



Info0911 - Administration de Systèmes d'Information

Luiz Angelo Steffene



Déroulement des Cours

- ▶ 12h de Cours Linux
- ▶ 12h de TPs Linux
- ▶ 16h de Cours/TP Windows Server (plus tard)
- ▶ Évaluation
 - ▶ 10% examen CISCO Network Fundamentals (seule la première note compte)
 - ▶ 60% examen pratique partie Linux
 - ▶ 30% examen partie Windows





Quelques sujets traités en cours

- ▶ Introduction/Révision des bases de l'administration Linux
- ▶ Programmation Shell
- ▶ Installation/administration de services réseau
- ▶ Mise en sécurité d'un système



Références

- ▶ Livres sur Unix à la bibliothèque de l'université
- ▶ The Linux Documentation Project (LDP), <http://www.tldp.org/>
Mirroir: <http://www.iitk.ac.in/LDP>
- ▶ Cours de Préparation LPI de IBM
http://www.ibm.com/developerworks/linux/lpi/101.html?S_TACT=105AGX03&S_CMP=LPLINUX
- ▶ Les transparents seront disponibles dans le Bureau Virtuel





INTRODUCTION AU MONDE UNIX



Histoire de l'UNIX

- ▶ La première version a été créée par le Bell Labs en 1969.
- ▶ Quelques ingénieurs de ce projet, dont Ken Thompson, **Dennis Ritchie**, Rudd Canaday, and Doug McIlroy ont implémenté la première version du système de fichiers dans une machine PDP-7, avec quelques outils. Le nom UNIX a été donné par **Brian Kernighan**.
- ▶ Le 1er janvier 1970 à 00:00:00 heures est la date « zéro » pour UNIX.
- ▶ En 1973 Unix est ré-écrit presque entièrement en C, un nouveau langage développé par Dennis Ritchie.
- ▶ L'utilisation d'un langage de haut niveau a rendu plus facile l'installation du système sur d'autres machines.



Histoire de l'UNIX

- ▶ En 1977 il y avait autour de 500 machines UNIX en tout le monde.
- ▶ En 1980 est sorti BSD 4.1 (Berkeley Software Development)
- ▶ En 1983, SunOS, BSD 4.2, System V
- ▶ En 1988 AT&T et Sun Microsystems s'allient pour développer System V Release 4 (SVR4). Celui-ci donnera naissance à UnixWare et Solaris 2.
- ▶ En 1991 (seulement) Linux est créée.





UNIX

- ▶ Unix est un système d'exploitation multi-utilisateur, multi-tâches
- ▶ Plusieurs utilisateurs peuvent être connectés simultanément, exécutant différents programmes
- ▶ C'est l'un des systèmes « par excellence » en ce qui concerne la mise en place de serveurs



C'est quoi LINUX

- ▶ Linux est un système d'exploitation ouvert de type Unix originalement créé par Linus Torvalds avec l'aide d'autres développeurs.
- ▶ Il a commencé en 1991 comme un projet d'études de Linus Torvalds, un étudiant d'informatique finlandais
- ▶ Le noyau (Kernel) version 1.0 a été lancé en 1994 (la dernière version stable est la 2.6.35-7 (le 2/10/2010))
- ▶ Développé sous la licence publique GPL (GNU General Public Licence), le code source Linux est ouvert et disponible à tous



GNU et Linux



- 1983 :
 - Richard M. Stallman est frustré par la privation de liberté accompagnant les systèmes propriétaires.
Il crée le projet GNU, dont l'objectif est de créer un système d'exploitation compatible avec Unix et composé uniquement de logiciels libres.
De nombreux composants de base sont développés : compilateur GCC, GNU Libc, GNU coreutils, éditeur Emacs... Mais il manque le noyau (le développement de GNU Hurd a pris du retard).
- 1991 :
 - Linus Torvalds démarre le développement du noyau Linux et le diffuse en tant que logiciel libre. Linux est moins ambitieux techniquement que GNU Hurd, et son développement rapide le rend rapidement utilisable.
- Avec les outils GNU et le noyau Linux, on a enfin un système entièrement libre : GNU/Linux



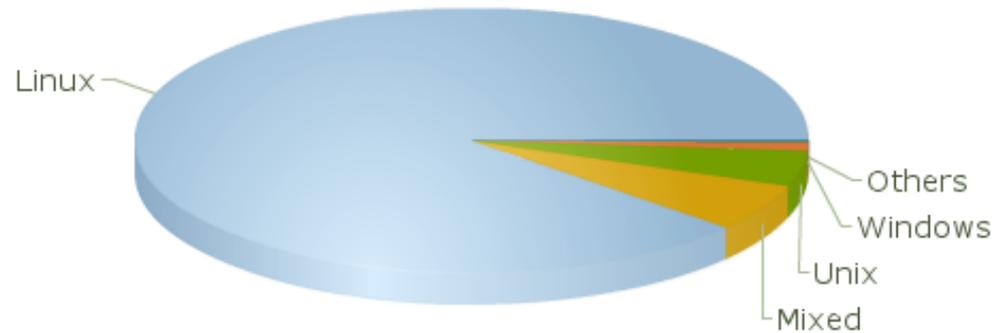
Les distributions Linux

- Rôle :
 - Rassembler les différents logiciels libres
 - Intégration : corriger les incompatibilités
 - Faciliter l'installation du système et des logiciels
 - Fournir un niveau de support intermédiaire
 - Vraie plus-value du monde du Libre par rapport au monde propriétaire
- Des centaines de distributions :
 - Généralistes
 - Adaptées à un type d'utilisation (serveur, netbook, embarqué, liveCD)
 - Adaptées à un domaine d'utilisation (Informatique scientifique, édition multimédia, magnétoscope numérique, sécurité)



Domaines d'Application

- Serveurs réseau, Calcul HPC



- Embarqué
 - Android, IOS, GPS, routeurs, webcams



La Famille Debian



- Debian
 - Une des plus anciennes distributions GNU/Linux (1993)
 - Développée par des bénévoles (environ 1000 développeurs)
 - Très respectée pour sa qualité, sa stabilité, son indépendance, l'expertise de ses développeurs
 - Défend les valeurs du Logiciel Libre (contrat social Debian)
 - Nouvelles versions tous les 2 ans (environ)
 - Utilisent le gestionnaire de paquet APT, et le format de paquet .deb



