

## ➤ **Sommaire**

- Les comptes utilisateur et l'authentification
- Fichiers: permissions, ACLs étendues, quotas
- Exécution décalée : cron, at et les scripts d'exploitation
- Le noyau
- X-Window
- Sauvegardes et restaurations
- Les impressions
- Le réseau
- Syslog et Accounting





# Exploitation d'un système Unix

---



## Gestion des comptes et authentification

## ➤ Plan

- Identification et authentification
- UID et GID
- Hashage et signature numérique
- Les fichiers de gestion de compte
- Synoptique d'authentification
- Les autres fichiers pour l'authentification
- Les Pluggable Authentication Modules (PAM)
- SU et SUDO



## ➤ Identification

- Savoir qui est qui, pour déterminer les accès autorisés
- Notion de login ou username

## ➤ Authentification

- Prouver qui on prétend être
- Notion de secret partagé ou password



## ➤ **UID : User IDentifier**

- Identifie de manière unique un utilisateur sur le système
- Nous sommes tous des numéros
- UIDs spécifiques
  - 0 : ROOT
  - Notions d'utilisateurs spécifiques pour gérer la séparation des privilèges des daemons (lp, mail, sshd, ...)

## ➤ **GID : Group IDentifier**

- Notion de groupes d'utilisateurs
- Tout utilisateur appartient à au moins un groupe
- Un utilisateur peut appartenir à plusieurs groupes
- Un groupe est aussi un numéro
- GID spécifiques
  - Privilèges pour accéder à des ressources spécifiques (audio, floppy, ...)



# Hashage et signature numérique

- Famille des algorithmes de chiffrement
- Appelés, aussi, algorithmes de chiffrement sans clé
- Transforme un message (chaîne de caractère, fichier) en une empreinte numérique de taille fixe (un grand entier de 128 bits ou plus)
- Transformation non réversible (fonction à sens unique)
  - Impossible de retrouver le message d'origine
- Problème des collisions
  - Différents messages peuvent avoir une empreinte identique
  - S'il est possible de générer deux messages ayant la même empreinte, alors l'algorithme est « cassé »
- Exemple : md5sum
  - md5sum /bin/bash (fichier de 511 Ko) :  
603492287ea2f26b9fb9266c961d5b0c    /bin/bash



➤ **/etc/passwd (lisible par tout le monde)**

➤ *fred:x:1000:100:F. Combeau:/home/fred:/bin/bash*

➤ Champ 1 : nom de login

➤ Champ 2 : mot de passe hashé ou x si mot de passe dans shadow (\* permet d'invalider un compte)

➤ Champ 3 : UID

➤ Champ 4 : GID

➤ Champ 5 : champs informatif (suivant l'OS)

➤ Champ 6 : répertoire d'accueil

➤ Champ 7 : exécutable du shell de commande

➤ Si présent dans `/etc/shells`, l'utilisateur peut le changer

➤ Si shell spécifique, l'utilisateur ne peut pas le changer



➤ **/etc/group (lisible par tout le monde)**

➤ *users:x:100:fred*

➤ Champ 1 : nom du groupe

➤ Champ 2 : mot de passe du groupe (x si mot de passe dans gshadow)

➤ Champ 3 : GID

➤ Champ 4 : liste de noms de login qui appartiennent au groupe





➤ **/etc/shadow (lisible uniquement par le système)**

➤ *fred:<password hashé>:12129:0:99999:7:::*

➤ Champ 1 : nom de login

➤ Champ 2 : mot de passe hashé (\* : invalider un compte)

➤ Champ 3 : dernier changement

➤ Champ 4 : autorisation de changement de mot de passe

➤ Champ 5 : mot de passe doit être changé

➤ Champ 6 : avertissement utilisateur mot de passe à changer

➤ Champ 7 : compte invalidé après expiration

➤ Champ 8 : compte expiré

➤ Champ 9 : réservé

➤ Les champs 3 à 8 sont réservés à l'*expiration* des comptes

➤ Commandes : `passwd` et `chage`



➤ **/etc/gshadow (lisible uniquement par le système)**

➤ *users:\*::*

➤ Champs 1 : nom de groupe

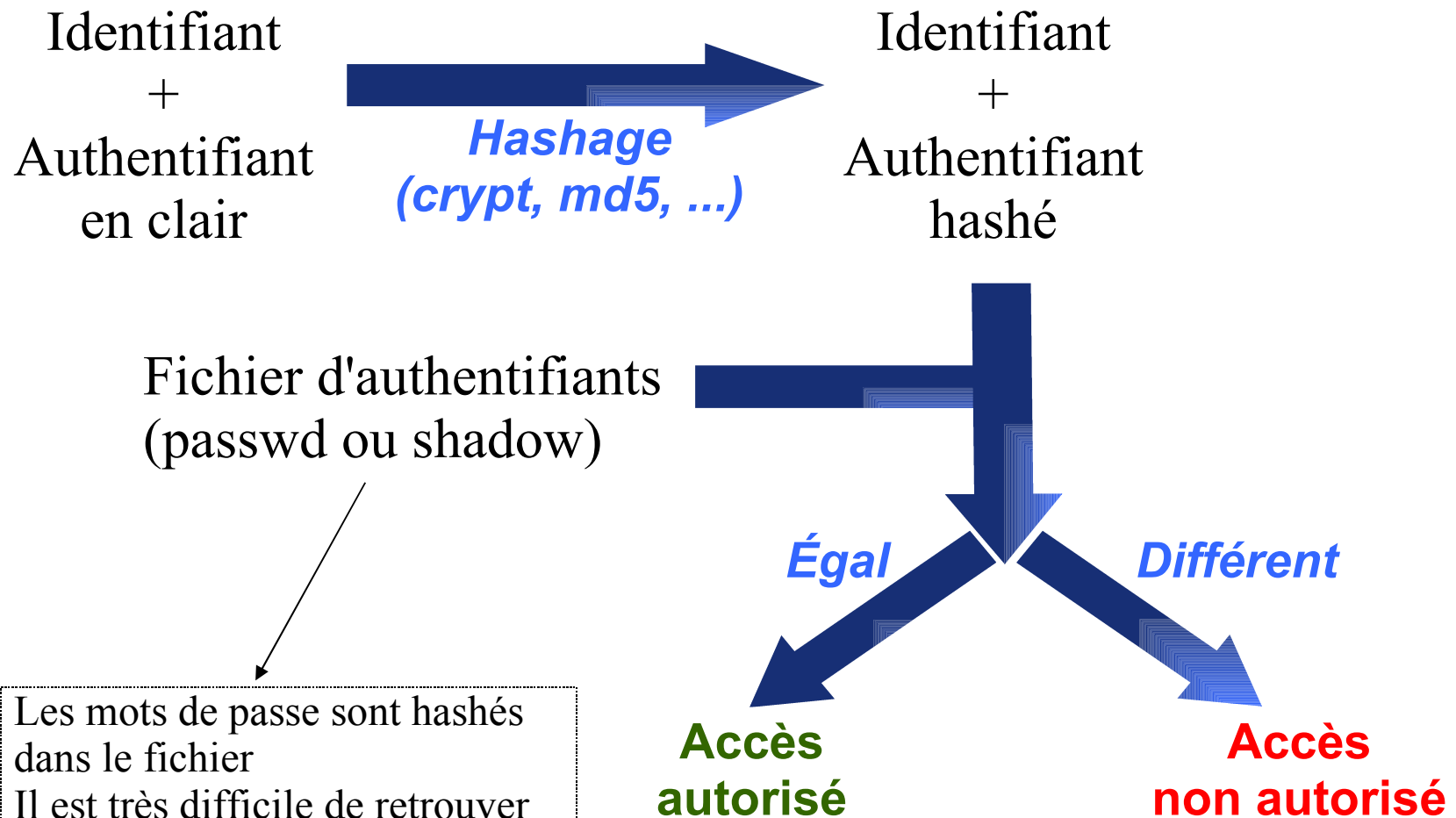
➤ Champs 2 : mot de passe hashé

➤ Champs 3 : administrateurs du groupe

➤ Champs 4 : membres du groupe



# Synoptique d'authentification



Les mots de passe sont hashés dans le fichier  
Il est très difficile de retrouver le mot de passe en clair à partir de la version hashée.

