Administration machine

Sécurité et Administration des Systèmes Informatiques

Administration machine

Fabrice Legond-Aubry
Fabrice.Legond-Aubry@u-paris10.fr

Introduction

Les ressources WEB

- Internet
  - http://www.commentcamarche.net/unix (intro)
  - http://www.ugu.Com
  - http://www.linux-france.org
  - http://developer.apple.com (MacOs X)
  - http://msdn.microsoft.com (windows 2k3, xp, 2k)
- Livres
  - Systèmes d'exploitation (2nde édition, A. Tanenbaum)
  - The C Programming Language, Second Edition (2nde édition, Brian W. Kernighan & Dennis M. Ritchie)
- Conférences
  - http://www.jres.org

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Linux: Etape 1

- Etape 1 – Firmware (BIOS)
  - Charge le MBR (Master Boot Record)
    - ✔Bootstrap (démarrage). Exécute le chargeur de démarrage
    - ✔Chargeur de démarrage (Boot Loader): LILO, GRUB, ...
- Le BIOS a des limitations (format 16 bits)
  - Il ne peut gérer des disque de plus de 2.3To (Table partition), 1Mo de RAM
- Remplacer par l'EFI qui joue le même rôle
  - Utilisation des GUID Partition Table (Taille max: 8Zio)
  - Permet un boot sécurisé, des MAJ sécurisés, un stockage des noyaux systèmes (utilisation de code signé)
  - Inclus la gestion réseau
  - Chaque partition à sa partition EFI (~100Mo) pour y stocker le bootloader. Formater en FAT32.

Démarrage du Système

Linux: Etapes 2

- Etape 2 – Exécution du « Boot Loader »
  - Charge une carte [map] du disque pour le système
    - ✔Trop bas niveau pour avoir une notion de système de fichiers (FS)
    - ✔Ce fichier contient une suite de blocs physiques (cylindre, tête, secteur) pointant sur le code du noyau (kernel)
    - ✔Lilo utilise les « maps », Grub sait interpréter les FS
  - Charge le noyaux des OS (Linux, Windows, etc...)
    - ✔Utilise les interruptions BIOS pour lire les secteurs disque (secteurs bas niveau)
- Pour l'EFI,
  - Charge le Bootloader sur la partition EFI

Démarrage du Système

Linux: Etapes 3 et 4

- Etape 3 [optionnel]– Créer et charge un disque virtuel (ramdisk)
  - « Indépendant du matériel »
  - Pas besoin de driver n'utilise que la ram
  - Utile pour charger une image minimale d'un disque qui lui contient des drivers (modules)
    - ✔Donner un accès à la machine (drivers disque) pour un noyaux générique
    - ✔Booter sur un CD, clé USB, …
- Etape 4 – Exécution du noyau (Kernel)

  Le bootloader exécute le noyau en transmettant les arguments
  - ACPI [gestion d'alimentation], APIC [interruption]
  - Démarrage en mode Single User
  - Spécification du script de démarrage

Démarrage du Système

Linux: Etapes 5, 6 et 7

- Etape 5 – Création du premier processus (nommé « init ») par le noyau
  - Son pid est 1 (« *ps auxww | grep init* »)
  - Premier créé, dernier stoppé
  - Offre le fork/exec (Voir les cours systèmes de L)
- Etape 6 – Interprétation du fichier « */etc/inittab* » par le noyau
  - Le processus Init lit le fichier « */etc/inittab* »
  - Exécution le script d'initialisation du système (par défaut /etc/rc.sysinit). Voir ligne sysinit de */etc/inittab*
  - *rc.sysinit* effectue les configurations de base
  - Passage du système dans le niveau d'exécution (RunLevel) par défaut
- Etape 7 – Exécution des scripts spécifique à chaque RunLevel
  - Init exécute les scripts spécifiques à chaque niveau d'exécution

Démarrage du Système

Structure de « /etc/inittab »

```
# Default runlevel.
id:5:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
# Specific runlevel init scripts (id:runlevel:action:process)
l0:0:wait:/etc/rc.d/rc 0
….
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
# Run gettys in standard runlevels (consoles)
1:2345:respawn:/sbin/mingetty tty1
```

Démarrage du Système

Niveau d'exécution (Runlevels) de Linux

- Utilisez « *man inittab* », « *man init* », « *man runlevel* »
- Niveau 0
  - Exécution de ce niveau lors de l'arrêt du système
  - Arrêt des services
- Niveau 1 (ou S)
  - Exécution du système en mode utilisateur seul
  - Utilisé pour l'administration
  - Seul root est connecté, sans réseau
- Niveau 2
  - Mode multi-utilisateurs
  - SANS les systèmes de fichier réseaux (i.e. NFS)
  - Seul les systèmes de fichiers locaux (FS) sont montés

Démarrage du Système

Niveau d'exécution (Runlevels) de Linux

- Niveau 3
  - Mode multi-utilisateurs
  - AVEC les système de fichiers réseaux (i.e. NFS)
  - Tous les systèmes de fichiers sont montés
- Niveau 4 – Inutilisé
- **Niveau 5**
  - Mode multi-utilisateurs
  - AVEC les système de fichiers réseaux (i.e. NFS)
  - Tous les systèmes de fichiers sont montés
  - Interface graphique (X11) et bannière de login graphique (kdm)
  - Niveau d'exécution du système linux par défaut
- Niveau 6
  - Redémarrage du système
  - Utilisé par la commande « *reboot* » ou « *shutdown –r* »

Variation des niveaux d'exécution Linux

Red Hat et Mandrake	SuSE	Debian	Slackware
0 Arrêt système	0 Arrêt système	0 Arrêt système	0 Arrêt système
1 Mono-user	1 Multi-user, sans réseau	1 Mono-user	1 Mono-user
2 Multi-user, sans réseau	2 Multi-user, réseau	2 Multi-user, sans réseau	2 Multi-user
3 Multi-user, réseau	3 Multi-user, réseau, X	3 Multi-user, réseau	3 Multi-user
4 Non utilisé	4 Non utilisé	4 Non utilisé	4 Multi-user, X
5 Multi-user, réseau, X	5 Non utilisé	5 Multi-user, réseau, X	5 Multi-user
6 Arrêt et reboot	6 Arrêt et reboot	6 Arrêt et reboot	6 Arrêt et reboot
.	S Mono-user	.	S Generic start

Détails sur /etc/rc.sysinit

- Configuration de base du système
  - Horloge, nom de machine (Hostname)
  - Configuration clavier (commande *loadkeys*)
  - Montage du système de fichier « /proc » (méta-data FS)
  - Montage des systèmes d'échange (swap)
- Calcul des dépendances des Modules (drivers)
  - Exécution de la commande *depmod*
    - ✔Création de la liste des symboles pour chaque module /lib/modules/<kernel_version>
    - ✔Le résultat est écrit dans le fichier modules.dep
- Initialisation des interfaces réseaux
  - DHCP ou adresses IP fixes

Niveau d'exécution (RunLevels) de linux

Détails sur les scripts dépendant du niveau d'exécution (RunLevel) :

- Init exécute le script « /etc/rc.d/rc » lorsqu'il entre/quitte un runlevel (Mandrake)
- Les scripts sont stockés dans « /etc/init.d/ »
- Ils doivent supporter les paramètres: start/stop/status/restart
- Le système crée des liens dans « /etc/rc.d/rcX.d » vers les scripts de « /etc/init.d »
- Ces scripts gèrent (arrêt/démarrage) TOUS les services

Niveau d'exécution (RunLevels) de linux

Détails sur les scripts dépendant du RunLevel sous Mandrake (méthode System V) :

- Les liens commençant par '**S**' (**start**) sont exécutés lors de l'entrée d'un runlevel
  - Les noms des liens commencent par '**S**' suivi d'un numéro de priorité suivi du nom du script
  - Les scripts de même priorités sont exécutés dans l'ordre alphabétique
    - ✔Ex: S03iptables, S03shorewall, S10network, S11portmap, ...
- Les liens commençant par '**K**' (**kill**) sont exécutés lors de la sortie d'un runlevel
  - Les noms des liens commencent par '**K**' suivi d'un numéro de priorité suivi du nom du script
  - Les scripts de même priorités sont exécutés dans l'ordre alphabétique
    - ✔Ex: K90network, K89portmap, ...

