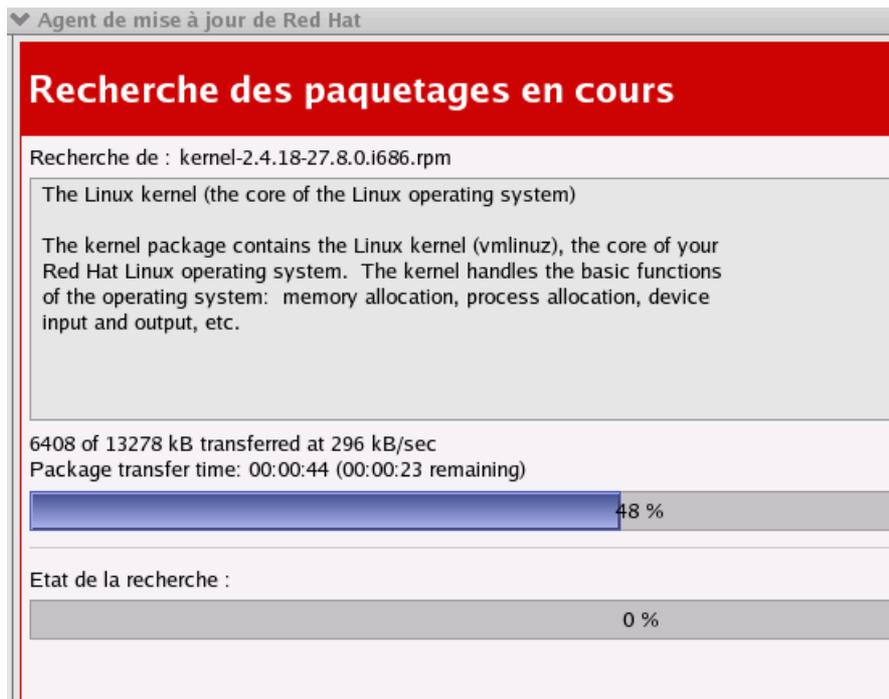
L'écran ci-dessus vous propose une mise à jour pour le noyau de votre système. En général ce sont les plus importantes car toutes les autres applications reposent sur ce noyau. Cochez l'option [**Tous les paquetages**] et faites [**Suivant**].



L'écran ci-dessus, vous donne la liste de toutes les mises à jour que vous pouvez faire. Cochez l'option [**Tous les paquetages**] et faites [**Suivant**].

