## Généralités

- rôles d'un système d'exploitation (OS ou opérating System)
  - machine virtuelle qui cache la machine physique
  - gestion des ressources matérielles (mémoire, périphériques, ...)
- Unix
  - multi tâches et multi utilisateurs
  - axiome: « tout est fichier »
  - interface standard: POSIX
  - unix: plusieurs famille d'OS : BSD, SYS V, Linux, ...
    - cf http://fr.wikipedia.org/wiki/Tableau_synoptique_des_syst%C3%A8mes_d'exploitation

## Rôle d'un administrateur système

- ajouts et suppressions d'utilisateurs
- ajout, suppression, configuration de matériel
- sauvegarde et restauration
- installation, mise à jour de logiciels
- surveillance du système:
  - sécurité
  - monitoring
- documentation locale
- rédaction de fiches de procédures, cahier des charges, ...
- aide aux utilisateurs

## Principes de base de l'administration

- tout système doit avoir un administrateur
- complexité :
  - de plus en plus de machines
  - des systèmes hétérogènes
- méthodes de travail
  - rigueur (doc, gestion de version, validation, ...)
  - automatiser les tâches qui peuvent l'être
  - documentation: fiches de procédure, cahier des charges, ...
  - pas de travail de fond: le vendredi soir, après un pot :-)

## Droits d'accès aux fichiers et aux processus

- fichiers: possède un propriétaire (UID) et un groupe propriétaire (GID)
- le propriétaire seul à pouvoir modifier les permissions du fichiers
- processus:
  - UID et GID réel: ceux de l'utilisateur du programme ( et pas ceux du propriétaire du fichier sur disque)
  - UID et GID effectifs: pour déterminer les droits d'accès
- bits SETUID ou SETGIUD

## Root

- root (superutilisateur): tout compte d'UID 0
- peut exécuter toute opération **valide** sur n'importe quel fichier ou processus
- devenir root
  - connexion directe en tant que root: déconseillé, notamment via réseau
  - su : changement d'identité
  - sudo: exécuter certaines commande en tant que root
- il est important de journaliser les actions effectuées en tant que root (sudo, snoopylogger, ...)

## les autres pseudo-utilisateurs

- daemon : propriétaire des logiciels systèmes (uid 1)
- nobody: utilisateur sans droit
  - utilisé par nfs notamment
  - utilisé par certains daemon
  - ne doit possèder aucun fichier
- squid, bind, ...: pratique actuelle

## méthodes d'administration

- installation de logiciels
  - des packages binaires via l'un des outils de gestion de packages du système d'exploitation :
  - des logiciels livrés en source à recompiler
  - des logiciels non livrés avec le système d'exploitation
- configuration:
  - via l'édition de fichier de configuration (=> il faut maîtriser un outil d'édition de texte par plateforme)
  - via des commandes d'administration
  - utilisation d'outils intégrés (smit (AIX), sam (HP UX), webmin, admintool (solaris)
  - via des scripts ou des extension d'outil (webmin) maison

## Documentation

- man: les pages de manuel d'unix, sections du manuel
  - man commande
  - exemples :
    - man 1 kill: kill commande utilisateur (section 1 du man)
    - man 2 kill: kill appel système (section 2 du man)
    - apropos ou man -k
- documentation du système d'exploitation, /usr/doc, /usr/share/doc, ...
- WeB (google est votre ami notamment pour les messages d'erreurs)
- groupe de news USENET, forums WeB
  - fr.comp.os.* par exemple

## le démarrage des PC

- MBR:
  - premier bloc de données du periph d'amorçage
  - 446 octets pour le gestionnaire d'amorçage
  - 64 octets pour la table des partitions (16 par partitions)
  - 2 octets à AA55 (valeur fixe)
- PBR: partition boot loader: en début de partition, chargé par le chargeur d'amorçage du MBR ou un autre chargeur d'amorçage
- MBR microsoft : charge le pbr de la première partition principale active
- gestionnaire d'amorçage: lilo, grub & Co

## démarrage du système

- chargement et initialisation du noyau
- détection des périphériques
- création des processus systèmes, montage de la partition racine /en lecture seule, création du processus init
- intervention de l'opérateur (en cas de démarrage en mode monoutilisateur)
- exécution des scripts de démarrage
- passage en mode multiutilisateur

