



Principes de l'installation et de la configuration d'un système

BUT Informatique 1ère année

Présentation

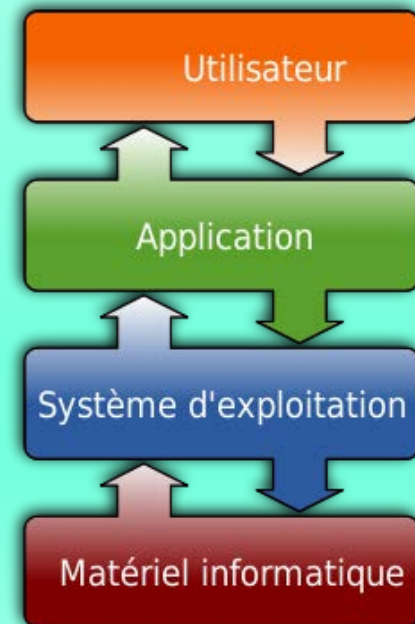
- TD 2 heures:
 - Introduction:
 - Théorie sur les SE : Noyau, FileSystem
 - Séquence de démarrage
 - BIOS et UEFI : POST, CMOS
 - Pilotes, Fichiers systèmes
 - Partitionnement
 - Virtualisation
 - Applications lors de TP:
 - Installation Windows
 - Installation Linux

SE : à quoi ça sert ?

- Faciliter
 - charger des programmes
 - accéder aux périphériques
 - enchaîner des tâches
 - gérer du stockage
 - partager des ressources
- Contrôler
 - maximiser le taux d'utilisation matériel
 - protéger le matériel et les données
 - distribuer équitablement les ressources
 - assurer la confidentialité des programmes et données

Système d'exploitation

- Ensemble d'instructions et de programmes préparés à l'avance constituant un système complexe permettant d'exploiter les capacités du matériel
- Interface entre l'utilisateur et le matériel pour l'exécution de programmes