La Famille Debian



ubuntu

- Ubuntu
 - Fondée en 2004 par Mark Shuttleworth
 - Développement se voulant ouvert, dirigé par Canonical
 - Processus de développement basé sur Debian (sync. Tous les 6mois)
 - Visant principalement le poste de travail
 - au début généraliste, maintenant (serveurs, cloud, netbooks)
 - A dépassé Debian en nombre d'utilisateurs
 - Nouvelles versions tous les 6 mois, versions Long Term Support tous les 2 ans
 - Utilisent le gestionnaire de paquet APT, et le format de paquet .deb



La Famille RedHat



A l'origine : Red Hat Linux (1994)

- Distribution commerciale, utilisant le gestionnaire de paquet RPM
- 2003 : séparation en deux versions : Red Hat Enterprise Linux et Fedora

Red Hat Enterprise Linux (RHEL) :

- Distribution commerciale (payante) pour serveurs et postes de travail
- Sorties tous les 2 ans, supporté es pendant 7 ans

Fedora :

- Distribution communautaire, développement dirigé par Red Hat
- Sorties tous les 6 mois
- Sert de *Technology Preview* pour RHEL

CentOS :

- RHEL est entièrement libre mais payante → code source disponible
- Distribution gratuite, clone de RHEL a` partir de ses sources

Autre clone de RHEL : Scientific Linux



Autres distributions populaires



Mandriva :

- Distribution commerciale française (anciennement Mandrake Linux)
- Originellement basée sur Red Hat, utilise RPM
- Avenir incertain ; *fork* communautaire : Mageia

Arch Linux :

- Distribution communautaire visant les utilisateurs avancés
- *Rolling release* (nouveaux paquets apparaissant chaque jour)

Gentoo :

- Distribution communautaire
- Compilation des logiciels depuis les sources ; permet des optimisations



SUSE Linux Enterprise :

- Distribution commerciale de Novell. Utilise RPM
- Distribution communautaire associée : openSUSE

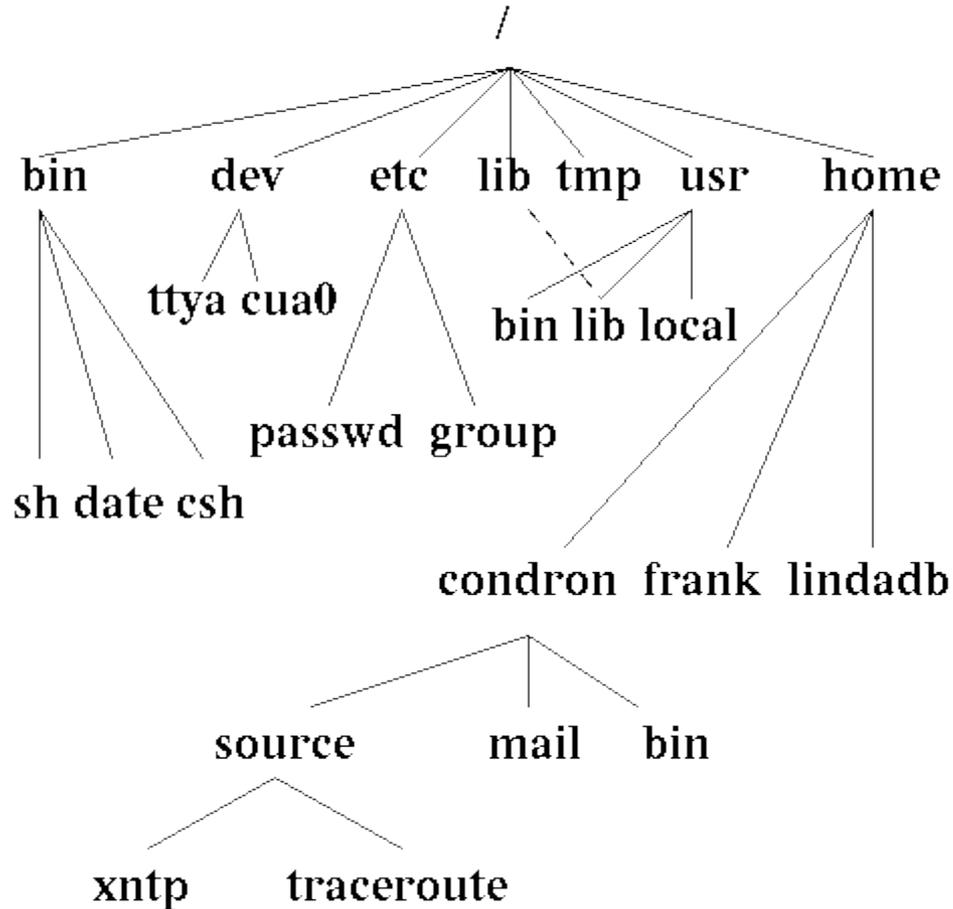




COMPRENDRE LA STRUCTURE UNIX



Le Système de Fichiers UNIX



Le Système de Fichiers

- ▶ Le système de fichiers UNIX ressemble un arbre inversé
 - ▶ Le répertoire « racine » est indiqué par /
- ▶ Un fichier ou répertoire peut être indiqué par son chemin absolu ou par un chemin relatif (selon une position de référence)
- ▶ Le chemin absolu commence par la racine, /, suivie des différentes branches (séparées aussi par des /) :
 - ▶ /home/angelo/source/xntp
- ▶ Un chemin relatif indique un chemin par rapport à une localisation dans l'arbre, normalement le répertoire courant. Deux répertoires spéciaux peuvent être utilisées pour aider :
 - ▶ . le répertoire courant
 - ▶ .. le répertoire père (supérieur)
- ▶ Alors, si vous êtes sur /home/frank et souhaitez indiquer le chemin relatif vers /home/angelo/source/xntp :
 - ▶ ../angelo/source/xntp



Structure Standard des Répertoires UNIX/Linux

```
/
|-- bin      exécutable de base
|-- dev      périphériques
|-- etc      fichiers de configuration
|-- home     répertoires des utilisateurs
|-- lib      bibliothèques partagées pour les exécutable de base
|-- mnt      point de montage des périphériques externes
|-- opt      logiciels optionnels
|-- proc     répertoire virtuel (informations du noyau)
|-- root     répertoire de l'utilisateur root
|-- sbin     exécutable de base pour l'administrateur
|-- sys      répertoire virtuel (informations du noyau)
|-- tmp      fichiers temporaires
|-- usr
|   |-- bin  exécutable pour les utilisateurs
|   |-- lib  bibliothèques partagées
|   |-- local logiciels installés à la main par l'admin
|   |-- sbin exécutable pour l'administrateur
'-- var
    '-- log  journaux
```