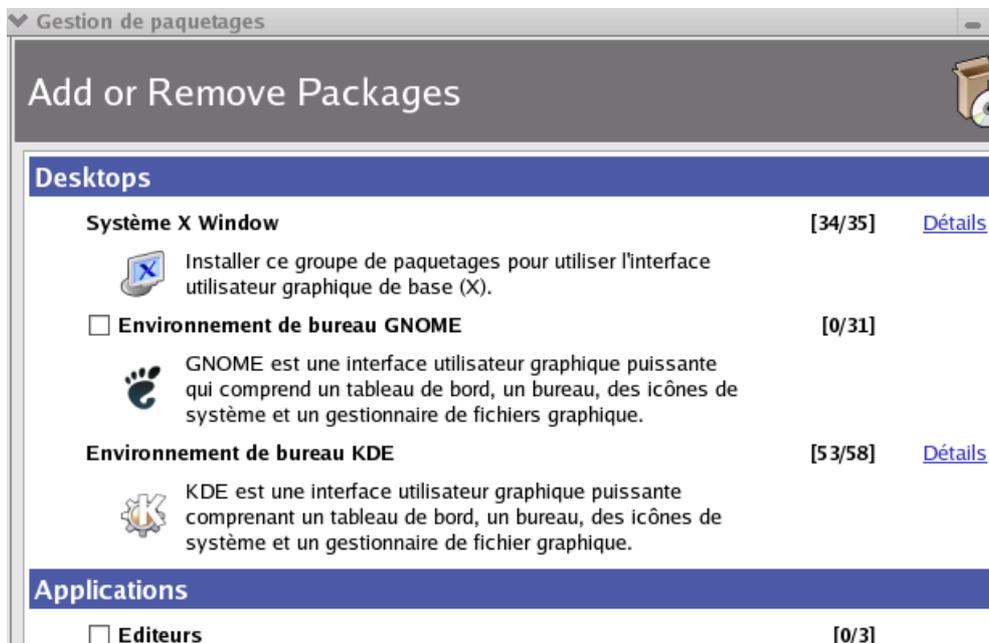


La boîte de dialogue ci-dessus vérifie les dépendances des paquetages à télécharger. Certains paquetages dépendent d'autres paquetages. L'utilitaire vérifie que les paquetages nécessaires à l'installation des mises à jour sont bien présents. Si ce n'était pas le cas, il les ajouterait à la liste des paquetages à télécharger et à installer. Ceci fait, le téléchargement des mises à jour commence. Lorsqu'il sera terminé, les paquetages téléchargés seront installés.



21.2 Ajout/retrait de paquetages

Pour ajouter des paquetages non inclus lors de l'installation ou supprimer des paquetages inutiles, on prend l'option **Menu KDE/Paramètres de système/Paquetages** :



On obtient ici, la liste des paquetages telle qu'elle nous a été présentée lors de l'installation et telle que nous l'avons validée. Les paquetages sélectionnés dans la fenêtre sont ceux qui sont présents sur notre système. Pour ajouter un paquetage, on le sélectionne. Pour en enlever un, on le désélectionne. Ci-dessus, ajoutons tous les éléments du bureau GNOME par exemple :



GNOME est une interface utilisateur graphique puissante qui comprend un tableau de bord, un bureau, des icônes de système et un gestionnaire de fichiers graphique.

Validons notre choix avec le bouton [Appliquer]. L'utilitaire nous donne des informations sur les paquetages qui vont être installés :



Nous utilisons le bouton [Continuer] pour lancer l'installation des paquetages. Les CD d'installation vont alors être réclamés et les paquetages installés. Une fois ceux-ci installés, vous pouvez vérifier s'il en existe des versions plus récentes avec **Menu/Outils de système/RedHat Network**.

21.3 Recherche et installation d'un paquetage sur internet

Parfois, on a besoin d'une application ne faisant pas partie de la distribution RedHat. On ne la trouve pas alors sur les CD. Elle n'est pas non plus proposée dans les mises à jour sur le réseau. Il faut alors la chercher ailleurs et l'installer. Il existe des sites spécialisés dans la recherche de paquetages pour différentes distributions de linux. L'un d'eux est à l'URL fr2.rpmfind.net. Nous utilisons Konqueror pour l'atteindre :



Ci-dessus, nous cherchons un paquetage appelé amavis. C'est un antivirus pour le courrier. On utilise le bouton [Search]. On obtient la réponse suivante :

RPM resource amavis

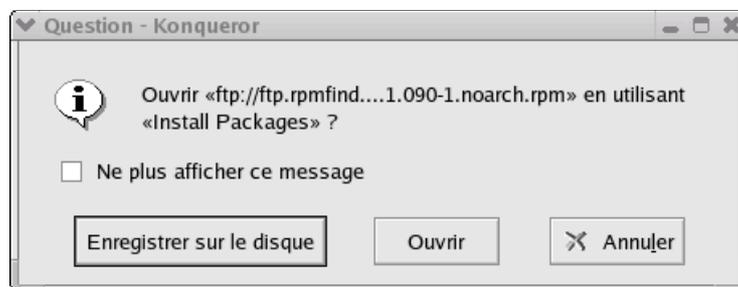
Found 2 sites for amavis

- <http://www.amavis.org/>
- <http://satan.oih.rwth-aachen.de/AMaViS/>

Found 9 RPM for amavis

Package	Summary	Distribution	Download
amavis-11-3.ppc.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.ppc.rpm
amavis-11-3.sparc.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.sparc.rpm
amavis-11-3.alpha.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.alpha.rpm
amavis-11-3.i686.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.i686.rpm
amavis-11-3.i586.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.i586.rpm
amavis-11-3.i386.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.i386.rpm
amavis-11-3.src.html	A Mail Virus Scanner Polish(ed)	Linux Distribution	amavis-11-3.src.rpm