# Autres fichiers intervenant dans l'authentification

---

- **/etc/securetty**
  - Points de connexion autorisés pour root (console, port série, tty...)
- **/etc/shells**
  - Shells autorisés (chemins complets)
- **/etc/login.defs (Linux)**
- **/etc/login.conf (BSD)**
  - Configuration des exécutable de la suite login
- **/etc/nologin**
  - La présence de ce fichier interdit la connexion des utilisateurs autres que root
- **/etc/ftpusers**
  - Liste de comptes interdits de connexion en FTP



## ➤ Mécanisme d'authentification centralisé

- Les applications « PAMifiées » délèguent leur authentification
- Les librairies PAM contrôlent cette authentification
- La configuration des librairies PAM est effectuée et contrôlée par l'administrateur pour chaque application
- Deux types de configuration :
  - `/etc/pam.conf` (fichier unique)
    - Plusieurs lignes par application
  - `/etc/pam.d` (dossier)
    - Un fichier de configuration par application utilisant les PAM

## ➤ Intégration des PAM suivant la distribution

- Support des PAM
  - Redhat, Mandriva, Debian, FreeBSD
- Pas de support des PAM
  - Slackware, OpenBSD



## ➤ 4 catégories gérées

- **Authentification** : vérification de l'identité (couple identifiant/authentifiant)
- **Compte** : vérification des informations de compte (mot de passe expiré, appartenance à un groupe)
- **Mot de passe** : mis à jour du mot de passe (obliger l'utilisateur à changer de mot de passe après expiration)
- **Session** : préparation de l'utilisation du compte (tracer la connexion, monter des répertoires utilisateurs)

## ➤ 4 contrôles de réussite

- **requisite** : tous ces modules DOIVENT réussir
- **required** : au moins un de ces modules doit réussir
- **sufficient** : la réussite de ce module suffit à valider la pile de modules
  - résultat ignoré si module “required” a échoué avant
- **optional** : résultat considéré seulement si aucun autre module n'a réussi ou échoué



➤ **Modules PAM : /lib/security**

➤ pam\_unix.so, pam\_nologin.so, pam\_rootok.so...

➤ **Configuration des modules : /etc/security**

➤ group.conf, limits.conf, pam\_env.conf, time.conf...


➤ **Exemple : slackware – dropline**

% ls /etc/pam.d

|                    |         |          |              |         |
|--------------------|---------|----------|--------------|---------|
| dropline-installer | halt    | poweroff | system-auth  | xserver |
| gdm                | login   | reboot   | time-admin   |         |
| gdm-autologin      | other   | samba    | useradd      |         |
| gdmsetup           | passwd  | shadow   | xdm          |         |
| gnome-system-log   | pkgtool | su       | xscreensaver |         |




## ➤ Exemple : login



```
auth      required      /lib/security/pam_securetty.so
auth      required      /lib/security/pam_stack.so service=system-auth
  auth      required      /lib/security/pam_env.so
  auth      sufficient    /lib/security/pam_unix.so likeauth nullok
  auth      required      /lib/security/pam_deny.so
auth      required      /lib/security/pam_nologin.so
```

```
account    required      /lib/security/pam_stack.so service=system-auth
  account    required      /lib/security/pam_unix.so
```



```
password    required      /lib/security/pam_stack.so service=system-auth
  password    required      /lib/security/pam_cracklib.so retry=3
  password    sufficient    /lib/security/pam_unix.so nullok md5 shadow use_auth tok
  password    required      /lib/security/pam_deny.so
```

```
session     required      /lib/security/pam_stack.so service=system-auth
  session     required      /lib/security/pam_limits.so
  session     required      /lib/security/pam_unix.so
session     optional      /lib/security/pam_console.so
```



## ➤ pam\_unix

- Reproduit l'authentification Unix classique

- auth

  - Compare le hash du mdp fourni avec passwd ou shadow

    - nullok : mdp vide accepté

    - nodelay : supprime le délai en cas d'échec

- account

  - Vérifie le statut du compte de l'utilisateur dans shadow

- password

  - Met à jour le mot de passe de l'utilisateur

    - md5 : utiliser le hash MD5 du mdp

    - shadow : utiliser le fichier shadow

    - nullok : permet de modifier le mdp vide (sinon impossible)

- session

  - Enregistre les événements de connexion



## ➤ **pam\_cracklib**

### ➤ password

#### ➤ Vérifie la robustesse du mot de passe

- retry : nombre d'essais en cas de mdp faible
- difok : nombre minimal de caractères différents
- minlen : taille minimale du mdp

## ➤ **pam\_env**

### ➤ auth

#### ➤ Configuration des variables d'environnement

## ➤ **pam\_limits**

### ➤ session

#### ➤ Configure les limites sur l'utilisation des ressources

## ➤ **pam\_stack**

### ➤ account, auth, password, session

#### ➤ Appelle la configuration PAM d'un autre service

- service : service dont la configuration est utilisée



- **Le mécanisme OPIE (One-time Passwords In Everything) est inclus dans FreeBSD et OpenBSD**
  - Il est supporté par login, ftpd et su
  - Un module PAM le prend également en charge
- **L'algorithme S/KEY est utilisé pour générer les mots de passe à usage unique**
  - Lors de la phase de login, l'utilisateur reçoit un challenge
  - Il doit recopier ce challenge dans un calculatrice
  - Celle-ci fournit la réponse au challenge
  - L'utilisateur peut s'authentifier avec cette réponse
- **Les commandes concernant l'usage de OPIE :**
  - `opiepasswd`
    - Initialise OPIE pour un utilisateur avec le mdp fourni
  - `opiekey`
    - Calcule les réponse aux challenges OPIE
  - `opieinfo`
    - Affiche le numéro de séquence et la graine courantes dans OPIE
    - Permet de générer une liste de futures réponses OPIE



## ➤ SU

- Permet de changer le compte sous lequel on est connecté (change UID et GID)
- Configuration classique :
  - `root` peut prendre n'importe quelle identité sans mot de passe
  - Tout autre utilisateur peut prendre une autre identité en fournissant le bon mot de passe
- La configuration de SU se fait par les PAM
  - Il est facile de changer la configuration par défaut
  - Exemple : restriction de SU à un groupe particulier
- Lancé sans nom d'utilisateur, demande le passage sous `root`
- Avec `-`, demande la création de l'environnement de l'utilisateur dont on demande l'identité
  - `% su -`
  - Passage sous le compte `root` avec l'environnement de `root`