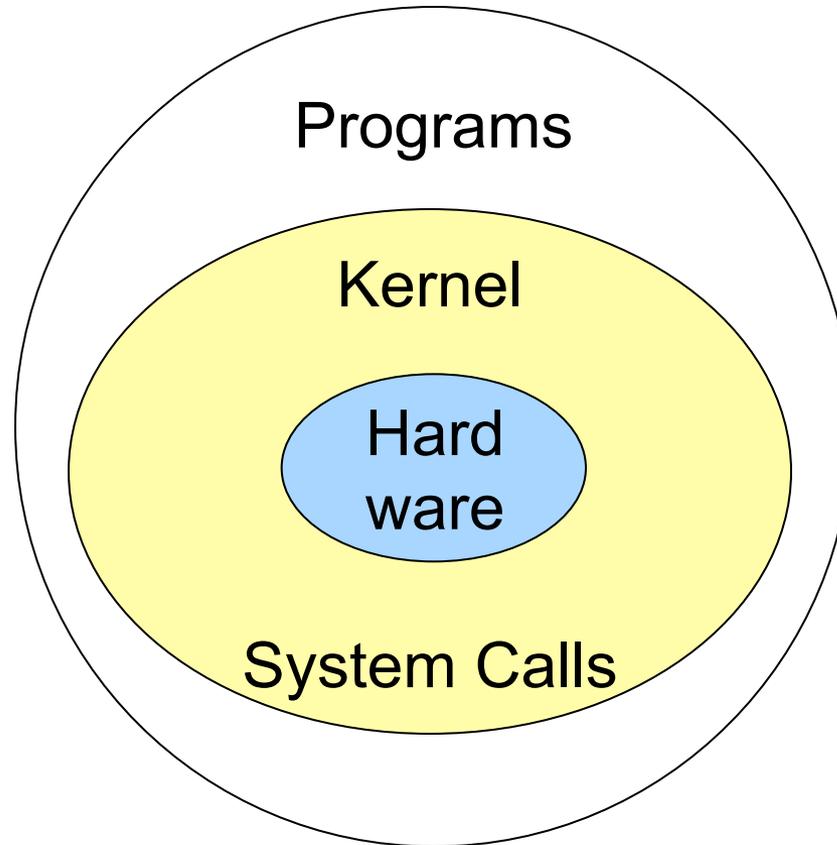


Répertoires, Fichiers et autres

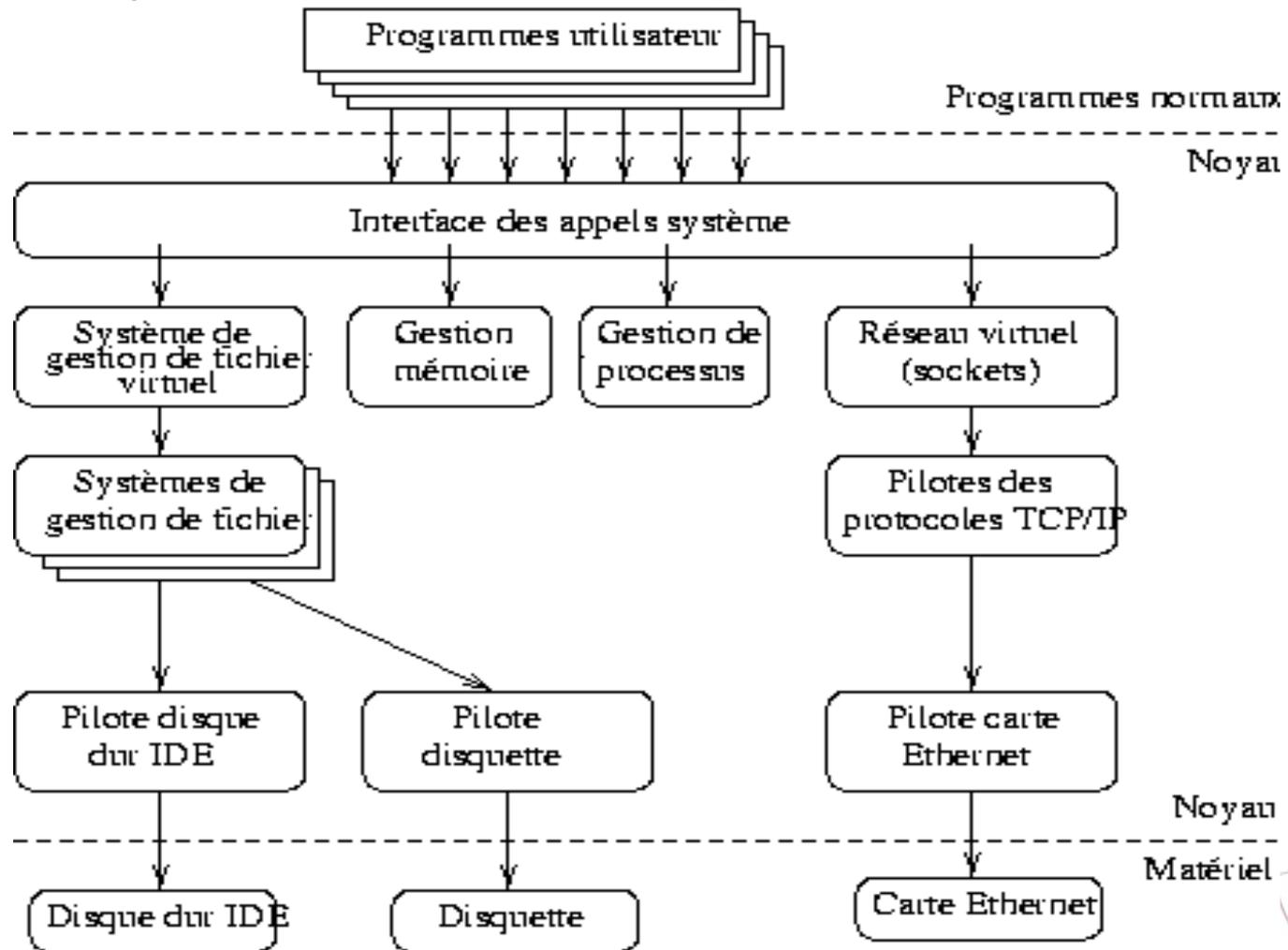
- **Fichiers classiques** : correspondant aux fichiers au sens où on l'entend généralement
- **Répertoires** : contiennent d'autres fichiers. Chaque répertoire contient au moins un répertoire parent (désigné par ..), et un répertoire courant (.), c'est-à-dire lui-même.
- **Liens symboliques** : pointeurs vers des fichiers réels (créés avec *ln -s*). Il existe un deuxième type de liens (liens durs, ou liens physiques, ou hardlinks) qui sont un nom alternatif pour le même ensemble de données.
- **Fichiers de périphériques** : situés dans le répertoire /dev, ils représentent les périphériques (disque dur, partition, etc)