Démarrage du Système

Niveau d'exécution (RunLevels) de linux

- Mais :
  - le nom des scripts peut varier
  - Les méthodes d'initialisation peuvent varier
- Sous netBSD,
  - On utilise le nom du script «=» (yes|no)
  - Les « yes » peut être suivi de paramètre pour le script
  - Le script de démarrage rc, calcul les dépendances et l'ordre de démarrage des scripts (via /sbin/rcorder)
- Sous Gentoo,
  - Tout passe par /sbin/rc avec des bonnes options
    - ✔shutdown, reboot, default, single, nonetwork
    - ✔Pour modifier la liste des service on utilise rc-update
  - On utilise une fonction depend() défini dans le script
    - ✔Utilisation de directive **need** (besoin), **use** (utilise), **provide** (fournit)
    - ✔Utilisation de **before** ou **after** pour se placer par rapport à un service

Démarrage du Système

Passage à systemd

- Démarrer un service (toto)
  - systemctl start toto.service
- Arrêter un service
  - systemctl stop toto.service
- Redémarrer un service
  - systemctl restart toto.service
- Désactiver un service
  - systemctl disable toto.service
- Activer un service
  - systemctl enable toto.service

Passage à systemd

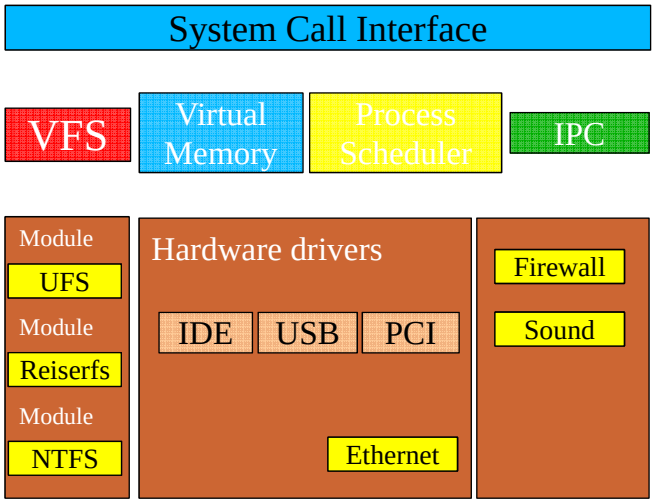
- Les runlevel n'existent plus
  - On utilise les target.
  - Les distributions choisissent leur target
  - systemctl isolate multi-user.target
  - systemctl isolate runlevel3.target
- Changer la target par défaut
  - Mettre à jour le lien : /etc/systemd/system/default.target
  - Fichiers de configuration : /lib/systemd

Passage à systemd

```
[Unit]
Description=Virtual Distributed Ethernet
After=syslog.target
[Service]
Type=forking
EnvironmentFile=/etc/conf.d/vde2
ExecStart=/usr/bin/vde_switch --daemon $VDE_OPTS
Restart=on-abort
Requires=network.service
[Install]
WantedBy=multi-user.target
```

Structure du noyau

- Le noyau Linux n'est pas Monolithique
- Les modules sont chargés par défaut ou sur demande
  - ✔Dynamic Linking (offre des vecteurs d'attaques)
  - ✔Noyaux plus petit, facilement distribuable et configurable
  - ✔Modules indépendants



Section : « Administration machine »

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Les répertoires de Linux

Organisation du système de fichier linux

- Le répertoire racine du FS linux est « / »
- Il existe un certains nombres de sous-répertoires ayant des rôles particuliers pour le système Linux
- Il est possible d'accrocher (monter) une partition (i.e. un disque dur) à n'importe quel endroit de l'arbre des répertoires.
- Il sera alors vu, par le système Linux, comme un sous-répertoire classique !
- On utilise la commande « *mount* »
- Il faut connaître le rôle de chaque répertoire !
- Il faut partitionner correctement les disques !

Les répertoires de Linux

Linux: Tout est fichier !!!

- Les répertoires importants pour Linux:
  - **/boot**
    - ✔Contient les noyaux [kernel.h, vmlinuz, System.map, config]
    - ✔Contient les images noyaux [initrd*.img]
  - **/bin** et **/sbin**
    - ✔Les utilitaires de base (sbin =« system bin», ils sont utiles pour root)
  - **/etc**
    - ✔Contient la plupart des fichiers de configuration de la machine et des services (fichiers /etc/*.conf)
    - ✔Contient les scripts de démarrage
  - **/mnt ou /media**
    - ✔Points de montages « ponctuels » des périphériques externes
    - ✔clef usb, disque, …
  - **/opt**
    - ✔Les logiciels optionnels sous licences ou GPL (ex: Solaris)

Les répertoires de Linux

Linux: Tout est fichier !!!

- Les répertoires importants pour Linux:
  - **/dev**
    - ✔Contient des fichiers spéciaux permettant de manipuler des périphériques
    - ✔**ATTENTION DANGER !!!!**
    - ✔Utilisation des commandes de lecture/écriture sur les fichiers (cat, …)
    - ✔On y trouve par exemple
      - Les disques: /dev/hda[X], /dev/sda[X], /dev/ram[X] (disque virtuel)
      - les lecteurs cd/dvd: /dev/cdrom[X], /dev/dvd[X]
      - Les lecteurs de disquettes: /dev/fd[X]
      - Souris: /dev/psmoue, /dev/psaux
      - La mémoire: /dev/kmem, /dev/mem
      - Le néant [/dev/null], le zéro [/dev/zero]
      - le hasard [/dev/random, /dev/urandom]

Les répertoires de Linux

Linux: Tout est fichier !!!

- Les répertoires importants pour Linux:
  - **/usr**
    - ✔Programmes et bibliothèques du constructeur ou distributeur
    - ✔Pour les utilisateurs
    - ✔/usr/local: Programmes et bibliothèques
  - **/tmp**
    - ✔Fichier temporaire à courte durée de vie
  - **/lib** et **/include**
    - ✔Répertoire contenant des librairies et des headers C (.h)
  - **/var**
    - ✔Données variables à longue durée de vie
    - ✔Le spool d'impression (/var/spool), les mails (/var/mail)
    - ✔Les verrous (/var/lock), les **LOGS (/var/log)**

Les répertoires de Linux

Linux: Tout est fichier !!!