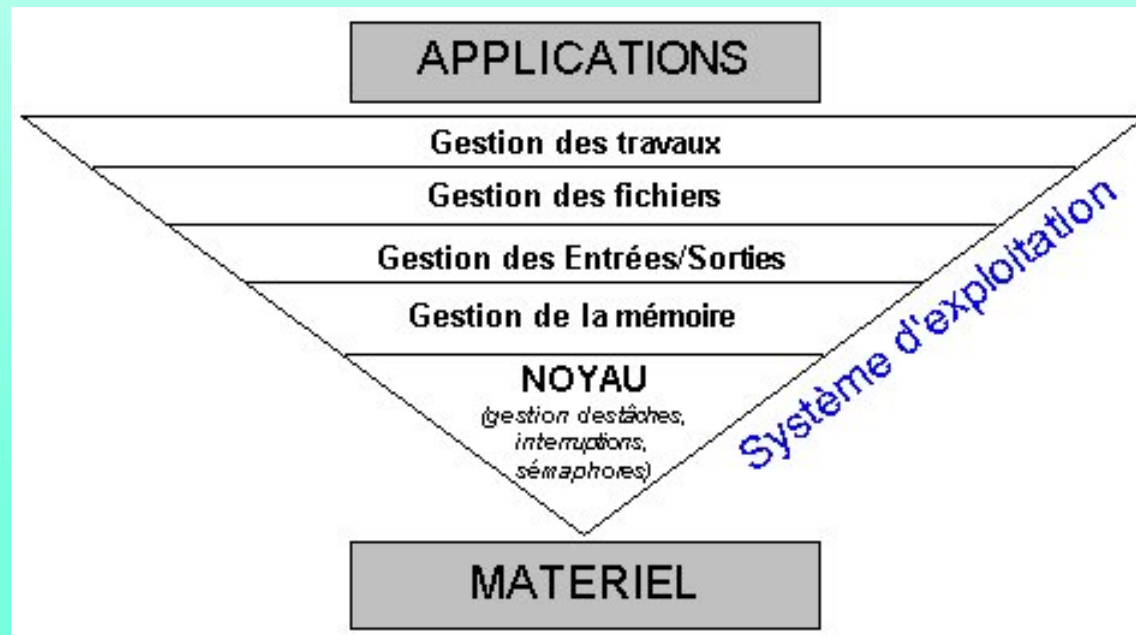
Fonctionnalités

- Suite de services facilitant l'utilisation d'applications:
 - Utilisation des périphériques :
 - permet à l'utilisateur de manipuler le périphérique par de simples demandes de lecture ou d'écriture
 - pas de perte de temps à traduire les opérations en instructions propres au périphérique
 - Accès aux fichiers :
 - tient compte du format propre à chaque support servant au stockage de fichiers
 - offre également des mécanismes de protection permettant de contrôler quel utilisateur peut manipuler quel fichier⁵

Systeme d'exploitation

Fonctions du système d'exploitation

Les principales fonctions du système d'exploitation peuvent être classées hiérarchiquement :



Classifications des systèmes

- **Par la configuration matérielle:**
 - Type de processeur
 - Largeur du registre interne du processeur(32, 64 bits)
 - Nombre de processeurs
- **Par le nombre d'utilisateurs simultanés :**
 - système mono-utilisateur ou multi-utilisateur
- **Système monoposte ou distribué**

Exemples de SE

Dos: Mono-tâche (traitement par lots), Mono-utilisateur

Windows 95, 98, Millenium: Multitâche, Mono-utilisateur

**Windows NT4, 2000, XP, Vista, Windows7, Windows8,
Windows 10, Windows 11 :** Multitâche, Multi-utilisateur

Unix, Linux: Multitâche, Multi-utilisateur

MacOs: anciennes version Mono-utilisateur

MacOS X: Multitâche, Multi-utilisateur

Les qualités d'un système

- La fiabilité :

Limiter les conséquences:

des défaillances matérielles

des erreurs des utilisateurs

En cas de panne, éviter les pertes d'information ou leur incohérence

- Efficacité :

optimiser l'utilisation des ressources:

sans en consommer trop pour lui-même

- Facilité d'emploi :

Offrir un langage de commande (dialogue usager/système) et des diagnostics d'erreurs (système/usager) clairs et précis

Avoir une interface graphique ergonomique et intuitive

Les qualités d'un système

- Adaptabilité :

Permettre des modifications matérielles et logicielles les plus simples possibles, à l'aide d'outils spécialisés

- Mesurabilité :

Enregistrer la comptabilité des ressources utilisées par les usagers, mesurer les paramètres de fonctionnement et de charge

Structure d'un système d'exploitation

- Typiquement, un SE est composé de :

Un noyau assurant les fonctionnalités suivantes :

- pilotes de périphériques
- gestion des processus
- gestion de la mémoire (il la distribue aux processus),
- ordonnancement des processus (répartition du temps processeur)

Un système de fichiers:

Méthode d'organisation des données persistantes sur un médium durable (disque dur, clef USB ...)

Le noyau (kernel en anglais):

- Gère la répartition du temps CPU entre les différents processus
- Gère la priorité des processus
- Sauvegarder l'état de la machine lorsque le processus s'interrompt
- Assure la gestion des interruptions:
 - déterminer la source de l'interruption
 - activer la procédure correspondante

Le système de fichiers

Le système de fichiers (*filesystem* en anglais):

- offre à l'utilisateur une vue abstraite sur ses données:
 - Arborescence fichiers répertoire
 - Unité de stockage fichier (octets)
- Masque l'organisation physique sous-jacente du médium utilisé:
 - Blocs, mécanismes d'entrée/sortie

Le système de fichiers

- gestion des droits d'accès aux fichiers
- compression et le chiffrement des données
- journalisation des écritures:
robustesse, en cas de crash système

filesystem,

SE(journalisés, associés et compatibles)