La Structure de l'UNIX



Composants principaux d'un système d'exploitation





INSTALLATION D'UN LINUX



Le choix d'une distribution LINUX

- ▶ Après avoir choisi votre distribution, vous avez une multitude de méthodes pour l'installation
 - ▶ CD/DVD – téléchargé et gravé ou acheté ou reçu gratuitement
 - ▶ FTP – requiert un disque de démarrage
 - ▶ Serveur PXE (boot par le réseau)
 - ▶ ...
- ▶ Vous devez réserver une (plusieurs) partitions de disque pour votre installation
- ▶ Si vous voulez faire chez vous mais vous n'avez jamais essayé Linux, il y a aussi la possibilité d'installer Linux dans une machine virtuelle (genre VMWare ou VirtualBox)
 - ▶ Ceci évite de bousiller votre disque si vous n'êtes pas trop expérimenté



C'est quoi une Partition ?

- ▶ Partitionner un disque signifie le diviser en plusieurs unités logiques.
- ▶ Une partition est une division contigüe du disque qui est considérée comme un disque indépendant.
- ▶ Les disques durs contiennent une table de partition qui regroupe les informations relatives aux partitions existantes.



Pourquoi faire plusieurs partitions dans un disque ?

- ▶ Possibilité d'installer différents systèmes d'exploitation (ou versions) dans un même disque
- ▶ L'utilisation de plusieurs partitions réduit aussi le risque d'une défaillance du système si la partition devient pleine.
 - ▶ Un utilisateur ou processus hors de contrôle peut consommer tout le space, ce qui empêche le système de fonctionner correctement. En faisant une séparation du space, on garantit une marge pour les processus du système.
- ▶ Encapsulation des données. Une corruption du système de fichiers reste local à une partition, ce qui peut sauvegarder une partie de vos données en cas d'accident.
- ▶ Lors de l'installation de Linux, vous avez la possibilité de partitionner le disque
 - ▶ Soit avec un outil graphique, soit avec l'outil **fdisk**



L'utilitaire FDISK

- `root@tty0[knoppix]# fdisk -l /dev/hda`
- Disk `/dev/hda`: 120.0 GB, 120034123776 bytes
- 255 heads, 63 sectors/track, 14593 cylinders
- Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
<code>/dev/hda1</code>	*	1	2611	20972826	7	HPFS/NTFS
<code>/dev/hda2</code>		2612	2624	104422+	83	Linux
<code>/dev/hda3</code>		2625	14593	96140992+	5	Extended
<code>/dev/hda5</code>		2625	2689	522081	82	Linux swap
<code>/dev/hda6</code>		2690	5180	20008926	83	Linux
<code>/dev/hda7</code>		5181	6426	10008463+	83	Linux
<code>/dev/hda8</code>		6427	6676	2008093+	b	W95 FAT32



L'utilitaire GPARTED

The screenshot shows the GParted utility interface. At the top, there is a menu bar with options: Refresh Devices (Ctrl+R), Devices, Show Features, and Quitter (Ctrl+Q). A dropdown menu is open over the 'Devices' button, listing several disk devices with their sizes: /dev/fd0 (0.00 o), /dev/sda (74.53 Gio), /dev/sdb (149.05 Gio), /dev/sdc (152.66 Gio), and /dev/sdd (298.09 Gio). The /dev/sdd device is selected. To the right of the menu, a toolbar shows the selected device: /dev/sdd (298.09 Gio). Below the menu, a graphical representation of the disk layout is shown with colored bars representing partitions. A table below the graphical view lists the partitions with their details.

Partition	Système	Label	Taille	Utilisé	Inutilisé	Marqueurs
/dev/sdd1	ext3	/	21.46 Gio	4.92 Gio	16.55 Gio	
/dev/sdd2	ext3	/media/sdd2	68.60 Gio	41.04 Gio	27.56 Gio	
/dev/sdd3	linux-swap		502.03 Mio	---	---	
▼ /dev/sdd4	extended		207.54 Gio	---	---	
/dev/sdd5	ext3	/media/sdd5	19.92 Gio	4.79 Gio	15.13 Gio	boot
/dev/sdd6	ext3	/home	19.89 Gio	3.42 Gio	16.46 Gio	
/dev/sdd7	ext3	/media/sdd7	19.79 Gio	15.59 Gio	4.21 Gio	
/dev/sdd8	ext3	/media/sdd8	19.83 Gio	3.60 Gio	16.22 Gio	
/dev/sdd9	reiserfs	/media/sdd9 Set	128.10 Gio	96.86 Gio	31.24 Gio	



Types de Filesystem

- ▶ ext2, ext3 et ext4 — Filesystem par défaut sur Linux, ce système a l'avantage de faire la journalisation des modifications, ce qui permet de récupérer plus rapidement les données lors d'une défaillance
- ▶ swap — partition utilisée pour la pagination de mémoire
- ▶ vfat — Le système de fichiers des anciens windows (jusqu'à Windows98) et certaines clés usb actuelles
- ▶ NTFS – Système utilisé par les Windows plus récents, requiert parfois l'installation de modules pour permettre l'écriture
- ▶ ReiserFS – un filesystem en arbre-B qui est spécialement performant pour les systèmes avec un grand nombre de petits fichiers
- ▶ XFS – système de fichiers optimisé pour la performance. XFS utilise intensivement le cache en RAM, ce qui peut causer des pertes en cas de panne de courant



RAID logiciel et LVM

- ▶ Software RAID (Redundant Array of Independent Disk)
 - ▶ RAID 0 (Striping)
 - ▶ RAID 1 (Mirroring)
 - ▶ RAID 5 (Striping with Parity)
- ▶ LVM (Logical Volume Manager)
 - ▶ Permet de combiner plusieurs partitions/disques en une seule unité logique. Par exemple, cela permet d'augmenter la taille de la partition / home par l'adjonction d'autres partitions sur un deuxième disque au lieu de recréer les partitions avec de tailles plus grandes