## ➤ SUDO

- Permet d'exécuter une commande sous l'identité d'un autre utilisateur
- Permet en particulier l'utilisation de commandes spécifiques sous le compte `root`
- Configuration par le fichier `/etc/sudoers`
  - utilisateur1 hôte = commande [utilisateur2]
  - Permet à l'utilisateur1 sur la machine hôte d'exécuter la commande commande en tant qu'utilisateur utilisateur2
  - Par défaut, utilisateur2 est `root`
- Exemple : `fred ALL=/sbin/shutdown -h now`
- Toute action effectuée à l'aide de la commande SUDO est enregistrée





# Exploitation d'un système Unix

---



## Gestion des fichiers

## ➤ Plan

- Permissions
- ACLs étendues
- Quotas



- A chaque fichier, sont associés un utilisateur propriétaire (UID) et un groupe propriétaire (GID)
- A chaque fichier, sont associés des permissions
- 3 types de permissions :
  - Lecture : R
  - Ecriture : W
  - Exécution : X
- **UMASK** permet de définir les droits hérités par défaut à la création des fichiers





## ➤ 3 groupes de droits qui s'appliquent :

- Au propriétaire du fichier
- Au groupe propriétaire du fichier
- Au reste du monde

➤ `-rw-r-xr--`      1 fred      users      51991 Mar 20 20:42 admin\_unix.sxi

## ➤ Droits additionels :

- Set UID : s à la place de x pour le propriétaire
- Set GID : s à la place de x pour le groupe propriétaire
- Sticky Bit : t à la place de x pour le reste du monde

➤ `-rwSr-sr--`      1 fred      users      51991 Mar 20 20:42 libre\_en\_fete.sxi



➤ **Interprétation des permissions pour les fichiers :**

- Lecture : permet de lire le contenu du fichier
- Ecriture : permet d'écrire dans le fichier
- Exécution : permet d'exécuter le fichier sous son identifiant
- Set UID : permet d'exécuter le fichier sous l'identifiant du propriétaire du fichier
- Set GID : permet d'exécuter le fichier sous l'identifiant de groupe du groupe propriétaire du fichier
- SUID et SGID sont à éviter car peuvent engendrer des problèmes de sécurité



## ➤ **Interprétation des permissions pour les dossiers :**

- Lecture : permet de lire le nom des fichiers composant un répertoire
- Ecriture : permet de créer et d'effacer des fichiers dans un répertoire
- Exécution : permet d'accéder aux informations des fichiers composant un répertoire et de s'y arrêter
- Set GID : permet de créer des fichiers dont le groupe propriétaire est celui du répertoire
- Sticky Bit : permet d'effacer uniquement les fichiers dont on est propriétaire



## ➤ Les commandes importantes

- `chown` : change le propriétaire d'un fichier (ainsi que le groupe)

- `chown toto /home/toto`

- `chown toto:users fichier`

- `chgrp` : change le groupe du fichier

- `chmod` : modifie les permissions sur un fichier

- deux façons de procéder

- spécifier les permissions

- `chmod g+rx,o-rwx fichier`

- utiliser le masque

- `chmod 755 fichier`



## ➤ Attributs de fichiers

- dépendent du système de fichiers
- meta-info influençant le comportement du système
- ext2/3 : `lsattr/chattr`

## ➤ Détails des attributs (Linux) [ASacDdlijsTtu]

- A
  - Access Time pas mis à jour (gain de performance)
- a (root only)
  - Append only : fichier ouvrable seulement en APPEND
- i (root only)
  - Immutable : fichier ne peut être modifié, supprimé, hard-linké, ...
  - (root only)
- S
  - Sync : comme le sync du mount(), mais pour des fichiers
- u
  - Undelete : sauvegarde du fichier pour récupération

## ➤ Les ACL permettent d'étendre le modèle des permissions:

- Il est possible de définir des listes d'accès, accordant des droits particuliers à certains groupes ou utilisateurs
- Toutefois, cela complique l'administration
- Tout utilisateur peut définir des ACL sur ses fichiers
- Disponible sous Linux comme patch des sources du noyau
- Commandes:
  - `getfacl`
  - `setfacl`

## ➤ Exemples

- ```
%setfacl -m u:titi:rwX fichier
%ls -l fichier
-r-xr-xr-x+  1 toto users    216112 Jan 27 09:54 fichier
%getfacl fichier
user::rw-
user:titi:rwX
group::r--
mask:rwX
other:r--
```



## ➤ Les quotas permettent de limiter l'utilisation des systèmes de fichier par les utilisateurs

- Limites sur inodes et sur blocs
- Deux types de limites :
  - Soft limit
    - Limite «souple» : permet d'accorder une période (grace period) pendant laquelle l'utilisateur peut dépasser la limite
  - Hard limit
    - Limite «dure» : l'utilisateur ne peut dépasser cette limite, le système refusera d'accorder les inodes ou blocs
- Quotas par utilisateur ou par groupe
- Les systèmes de fichier doivent être montés avec l'option quota (ou plus spécifiques usrquota et grpquota)
- Les quotas doivent être gérés par le système de fichier
  - ext(2|3) et reiserfs sous Linux



## ➤ Commandes

- `quotacheck` : met à jour les quotas, à utiliser au démarrage du système, puis périodiquement (toutes les semaines)
- `quotaon` : active les quotas pour un système de fichier donné
  - `quotaon -a` (active les quotas suivant fstab)
- `quotaoff` : désactive les quotas
- `edquota` : permet de définir les quotas par utilisateur (lance l'éditeur sur le fichier de quota)
  - `edquota -u toto` (définition du quota de l'utilisateur toto)
  - `edquota -g users` (quota pour le groupe users)
  - `edquota -t` (configuration de grace period)
- `repquota` : rapport sur les quotas







# Exploitation d'un système Unix

---



## Planification de tâches

## ➤ Plan

- Cron
- At
- Scripts d'exploitation



- **cron est un démon permettant la programmation de tâches exécutées périodiquement**
  - cron est lancé au démarrage du système
  - cron peut être configuré pour lancer des travaux chaque jour, chaque semaine, chaque mois
  - La configuration est faite par des fichiers `crontab`
  - Configuration globale : `/etc/crontab`
    - Parfois la crontab de root est utilisée
  - Chaque utilisateur peut définir des tâches périodiques
    - Emplacement des fichiers crontab :
      - Sous Linux : `/var/spool/cron/<user>`
      - Sous BSD : `/var/cron/tabs/<user>`
  - L'utilisateur utilisé pour lancer les commandes est le propriétaire du fichier, sauf pour la crontab globale



## ➤ Structure du fichier crontab :

- Commentaires avec #
- Liste de variables d'environnement
  - SHELL=<commande shell à utiliser>
  - HOME=<dossier d'exécution par défaut>
  - MAILTO=<adresse d'envoi du rapport>
- Lignes de tâches
- `minutes heures jours-du-mois mois jours-de-la-semaine tâche`
- Dans `/etc/crontab`, l'utilisateur utilisé pour lancer la tâche est spécifié avant la tâche elle-même