- Les répertoires importants pour Linux:
  - **/home**
    - ✔Données des utilisateurs
  - **/root**
    - ✔Le compte root, local à la machine (n'est pas dans /home)
  - **/proc**
    - ✔Utilisation des commandes de lecture de fichiers
    - ✔Une foule d'informations sur l'état du système !
      - CPU (/proc/cpuinfo), Mémoire (/proc/meminfo)
      - Les processus: /proc/PID
      - Les disques: /proc/partitions, /proc/ide, /proc/scsi
      - Les modules: /proc/modules

Démarrage du Système

- /SYS ??????
- /PROC LECTURE SEULE POSSIBLE ? Mount –rebind
- Verifier /USR/LOCAL

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

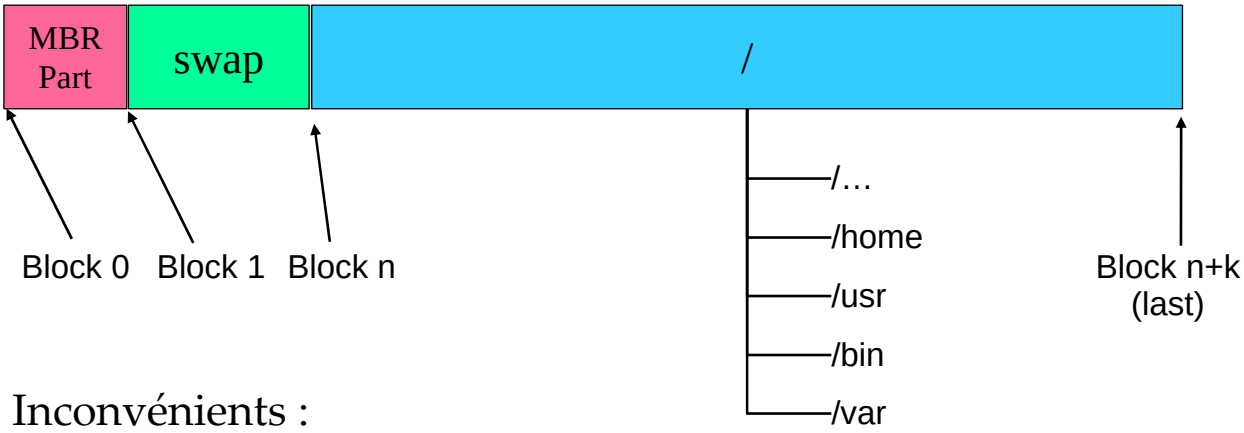
Gestion des droits

Planification

Journalisation

Stratégies d'association partitions / répertoires

- Modèle avec 2 partitions : Machine cliente



Inconvénients :

Lors des MAJ, sauvegarde de /home

Pollution possible de la partition par les logs, les utilisateurs

Le système peut planter à cause des logs

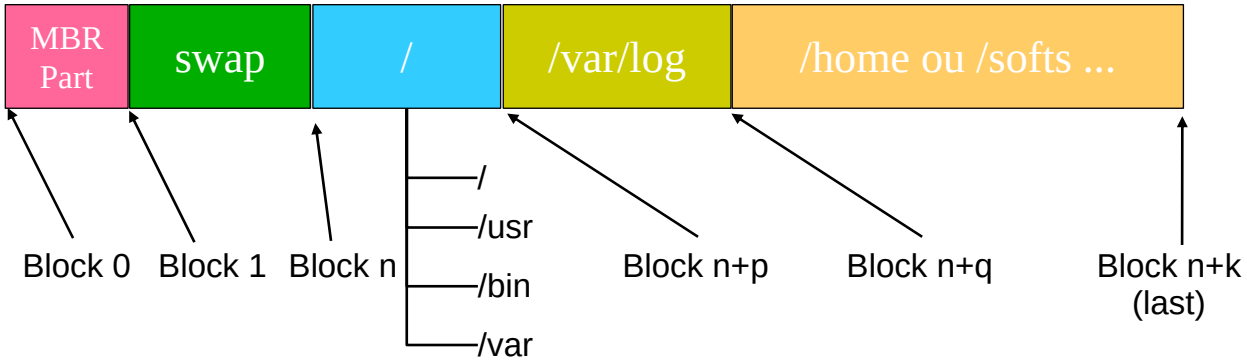
Avantages:

Partage de l'espace

Simplicité

Stratégies d'association partitions / répertoires

- Modèle à 4 partitions : Pour une machine serveur
  - Isolation extrême, sécurité importante
  - Facilité de mise à jour
  - Les fichiers de log (/var/log) ont leur partition !
  - Les comptes (/home) ont leur partition !
  - Le système a sa partition (/) !



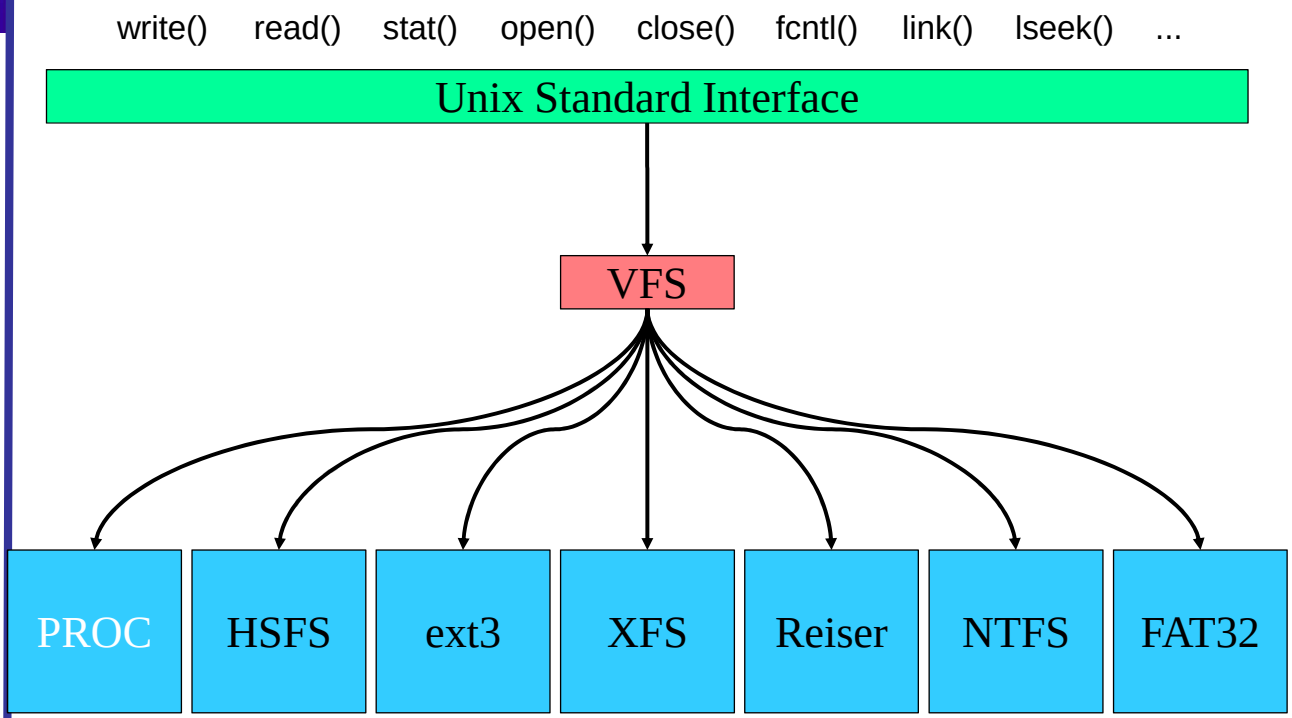
Taille de la partition d'échange

- Revoir vos cours de Système (licence) !!!!
- Rôle : pallier le manque de mémoire physique
- Sa taille dépend de l'algorithme de swap
- Pour Linux :
  - Utilisé lors qu'il n'y a plus de RAM
  - Mémoire libre total = mémoire RAM + taille du swap !
  - Possibilité d'ajouter du swap à postériori.
- Autres systèmes Unix : BSD, Solaris, ...
  - La mémoire doit être superposée au swap (swap pre-allocation)
  - Mémoire libre total = taille de la partition d'échange (Swap)
- Taille de la partition swap Linux = $n * RamSize$
- n=0..3 (Client), n=4..6 (server)
- n=7..10 (gros serveurs) désuet car beaucoup de mémoire
  - On a tendance maintenant a ne plus mettre de swap car trop lent