Boot Loader

- ▶ Afin de charger un Système d'Exploitation, la BIOS lit des instructions dans le premier secteur du disque.
- ▶ Dans ce premier secteur réside le master boot record (MBR), et le boot loader normalement est installé dans ce secteur.
- ▶ Selon le boot loader, des fichiers supplémentaires sont chargés à partir d'une partition dans le disque dur.
- ▶ Le boot loader peut vous donner la possibilité de choisir entre les systèmes d'exploitation installés dans le disque.
- ▶ Une fois que le boot loader démarre le système, celui-là reste inutilisé jusqu'au prochain boot.
- ▶ Exemples de boot loader
 - ▶ LILO – le plus ancien, capacités limitées (menu texte seulement, etc)
 - ▶ **GRUB** – boot loader utilisé actuellement



Installation et Suppression de Logiciels

- ▶ Télécharger un exécutable binaire
- ▶ Télécharger le code source et le compiler
- ▶ Utiliser des gestionnaires de paquets
 - ▶ CentOs/Redhat – rpm, yum
 - ▶ yum install <paquet>
 - ▶ debian/ubuntu – apt, aptitude
 - ▶ apt-get install <paquet>
- ▶ Utiliser une interface graphique
 - ▶ synaptic





COMMANDES LINUX





Le shell UNIX

- ▶ Au début il y avait le shell...
 - ▶ Une commande est un programme qui interagit avec le noyau afin d'effectuer des actions appelées par l'utilisateur
 - ▶ Une commande peut être : interne au shell (buil-in); un script shell exécutable, ou un code compilé
 - ▶ Le **shell** est l'interpréteur de commandes. L'utilisateur interagit avec le noyau à travers le shell, soit directement par ligne de commande soit par le biais d'interfaces graphiques. Il est courant d'écrire des scripts (commandes en mode texte) afin d'automatiser certaines actions sur le shell.



Le Shell UNIX

- ▶ D'habitude, chaque machine UNIX est dotée d'un shell compatible **Bourne shell**
- ▶ Le shell Bourne original est disponible sur `/bin/sh`
 - ▶ Dans la plupart des Linux actuels ce fichier est un lien vers `/bin/bash`
- ▶ L'invité de commande Bourne shell est identifié par un **\$** (ou **#** si vous êtes logué en tant que root)
- ▶ Un autre shell autrefois populaire est le C Shell (sur les machines Solaris, par exemple). L'invité de commande C shell est identifié par un **%**



Comment obtenir de l'aide sur les commandes

- ▶ La plupart des commandes est associée à une aide en ligne. La commande **man** permet d'obtenir la description de la commande et de ses options
 - ▶ **man ls**
- ▶ Il est aussi possible d'utiliser le GNU Info System (commande **info**)
- ▶ Des commandes telles que **whatis** et **apropos** permettent de chercher des références sur une commande
- ▶ Plusieurs outils ont des option de type long, dont **--help**, ou **--usage** qui donnent des informations sur les options et les arguments
 - ▶ **whoami --help**



Structure d'une commande

- ▶ commande <options> <arguments>
 - ▶ Ne pas oublier les espaces entre les commandes, options et arguments
 - ▶ Les commandes sont généralement **sensibles à la casse**
- ▶ Dans une même ligne, nous pouvons plusieurs commandes, séparées par des ;
 - ▶ Celles-ci seront exécutées l'une après l'autre, dans l'ordre
- ▶ Dans le prochain cours nous allons étudier un peu la programmation Shell



Lister le Contenu d'un Répertoire

- ▶ La commande **ls** est utilisée pour lister le contenu d'un répertoire.
- ▶ L'utilisation de paramètres permet d'afficher des informations supplémentaires :
 - ▶ \$ **ls -l** listage longue (avec permission, propriétaire, groupe)
 - ▶ \$ **ls -a** listage des tous les fichiers (même occultes)
 - ▶ \$ **ls -t** listage des dates de modification des fichiers
 - ▶ \$ **ls -u** listage des dates de dernière accès
 - ▶ \$ **ls -R** listage récursive (entre dans les répertoires)
 - ▶ \$ **ls -S** listage ordonné par taille des fichiers
- ▶ Les paramètres peuvent être combinés
 - ▶ Ex : **ls -lR**, **ls -la**



Se Déplacer dans les Répertoires

- ▶ **cd try_it**
« entre » dans le répertoire try_it (qui devient le répertoire courant)
- ▶ **pwd**
affiche le chemin absolu du répertoire courant (/home/smith/try_it)
- ▶ **cd ..**
retourne au répertoire supérieur
- ▶ **cd /home**
se déplacer vers le répertoire /home (en utilisant le chemin absolu)
- ▶ **cd ~**
revient à votre répertoire home (~ est un alias pour /home/votrenom)
- ▶ **cd ~/source**
entre dans le répertoire source dans votre espace home



Opérations sur le Répertoire

- ▶ La commande **mkdir my_dir**
 - ▶ Crée un nouveau répertoire my_dir (le chemin donné est relatif) comme sous-répertoire du répertoire courant.
- ▶ La commande **rmdir your_dir**
 - ▶ Supprime le répertoire your_dir (uniquement si celui est entièrement vide).



Opérations sur les fichiers : Copy / Move

- ▶ La commande

- **cp file_1 file_2**

- ▶ Copie le contenu de file_1 vers file_2. Les deux fichiers doivent être dans le même répertoire, sinon l'utilisation de chemins est nécessaire. Si file_2 n'existe pas il sera créé; si file_2 existe déjà, il sera remplacé

- ▶ La commande

- **mv file_1 file_2**

- ▶ Transmet le contenu de file_1 vers file_2
- ▶ Le fichier file_1 est supprimé du disque.
- ▶ Utilisé pour renommer un fichier

- ▶ La commande

- **mv file_1 /home/chemin**

- ▶ Déplace le fichier file_1 vers le répertoire /home/chemin



Suppression d'un fichier

- ▶ La commande

- **rm file_a**

- ▶ Supprime le fichier file_a du système

- ▶ Si vous utilisez des jokers (*)

- **rm h*c**

- ▶ Tous les fichiers commençant par h et finissant par c seront supprimés du répertoire courant

- ▶ Si vous utilisez uniquement le joker

- **rm ***

- tous les fichiers du répertoire courant seront supprimés



Utilisateurs, Groupes et Permissions d'Accès

- ▶ UNIX/LINUX implémente le concept d'utilisateur et de groupes associés aux fichiers
- ▶ Le système détermine si un utilisateur ou groupe ont des droits d'accès à un fichier ou répertoire selon les permissions attribués à ces fichiers
- ▶ Le super-utilisateur garde néanmoins le droit d'accès sur la totalité des fichiers et répertoires.