- FAT (DOS / Windows, Linux, BSD)
- NTFS (Windows, Linux « NTFS3-G », Mac OS10 « NTFS3-G »,)
- ext2, ext3, ext4 (Linux, BSD)
- HFS, HFS+, APFS (Mac OS, Mac OS X)
- ReiserFS (Linux)
- Resier4 (Linux expérimental)
- JFS (Linux, AIX, OS/2)
- UFS, UFS+ (BSD, Linux en lecture seule)
- FFS (BSD)
- S5 (Unix System V)
- XFS (Irix, Linux)
- HPFS (OS/2)
- ODS (VMS, OpenVMS)
- NSS : *Novell Storage Services* (Netware et Suse Linux)

Séquence de démarrage (boot)

- L'objectif de la séquence de démarrage est de démarrer le SE
- Où est-il situé?
 - Sur le disque dur
- Peut-on l'exécuter s'il est sur le disque dur?
 - Non, le disque dur est un périphérique de stockage lent
- Donc, il nous faut le transférer dans la mémoire principale (RAM)

Séquence de démarrage (boot)

- Problématique:
 - La RAM n'est pas initialisée au démarrage (elle perd son contenu lorsqu'on coupe l'alimentation)?
 - Utiliser une ROM qui contient un petit programme qui sera exécuté au démarrage
 - Comment savoir où le SE est situé sur le disque dur?
 - Un programme «spécial» (nommé le «bootloader») est toujours situé au même endroit sur le disque
 - Que faire s'il y a plusieurs SE sur le disque dur (multiboot)
 - C'est le «bootloader» qui demandera à l'utilisateur de faire un choix