Taille des autres partitions et nature des FS

- La partition racine (/)
  - ✔Un système linux complet va de 1,44Mo à 10Go
  - ✔Serveurs: réduire le nombre d'applications au minimum !
  - ✔Il faut prévoir un peu d'espace supplémentaire: **~ +50% (6– 10 Go)**
- La taille des autres partitions dépend des besoins !!
- Choisir un FS pour les partitions:
  - ✔Eviter les vénérables ancêtres (UFS/ext2/ext3)
  - ✔Toujours utiliser des FS journalisés (Reiserfs, xfs, jfs) (sauf si vous ne voulez pas laisser de traces !!)
  - ✔Resierfs: des partitions de plusieurs To (10^{12})
  - ✔Xfs: des partitions de plusieurs Po (10^{18})
  - ✔Une fois montés, tous les FS sont identiques pour linux grâce au VFS

L'API du FS : le VFS



Section : « Administration machine »

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

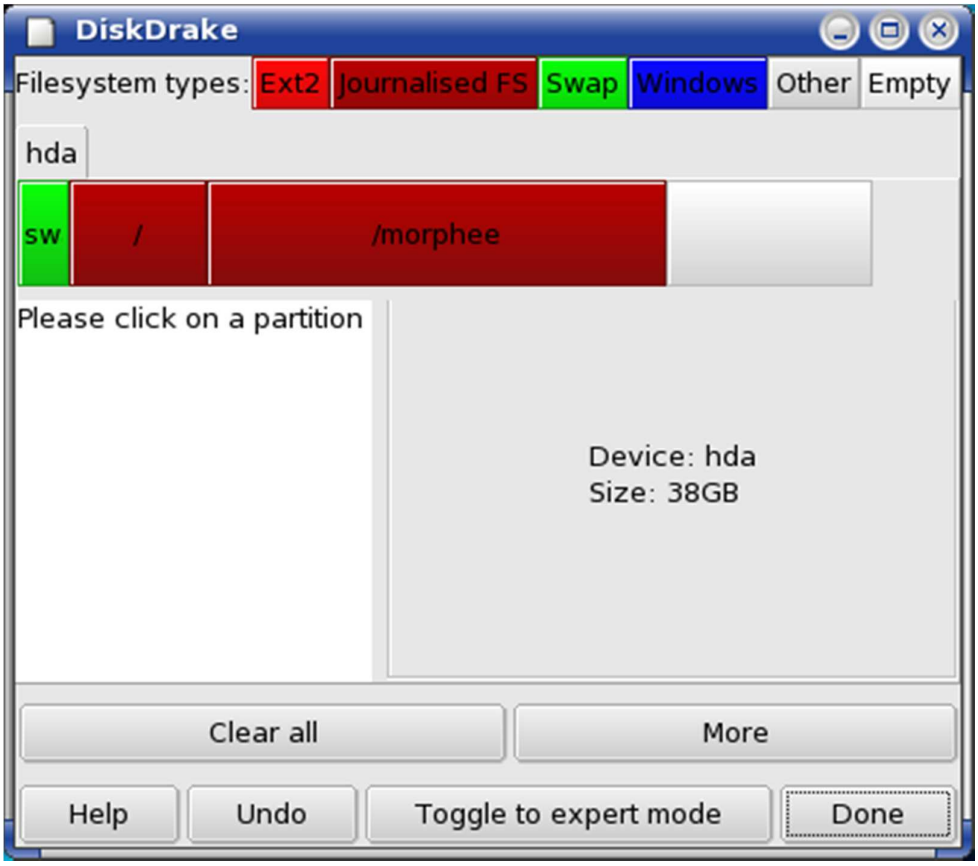
Planification

Journalisation

Créer les partitions: sfdisk/fdisk

- Créer des partitions: commande fdisk /sfdisk
  - Permet de gérer les partitions sur chaque disque
    - ✔*fdisk <device>*
  - Nom des disques sous linux:
    - ✔Disque IDE ATA : /dev/hdXN
    - ✔Disque SCSI / SATA : /dev/sdXN
    - ✔X=a..h (lettre du disque), N=1..8 (numéro de la partition)
  - ATTENTION DANGER !!!!
- Il existe des outils graphiques: DiskDrake (mandrake)
  - Gestion graphique (clicodrome)
  - Peut modifier la taille des partitions (DANGEREUX !!)

Diskdrake



Créer le système de fichiers (formatage): mkfs

- Après avoir sélectionné son FS
  - Utilisation de la commande *mkfs* générale
    - ✔mkfs [-V] [-c] [-t fstype] [fs-options] filesys [blocks]
    - ✔« -c » force la vérification, « -t » pour spécifier le type du FS
    - ✔Filesys est la partition a installer (soit un /dev soit un point de montage si la parition est déjà monté)
    - ✔“fs-options” sont des options spécifiques à chaque FS
    - ✔`mkfs -t xfs /dev/hda3`
  - mkfs est simplement une interface vers des programmes spécifiques à chaque FS: *mkfs.ext2, mkfs.reiserfs*
  - Vérification des secteurs défectueux: *badblocks*
  - Vérification de la structure du FS:
    - ✔*fsck, fsck.ext2, fsck.xfs, …*

Monter un FS: mount

- Automatiquement en utilisant « /etc/fstab »
  - Décrire les points de montage ainsi que leurs options (comme le montage automatique au boot)
- Structure de chaque ligne :

  "Device Mount_point FS_type FS_options Dump FSCK_order"

Périphérique	Point de montage	Type fs	options	Sauvegarde	Vérif.
/dev/hda7	/	xfs	defaults	1	1
none	/proc	proc	defaults	0	0
/dev/sda5	swap	swap	defaults	0	0
/dev/sdb1	/mnt/removable	auto	users, defaults	0	0

Monter/Démonter un FS "à la main"

- Montage manuel
  - Lire le man. De nombreuses options.
  - Ex: `mount -t auto -o ro /dev/fd0 /mnt/floppy`
- Utilisation de la commande
  - `umount peripherique`
  - `umount point_de_montage`
- Pourquoi démonter un FS ?
  - Effectuer une vérification de cohérence (fsck)
  - Sauvegarder en toute sécurité !
  - Interdire l'utilisation d'une partition par les utilisateurs
  - Gestion des disques amovibles (CD, clefs USB)
- Exemples

  ✔`umount /data` ou `umount /dev/hda3`

Gestion des disques

Outils de gestion d'espace: df

- Espace utilisé et libre sur les partitions
  - Commande: **`df [options] [FILE]`**
  - **`[FILE]`** peut être un fichier, un périphérique, un point de montage
  - Une option intéressante « -h » pour une lecture plus aisée
  - Exemples

```
[legond@hebe]> df /dev/hda5 /dev/sda1
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda5             38909396  32879472   6029924  85% /
/dev/sda1             27550256    267588  27282668   1% /data
```

```
[legond@hebe]> df  -h /dev/hda5 /dev/sda1
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda5              38G   32G  5.8G  85% /
```


Gestion des disques

Outils de gestion d'espace: du/ls

- Occupation **réelle** d'un fichier, d'un répertoire ou d'un périphérique
- Commande: ***du***
  - Options utiles : -h (humain) -s (total/argument) -c (grand total)
  - Différent de ***ls -l*** ! (Fichiers creux / Sparse files)
  - Exemple de création d'un fichier creux
    - ✔ **dd if=if/dev/zero of=/tmp/Essai seek=1000000 bs=1 count=1**
    - ✔ Ou le programme C

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main( int argc, char *argv[] )
{
 int file;

 file = creat("/tmp/essai", O_CREAT | O_WRONLY );
 lseek( file, 1000000, SEEK_SET );
 write( file, 'a', 1);
 close( file );
}
```

Gestion des disques

Outils de gestion d'espace: du/ls