Permissions d'Accès

- ▶ Chaque fichier, répertoire ou application contient trois types de permission :
 - ▶ **r** — Indique le droit de lecture sur le fichier (**Read**).
 - ▶ **w** — Indique le droit d'écriture sur le fichier (**Write**).
 - ▶ **x** — Indique le droit d'exécution sur le fichier (**eXecute**).
- ▶ Chacune de ces trois permissions sont attribuées à trois catégories différentes d'utilisateurs :
 - ▶ User – Le propriétaire du fichier ou de l'application.
 - ▶ Group – Les membres du groupe qui détient le fichier ou l'application.
 - ▶ On peut aussi dire « les autres membres du groupe du propriétaire »
 - ▶ Others – Tous les autres utilisateurs ayant accès au système.



Permissions d'Accès

- ▶ La liste des permissions peut être facilement accédée à partir de la commande de listage longue de fichiers **ls -l**.
- ▶ Par exemple, si l'utilisateur juan crée un exécutable nommé test ,la sortie de la commande **ls -l test** serait similaire à :

```
-rwxrwxr-x 1 toto student 0 Sep 26 12:25 test
```

- ▶ Cette ligne indique que le fichier est disponible pour lecture, écriture et exécution par le propriétaire (toto) aussi que les membres du groupe student.
- ▶ Le fichier est aussi lisible et exécutable par tous les autres utilisateurs, qui n'ont pas le droit d'écrire dessus.



Permissions d'Accès

-rwxrwxr-x 1 toto student 0 Sep 26 12:25 test

- ▶ Les permissions pour ce fichier sont indiquées au début de la ligne, initiant par rwx.
 - ▶ Le premier signe indique un fichier (-), un répertoire (d) ou autres
 - ▶ Les trois signes suivants indiquent les permission du propriétaire.
 - ▶ Les trois signes suivants indiquent les permissions du groupe.
 - ▶ Les trois derniers signes indiquent les permissions des autres.



Modification des droit d'accès

- ▶ Le propriétaire d'un fichier ou répertoire peut être changé avec
 - chown <proprietaire> <fichier>**
- ▶ Le groupe propriétaire d'un fichier peut être modifié avec
 - ▶ **chgrp <groupe> <fichier>**
- ▶ Les permissions d'un fichier peuvent être modifiées avec la commande
 - chmod -R ### <fichier ou répertoire>**
- ▶ L'option -R permet de changer récursivement la permission de tous les fichiers sous un répertoire.



Comment Changer les Permissions ?

- ▶ Les permissions ont des valeurs $r=4$ $w=2$ et $x=1$
- ▶ 0 = Rien
- ▶ 1 = Exécution
- ▶ 2 = Écriture
- ▶ 3 = Exécution & Écriture ($2 + 1$)
- ▶ 4 = Lecture
- ▶ 5 = Exécution & Lecture ($4 + 1$)
- ▶ 6 = Lecture & Écriture ($4 + 2$)
- ▶ 7 = Exécution & Lecture & Écriture ($4 + 2 + 1$)
- ▶ Ex : **chmod 751 fichier**
- ▶ On peut aussi attribuer des permissions
 - ▶ `chmod u=rwx,g=rx,o=x fichier`



Gestion de fichiers

- De surcroît, existent trois type de permission spéciales existent :
 - ▶ **setuid** — utilisée seulement par les applications, indique que l'application sera exécutée avec les mêmes permissions du propriétaire au lieu de l'utilisateur courant. Est indiquée par un **s** à la place du **x** du propriétaire. Si le propriétaire n'a pas le droit d'exécution, on voit la lettre **S** majuscule.
 - ▶ **setgid** — fonction similaire à setuid, le bit setgid indique que les permissions du groupe seront utilisées (à la place des permissions de l'utilisateur courant).
 - ▶ **sticky bit** — normalement utilisé pour les répertoires, indique qu'un fichier crée dans ce répertoire ne pourra être supprimé que par son propriétaire. La lettre **t** remplace le **x** dans toutes les catégories.
 - ▶ Utile pour permettre aux membres du groupe d'écrire sur le fichier mais jamais l'effacer



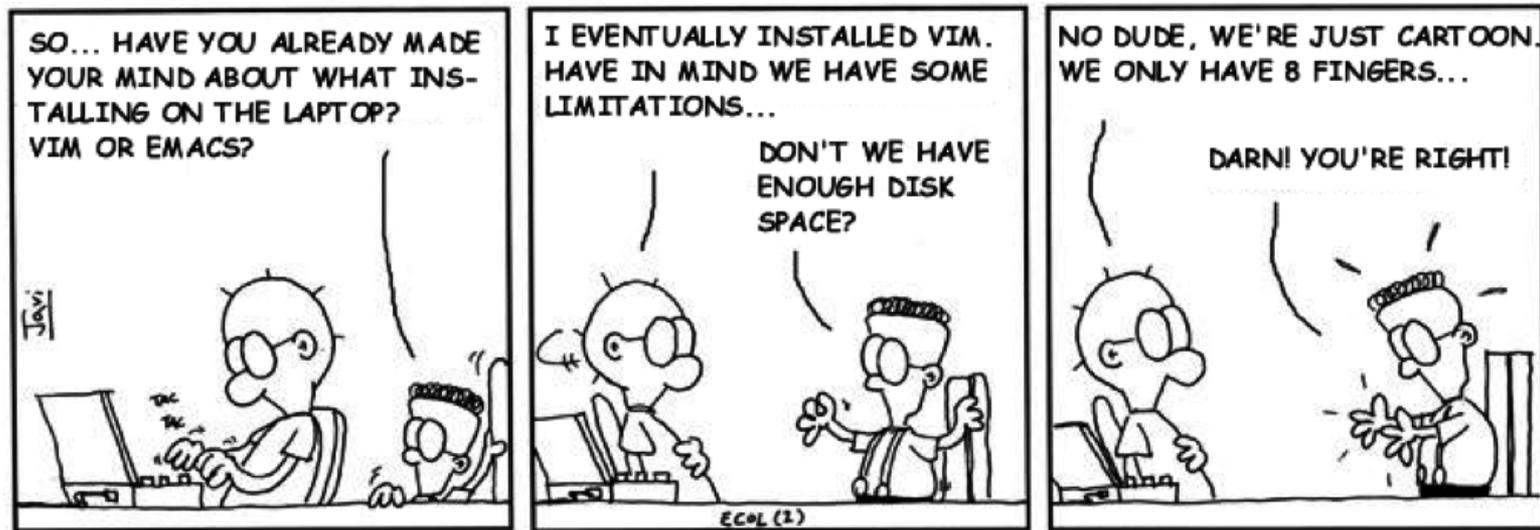
Gestion de fichiers

- ▶ **chown** Modifie le propriétaire d'un fichier.
 - ▶ **chown <owner1> <filename>** : owner1 devient le propriétaire du fichier.
- ▶ **chgrp** Modifie le groupe d'un fichier.
 - ▶ **chgrp <group1> <filename>** : le fichier appartient désormais à group1.
- ▶ **tail** Affiche les 10 dernières lignes d'un fichier.
 - ▶ **tail <filename> = tail -10 <filename>**
 - ▶ **Tail -100 <filename>** - affiche les 100 dernières lignes
- ▶ **head** Affiche les 10 premières lignes d'un fichier
 - ▶ **head <filename>**
 - ▶ **head -50 <filename>** - affiche les 50 premières lignes