## ➤ Exemples de tâches programmées :

- `* * * * 1 quotacheck` (quotacheck tous les lundis)
- `* * 1 * * updatedb` (mise à jour de la base locate le 1<sup>er</sup> du mois)
- à noter : il existe des alias `@daily`, `@weekly`, `@monthly`, `@yearly`



## ➤ Commandes

### ➤ crond

- C'est le démon lui-même, il est lancé par l'un des scripts de démarrage, sans option

### ➤ crontab

- C'est le programme de manipulation des crontab, qui modifie la crontab de l'utilisateur courant
- `crontab -l` : affiche la crontab
- `crontab -r` : efface la crontab
- `crontab -e` : lance l'éditeur sur la crontab et valide le contenu en sortie



- **at permet de lancer ponctuellement des commandes, à une date et une heure données**
  - Le démon `atd` est lancé par les scripts de démarrage
  - La commande `at` permet de programmer le lancement d'une tâche (une seule fois, pas de périodicité)
    - Utilisation : `at HEURE`  
`at midnight, at noon, at 4pm`  
`at + 5 minutes, at + 2 days`
    - puis saisir la liste des commandes à effectuer
    - `atq` affiche la liste des tâches programmées
    - `at -c <numéro de tâche>` affiche le contenu d'une tâche



- **L'ensemble des tâches d'exploitation du système peut être effectué par des scripts**
  - Les scripts peuvent être programmés dans tout langage, par exemple le shell, Perl, python
  - Le lancement des scripts peut être automatisé par `cron` et `at`.
  - Exemples : vérification des comptes sans mot de passe, utilisation du disque, sauvegardes du système
  - Les scripts doivent être testés le plus possible avant d'être déployés : test sur des copies des vrais fichiers, dans une arborescence à part, tester les limites, ...
  - Les scripts ne peuvent pas tout faire, il faut parfois revenir à un langage de programmation (bien que le cas se présente rarement)





# Exploitation d'un système Unix

---



**Le noyau**



## ➤ Plan

- Fonctionnement
- Modules
- Configuration et compilation



## ➤ Le noyau est le coeur du système d'exploitation

- Il est chargé au démarrage de la machine (cf. Boot)
- Il fournit les interfaces de communication avec les périphériques matériels
- Il peut être monolithique (un seul bloc) ou modulaire
- Emplacement
  - Sous Linux : /boot/vmlinuz (compressé)
  - Sous BSD : /boot/kernel/kernel (non compressé)

## ➤ Chargement

### ➤ lilo

```
image = /boot/vmlinuz
root = /dev/hda7
label = Linux
read-only
vga = 834
initrd = /boot/initrd
```

### ➤ grub

```
title Linux
kernel (hd0,6)/boot/vmlinuz root=/dev/hda7 ro vga=834
initrd (hd0,6)/boot/initrd
```



- **Linux : le boot loader (lilo, grub) peut passer des paramètres de démarrage au noyau (*ligne de commande du noyau*)**
  - `single` : démarrage en mode single user (runlevel 1)
  - `emergency` : un shell est lancé à la place d'init
  - `ro` : système de fichier racine monté en lecture seule (recommandé)
  - `vga=xxx` : choix du mode graphique (`normal` pour le mode texte)
  - `initrd` : chargement d'un système de fichier en mémoire avant le lancement du noyau
- **BSD : le boot loader contient un shell permettant de modifier l'environnement du noyau**
  - `lsmod` : examen de l'espace mémoire (kernel + modules)
  - `load/unload` : chargement de fichiers en mémoire
  - `boot [-s]` : démarrage du noyau (mode single user)



- La commande `sysctl` permet de modifier les paramètres du noyau pendant le fonctionnement du système
  - Initialisation selon le fichier `/etc/sysctl.conf`
  - `sysctl <variable>` affiche la valeur d'une variable
  - `sysctl -a` pour voir toutes les variables (combiner avec `grep`)
  - Pour modifier une valeur :
    - `sysctl -w variable=valeur` sous Linux
    - `sysctl variable=valeur` sous \*BSD
- `/proc` contient des informations sur les processus en cours d'exécution
  - Particularité de Linux : les paramètres du système peuvent être configurés par des entrées dans `/proc`
  - Exemples :
    - `/proc/sys/net`, `/proc/modules`, `/proc/meminfo`



- Si le noyau n'a pas été compilé de façon monolithique, il est possible de charger des modules pour étendre les fonctionnalités du noyau après le démarrage du système

- Par exemple, pour les pilotes de périphériques

- Commandes :

- Sous Linux

- `lsmod` (affiche la liste des modules chargés)

- `insmod` (charge un module)

- `modprobe` (charge un module avec les modules pré-requis)

- `rmmod` (retire un module à condition qu'il ne soit plus utilisé)

- `/lib/modules/<version du noyau>` (contient les modules)

- Sous \*BSD

- `kldstat`, `kldload`, `kldunload`

- `/boot/kernel` contient les modules

- Chargement automatique possible



- **Le code source du noyau de Solaris n'est pas fourni, cependant des modules peuvent être insérés**
  - Il est impossible de recompiler le noyau Solaris
  - Ce noyau fournit des interfaces pour le chargement de modules, il est donc possible de charger de nouveaux pilotes
  - Les modules ne sont chargés que lors du démarrage, il faut donc spécifier dans le fichier de configuration `/etc/system` le nouveau module à charger puis relancer la machine
  - Le noyau et les modules sont dans `/kernel`, le noyau est `/kernel/unix`



## ➤ Requis dans certains cas

- Besoin d'une fonctionnalité d'une nouvelle version
- Application d'un patch
- Correction d'un problème de sécurité
- Pas de mise à jour constructeur
- Optimisation pour une architecture

## ➤ Précautions à prendre

- Conserver un ancien noyau fonctionnel
  - pas toujours possible
  - sauvegarder la configuration de l'ancien noyau
- Sauvegarde éventuelle des partitions
  - en cas de corruption du système de fichiers

## ➤ Opération longue

- Nombre d'options activées
- Nombre de processeurs (`make -j N`)

## ➤ Dossier Documentation



## ➤ Avant la configuration

### ➤ Enumération des périphériques matériels

➤ `lspci`

➤ `dmesg`

### ➤ Choix des systèmes de fichiers

### ➤ Déterminer les protocoles réseau nécessaires

### ➤ Installer les sources du noyau

### ➤ Appliquer les patches

## ➤ Configuration

### ➤ Copier le fichier de configuration choisi en `.config`

### ➤ `make menuconfig`

### ➤ Modulaire / Monolithique

### ➤ Important

#### ➤ Pilotes pour le disque dur et le système de fichier racine

➤ ou utiliser un `initrd`

#### ➤ `CONFIG_HIGHMEM4G` si 1Go de RAM ou plus





## ➤ Avant la compilation

- Vérifier l'espace libre : ~ 200 Mo nécessaires
- Editer `Makefile` pour certaines options spécifiques
  - Suffixe de version du noyau

## ➤ Compilation

- Noyau 2.6.xx : `make`
  - `bzImage` modules
    - compilation du noyau compressé
    - compilation des différents modules sélectionnés
  - `make O=<dir>` : fichiers produits dans *dir*
- Noyau 2.4.xx
  - `make bzImage`
  - `make modules`
  - Très verbeux



## ➤ Avant l'installation

- Monter éventuellement la partition `/boot`
- Vérifier si lilo ou grub est utilisé

## ➤ Installation

- Automatique : `make install`
  - exécute `/sbin/installkernel`
  - Comportement suivant la distribution
- A la main
  - copier le noyau : `arch/<arch>/boot/bzImage`
    - `/boot/vmlinuz-<version>`
  - copier `System.map`
    - `/boot/System.map-<version>`
  - copier `.config`
    - `/boot/config-<version>`



- **Mettre à jour le bootloader**
  - Ajouter le nouveau noyau
  - Préserver les anciennes entrées
  - Relancer s'il s'agit de lilo
- **Redémarrer la machine**
  - Sélectionner le nouveau noyau
  - Diagnostiquer les erreurs...