```
[legond@morphee]>ls -l /tmp/essai
---x-----x  1 bonnaire src 1000001 Oct 18 14:25 /tmp/essai
```

Taille apparente du fichier en octets

```
[legond@morphee]>du -B 4096 /tmp/essai
1       /tmp/essai

[legond@morphee]>du -h /tmp/essai
4,0K    /tmp/essai
```

Taille réelle: 4 Ko soit 1 bloc (Reiserfs block size = 4096 octets)

- L'option "-s" permet de faire la somme des répertoires

```
[legond@morphee]>du -s -h /usr
3,1G     /usr

[morphee 14:40]>du -s -h /usr/*
176M    /usr/bin
36M     /usr/include
690M    /usr/lib
...
```


Gestion des disques

Outils de gestion d'espace: quota

- Quotas empêche les abus des utilisateurs
- Quotas sont donnés en nombres de blocs disques
- 1re limite : limite douce (Soft Limit)
  - ✔L'utilisateur peut excéder cette limite pendant une période limitée (grace period).
  - ✔Au bout de la période de grâce la limite devient « dure »
- 2e limite : limite dure (Hard Limit)
  - ✔L'utilisateur ne peut plus créer de fichier
  - ✔L'utilisateur ne peut plus ajouter des données à un fichier
  - ✔L'utilisateur ne peut qu'effacer
- Root n'est pas soumis aux quotas ! ☺
- Différents quotas pour différents type d'utilisateurs !

Gestion des disques

Outils de gestion d'espace: quota

- Les quotas peuvent être assignés à des groupes de personnes
- Activer les quotas sur un FS:
  - Commande: **`quotaon [options] <file system>`**
  - Ex: **`quotaon /dev/hda3`**
- Désactiver les quotas sur un FS:
  - Commande: **`quotaoff [options] <file system>`**
  - Ex: **`quotaoff /dev/hda3`**
- Pour les options de quotaon/quotaoff : RTFM !
- Voir aussi : **`quotacheck, quotastats`**
- Voir aussi : **`edquota, repquota`**

Gestion des disques

Encore des outils de gestion de disque

- **dd** (disk dump): copie de données. Très souvent utilisé !
- **tar**, **un/zip**, **cpio**: création d'archives

  `tar cvBf - directory | (cd /backupdir; tar xpBf -)`
- **gzip**, **bzip2**: compression de fichier
- **tee**: duplication d'un flux de données
- **mc**: midnight commander (manipulation de fichiers)
- **rsync**: copie/synchronisation de répertoire
- **dump**: sauvegarde de fs de type ext2/ext3
- **losetup**: permet de manipuler des images de disque (fichier .iso)
- **chown**, **chmod** : changer les propriétaires, les attributs des fichiers
- **touch, stats** : manipuler la date des fichiers
- **lsof**: liste des fichiers ouverts par un processus (UTILE!!)

Gestion des utilisateurs

Section : « Administration machine »

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Gestion des utilisateurs

Linux : un système multi-utilisateurs !

- Le système linux peut être utilisé par plusieurs utilisateurs.
- Le système Linux définit un utilisateur par:
  - Un login: un identifiant texte qui est unique
  - Un identifiant numérique : l'User ID (UID) qui est unique
  - Un identifiant numérique de groupe: Group ID (GID)
    - ✔C'est le groupe principal de l'utilisateur (Primary Group)
    - ✔Il est obligatoire et est utilisé lors de de la connexion
    - ✔L'utilisateur peut appartenir à d'autres groupes
    - ✔Ce sont les groupes secondaires (Secundary groups)
  - Un répertoire de travail (répertoire Maison)
    - ✔Habituellement /home/login
  - Un shell de connexion
    - ✔Par défaut c'est /bin/bash
  - Le nom réel de l'utilisateur (GECOS)

Les identités !

- Attention aux conflits entre les UID des utilisateurs locaux et externes.
  - Pose des problèmes de sécurité en NFS
  - Pose des problèmes lors de l'authentification
    - ✔Voir le fichier **/etc/nsswhich.conf**
- "id" permet d'obtenir des informations sur sa propre identité.

```
root@morphee> id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
root@framekit-dev> id -u
0
legond@framekit-dev> whoami
legond
```

- S'authentifier sous un autre utilisateur (changer son identité) :

```
legond@framekit-dev > su apache
Password:
```


Linux : un système multi-utilisateurs !

- "getent" permet d'obtenir des informations sur un utilisateur ou un groupe.

```
legond@hebe > getent passwd apache
apache:x:74:74:system user for apache2:/var/www:/bin/sh

legond@hebe > getent group src
src:*:300:busca,darche,jlm,cdu,bf,cg,vevar,root
```

- "testsaslauthd" permet de vérifier l'authentification sur un service (login, ftp, …)