## init

- cf man telinit
- Rôle de init
  - exécution ds scripts d'initialisation
  - gestion des terminaux
  - ancêtre de tous les processus
- 2 types de démarrage :
  - BSD
  - SYSV

## mode monoutilisateur

- init lance un processus qui lance un shell root
- le processus de démarrage reprend après
- ce modecorrespond aux runlevel 1 ou S (le sens dépend de la distribution de l'OS)
  - S: réparation des systèmes de fichiers ou ~: pas de daemon, racine en lecture seule est la seule partition montée
    - il faut la remonter en lecture/écriture et monter les autres partitions si nécessaire
  - 1: administration, certains daemon sont actifs, les partitions sont toutes montées (Sys V, Suse, ...ràf: à préciser sous debian)

## états du système

- BSD: monoutilisateur ou multiutilisateur
- Sys V, Linux:
  - plusieurs états (Run Level)
  - sens (dépend du système, voire de la distribution) :
    - 0: arrêt de la machine (halt)
    - 1: monoutilisateur (administration)
    - 2 à 5 : multiutilisateur
    - 6: redémarrage (reboot)
    - s ou S: monoutilisateur (maintenance)
  - le « run level » par défaut est défini dans /etc/inittab
  - connaître son « Run level »: who -r, runlevel (linux)
  - changer de « run level »: telinit No

## scripts de démarrage BSD

- varient selon le système BSD
- l'idée de base: deux scripts (/etc/rc et /etc/rc.local) pilotent le démarrages du sytème
- Scripts de démarrage FreeBSD et autres BSD libres (ràf: dans une version ultérieure de ce document)

## scripts de démarrage Sys V et Linux

- le système le plus répandu
- /etc/inittab:
  - indique à init ce qu'il faut faire pour chaque niveau d'exécution
  - au démarrage: init passe par tous les niveaux entre 0 et le niveau d'exécution par défaut;
  - idem dans l'autre sens à l'arrêt
  - les scripts de démarrages situées dans /etc/rcN.d sont démarrés via inittab
  - en pratique: de nos jours, on ne modifie que rarement inittab, on s'appuie sur les scripts (cf ci-dessus)

## inittab

- Demo:
  - syntaxe du fichier
  - man inittab

## Scripts de démarrage Sys V (suite)

- /etc/init.d ou /sbin/init.d (HP-UX):
  - contient la copie d'origine des scripts de démarrage
  - un script par daemon ou aspect du système
  - les scripts comprennent les arguments start et stop (et parfois d'autres arguments comme restart)
- /etc/rcN.d (N=0, 1, 2, ... : run level)
  - contient un lien symbolique vers le scripts d'origine. Le nom du lien peut avoir deux formes:
    - KXX: le service doit être arreté dans ce run level
    - SXX: le service doit être démarré dans ce « run level »
  - outils de gestion des scripts de démarrage : update-rc.d (linux debian)

## Scripts de démarrage Sys V (suite)

- processus d'appel des scripts par init
  - recherche de /etc/rcN.d
  - si init augmente son niveau d'exécution: execution des scripts commençant par S dans l'ordre des No
  - si init diminue son niveau d'exécution: execution des scripts commençant par K dans l'ordre des No
- les scripts peuvent être utilisés pour arrêter ou redémarrer manuelle un service
  - exemple: /etc/inid.d/networking restart
- DEMO: exemple d'ajout de script avec update-rc.d

## arrêt du système

- init est à la source de tout
  - arrêt du système: demander à init de tuer proprement ses processus fils
  - shutdown [-r|-h] heure message : pour arrêter ou rebooter le système à l'heure indiqué
  - reboot: équivaut à shutdown -r ou à telinit 6
  - halt: équivaut à shutdown -h ou à telinit 0

## processus d'arrêt du système

- écriture de l'évènement dans les journaux
- exécute les scripts KXX (init SYSV)
- init tue ses processus fils
- vide les tampons disque (sync)
- démonte les systèmes de fichiers  quand les écritures sont terminées
- arrête la machine