Nous voyons ci-dessus plusieurs paquetages. Ceux qui ont le suffixe **src** dans leur nom sont des paquetages contenant du code source à compiler. Les autres sont des binaires. Nous décidons ici de télécharger le binaire [amavis-11-3.i386.rpm](#). Nous cliquons sur son lien. Konqueror détecte alors qu'on télécharge un paquetage et nous demande comment on veut le gérer :



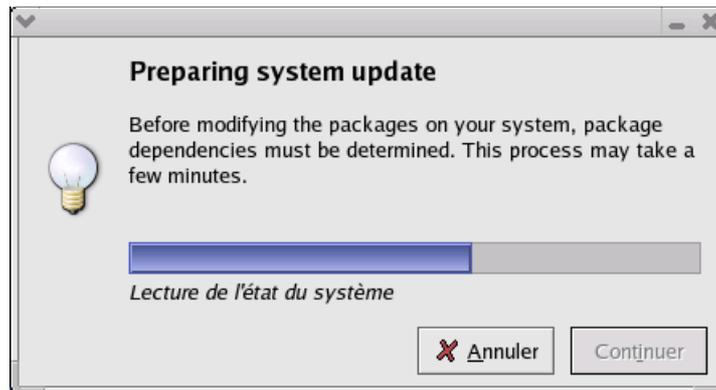
On peut soit

- installer immédiatement le paquetage via l'outil de gestion des paquetages
- soit l'enregistrer sur le disque

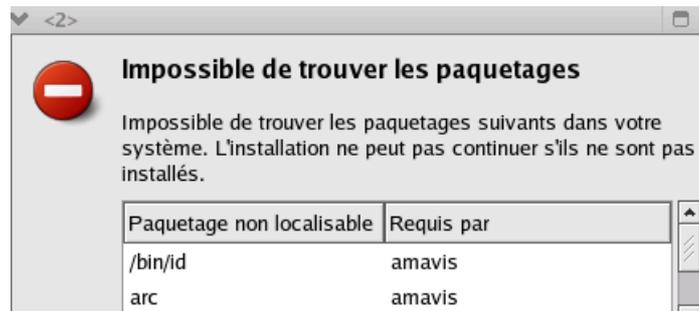
Nous prenons la deuxième solution et choisissons de le télécharger dans `/root/downloads`. Le téléchargement du paquetage commence alors. Lorsqu'il est fini, le fichier téléchargé est dans le dossier `/root/downloads` :



Maintenant double-cliquons sur le paquetage amavis ci-dessus. Cela lance son installation :



Le système vérifie les dépendances du paquetage. Un paquetage dépend parfois ici d'autres paquetages qui doivent être présents pour qu'il puisse être installé. C'est le cas ici :

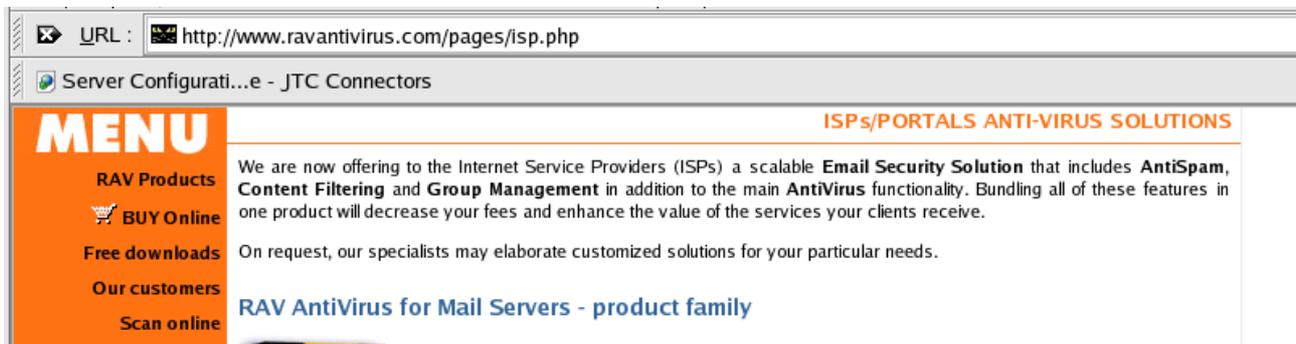


Il nous faut maintenant refaire ce que nous venons de faire pour tous les paquetages indiqués ci-dessus. C'est une limitation des paquetages de RedHat. Pour un paquetage appartenant à la distribution, le système de mise à jour résoud les dépendances des paquetages à installer et va ensuite les chercher sur un site RedHat. Pour un paquetage n'appartenant pas à la distribution, cela n'est pas fait et doit alors être fait manuellement.