```
legond@hebe > testsaslauthd -u legond -s su -p monpass
0: OK "Success."
```

Le Super utilisateur, dieu, l'admin, ... : root

- login root, groupe root (uid=0, gid=0)
- tous les droits sur tous les objets sur une machine locale
- aucun droits sur les machines distantes (NFS)
- droit de contrôle sur les processus
- droit d'accès à tous les fichiers (rwx)
- inconvénient: pas d'identité de la personne
- possibilité d'interdire ce login (**/etc/ttytab**)
- seul à pouvoir travailler en mode "single-user"
- responsable de la cohérence des informations au boot
- **ATTENTION: toutes les bêtises lui sont permises !!!**

Rappel :Choisir un mot de passe

- La fonction crypt() / MD5 /SHA1 sont des fonctions de hachage à sens unique (voir les précédents transparents)
- Cette fonction permet de crypter un mot de passe
- Il n'existe pas de fonction qui permette de revenir de la forme cryptée à la forme claire.
- Il reste possible d'attaquer le mot de passe en cryptant des mot de passes et en les comparant à la forme cryptée.
- Alors peut-on choisir n'importe quel mot de passe ?

NON

- Il est important de bien choisir ses mots de passe
- Un mot de passe de passe doit être unique sur chaque système

Gestion des utilisateurs

Choisir un mot de passe

- Certains mots de passes peuvent être devinés:
  - On crypte un ensemble de mdp préexistants
  - Dictionnaires anglais, français, …
  - Noms de films/acteurs/produits/personnes célèbres
  - Génération de mdp par des règles de transformations (Librairies de crack)
- **Un bon mdp ne doit pas :**
  - Utiliser le dictionnaire
  - Indépendant de l'environnement de l'utilisateur (login name, nom réel)
    - ✔ Pas de nom, surnom, d'identifiant (même modifier), de mot connu, marque
    - ✔ Pas de date (textuelle ou numérique), de clavier
- **Un bon mdp doit :**
  - Etre composé de ponctuations, chiffres, lettres majuscules et minuscules
  - Avoir au moins 7 caractères (si possible 10 à 14)
  - Vous pouvez vous trouver un algo (mais il doit rester secret)
  - Changer son mdp régulièrement

Gestion des utilisateurs

Choisir un mot de passe

- Exemples

  zqhYtds/965 moNpass,67$ Hg#91__bb
- On peut en générer de façon « pseudo-aléatoire »
  - Utilisation de « pwgen » et de « mkpasswd »
- Vérification des mots de passes :
  - Utilisation de la cracklib pour vérifier les mdp « cracklib-check »
- Il existe des logiciels pour attaquer les hash
  - John the Ripper
  - Par dictionnaire, par mutation (I/1/i/l, 0/o/O/@, z/2, E/3)

Ajouter et modifier les utilisateurs

- Changer de mot de passe:
  - ✔ `passwd [-k] [-l] [-u [-f]] [-d] [-S] [nom_utilisateur]`
- Changer de shell:
  - ✔ `chsh [-s shell] [-l] [-u] [-v] [utilisateur]`
- Changer l'âge d'un mot de passe
  - ✔ `chage`
  - ✔ Expiration du mot de passe (0 ou 99999 ➔ pas d'expiration)
- Commande d'ajout d'utilisateur: **useradd**
  - ✔
```
useradd  [-u uid [-o]] [-g group] [-G group,...]
  [-d home] [-s shell] [-c comment] [-m [-k template]]
  [-f inactive] [-e expire ] [-p passwd] name
```
- Commande de modification d'utilisateur: **usermod**
  - ✔
```
usermod  [-u uid [-o]] [-g group] [-G group,...]
  [-d home [-m]] [-s shell] [-c comment] [-l new_name][-f
  inactive] [-e expire ] [-p passwd] [-L|-U] name
```


Effacer un utilisateur

- Commande de modification d'utilisateur: **`userdel`**
  - Effacer un utilisateur et, éventuellement, tous ses fichiers
    - ✔ `userdel [-r] name`
  - L'option "-r" efface tous les fichiers associés
    - ✔ Le répertoire de travail (homedir) et sa boîte mail, …
    - ✔ **ATTENTION: ne peut être annulé !**
- Vérifier l'intégrité de « /etc/passwd »: **`pwck`**
- Le clicodrome mandrake (l'outil graphique): **`userdrake`**
- Encore et toujours: RTFM !

Les fichiers « shadow »

- Tout le monde peut lire /etc/passwd.
  - Problème : les clefs des mdp sont stockés en clair
  - **Dangereux !!! ➔ brute force attack**
- Les clefs des mdp ne sont plus stockés dans « /etc/passwd ».
- Elles sont stockés dans « /etc/shadow » que seul root peut lire.
- Pour convertir les fichiers « /etc/passwd » classiques vers le format shadow : *pwconv, pwunconv*.

Les fichiers « shadow »

- « /etc/shadow » contient :
  - Nom de connexion (login)
  - Mot de passe crypté
  - Nombre de jours écoulés depuis le 1er janvier 1970 jusqu'au dernier changement de mot de passe
  - Nombre de jours durant lesquels le mot de passe est encore valide
  - Nombre de jours après lesquels le mot de passe doit être changé
  - Nombre de jours avant l'expiration du mot de passe impliquant l'avertissement de l'utilisateur
  - Nombre de jours après l'expiration provoquant la désactivation du compte
  - Numéro du jour depuis le 1er janvier 1970 à partir duquel le compte a été désactivé
  - Champ réservé

Les Groupes

- Un Groupe est défini par:
  - Un identifiant [unique]: Group ID (GID)
  - Un nom de groupe [unique]
  - Un mot de passe de groupe (peut être vide)
  - Des membres: une liste d'utilisateurs appartenant au groupe
- Un utilisateur peut appartenir à plusieurs groupes
- Rappel: Un utilisateur appartient à au moins un groupe

Ajouter/modifier/effacer les groupes

- Afficher la liste des groupes auxquels appartient un utilisateur: ***groups***
- Créer un groupe : Commande d'ajout de groupe
  - ***groupadd [-g gid [-o]] [-r] [-f] groupname***
  - Les options :
    - ✔-g : id du groupe
    - ✔-o : permet la création d'un groupe dont l'id n'est pas unique
    - ✔-r : crée un groupe système (GID < 499)
    - ✔-f : force une erreur en cas d'existance du groupe
- Modifier un groupe : Commande de modification
  - ***groupmod [-g gid [-o]] [-n nom_du_groupe] groupe***
- Effacer un groupe : Commande de modification
  - ***groupdel groupe***
- Enlever un utilisateur d'un groupe: ***gpasswd***
- Vérifier l'intégrité de « /etc/group »: ***grpck***

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Un peu de sécurité: droits d'accès

- Chaque fichier/répertoire possède un propriétaire et un groupe
- Chaque fichier possède des autorisations applicables à trois classes d'utilisateurs :
  - u (user) : propriétaire du fichier/répertoire
  - g (group) : groupe du fichier/répertoire
  - o (others) : les autres
- Pour ces 3 catégories, il existe 3 types d'autorisation :
  - r (read) : pour la lecture (Le contenu du fichier peut être lu)
  - w (write) : pour l'écriture (Le contenu est modifiable)
  - x (execute): pour l'exécution (Le fichier peut être exécuté)
- Possibilité d'utiliser les ACL

Gestion des droits

Fichiers et droits d'accès

```
legond@morphee > ls -al .xfig
- rwx r-x r-x  1  legond  src  132  avr 23  2003  .xfig
```

Groupe propriétaire du fichier
Utilisateur propriétaire du fichier
Droits pour les autres utilisateurs
Droits pour les personnes appartenant au groupe du fichier (« src »)
Droits pour l'utilisateur du fichier (ici « legond »)

- Droits pour un fichier:
  - r : Le contenu du fichier peut être lu.
  - w : Le contenu est modifiable.
  - x : Le fichier peut être exécuté.
  - s : utiliser l'ID du propriétaire ou du groupe propriétaire du fichier lors de l'exécution,
  - Le choix le plus sécuritaire pour les fichiers : « **-rw------** » **(devrait être le choix par défaut)**

Gestion des droits

Répertoires et droits d'accès

```
legond@morphee 18:50 > ls -adl ~/.ssh
d rwx r-x r-x  1  legond  src 132 avr 23  2003 .ssh
```

Groupe propriétaire du fichier
Utilisateur propriétaire du fichier
Droits pour les autres utilisateurs
Droits pour les personnes appartenant au groupe du fichier (src)
Droits pour l'utilisateur du fichier (legond)

- Droits pour un répertoire:
  - r : Le contenu est accessibles en lecture. Nécessaire pour la commande « ***ls*** ».
  - w : Les éléments du répertoire sont modifiables. Création/Suppression possible des fichiers contenus par le répertoire. Indépendant des droits de manipulation et d'accès au contenu des fichiers.
  - x : Le répertoire peut être traversé grâce à la commande «***cd*** »
- **ATTENTION :**
  Un fichier est protégé des modifications par ses PROPRES autorisations, et des suppressions par les autorisations du répertoire qui le contient.
- **Le choix le plus sécuritaire est: « drwx--x--x »**

Gestion des droits

Outils de gestion des droits

- Changer les droits :
  `chmod [options] [augo] [+-=] [rwxstugo] fichier(s)`
- Signification :
  - ✔« **a** » : S'applique à tout le monde (user, group et other)
  - ✔« **u** » : S'applique au propriétaire du fichier (user)
  - ✔« **g** » : S'applique au groupe (group)
  - ✔« **o** » : S'applique aux autres (other)
  - ✔« + » : Ajout de nouveaux droits
  - ✔« - » : Suppression d'anciens droits
  - ✔« = » : Redéfinition complète des droits sans tenir compte des anciens
  - ✔Fixer les droits par leurs valeurs octales pour u/g/o : r(4), w(2), x(1)

```
legond@scylla > ls -al unfichier
-rw-------  1 legond src 0 oct  1 22:59 unfichier
legond@scylla > chmod go+w unfichier
legond@scylla > ls -al unfichier
-rw--w--w-  1 legond src 0 oct  1 22:59 unfichier
legond@scylla > chmod 644 unfichier
legond@scylla > ls -al unfichier
-rw-r--r--  1 legond src 0 oct  1 22:59 unfichier
```


Gestion des droits

Outils de gestion des droits

- Définir les droits par défaut des fichiers créés:
  - ***umask [-p] [-S] [mode]***
  - Fixer les droits par leurs valeurs octales pour u/g/o : r(4), w(2), x(1)
  - Ajouter tous les droits que vous voulez interdire !