Édition de fichiers

- Visualisation et édition
- ▶ **emacs** Éditeur « full screen », puissant mais parfois lourd
- ▶ **pico, nano** Editeurs simple
- ▶ **vi** Editeur de texte puissant mais pas très simple à utiliser. Toujours installé dans les Linux
- ▶ **gedit** Éditeur de texte graphique



Gestion de fichiers

- Compression, sauvegarde et restauration
- ▶ **compress / uncompress** Compression (dans le format .Z)
- ▶ **gzip / gunzip** – compression vers le format .gz
- ▶ **zip / unzip** – compression vers le format .zip (comme sous windows)
- ▶ **bzip2 / bunzip2** – compression vers le format bzip2 (le plus efficace)
- ▶ Sauf **zip**, les autres font d'habitude la compression d'un seul fichier
- ▶ **tar** Commande pour archiver (regrouper) des fichiers et répertoires
 - ▶ Possibilité de compression intégrée avec compress (Z), gzip (z) et bzip2 (j)
 - ▶ **tar -cvzf <destination> <files/directories>** - regroupe et compresse les fichiers avec le format gzip.
 - ▶ **tar -xvzf <compressed file>** - extrait les fichiers
 - ▶ **tar -tvzf <compressed file>** - liste le contenu uniquement



Gestion de fichiers

- ▶ **cat** affiche le contenu d'un fichier
 - ▶ **cat filename**
 - ▶ Très utilisé avec les tubes (ex: **cat /etc/passwd | grep root**)
- ▶ **cmp** Compare deux fichiers.
- ▶ **diff** Affiche les différences entre deux fichiers.
 - ▶ **diff file1 file2**
- ▶ **cut** Permet de découper des données dans une ligne
 - ▶ **cat /etc/passwd | cut -d ':' -f 1** - affiche le nom des utilisateurs
- ▶ **echo** Affiche une ligne de texte



Gestion des fichiers

- ▶ **grep** affiche tous les fichiers/lignes avec une patron donné
 - ▶ **grep pattern <filename/directorypath>**
 - ▶ **ls -l |grep sidbi** : identifie toutes les lignes avec le mot sidbi
 - ▶ **grep " R "** : recherche la lettre R entourée de spaces
- ▶ **sleep** rajoute une pause pendant un temps donné
- ▶ **sort** permet de trier une liste
- ▶ **uniq** supprime des entrées répétées dans une liste ordonnée
- ▶ **wc** compte le nombre de lignes (-l), mots (-w) ou lettres (-c) dans une entrée.



Administration et sudo

- ▶ L'utilisateur root est trop puissant pour être utilisé comme un compte utilisateur
- ▶ La commande **su** permet de devenir root mais ce n'est pas une solution efficace
 - ▶ Parfois on « oublie » de se déconnecter
 - ▶ Le mot de passe du root est connu par l'utilisateur
 - ▶ Il a tous les pouvoirs du root
- ▶ La commande sudo permet d'accorder aux utilisateurs des droits sur les appels individuels. C'est leur propre mot de passe qui est demandé
 - sudo ifconfig
password: *****
- ▶ Les utilisateurs (et leurs droits) sont indiqués dans le fichier /etc/sudoers



Commandes d'Administration du Réseau

- ▶ **arp** Ce programme affiche et permet de modifier le cache ARP.
- ▶ **dig** Envoie des requêtes de nom de domaine aux serveurs DNS – util pour des test ou du débogage.
- ▶ **ftp** Permet le transfert de documents/fichiers.
- ▶ **ifconfig** Affiche les interfaces réseau, leurs configurations et aussi permet de modifier les configurations.
 - ▶ **ifconfig eth0 address 172.31.2.2 netmask 255.255.0.0**
- ▶ **ifdown** Désactive une interface réseau. Ex: **ifdown eth0**
- ▶ **ifup** Permet d'activer une interface réseau. Ex: **ifup eth0**
 - ▶ Sous ubuntu : le processus NetworkManager est activé par défaut, il se superpose généralement aux configurations « faites à la main »



Commandes d'Administration du Réseau

- ▶ **netstat** Affiche des informations sur les connexions réseaux en cours (port, origine/destination, type de protocole, etc.). L'option "**netstat -r**" affiche la table de routage
- ▶ **Ping** Permet de tester une connexion à l'aide d'échos en ICMP
- ▶ **Traceroute/tracepath** Affichent le chemin suivi par des paquets IP
- ▶ **Tcpdump** Permet de surveiller/capturer les paquets qui transitent sur une machine



Commandes d'Administration du Réseau

- ▶ **route** Outil pour manipuler la table de routage (route add, route del)
- ▶ **telnet** Permet d'établir une connexion terminal distante à l'aide du protocole TELNET
 - ▶ telnet <remote system name/ip>
 - ▶ Encore largement utilisé pour se connecter à l'interface d'administration du matériel réseau (routeurs, switches)
- ▶ **ssh** Version sécurisée de telnet.
 - ▶ ssh <remote system name/ip>
- ▶ **sftp / scp** – outils pour le transfert de données à l'aide du protocole SSH



Commandes d'Administration du Réseau

- ▶ Les fichiers clé
 - ▶ /etc/hostname – le nom de votre machine
 - ▶ /etc/resolv.conf – l'adresse IP des serveurs DNS
 - ▶ /etc/hosts – fichier avec des noms « locaux » (similaire à DNS)
 - ▶ répertoire /etc/network – fichiers de configuration statique du réseau



Administration du Système

- Gestion des niveaux d'exécution du système
 - ▶ **exit** quitte le shell
 - ▶ **halt** Arrête le système
 - ▶ **init** Permet de contrôler le niveau d'exécution. (init 3)
 - ▶ **logout** Déconnecte l'utilisateur
 - ▶ **poweroff** Arrête le système
 - ▶ **reboot** Réinitialise le système
 - ▶ **runlevel** Affiche le runlevel courant (et le précédent)



Administration du Système

- ▶ **shutdown** Permet d'étendre la machine ou la rebooter. Shutdown permet aussi de programmer l'arrêt et d'envoyer des messages aux utilisateurs connectés.
shutdown -h +10 'Un nouveau disque sera installé. Nous retournons d'ici une heure.'