Prenons un autre exemple. Nous nous intéressons toujours aux antivirus pour RedHat. Ouvrons Konqueror pour faire une requête Google :



Suivons le lien www.ravantivirus.com ci-dessus :



Ci-dessus, suivons le lien [Free downloads] :



Suivons le lien [RAV Antivirus Desktop for Linux] :

Please take a moment to register for your **free 30-day fully functional evaluat software**. GeCAD Software protects the information that is gathered on this site Is not to be sold, rented, leased or traded.

RAV AntiVirus Desktop for Linux [v8.1.4]

Filesize: 2244KB

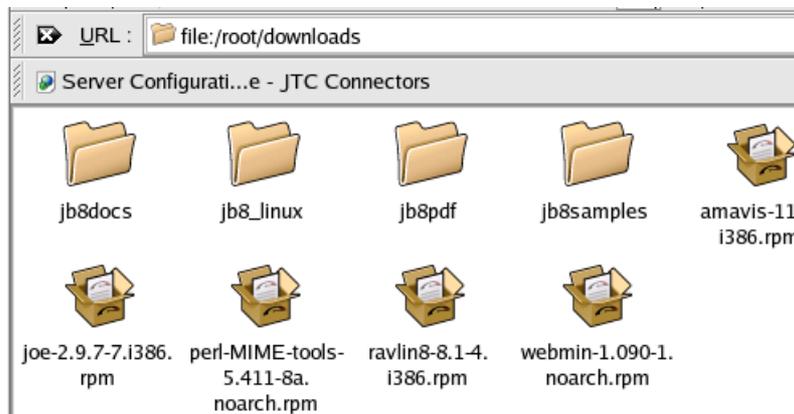
(* represents required fields)

Name:* Serge Tahé
E-mail:* serge.tah@istia.univ-angers.f
City:* Angers
Country:* France
State: [Non US state] (required for USA)
Address:
Zip code:
Profession:
Company:
Position:

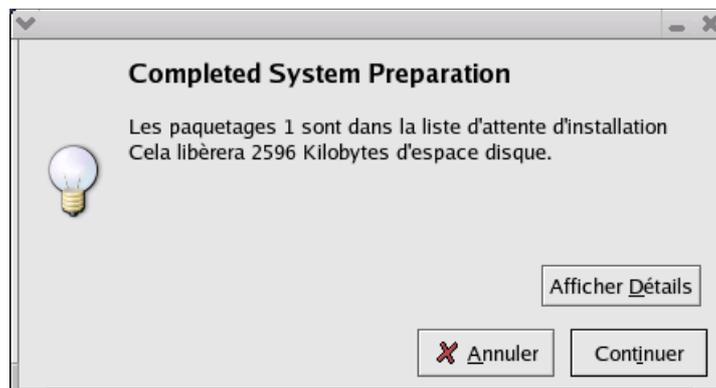
- Save this information for automatic fill in
- Yes, subscribe me to RAV newsletter.
- Yes, subscribe me to rav-desktop-unices discussion list.

Start Download

On remplit le formulaire ci-dessus et on commence le téléchargement avec [Start Download]. On récupère alors un fichier rpm :



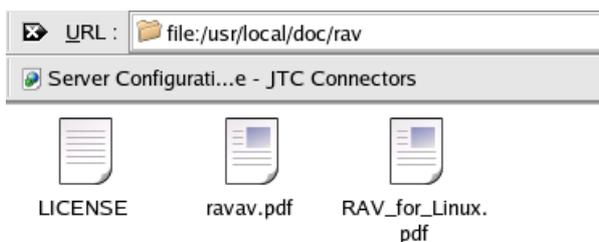
C'est le fichier ravlin8-8.1-4.i386.rpm ci-dessus. Double-cliquons dessus. L'installation du paquetage est opérée :