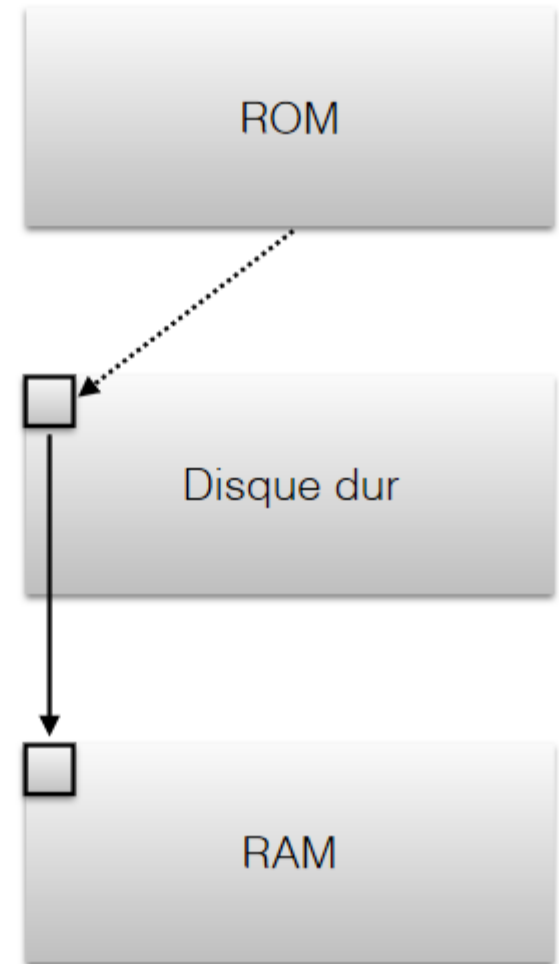
Séquence de démarrage simplifiée

1. Exécution du BIOS (stocké dans la ROM)

2. Le BIOS trouve l'emplacement du système d'exploitation sur le disque dur

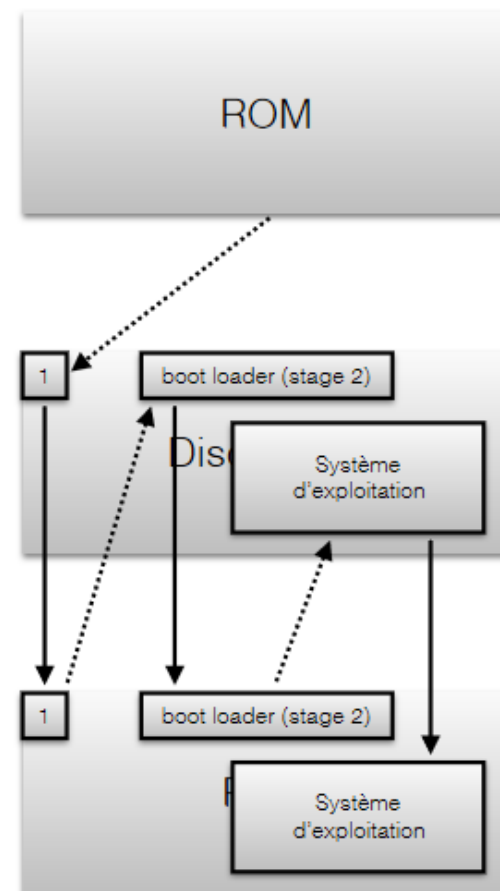
3. Le BIOS copie le SE en RAM

4. Branchement vers la RAM pour que le SE puisse s'exécuter



Séquence de démarrage réelle

1. Exécution du BIOS (stocké dans la ROM)
2. Le BIOS charge le « boot loader stage 1 » situé dans le « Master Boot Record » (MBR) dans la RAM, et démarre l'exécution de ce programme
3. Le programme charge un autre programme, le « boot loader stage 2 » en mémoire, et démarre l'exécution de ce programme
4. Le « boot loader stage 2 » demande à l'utilisateur quoi faire (si désiré). Il charge le SE en mémoire, et démarre son exécution



LE BIOS

- BIOS (Basic Input Output System):
 - permet au PC de booter (démarrer) et d'initialiser les périphériques avant de « passer le relais » au système d'exploitation
 - situé dans une puce reprogrammable (mémoire flash) EEPROM
 - les paramètres utilisateur (heure...) du BIOS étant eux stockés dans une mémoire CMOS:
 - besoin d'être alimentée pour conserver ses informations
 - pile plate présente sur la carte-mère
 - POST (Power-on-self-Test): amorçage
 - Disparition du BIOS (UEFI)

L'UEFI

- UEFI (Unified Extensible Firmware Interface):
 - fonctionnalités réseau intégrées en standard
 - interface graphique haute résolution
 - gestion intégrée du multiboot
 - affranchissement de la limite des partitions à 2,2 To
 - BIOS écrit en assembleur. L'UEFI est écrit en C:
 - maintenance plus aisée
 - Développé pour assurer l'indépendance entre système d'exploitation et plate-forme matérielle sur laquelle il tourne

Le rôle du BIOS ou de l'UEFI

1. Test de l'ordinateur (POST)
2. Lancer le système d'exploitation
3. Acquérir et maintenir l'information sur les périphériques
4. Fournir une librairie de fonctions au SE pour contrôler les périphériques (sous la forme d'interruptions)

Pratiquement plus utilisé avec les SE récents

5. Tâches secondaires, reliées aux périphériques: gestion de la puissance, gestion de la température, Plug and play, etc...

Le POST (Power On Self Test)

- Les tests effectués dans l'ordre sur les PC :
 1. Test microprocesseur (test des registres en écriture et en lecture)
 2. Test ROM contenant le programme de démarrage et le BIOS
 3. Initialisation du contrôleur de mémoire
 4. Test des 16 premiers Kilo de mémoire RAM
 5. Initialisation du contrôleur d'interruption et des interruptions
 6. Test du temporisateur servant à compter le temps
 7. Test de l'adaptateur d'écran et affichage du curseur
 8. Test de l'ensemble de la mémoire RAM
 9. Test du clavier
 10. Vérification de la présence d'un lecteur amovible ou d'un disque dur
 11. Test de l'imprimante et des ports de communication
 12. Comparaison des résultats obtenus lors des tests avec la configuration sauvegardée en mémoire CMOS
 13. «Beep» du haut-parleur