Administration du Système

- L'heure du système
- ▶ **cal** affiche un calendrier
 - ▶ cal
 - ▶ cal 2009
- ▶ **date** Affiche la date et l'heure du système (avec des options de format de sortie).
date permet aussi de modifier la date du système
date MMDDhhmm[[CC]YY][.ss])
- ▶ **hwclock** Affiche ou modifie la date dans la CMOS de l'ordinateur
- ▶ **uptime** Indique combien de temps la machine est allumée



Administration des Comptes

- ▶ **useradd** rajoute ou modifie les informations d'un compte utilisateur.
 - ▶ **useradd -g <group> -s <shell> -c <comment> -d <home directory> <username>**
- ▶ **userdel** supprime un compte utilisateur et tous ses fichiers.
 - ▶ **userdel <user name>**
- ▶ **passwd** Permet de changer le mot de passe d'un utilisateur
 - ▶ **passwd**
 - ▶ **passwd <username>**
- ▶ **users** Affiche le nom des utilisateurs connectés
- ▶ **wall** envoie un message aux terminaux de tous les utilisateurs
 - ▶ **wall "text message"**
- ▶ **who** Affiche les utilisateurs connectés et dans quel terminal
- ▶ **whoami** Affiche l'identifiant de l'utilisateur courant



La gestion des utilisateurs

- ▶ Le fichier `/etc/passwd` est la base de données sur les utilisateurs.
 - ▶ Exemple de ligne :

```
toto:DcoJmNQ:1003:1003:Martin,,,:/home/toto:/bin/bash
```

- ▶ Suite de champs séparés par des ':'
 - ▶ login de l'utilisateur
 - ▶ mot de passe crypté. Il peut être remplacé par x et déporté dans `/etc/shadow` pour le cacher (seul root peut lire)
 - ▶ uid : numéro d'utilisateur utilisé dans les inodes, entre autres. l'uid du super-utilisateur est 0
 - ▶ gid : numéro de groupe principal
 - ▶ commentaire : en général nom et prénom
 - ▶ répertoire personnel
 - ▶ commande de connexion : shell à lancer lors de la connexion



La gestion des utilisateurs

- ▶ Fichier `/etc/group` est la base de données sur les groupes et les utilisateurs qui appartiennent à ces groupes

- ▶ Exemple de ligne :

```
audio:DcoJmNQ:29:plateau,toto
```

- ▶ Suite de champs séparés par des ':'
 - ▶ Nom du groupe
 - ▶ Mot de passe crypté : peu utilisé
 - ▶ Identifiant du groupe (gid)
 - ▶ Liste des utilisateurs du groupe (connexion sans mot de passe)



La gestion des utilisateurs

- ▶ /etc/passwd et /etc/group sont lisibles par tous les utilisateurs
 - ▶ **risque d'attaque des mots de passe en force brute**
- ▶ solution : cacher les mots de passe dans /etc/shadow

```
toto:$1$wfl1guEva$6hednbGEed1NqCuqKVwQN1:12663:0:99999:7:::
```

- ▶ login
- ▶ mot de passe crypté
- ▶ dernière modification de mot de passe (jours depuis 1/1/70)
- ▶ jours à attendre avant de pouvoir changer le mot de passe
- ▶ jours avant avertissement de fin de validité du mdp
- ▶ jours entre avertissement et désactivation
- ▶ date de désactivation
- ▶ champs réservés



Gestion des processus

- ▶ **ps** Affiche le status des processus
 - PPID**- ID du processus père ; **PID**- ID du processus
 - ▶ **ps ax | more** liste tous les processus, daemons inclus
 - ▶ **ps -ef | grep <process>** recherche un processus spécifique
- ▶ **pstree** Affiche l'arbre des processus courants
- ▶ **top** Outil qui permet une vue dynamique de l'état des processus (activité, consommation CPU, mémoire).
- ▶ **lsuf** liste les fichiers ouverts pour chaque processus
- ▶ **fuser** liste des processus associés à un dispositif



Gestion des processus

- Gestion de processus sous Linux
 - ▶ **<commande/programme> &** lance le programme comme tâche de fond (ne bloque pas le terminal)
 - ▶ **Ctrl-z** permet de suspendre un processus actif (dans le terminal)
 - ▶ **bg** relance en tâche de fond un processus suspendu
 - ▶ **fg** relance dans le terminal un processus suspendu
 - ▶ **kill** permet de « tuer » ou d'arrêter un processus
 - ▶ **kill 34** – arrêt ou suspend le processus numéro 34.
 - ▶ **kill -9 <pid>** - force l'arrêt d'un processus
 - ▶ **killall** permet de tuer un processus en appelant son nom (au lieu du PID)
 - ▶ **pid** Affiche le PID du processus courant



Automatisation de tâches

- ▶ Deux types de tâches automatiques :
 - ▶ retardées (delayed)
 - ▶ répétitives

at(1) : programme une exécution retardée

at 10am Jul 31 ; at 1am tomorrow ; at now ; at 4pm

(ouvre un éditeur interactif)

atq, at -l : liste les exécutions prévues, en annule une

atrm : annule une exécution prévue



Automatisation de tâches

- **crontab(1)** : programme une tâche répétitive
- Une table par utilisateur (édition: `crontab -e`) dans `/var/spool/cron`
 - Le démon *crond* se charge d'appliquer ces configurations.
 - `PATH=/usr/bin:/bin:/home/toto/SCRIPTS`
 - **# Minutes Heures Jours du mois Mois Jours de la semaine Commande**
 - `2 0,4,8,12,16,20 * * * p=`pidof sauvegarde`; [-n "$p"] || sauvegarde`
 - `49 10 * * 1 linux-counter-machine-update -m`
- **#** Le caractère `*` permet d'indiquer le plus grand intervalle possible.



Automatisation de tâches

- Certaines distributions comportent les répertoires :
 - `/etc/cron.hourly`
 - `/etc/cron.daily`
 - `/etc/cron.monthly`
- `/etc/crontab` est alors configuré pour exécuter les scripts se trouvant dans ces répertoires respectivement toutes les heures, tous les jours et tous les mois.