Ceci fait, on ne sait pas comment utiliser l'antivirus... Il est possible de connaître le contenu d'un paquetage avec la commande **rpm** :

```
[root@tahe root]# rpm -ql ravlin8-8.1-4 | more
/usr/local/doc/rav
/usr/local/doc/rav/LICENSE
/usr/local/doc/rav/RAV_for_Linux.pdf
/usr/local/doc/rav/ravav.pdf
/usr/local/man/man8
....
```

On découvre ainsi que le paquetage contenait un fichier PDF qui a été rangé dans `/usr/local/doc/rav` :



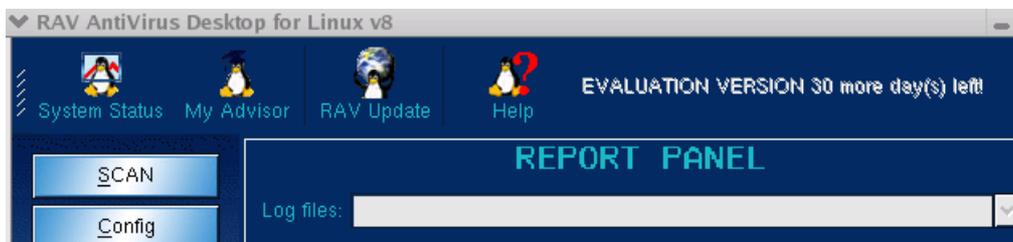
Nous double-cliquons sur le fichier `RAV_for_Linux.pdf` ci-dessus :



Nous lisons ce document et découvrons que pour lancer l'antivirus il faut taper la commande suivante :

```
[root@tahe root]# ravlin8 &
```

Ceci lance l'application qui affiche alors une interface graphique :



Nous laissons au lecteur le soin de découvrir l'application...

Table des matières

1 NOTIONS DE BASE DES RÉSEAUX TCP-IP.....	3
1.1 UN RÉSEAU ...DE RESSOURCES.....	3
1.2 GÉNÉRALITÉS.....	3
1.3 UN PEU DE FORMALISME : LES RÉSEAUX, VUS SELON L'ARCHITECTURE OSI.....	4
1.4 ZOOM SUR LA COUCHE DE NIVEAU 3 : LE PROTOCOLE IP.....	5
1.5 ILLUSTRATION.....	6
1.6 RÉSEAUX PUBLICS, RÉSEAUX PRIVÉS.....	7
2 LE TRAVAIL À RÉALISER.....	10
2.1 INTRODUCTION.....	10
2.2 LES DIFFÉRENTES VERSIONS DE LINUX.....	10
2.3 INSTALLATION DE LINUX SUR UN PC.....	11
2.4 INSTALLATION DES SERVICES.....	11
2.5 INSTALLATION D'APPLICATIONS COMPLÉMENTAIRES.....	11
3 INSTALLATION DE LINUX REDHAT 8.0.....	11
3.1 CONFIGURATION DU BIOS.....	11
3.2 RÉCUPÉRER L'ADRESSE IP DU POSTE WINDOWS.....	12
3.3 INSTALLATION PAR FTP OU PAR CD.....	13
3.4 INSTALLATION AVEC CD.....	19
3.5 FIN DE L'INSTALLATION.....	23
4 DÉCOUVRIR L'ENVIRONNEMENT DE TRAVAIL LINUX.....	25
4.1 L'ENVIRONNEMENT DE TRAVAIL KDE.....	25
4.1.1 Le terminal KONSOLE.....	25
4.1.1.1 Commandes de base d'Unix.....	25
4.1.1.2 Aide sur les commandes Unix.....	29
4.1.2 L'ÉDITEUR DE TEXTE KWRITE.....	30
4.1.2.1 Tests.....	31
4.1.3 Le gestionnaire de fichiers KONQUEROR.....	31
4.1.4 Le navigateur MOZILLA.....	35
4.1.4.1 Configuration du proxy HTTP.....	35
4.2 SYSTÈMES DE FICHIERS, PÉRIPHÉRIQUES ET ARBORESCENCE.....	35
4.2.1 Gestion des périphériques.....	35
4.2.2 Gestion des systèmes de fichiers.....	35
4.2.3 Organisation des données sur le système de fichiers.....	38
4.3 GESTION DES COMPTES UTILISATEURS.....	39
4.3.1 Création de groupes d'utilisateurs.....	40
4.3.2 Création d'utilisateurs.....	40
4.3.3 Les fichiers /ETC/PASSWD, /ETC/SHADOW, /ETC/GROUP.....	44
4.3.4 Gestion des utilisateurs en mode texte.....	45
4.3.5 Droits d'accès attachés aux utilisateurs.....	45
5 INTRODUCTION À L'INSTALLATION DE SERVICES.....	50
5.1 MODE OPÉRATEUR.....	50
5.2 APPLICATIONS CLIENT-SERVEUR DANS LE MONDE TCP-IP.....	51
5.3 LE RÉSEAU PRIVÉ ÉTUDIANT.....	52
5.4 IDENTIFIER UNE MACHINE DE L'INTERNET.....	52
5.5 LES FICHIERS DE CONFIGURATION RÉSEAU DU POSTE LINUX.....	53
5.6 LES SERVICES À INSTALLER.....	54
5.7 LES TESTS DES SERVICES INSTALLÉS.....	54
5.8 LES FICHIERS DE LOGS.....	55
5.9 LES DÉMONS.....	55
5.9.1 Quelques démons.....	55
5.9.2 Le super démon réseau XINETD.....	56
5.9.3 Configuration du super démon XINETD.....	56