# Exploitation d'un système Unix

---



## X-Window System

## ➤ Plan

- Architecture client-serveur
- Modules
- Configuration



- **X11 est X Window System version 11**
- **X-Window est une architecture client-serveur, destinée à la gestion des environnements graphiques**



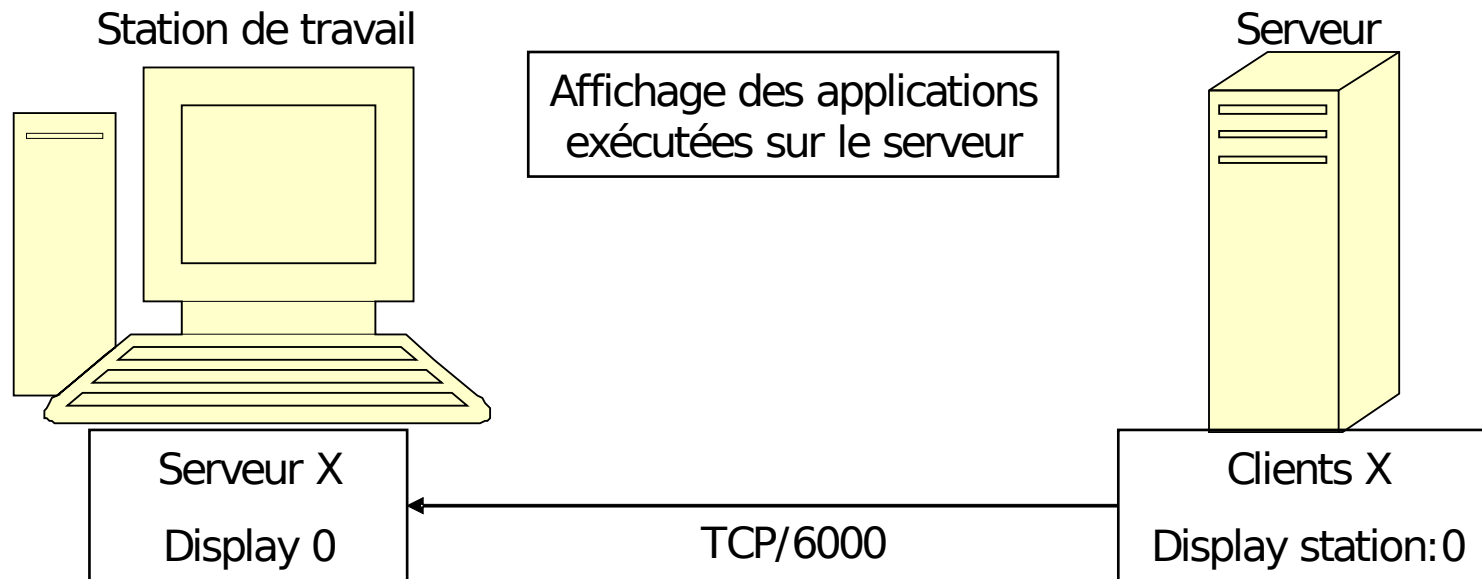
- Le serveur X effectue le rendu graphique sur une machine donnée, en utilisant le matériel (carte graphique, écran, clavier, souris...)
- Les clients X (gestionnaires de fenêtres, terminaux X...) se connectent au serveur X pour leur affichage
- La communication se fait localement par un socket UNIX, mais peut aussi être faite par TCP/IP pour un serveur distant (ports TCP 6000 et suivants)
- La variable d'environnement qui désigne le serveur X est `DISPLAY` (sur une machine autonome, `DISPLAY=:0`)

- **Implémentations libres actuelles (toujours X11R6)**

- XFree86 (version 4.4)
- X.org (version 6.9.0)
- Et maintenant X.org version 7



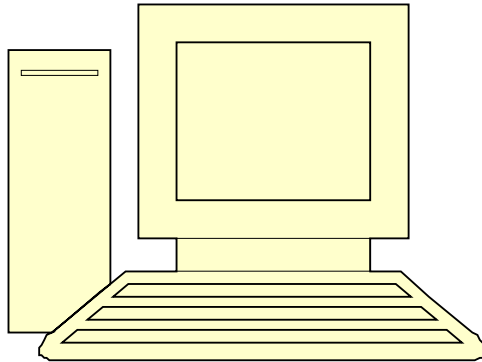
# Architecture client/serveur



# Architecture client/serveur + SSH

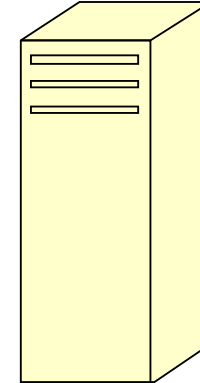


Station de travail



Affichage des applications  
exécutées sur le serveur  
avec ssh et X11-forward

Serveur



Client SSH

Serveur SSH

Tunnel SSH

Serveur X  
Display 0

Clients X  
Display station:10





## ➤ Par adresse IP

- La commande `xhost` permet d'accorder l'accès par adresse IP
  - `xhost +` : toutes les machines ont accès
  - `xhost -` : aucune machine n'a accès
  - `xhost + toto` : ajout de la machine toto en accès

## ➤ Avec une clé

- L'utilisateur obtient un cookie auprès du serveur, qui lui permet de s'y connecter
- Les cookies sont stockés dans `~/.Xauthority`
- La commande `xauth` permet de gérer les cookies



## ➤ Le serveur X a une architecture modulaire

- Les différents pilotes utilisés par X sont des modules
  - Cartes graphiques
  - Périphériques d'entrée (clavier, souris, ...)
  - Extensions (freetype, GLX, ...)
  - Situés dans `/usr/X11R6/lib/modules/`
- Les modules sont chargés au démarrage du serveur X suivant la configuration
- L'échec du chargement d'un module n'est pas forcément fatal, mais entraîne le non-fonctionnement du périphérique associé
  - Fatal s'il s'agit d'un module de carte graphique



# Configuration

- **La configuration du serveur X se fait par le fichier `/etc/X11/xorg.conf` (anciennement `/etc/X11/XF86Config[-4]`)**
  - Le fichier se compose de sections
    - Section Files : définition des ressources rgb, polices
    - Section Module : extensions à charger (GLX, extmod)
    - Sections InputDevice : définition des périphériques d'entrée et de leurs pilotes (souris, clavier, autres)
    - Sections Monitor : définition des écrans et de leurs modes de fonctionnement
    - Sections Device : définition des cartes graphiques et pilotes
    - Sections Screen : associe cartes et écrans à des résolutions de fonctionnement
    - Sections ServerLayout : associe une section Screen à des périphériques d'entrée, constitue le point d'entrée pour le démarrage du serveur X (on peut spécifier le layout sur la ligne de commande : `X -layout <layout>`)