```
legond@scylla 12:09 > umask 000
legond@scylla 12:09 > umask -s
u=rwx,g=rwx,o=rwx
legond@scylla 12:11 > umask 222
legond@scylla 12:11 > umask -s
u=rx,g=rx,o=rx
```

- Changer le propriétaire / le groupe d'un fichier :
  - ***chown [options] propriétaire[:groupe] fichier(s)***
  - ***chgrp [options] groupe fichier...***

Gestion des droits

Les sudoers

- Ces sont les exceptions.
- « sudo » permet l'exécution de certains programmes critiques par des utilisateurs courants (non root)
- Le fichiers « **/etc/sudoers** » contient la liste de qui peut exécuter quoi.
- Utilisation du groupe « wheel » (man sudoers)
- Structure :
  - User_List Host_List = Runas_Spec [Tag_Spec] Cmnd
  - User_list : les personnes autorisées pour faire sudo
  - Host_list : les machines sur lequels le sudo est autorisé
  - Runas_spec : identité prise par la personne
  - Tag_Spec : options pour le sudo
  - Cmnd : Commande autorisée a être exécutée

Gestion des droits

Les sudoers

- On peut définir des alias pour les users, les hosts, les commandes, les identités prises
- On peut appliquer des options particulières pour une catégorie de machine, d'utilisateur ou d'identité prise
  - 'Defaults' '@' Host paramètres
  - 'Defaults' ':' User paramètres
  - 'Defaults' '>' User paramètres
  - Les paramètres sont légions !
  - Les paramètres sont utiles pour la sécurité !

Les sudoers

- Fichier « **/etc/sudoers** »
  - Ajout d'un programme pour le groupe wheel (sudo sans password)
    - ✔`%wheel ALL=(ALL) NOPASSWD:ALL  prog`
  - Ajout d'un programme pour le netgroupe src (sudo avec password)
    - ✔`+src ALL=(ALL) ALL prog`
  - Ajout d'un programme pour l'utilisateur « newuser » seulement en tant que « operator »
    - ✔`newuser ALL=(operator) ALL prog`
- Exécution via « sudo –u utilisateur commande »

Section : « Administration machine »

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Planification

Programmation de tâches

- Commande « at »
  - permet l'exécution d'une tâche donnée, une seule fois, à un moment donné
  - Si la machine est éteinte à ce moment-là, la tâche ne sera pas exécutée.
  - Dès le redémarrage machine, elle est exécutée
  - La commande est utilisable ***par tout utilisateur*** déclaré sur la machine.
- Commande « cron »
  - permet l'exécution d'une ou plusieurs tâches selon un intervalle de temps fixé et répété
  - Si la machine est éteinte à ce moment-là, la tâche ne sera pas exécutée.
  - La commande est utilisable, par défaut, ***par tout utilisateur*** déclaré sur la machine.

Planification

Programmation de tâches

- Commande « anacron »
  - permet l'exécution d'une ou plusieurs tâches après une période de temps déterminée.
  - Si la machine n'est pas allumée à ce moment-là, la tâche sera exécutée dès que possible.
  - La commande est utilisable ***uniquement par root***.
  - **Exemple :** programmer la sauvegarde de /home tous les 7 jours. Si la machine reste éteinte 9 jours, la tâche s'exécute lors du démarrage de la machine au 10e jour..
- Utilisation de scripts (bash, perl, python)
  - Boucles, sleep, signaux (SIGALARM), ….

```
exec 2>/dev/null 1>/dev/null
while [ 1 ] ;
do
     action;
     sleep delai_a_fixer
done
```

Planification

Commande « at »

- Syntaxe 1 : ***at HEURE***
  - Saisir autant de lignes de commandes que nécessaire
  - terminer par Ctrl+D
- Syntaxe 2 : ***at -f commande HEURE***
- Le format de HEURE peut être :
  - HHMM ou HH:MM
  - midnight / noon / teatime : minuit / midi / 16h (sacrés anglais ;))
  - MMJJAA ou MM/JJ/AA ou MM.JJ.AA : attention, les jours sont à noter après le mois (logique non ? :))
  - « now + x minutes / hours / days / weeks » à partir de maintenant
- Liste des tâches : ***atq***
- Suppression d'une tâche : ***atrm***
- Contrôle d'accès au planificateur : ***/etc/at.deny*** et ***/etc/at.allow***

Planification

Commande « cron », fichier « crontab »

- Il existe une crontab par utilisateur
  - Elles sont dans « ***/var/spool/cron*** »
  - « root » peut agir sur les crontab de tout le monde
  - Contrôle d'accès par ***/etc/cron.deny*** et ***/etc/cron.allow***
- Lister les tâches (« ***crontab -l*** »)
- Supprimer une tâche (« ***crontab -r*** »)
- Editer une tâche (« ***crontab -e*** »)
- Il existe un fichier de tâches pour le système
  - ***/etc/crontab***

Planification

Fichier « /etc/crontab »

- Le fichier « crontab ». Une ligne= Une tâche programmée
- Chaque ligne doit comporter obligatoirement 6 colonnes :
  - 1re colonne, les minutes : de 0 à 59
  - 2e colonne, les heures : de 0 à 23
  - 3e colonne, le jour du mois : de 0 à 31
  - 4e colonne, les mois : de 0 à 12
  - 5e colonne, le jour de la semaine : de 0 à 7 (dimanche correspondant à 0 ou 7)
  - 6e colonne, la tâche à exécuter
- Valeurs possibles pour les champs
  - * : toutes les valeurs possibles
  - 2 nbs séparés par un - : un intervalle de temps
  - nbs séparés par des , : une liste de valeurs
  - / valeur : fixer un pas sur * ou sur un intervalle
- **Exemple :** exécuter « ***/root/backup.sh*** » tous les jours à 23h55

```
55    23    *    *    *    /root/backup.sh
```


Journalisation

Section : « Administration machine »

Démarrage du Système

Les répertoires de Linux

Stratégies de partitionnement

Gestion des disques

Gestion des utilisateurs

Gestion des droits

Planification

Journalisation

Journalisation

Gestion des traces (logs)

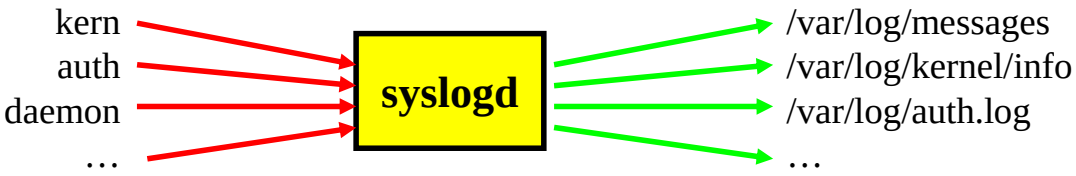
- Principes
  - Notifier toutes les actions du système dans un fichier qui est en perpétuel extension
  - Les journaux peuvent être gérés par les daemons/services « ***syslogd*** » ou « ***syslog-ng*** »
  - Type des journalisation possible: console, mail, fichier, machine
- Niveaux possibles de sévérité/criticité des erreurs (« level ») sont dans l'ordre :
  - **emerg(ency):** Erreurs diffusées à tout le monde. Le système ne sera plus en état de fonctionner
  - **alert:** Erreurs qui doivent être corrigées immédiatement.
  - **crit(ical):** Erreurs critiques à corriger le plus tôt possibles (erreurs de périphériques)
  - **err(or):** Erreurs de niveau « standard »
  - **warning:** Erreurs légères
  - **notice, info, debug:** Informations avec plus ou moins de détails ou d'importance
- Fonctions C pour « logger » des évènements: openlog(), syslog(message), closelog()
- Voir les derniers logs du noyau : « ***`dmesg`*** »
- Voir les dernières connexions : « ***`last`*** »

Journalisation

Les « facilities »

- Les évènements générés par une application/un service peuvent être envoyés à différentes destinations appelées usines de traces (« facilities »).
- Usines de traces (« facilities »):
  - user : messages des processus utilisateurs
  - kern : messages du noyau (et du firewall!)
  - daemon : messages des services
  - mail : messages du services de mail (pop/smtp)
  - news : messages du services de news
  - auth : messages d'authentification et de login
  - cron : messages des planificateurs de tâches
  - local0-7 : des facilities en libre services
  - lpr / uucp / mark (timestamp)

Le démon / service « syslog »



- C'est un commutateur (« dispatcher ») de lignes de traces
- Lit les données envoyées, par les applications, aux facilities
- Les données peuvent être envoyées à « syslog » localement ou par le réseau (UDP port 514, non crypté)
- Ecrit les traces dans des endroits désignés par l'administrateur.
- Capable de séparer les informations
- Configuration par le fichier « /etc/syslog.conf »

Journalisation

« /etc/syslog.conf »

- Lignes de configuration de la forme

  "facility_source_filter destination"

- ***Facility_source_filter*** : permet de sélectionner et d'extraire des données provenant des facilities.
  - Une forme simple est: « facility.level »
  - Le niveau (« level ») peut être crit, emerg, *, …
  - L'usine (« facility ») peut être auth, kern, *, …
  - * désigne tous les éléments ou tous les niveaux
  - Possibilité de séparer, d'agréger et/ou de dupliquer les logs
- ***Destination*** : vers quoi envoyer les données reçues par syslog
  - Un fichier simple. Ex: /var/log/messages
  - Un fichier particulier. Ex: /dev/tty5 (la 5e console)
  - Vers le port UDP 514 d'une machine. Ex: @diane (envoie à la machine diane)

Journalisation

Journalisation

Fichiers de logs

- Exemple :