CMOS (complementary metal-oxide semiconductor)

- zone de mémoire (64 octets initialement) maintenue alimentée par une pile lorsque l'ordinateur est éteint
- contient des informations relatives au matériel d'un PC: types de disque dur, de clavier, d'écran...
- contient aussi la date et le temps du système qui est mis à jour par un circuit qui compte le temps, alimenté par la pile.
- L'information contenue dans la RAM CMOS est utilisée par le BIOS

Les pilotes (drivers)

- Programmes qui permettent au système d'exploitation de gérer le matériel
- Développer par les fabricants du matériel (en général)
- Evolue très souvent dans le temps
 - Correction de bugs ou de failles de sécurité

Les pilotes (drivers)

- Comment cela fonctionne?
 - Le BIOS (ou UEFI) démarre les composants fondamentaux (HDD, GPU intégré, clavier, souris...) avec de pilotes standards=> performances minimum
 - Une fois le système amorcé, il prend le contrôle sur les périphériques:
 - avec des pilotes standards (fourni dans le système)
 - ou des pilotes spécifiques (fournies par le constructeur du matériel) => généralement + performant
 - Attention il existe très souvent des versions:
 - Suivant le système (version, 32bits, 64 bits...)

Les pilotes (drivers)

- Plug and play:
 - Norme
 - Simplifie la détection du matériel et son installation
 - Matériel contient un firmware qui permet au bios de récupérer les informations suivantes:
 - Numéro unique caractérisant le matériel
 - Ressources nécessaires pour fonctionner
 - Le bios fournit ces informations au système (aussi à la norme plug and play) quand il prend le relais.

Les fichier systèmes

- Fichiers contenant des données nécessaires au fonctionnement du SE
- Windows:
 - Souvent dans C:\Windows\system32 mais pas obligatoire.
 - Extensions du type
 - .sys (ex: pagefile.sys)
 - .ini (ex: boot.ini)
- Linux:
 - Dans /etc
 - fstab, network, resolve.conf, passwd...

Le Partitionnement.

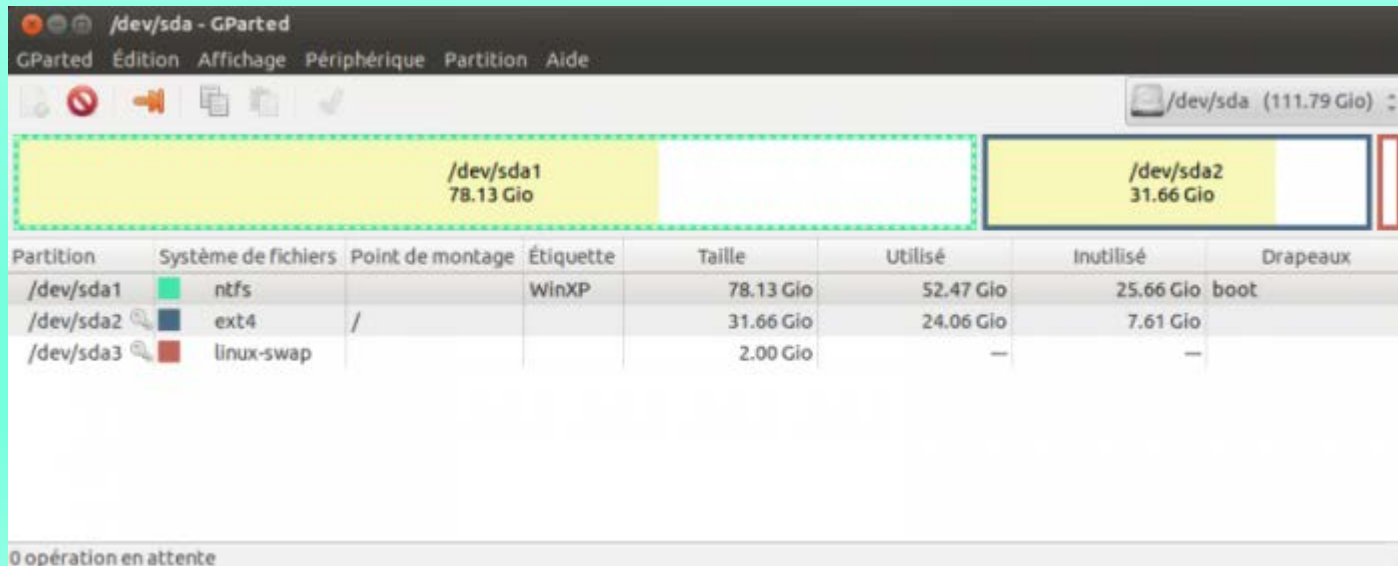
- Différencier les données du système:
 - Augmentation de la fiabilité en cas de défaillance système
 - Evite le remplissage accidentel du système
 - Permet de réinstaller l'OS sans détruire les données.
 - Multiboot