5.9.4 CONCLUSION.....	58
6 LES SERVICES DE CONNEXION À DISTANCE TELNET ET SSH.....	59
6.1 FONCTION.....	59
6.2 INSTALLATION DU SERVICE TELNET.....	59
6.3 TEST 1 DU SERVICE TELNET.....	59
6.4 TEST 2 DU SERVICE TELNET.....	60
6.5 INSTALLATION DU SERVICE SSH.....	60
6.6 TEST DU SERVICE SSH.....	61
7 LES SERVICES DE COURRIER ÉLECTRONIQUE.....	62
7.1 FONCTION.....	62
7.2 INSTALLATION ET CONFIGURATION.....	62
7.3 TEST 1.....	62
7.4 TEST 2.....	63
7.5 TEST 3.....	66
7.6 UTILISATION D'ALIAS.....	66
7.7 GÉRER LE RELAYAGE DE COURRIER.....	68
7.8 CRÉER UN RÉPONDEUR.....	72
8 LE TRANSFERT DE FICHIERS.....	76
8.1 FONCTION.....	76
8.2 INSTALLATION ET CONFIGURATION.....	76
8.3 TEST 1.....	76
8.4 TEST 2.....	77
8.5 TEST 3.....	77
8.6 TEST 4.....	78
8.7 TEST 5.....	78
8.8 COMMENTAIRES.....	78
9 LE SERVICE WORLD WIDE WEB (WWW).....	79
9.1 FONCTION.....	79
9.2 INSTALLATION.....	79
9.3 LE FICHIER DE CONFIGURATION HTTPD.CONF.....	79
9.4 DÉMARRAGE/ARRÊT DU SERVICE HTTP.....	80
9.5 TEST 1.....	81
9.6 TEST 2, NAVIGATEUR SOUS LINUX.....	81
9.7 TEST 3 - MODIFIER LA PAGE D'ENTRÉE DU SITE.....	82
9.8 SITES WEB PERSONNELS.....	82
10 LE SERVICE SAMBA.....	85
10.1 FONCTION.....	85
10.2 INSTALLATION ET CONFIGURATION.....	85
10.3 COMMENTAIRES.....	86
10.4 PRÉPARATIFS.....	87
10.5 DÉMARRAGE DU SERVICE.....	88
10.6 UTILISATION DU CLIENT SMBCLIENT.....	88
10.6.1 CONNEXION À UN SERVICE D'ACCÈS PUBLIC.....	88
10.6.2 CONNEXION À UN SERVICE D'ACCÈS PRIVÉ.....	90
10.6.3 CONNEXION AU RÉPERTOIRE PERSONNEL.....	91
10.7 UTILISATION D'UN CLIENT WINDOWS.....	92
10.7.1 VOISINAGE RÉSEAU.....	92
10.7.2 ACCÈS À LA RESSOURCE [PUBLIC] EN LECTURE SEULE.....	93
10.7.3 ACCÈS AUX RÉPERTOIRES PERSONNELS SUR LINUX, RESSOURCE [HOMES].....	93
10.7.4 ACCÈS À LA RESSOURCE [PRIVATE].....	94
10.7.5 ACCÈS À LA RESSOURCE [PUBLIC] EN LECTURE ET ÉCRITURE.....	95
10.7.6 DEUX UTILISATEURS WIN9X PARTICULIERS.....	95
10.8 CONCLUSION.....	95
11 LE SERVICE DES QUOTAS.....	97
11.1 FONCTION.....	97
11.2 INSTALLATION.....	97