```
*.*;auth,authpriv.none                  -/var/log/syslog
kern.=debug;kern.=info;kern.=notice     -/var/log/kernel/info
kern.=warn                              -/var/log/kernel/warnings
kern.err                                /var/log/kernel/errors
```

- Les Fichiers de logs importants sous linux sont :
  - /var/messages ➔ messages du systèmes
  - /var/syslog ➔ fichier de log important !
  - /var/auth.log ➔ tout ce qui est authentification
  - /var/boot.log ➔ log du démarrage du système
  - /var/kernel/* ➔ tous les messages du noyau
  - /var/daemons/* ➔ tous les messages des services
  - ….
- **ATTENTION : La structure du /var/log varie suivant les systèmes et les administrateurs !!!!**

Journalisation

Un autre service : Syslog-ng

- syslog-ng pour syslog-nextgen
- Configuration plus complexe mais mieux structurée
- Bien plus puissant au niveau de ses filtres
  - Capable de filtrer suivant le contenu textuel des logs
  - Capable de filtre suivant la machine émettrice
  - Rotation automatique des logs (sans effacement)
- Bien plus puissant au niveau de ses possibilités !
  - Capable d'envoyer les données par udp et tcp
  - Capable d'utiliser n'importe quel port (autre que 514)
  - Compatible avec syslog en réseau

Journalisation

La collecte d'informations et la loi !

- **LA LOI VOUS OBLIGE A CONSERVER CES DONNEES !**
- **LA LOI VOUS INTERDIT DE LES CONSERVER TROP LONGTEMPS !**
  - **Allez voir: « www.droit-technologie.org », « www.cnil.fr »**
- **Les obligations et devoirs :**
  - **La collecte des données**
    - ✔**Interdiction de collecter des données sensibles (races, opinion politique, …)**
    - ✔**Le fait de collecter des données à caractère personnel par un moyen frauduleux, déloyal ou illicite est puni de 5 ans d'emprisonnement et de 300 000 € d'amende (art. 226.18 du code pénal)**

Journalisation

La collecte d'informations et la loi !

- **Les obligations et devoirs :**
  - **La durée de conservation des informations**
    - ✔**Les données personnelles ont une date de péremption**
    - ✔Le code pénal sanctionne la conservation des données pour une durée supérieure à celle qui a été déclarée de 5 ans d'emprisonnement et de 300 000 € d'amende **(art. 226-20 du code pénal)**
  - **La sécurité des fichiers**
    - ✔**Vous devez contrôler les accès aux informations**
    - ✔Le non-respect de l'obligation de sécurité est sanctionné de 5 ans d'emprisonnement et de 300 000 € d'amende. **(art. 226-17 du code pénal)**
    - ✔La divulgation d'informations commise par imprudence ou négligence est punie de 3 ans d'emprisonnement et de 100 000 € d'amende. (art. 226-22 du code pénal)

Journalisation

La collecte d'informations et la loi !

- **Les obligations et devoirs :**
  - **La durée de conservation des informations**
    - ✔**Les données personnelles ont une date de péremption**
    - ✔Le code pénal sanctionne la conservation des données pour une durée supérieure à celle qui a été déclarée de 5 ans d'emprisonnement et de 300 000 € d'amende **(art. 226-20 du code pénal)**
  - **La sécurité des fichiers**
    - ✔**Vous devez contrôler les accès aux informations**
    - ✔Le non-respect de l'obligation de sécurité est sanctionné de 5 ans d'emprisonnement et de 300 000 € d'amende. **(art. 226-17 du code pénal)**
    - ✔La divulgation d'informations commise par imprudence ou négligence est punie de 3 ans d'emprisonnement et de 100 000 € d'amende. (art. 226-22 du code pénal)
- Voir la nouvelle loi sur la conservation des données de connexions
  - *loi n°2006-64 du 23 janvier 2006, JO du 24/01/2006*

Journalisation

Que faire de tous ces logs ?

- logrotate permet :
  - d'éviter l'explosion des logs et la saturation des disques
  - de sauvegarder et de compresser les logs
  - de définir un délai de conservation des informations
  - d'effacer automatiquement les données trop anciennes

Configurer la « rotation » des logs

- Fichier « /etc/logrotate.conf » et répertoire « /etc/logrotate.d »
- Ils contiennent les informations sur comment effectuer la rotation

```
# fichiers ou liste de fichiers a archiver
/var/log/auth.log /var/log/kernel/*.log
{
        rotate 5 # conserve 5 rotations successives
        weekly # archive toute les semaines
        compress # demande la compression des fichiers archivés
        missingok # ne sort pas si le fichier n'existe pas
        # Avertir le service de la sauvegarde des logs
        postrotate
        /usr/bin/killall -HUP syslogd #
        endscript
}
```

- Résultat :

```
[root@scylla root]# ll /var/log/auth*
-rw-r-----  1 root root 185574 Oct  3 18:24 /var/log/auth.log
-rw-r-----  1 root root  23466 Oct  2 04:02 /var/log/auth.log.1.gz
-rw-r-----  1 root root   1731 Sep 25 04:02 /var/log/auth.log.2.gz
-rw-r-----  1 root root   6631 Sep 18 04:02 /var/log/auth.log.3.gz
-rw-r-----  1 root root   7062 Sep 11 04:02 /var/log/auth.log.4.gz
-rw-r-----  1 root root   3472 Sep  4 04:02 /var/log/auth.log.5.gz
```


Objectif : Gestion & Conservation

- Toujours avoir plusieurs endroits de stockage
- La cible favorite des pirates
- Toujours assurer le contrôle d'accès à log
- Attention cependant :
  - à la capacités des BD et aux délais de conservation !
  - à la capacité de traitement des informations !
  - Aux coûts de cryptage et à la bande passante du réseau consommée (un réseau parallèle peut être envisagé)
  - A la loi !