# Exploitation d'un système Unix

---



## Sauvegarde et restauration

## ➤ Plan

- Politiques de sauvegardes
- Commandes : dump, tar, cpio, pax, dd



## ➤ Pourquoi sauvegarder?

- parce que le matériel tombera en panne (disque dur défectueux, panne générale d'un système)
- parce que les utilisateurs ne sont pas infallibles (ils effacent des fichiers par erreur)

## ➤ Quand sauvegarder?

- Il faut adapter la politique de sauvegarde à l'importance des données (station de travail, serveur de calcul, base de données)
- Il faut autant que possible sauvegarder pendant les heures creuses (nuit, week-end)
- Bien utiliser les sauvegardes incrémentales
- Par exemple :
  - Sauvegarde complète le dimanche
  - Sauvegarde incrémentale tous les soirs de la semaine
- Aucune politique n'est applicable partout, il faut toujours s'adapter à la situation
- Fil conducteur : envisager la perte de tous les disques durs pendant que vous n'êtes pas connecté sur le système



## ➤ Questions importantes :

- Quels fichiers doivent être sauvegardés ?
- Où sont ces fichiers ?
- Qui sauvegarde les fichiers ?
- Où sont faites les sauvegardes ?
- Quelle est la fréquence de modification des fichiers à sauvegarder ?
- Quel est le délai de restauration à tenir ?
- Les fichiers seront-ils restaurés sur le système où ils ont été sauvegardés ?
- Où les sauvegardes sont-elles stockées ?

## ➤ Stratégies de haute disponibilité

- La restauration du système doit respecter des délais très courts en cas de panne



- **dump est un outil permettant de sauvegarder des systèmes de fichier entier**
  - Le programme dump doit être adapté au système de fichier à sauvegarder
    - Exemple : dump sous Linux ne sauvegarde que ext2/3
  - Il permet des sauvegardes incrémentales
    - `dump -0` : garantit une sauvegarde complète
    - `dump -1` : sauvegarde les fichiers modifiés depuis la dernière sauvegarde de niveau 0
    - Attention : le niveau par défaut est 9
  - Le fichier `/etc/dumpdates` contient les dates et niveaux des dernières sauvegardes
    - `dump -W` : informations sur les sauvegardes à faire
    - `dump -u` : met à jour `dumpdates` si la sauvegarde réussit
  - Dump sous Linux peut également sauvegarder des fichiers ou dossiers particuliers, mais seulement de niveau 0 et sans mettre d'entrée dans `dumpdates` (préférer `tar/cpio`)





- **restore est le pendant de dump, il permet la restauration des fichiers sauvegardés avec dump**
  - -C : compare les fichiers sauvegardés et les fichiers actuels
  - -r : restaure le système de fichier entier (formaté au préalable)
  - -i : restauration interactive
  - -x : restauration de certains fichiers (préférer tar ou cpio)
  - -f : spécifie le fichier



- **tar (tape archiver) est un outil de création d'archives adapté à la sauvegarde d'arborescences de fichiers**
  - A l'origine, tar est fait pour copier des arborescences de fichier sur un lecteur de bande
  - Sous Linux et \*BSD, tar est aujourd'hui un utilitaire d'archivage très souple, avec possibilité de compression
  - options:
    - c : création d'archive
    - x : extraction d'archive
    - f : spécification du fichier
    - v : mode verbeux
    - z : compression gzip
    - j : compression bzip2 (plus lent mais plus efficace)
  - Anecdote :
    - tar c : écriture type BSD
    - tar -c : écriture type GNU



## ➤ Exemples

- `tar c /home` : simple sauvegarde de /home sur le périphérique de bande par défaut (/dev/rmt0)
- `tar czf projet.tar.gz projet` : sauvegarde le dossier projet dans le fichier projet.tar.gz (compressé au format gzip)
- `tar xvjf linux-2.4.24.tar.bz2 -C /usr/src` : extraction des fichiers de l'archive linux-2.4.24 (compressée bzip2) dans le dossier /usr/src, en mode verbeux (les fichiers extraits sont écrits au fur et à mesure sur le terminal)
- `tar xj -p -f sauvegarde.tar.bz2` : extraction de l'archive sauvegarde avec restauration des permissions
- `tar x <fichier à restaurer>` : restauration d'un fichier précis à partir de la bande



- **cpio est comparable à tar, mais il est conçu pour sauvegarder des listes de fichier plutôt que des arborescences**
  - cpio est habituellement combiné avec `find`
    - Exemple : `find /chemin | cpio`
  - cpio peut utiliser de nombreux formats d'archive, dont `tar`
  - cpio ne gère pas les formats compressés, mais il peut être combiné avec `zcat` ou `bzcat`
  - Options :
    - `cpio -o` : création d'archive
    - `cpio -i` : restauration
    - `-F` : spécification de l'archive
    - `-H` : format d'archive (tar par défaut)



- **pax est un hybride de tar et cpio, il permet de sauvegarder indifféremment des fichiers ou des arborescences**
  - `pax` : sans argument, donne la liste des fichiers dans l'archive
  - `-r` : extrait des fichiers d'une archive
  - `-w` : crée une archive
  - `-f` : spécifie le fichier



- **dd est un outil de copie de bas niveau, il copie des blocs de données (disc dump)**
  - On utilise dd pour faire des copie bas niveau de système de fichiers
    - `dd if=/dev/hda of=fichier_de_sauvegarde`
    - Cette commande extrait l'image brute du disque hda (bloc par bloc) vers un fichier de sauvegarde





# Exploitation d'un système Unix

---



## Gestion des impressions

## ➤ Plan

### ➤ Les différents systèmes

➤ lpd

➤ cups





## ➤ lpd est le système d'impression \*BSD

- Il repose sur le démon d'impression `lpd`, et le fichier de configuration `/etc/printcap`
- Le fichier `/etc/printcap` définit pour chaque imprimante comment on y accède (port parallèle, réseau...), les filtres à appliquer pour les travaux d'impression
- Le démon `lpd` est responsable du *spooling* et de l'envoi des travaux à l'imprimante
- Les commandes accessibles aux utilisateurs :
  - `lpr` : envoie un travail d'impression au démon
  - `lpq` : affiche la file d'attente des travaux d'impression
  - `lprm` : permet d'annuler un travail
- L'implémentation utilisée actuellement est **lpr-ng**



## ➤ **Cups hérite du système d'impression de System V**

- cups définit des classes d'imprimantes
- Ensuite, on peut attacher des imprimantes à des classes, avec un emplacement précis (port parallèle, réseau...)
- Les fichiers de configuration sont dans `/etc/cups`
  - Le fichier `printers.conf` contient la définition des imprimantes avec leur emplacement
  - Le fichier `classes.conf` contient la définition des classes avec les imprimantes qui appartiennent à ces classes
- Commandes utilisateur :
  - `lp` : envoie un travail d'impression
  - `lpstat` : informations sur les files d'attente des imprimantes
  - `cancel` : annule un travail d'impression