Le partitionnement

- Table de partitionnement:
 - MBR (Master Boot Record) si BIOS:
 - Max 4 partitions primaires (amorçable)
 - Taille maxi d'une partition 2,2To
 - GPT (GUID Partition Table) si UEFI:
 - Max 128 partitions
 - Taille maxi d'une partition 256 To
 - contient également une routine d'amorçage dont le but est de charger le système d'exploitation, ou le chargeur d'amorçage (*boot loader*)

Le Partitionnement.

- La partition d'échange (SWAP):
 - sert à étendre la mémoire utilisable par un système d'exploitation
 - Windows n'utilise pas une partition mais un fichier d'échange (pagefile.sys)
- Outil de partitionnement gratuit Gparted (Linux)



Distribution live

- OS exécutable depuis un support amovible.
- Avantages:
 - *utilisable en direct*, sans passer par phase d'installation.
 - Très pratique pour des petits tests.
 - Sauver un autre OS.
- Inconvénients:
 - Nécessite de la RAM.
 - Plus long à démarrer (qu'un système installé).
 - Pas de modification de l'OS.
- Exemples: Ubuntu, SystemRescueCd...

La virtualisation d'OS

- consiste à faire fonctionner plusieurs OS par machine.
- Avantages:
 - Installation d'OS de tests facile à mettre en œuvre.
 - Changer d'OS sans redémarrer.
 - Continuité de service (sauvegarde machine virtuelle).
- Inconvénients:
 - Nécessite beaucoup de RAM.
 - Pas de multiboot sur une VM (virtual machine).
- Exemple de produits de virtualisation:
 - VmWare, VirtualBox, Parallels...

La virtualisation d'OS

- L'hyperviseur peut se substituer à l'OS.
- La machine virtuelle est située dans un fichier.
 - Différents formats en fonction des applications:
 - vmdk – format VMware
 - vdi pour "Virtual Disk Image", format native de VirtualBox
 - vhd – Format Microsoft

La virtualisation d'OS

- La machine virtuelle est située dans un fichier:
 - Pour VMWare:
 - **myvm.vmdk** : fichier texte qui contient la description du disque dur virtuel
 - myvm-flat.vmdk : fichier qui contient le contenu du disque dur virtuel
 - myvm.nvram : fichier contenant le bios de la machine
 - **myvm.vmx** : fichier de configuration de la machine virtuelle
 - myvm.vmxs : fichier de configuration supplémentaire
 - myvm.vswp : fichier contenant la mémoire principale de la machine virtuelle
 - vmware.log : journal de la machine virtuelle
 - myvm.vmsd : fichier texte contenant la description du snapshot
 - myvm-Snapshot#.vmsn : fichier contenant l'état exact de la machine au moment de la prise du snapshot
 - myvm-000001.vmdk : fichier texte qui contient la description du disque dur virtuel après snapshot
 - myvm-000001-delta.vmdk : fichier qui contient la différence de contenu entre le disque dur virtuel actuel et le snapshot
 - myvm.hlog : fichier résultant d'une opération de vMotion