<u>11.3</u>	<u>LES FICHIERS DE CONFIGURATION.....</u>	<u>97</u>
<u>11.4</u>	<u>COLLECTE DES ESPACES DISQUES OCCUPÉS.....</u>	<u>97</u>
<u>11.5</u>	<u>COMMENTAIRES.....</u>	<u>99</u>
<u>12</u>	<u>FILTRE L'ACCÈS AUX SERVICES LANCÉS PAR XINETD.....</u>	<u>100</u>
<u>12.1</u>	<u>FONCTION.....</u>	<u>100</u>
<u>12.2</u>	<u>LES FICHIERS DE CONFIGURATION.....</u>	<u>101</u>
<u>12.3</u>	<u>DÉMARRAGE DU SERVICE.....</u>	<u>102</u>
<u>12.4</u>	<u>TEST 1.....</u>	<u>102</u>
<u>12.5</u>	<u>TEST 3.....</u>	<u>103</u>
<u>12.6</u>	<u>COMMENTAIRES.....</u>	<u>103</u>
<u>13</u>	<u>LES SYSTÈMES DE FICHIERS.....</u>	<u>106</u>
<u>13.1</u>	<u>LES SUPPORTS PHYSIQUES DES FICHIERS.....</u>	<u>106</u>
<u>13.2</u>	<u>MONTER DES SYSTÈMES DE FICHIERS.....</u>	<u>107</u>
<u>13.2.1</u>	<u>MONTAGES FAITS AU DÉMARRAGE DU SYSTÈME.....</u>	<u>107</u>
<u>13.2.2</u>	<u>MONTAGES MANUELS.....</u>	<u>109</u>
<u>13.2.3</u>	<u>CONNAÎTRE LE PARTITIONNEMENT D'UN DISQUE : FDISK.....</u>	<u>109</u>
<u>13.2.4</u>	<u>MONTAGE D'UNE PARTITION DOS.....</u>	<u>109</u>
<u>13.2.5</u>	<u>MONTAGE DU CDROM.....</u>	<u>110</u>
<u>13.2.6</u>	<u>MONTAGE D'UNE DISQUETTE LINUX.....</u>	<u>111</u>
<u>13.2.7</u>	<u>MONTAGE D'UN LECTEUR ZIP.....</u>	<u>111</u>
<u>13.2.8</u>	<u>LES OUTILS MTOOLS.....</u>	<u>112</u>
<u>13.3</u>	<u>MONTAGE DE FICHIERS EXPORTÉS PAR LE SERVICE SAMBA.....</u>	<u>114</u>
<u>14</u>	<u>LA RECHERCHE RAPIDE DE FICHIERS.....</u>	<u>118</u>
<u>14.1</u>	<u>INTRODUCTION.....</u>	<u>118</u>
<u>14.2</u>	<u>LA COMMANDE LOCATE.....</u>	<u>118</u>
<u>14.3</u>	<u>CONCLUSION.....</u>	<u>118</u>
<u>15</u>	<u>SOUMETTRE DES TRAVAUX À DES HEURES PRÉCISES.....</u>	<u>119</u>
<u>15.1</u>	<u>INTRODUCTION.....</u>	<u>119</u>
<u>15.2</u>	<u>LA COMMANDE CRONTAB.....</u>	<u>119</u>
<u>15.3</u>	<u>LES RÉPERTOIRES HOURLY, DAYLY, MONTHLY.....</u>	<u>122</u>
<u>15.4</u>	<u>AFFICHAGES ÉCRAN DES SCRIPTS SOUMIS À CROND.....</u>	<u>123</u>
<u>16</u>	<u>LE SGBDR MYSQL.....</u>	<u>125</u>
<u>16.1</u>	<u>INTRODUCTION.....</u>	<u>125</u>
<u>16.2</u>	<u>INSTALLATION DE MySQL.....</u>	<u>125</u>
<u>16.3</u>	<u>LANCLEMENT DU SERVEUR MYSQLD.....</u>	<u>126</u>
<u>16.4</u>	<u>LA DOCUMENTATION DE MySQL.....</u>	<u>126</u>
<u>16.5</u>	<u>UTILISATION DU CLIENT MySQL.....</u>	<u>127</u>
<u>16.6</u>	<u>POURSUIVRE.....</u>	<u>134</u>
<u>17</u>	<u>LE SGBDR POSTGRES.....</u>	<u>134</u>
<u>17.1</u>	<u>INTRODUCTION.....</u>	<u>134</u>
<u>17.2</u>	<u>INSTALLATION DE POSTGRES.....</u>	<u>134</u>
<u>17.3</u>	<u>LA DOCUMENTATION DE POSTGRES.....</u>	<u>135</u>
<u>17.4</u>	<u>LANCLEMENT DU SERVEUR POSTGRES.....</u>	<u>136</u>
<u>17.5</u>	<u>L'UTILISATEUR UNIX POSTGRES.....</u>	<u>136</u>
<u>17.6</u>	<u>PREMIÈRE UTILISATION DU CLIENT PSQL.....</u>	<u>136</u>
<u>17.7</u>	<u>CRÉATION DES PREMIÈRES BASES DE DONNÉES (INSTALLATION FAITE SANS RPM).....</u>	<u>137</u>
<u>17.8</u>	<u>LE CLIENT PSQL.....</u>	<u>138</u>
<u>17.9</u>	<u>CRÉATION DE BASES DE DONNÉES.....</u>	<u>141</u>
<u>17.10</u>	<u>CRÉATION D'UTILISATEURS.....</u>	<u>141</u>
<u>17.11</u>	<u>AUTHENTIFICATION DES UTILISATEURS.....</u>	<u>142</u>
<u>17.12</u>	<u>COMMANDES SQL.....</u>	<u>145</u>
<u>18</u>	<u>OUTILS DE DÉVELOPPEMENT.....</u>	<u>148</u>
<u>18.1</u>	<u>PERL.....</u>	<u>148</u>
<u>18.2</u>	<u>PYTHON.....</u>	<u>148</u>
<u>18.3</u>	<u>C.....</u>	<u>148</u>