## ➤ Plan

- TCP/IP

- DNS





# Exploitation d'un système Unix

---



## Configuration du réseau

## ➤ Configuration des interfaces TCP/IP

- `ifconfig <interface> <adresse IP> netmask <masque réseau>`
- La passerelle par défaut est configurée par les commande de routage
  - Linux : `route add default gw <adresse IP>`

## ➤ Listing des services réseau

- `netstat`
  - `-a` : liste les sockets non connectés (en écoute)
  - `-t` : TCP
  - `-u` : UDP
  - `-p` : donne le PID associé au socket



## ➤ La configuration DNS est importante pour le bon fonctionnement d'un système UNIX

### ➤ Fichier `/etc/host.conf`

➤ définit l'ordre de recherche des noms DNS

➤ `order hosts, bind`

➤ signifie que l'on regarde d'abord dans le fichier `/etc/hosts` puis ensuite on interroge le serveur DNS

### ➤ Fichier `/etc/resolv.conf`

➤ Spécifie l'adresse IP des serveurs DNS

➤ `nameserver <adresse IP>`

➤ Spécifie les domaines de recherche de noms DNS

➤ `search <domaine>`





# Exploitation d'un système Unix

---



## Les journaux d'évènements système : syslog et accounting

## ➤ Plan

- Syslog
- Logrotate
- Les outils d'accounting





- **syslog reçoit l'ensemble des logs du système et les répartit dans différents fichiers suivant sa configuration**
  - Le démon `syslogd` est lancé au démarrage
  - Deux façons de recevoir les logs :
    - `/dev/log` : socket unix
    - 514/UDP : pour le réseau
  - Chaque message possède une *facility* et une *priority* :
    - *facility* indique le type de log (kernel, daemon, ...)
    - *priority* indique la gravité du message (debug, emerg, ...)
  - Le fichier `/etc/syslog.conf` définit la configuration de syslog
    - Par *facility* et/ou par *priority*
    - Possibilité de diriger les logs dans des fichiers ou de les transmettre à d'autres programmes



# Logrotate

- **Logrotate contrôle la taille des fichiers de log et les sauvegarde au besoin**

- **Lancement périodique par cron**

- **Dans la crontab de root**

```
40 4 * * * /usr/bin/run-parts /etc/cron.daily 1> /dev/null
```

- **Dans le fichier /etc/cron.daily/logrotate**

```
#!/bin/sh
```

```
/usr/sbin/logrotate /etc/logrotate.conf
```

- **Configuration de logrotate : /etc/logrotate.conf**

```
/var/log/wtmp {  
    monthly  
    create 0664 root utmp  
    rotate 1  
}  
  
/var/log/syslog /var/log/messages {  
    sharedscripts  
    postrotate  
        /bin/pkill -HUP syslogd  
    endsript  
}
```



- **Les système de comptabilité (ou *accounting*) permettent de surveiller l'utilisation des ressources par les utilisateurs à des fins de statistiques ou de facturation (parfois le temps CPU n'est pas donné)**
  - La comptabilité classique sous \*BSD stocke ses informations dans `/var/adm` (sous Linux `/var/log/*acct`)
  - Les commandes permettant l'extraction des informations :
    - `accton` : activer ou désactiver l'accounting
    - `sa` : informations sur l'utilisation CPU et mémoire
    - `ac` : temps de connexion des utilisateurs
    - `lastcomm` : dernières commandes lancées
    - `acctail` : surveillance interactive de l'accounting
  - **Linux : options `CONFIG_BSD_PROCESS_ACCT` et `CONFIG_BSD_PROCESS_ACCT_V3`**





# Exploitation d'un système Unix

---



**Tuning :  
Systèmes de Fichiers,  
Processus et  
Mémoire Virtuelle**

## ➤ **Qu'est-ce que le tuning ?**

- C'est, le plus souvent, pouvoir répondre à la question :
  - « Pourquoi le système est-il si lent ? »
- En fait, c'est étudier l'affectation des ressources d'un système ...
- ... dans le but, de l'optimiser
  
- Le tuning consiste, donc, à paramétrer au mieux un système pour pouvoir l'utiliser au mieux de ses performances
- Un autre aspect du tuning est l'achat de matériels ou la mise à niveau matériel pour améliorer les performances

## ➤ **Les performances du système dépendent de l'affectation des ressources**

## ➤ **Les ressources intervenant dans les performances :**

- CPU
- Mémoire
- Entrées/Sorties Disques
- Réseau et Périphériques

*Trouver la bonne fréquence*

- **Problème de performance = insuffisance de ressources**
- **Une insuffisance de ressources ne peut se régler que de 2 façons :**
  - Ajouter des ressources (achat ou MAJ de matériels)
  - Rationner les ressources (tuner le système)
- **Les mécanismes de contrôles des ressources système**
  - CPU
    - Gestion des priorités
    - Traitement par lot et files d'attente
    - Ordonnancement des processus
  - Mémoire
    - Architecture du swap
    - Limitation des ressources utilisées
    - Paramètres de la gestion de la mémoire
  - Entrées/Sorties Disques
    - Architecture du système de fichiers (disques, contrôleurs, ...)
    - Placement des fichiers sur les disques
    - Paramètres des Entrées/Sorties



➤ **Une bonne approche pour gérer des problèmes de performances :**

- Poser le problème de manière la plus détaillée
- Déterminer la ou les causes du problème
- Formuler les améliorations des performances à atteindre
- Concevoir et implémenter les modifications au niveau du système et des applications
- Surveiller le système pour vérifier si les modifications ont fonctionné
- Et recommencer

➤ **Dans tout problème de performances**

- Il faut savoir ce qu'il se passe sur son système, en temps normal
- Nécessite de surveiller le système ...
- ... et de savoir quand le système dévie ou se comporte anormalement
- Permet, également, d'avoir un historique du comportement du système

➤ **Pour chaque ressource, voyons comment la surveiller**



## ➤ Surveiller le CPU

- C'est, en fait, surveiller les processus
- *uptime* donne des valeurs moyennes de charge
- Charge : nombre moyen de processus actifs
  - Si  $> 3$ , souvent problématique sur un système interactif
- *ps* donne la liste des processus du système et le pourcentage de CPU qu'ils prennent
- *top* permet d'avoir la liste des processus en temps réel, ainsi que des informations importantes sur la charge du système
- *pstree* permet de créer l'arborescence des processus (en intégrant les liens de parentés)
- *vmstat* donne des informations sur l'utilisation CPU (mode utilisateur, mode noyau et idle)
- *ps* peut montrer les threads noyau, qu'il ne faut pas confondre avec des processus
- Sur les systèmes multi-threadés, il ne faut pas confondre processus et threads (un thread est un composant d'un processus et le scheduler système ne voit que les processus)





## ➤ Surveiller le CPU

### ➤ Surveiller les entrées/sorties des processus

- Quels sont les fichiers ouverts par les processus ?
  - répertoires, fichiers, fichiers temporaires, exécutables, bibliothèques, devices, sockets réseau, pipes, ...
- *lsuf* ou *fuser* sont des commandes Linux permettant de voir quels fichiers sont ouverts par un processus
- *fstat* est l'équivalent sous FreeBSD et OpenBSD