<u>18.4</u> C++.....	149
<u>18.5</u> JAVA.....	149
<u>18.6</u> JBUILDER.....	151
<u>19</u> DÉVELOPPEMENT WEB.....	159
<u>19.1</u> LES COMPOSANTES D'UNE APPLICATION WEB.....	159
<u>19.2</u> LES ÉCHANGES DANS UNE APPLICATION WEB COMPORTANT UN FORMULAIRE.....	160
<u>19.3</u> QUELQUES RESSOURCES.....	160
<u>19.4</u> APACHE-PERL.....	161
<u>19.5</u> APACHE-PHP.....	165
<u>19.6</u> APACHE - JAVA.....	168
<u>19.6.1</u> INSTALLATION DU SERVEUR WEB TOMCAT.....	168
<u>19.6.2</u> UN SCRIPT JSP.....	170
<u>20</u> CONFIGURATION D'UNE IMPRIMANTE HP JET DIRECT.....	173
<u>20.1</u> CONFIGURER L'IMPRIMANTE AVEC MENU KDE.....	173
<u>20.2</u> FAIRE UN TEST AVEC OPENOFFICE.....	175
<u>21</u> GESTION DES PAQUETAGES.....	176
<u>21.1</u> MISE À JOUR DES PAQUETAGES INSTALLÉS.....	176
<u>21.2</u> AJOUT/RETRAIT DE PAQUETAGES.....	181
<u>21.3</u> RECHERCHE ET INSTALLATION D'UN PAQUETAGE SUR INTERNET.....	182