### ➤ Surveiller les appels système d'un processus

- Permet une surveillance très fine (trop fine !)
- Une bonne connaissance du fonctionnement de l'OS est indispensable pour interpréter les informations recueillies
- Très utile pour déboguer dans, presque, toutes les situations
- *strace* est la commande Linux et FreeBSD de visualisation des appels système
- *ktrace* est l'équivalent pour OpenBSD



## ➤ Surveiller le CPU

### ➤ On peut limiter les ressources d'un processus

- Temps CPU utilisé
- Taille maximale du segment de données
- Taille maximale de segment de pile
- Taille maximale du fichier core
- Quantité de mémoire virtuelle utilisée

### ➤ Toutes ces limitations peuvent être définies par les commandes *limit* ou *ulimit*

### ➤ Mais ces limitations ne s'appliquent qu'aux processus

- Pas de notion d'une connexion utilisateurs
- Pas de notion de limitations par utilisateur
- Donc peu utilisable



- Lors de problèmes de performance CPU, 3 solutions peuvent être utilisées :

- Gérer les priorités des processus

- Chaque processus possède deux priorités
  - Priorité de base (définie à la création du processus)
  - Priorité courante (calculée pour allouer le CPU)
- Commandes *nice* ou *renice* (modifier la priorité de base)
- Un processus peut être arrêté ou suspendu en lui envoyant un signal : commandes *kill* et *killall*
- Certains processus ont du mal à mourir
  - Zombis
  - Attente de ressources réseau (type NFS)
  - Attente de ressources E/S (disque ou bandes)



- Lors de problèmes de performance CPU, 3 solutions peuvent être utilisées :

- Déplacer les tâches consommatrices de CPU vers d'autres systèmes ou les exécuter lorsque le système est moins chargée (batch)
  - Utilisation de *cron* (exécution périodique)
  - Utilisation de *at* ou *batch* (exécution différée)
- Modifier les paramètres d'ordonnancement pour privilégier certains processus
  - Très compliqué
  - Et surtout très risqué pour la stabilité du système
  - A faire, si l'on sait EXACTEMENT ce que l'on fait



## ➤ Surveiller la mémoire

- Les Unix utilisent des mécanismes de pagination (gestion de l'espace mémoire d'un processus)
- Ne pas confondre :
  - Pagination (allocation d'une unité de mémoire virtuelle)
  - Swap (transfert d'un processus entier de/vers la zone de stockage secondaire)
  - Trashing (système ne possédant plus assez de mémoire virtuelle pour fonctionner)
- *vmstat* permet de surveiller l'utilisation de la mémoire
- *ps* donne la liste des processus avec le pourcentage de mémoire utilisée
- *top* donne la liste des processus en temps réel avec le pourcentage de mémoire utilisée
- *free* (sous linux) donne un bon aperçu de l'utilisation globale de la mémoire



## ➤ Surveiller la mémoire

- L'espace de pagination (swap) est important
- Déterminer la taille adéquate pour le swap
  - Difficile
  - Dépend de ce que fait le système
  - Dépend de la configuration globale matérielle du système
- L'espace de pagination peut être
  - Une partition dédiée (à préférer pour de meilleure performance)
  - Un fichier
- Penser à bien architecturer l'espace de pagination
  - Répartir sur plusieurs disques
  - Ne pas créer de goulots d'étranglement
- Des priorités peuvent être définies pour assurer un ordre séquentielle d'utilisation d'un espace de pagination



## ➤ Architecture des E/S disques

- Pour optimiser les E/S, utiliser l'arborescence UNIX
  - / et /usr sont utilisés en parallèle
    - Il est judicieux de placer les deux partitions sur deux disques différents sur des contrôleurs différents
  - /var est plutôt utilisé à la fois en lecture et en écriture
    - Dans le cas d'un serveur, on pourra placer /var sur un disque dédié
- L'utilisation de RAID (même logiciel) peut améliorer les performances
- La fragmentation des fichiers diminue les performances
- L'utilisation de types de systèmes de fichiers peut améliorer les performances
  - Ext2 est très rapide mais peu robuste en cas de crash
  - Ext3 est très robuste mais moins rapide que ext2
  - Reiserfs est moins rapide que ext3
  - Le paramétrage du système de fichiers peut avoir un impact (taille des clusters, taille réservée à root)



## ➤ Gérer l'espace disque

- Les commandes *du* et *df* permettent de surveiller l'espace disque utilisé
  - *df* donne des informations sur les systèmes de fichiers
  - *du* donne la taille d'un répertoire sur disque
  - *quot* permet d'avoir la consommation d'espace disque par utilisateur
- Définir les fichiers inutiles (scripts automatiques)
- Définir une politique pour les fichiers inutiles
- Compresser les fichiers peut utilisés
- Convaincre les utilisateurs de faire le ménage
- Proposer des outils d'archivage de fichiers transparents pour l'utilisateur
- Définir une politique pour les fichiers non accédés depuis un certain temps :
  - Archivage (disque, CD-ROM, bandes)
  - Récupération (transparente si possible)





## ➤ Gérer l'espace disque

### ➤ Limiter la taille des fichiers de log

- Utilisation de logrotate
- Limitation des fichiers core
- Surveiller `/var` et le purger le cas échéant

### ➤ L'utilisation des quotas

- Permet de limiter l'espace utilisé pour un utilisateur
- Permet de limiter l'espace utilisé pour un groupe d'utilisateur





# Exploitation d'un système Unix

---



## Bibliographie

## ➤ Les deux Bibles :

- Les bases de l'administration système, 3<sup>ème</sup> Edition
  - Elen Frich – O'Reilly – 2841772225
- Unix System Administration Handbook, 3<sup>rd</sup> Edition
  - Evi Nemeth, Scott Seebass, Garth Snyder
  - Prentice Hall PTR – 0130206016

## ➤ Le système Linux :

- Le système Linux, 4<sup>ème</sup> Edition
  - Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson et Lar Kaufman
  - O'Reilly – 2-84177-241-1
- Administration réseau sous Linux, 2<sup>ème</sup> Edition
  - Olaf Kirch et Terry Dawson – O'Reilly – 2-84177-125-3
- Linux Administration Handbook
  - Evi Nemeth, Garth Snyder, Adam Boggs
  - Prentice Hall – 0130084662

## ➤ Les BSDs :

- Absolute Openbsd: Unix for the Practical Paranoid
  - Michael W. Lucas - No Starch Press – 1886411999
- FreeBSD Unleashed
  - Michael Urban, Brian Tiemann - Sams Publishing – 0672322064

## ➤ Les Basics :

- Conception du système Unix
  - Maurice J. Bach - Dunod – 2225815968
- Conception et implémentation du système 4.4 BSD
  - Stephen L. Nelson – Addison Wesley - 284180142X

## ➤ Références Internet :

- [www.ugu.com](http://www.ugu.com) : Unix Guru Universe (infos pour l'administration)
- [www.sun.com/bigadmin/docs](http://www.sun.com/bigadmin/docs) : Docs online sur l'administration Solaris
- [www.sysadminmag.com](http://www.sysadminmag.com) : Articles sur l'administration Unix
- [www.tldp.org](http://www.tldp.org) : Linux Documentation Project