

# Introduction aux OS

## M1105 - Systèmes d'exploitation

Sebastien.Kramm@univ-rouen.fr

IUT de Rouen, dépt. Réseaux & Télécoms

Version du 19 novembre 2017

# OS : Operating System

Plusieurs définitions coexistent :

Définition grand public :

Logiciel qui permet de faire fonctionner un ordinateur et d'exécuter des programmes applicatifs.

Définition stricte

Logiciel système qui doit :

- ▶ gérer les **ressources** de la machine et leur utilisation par les programmes applicatifs en leur fournissant une interface homogène ;
- ▶ fournir une Interface Homme Machine (IHM) pour permettre l'utilisation de la machine.

## Concept de ressources

Les ressources d'une machine sont multiples :

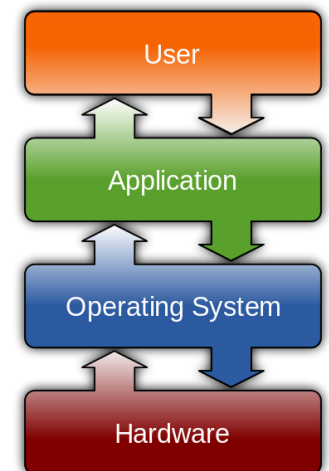
- ▶ Mémoire vive
- ▶ processeur
- ▶ Stockage de masse, avec souvent plusieurs unités (DD, USB, ...), et organisé en **système de fichiers**
- ▶ Périphériques d'E/S "simples" (clavier, souris, écran en mode "console")
- ▶ Périphériques d'E/S complexe : carte graphique pour affichage graphique
- ▶ connexions à d'autres machines distantes (imprimante, réseau local, réseau distant, ...)
- ▶ Capteurs (température, accélération, GPS, etc.)
- ▶ ...

Tout les programmes ne peuvent pas y accéder de façon autonome : l'OS fournit au programme une **interface** pour utiliser ces ressources.

## Concept de ressources

- ▶ L'OS interdit aux programmes d'accéder directement aux ressources matérielles

- ▶ Objectifs :
  - ▶ **Sécurité** des applications : pas d'accès concurrent possible à la même ressource
  - ▶ **Partage** des ressources entre applications : pas de possibilité d'accès exclusif

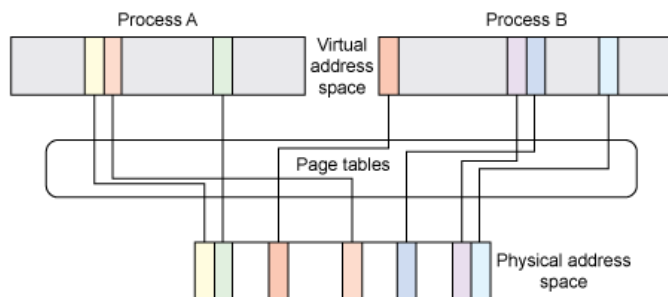


Un OS fournit aussi un ensemble de services logiciels permettant son fonctionnement :

- ▶ Ordonnanceur de tâche
- ▶ Gestionnaire de mémoire virtuelle
- ▶ Pilotes de périphériques
- ▶ Gestion de la sécurité (des fichiers, des programmes, des utilisateurs, ...)
- ▶ Bibliothèque de fonctions précompilées pour les programmes applicatifs

- ▶ Utilisateur (*user*) : reel ou logiciel, possède des droits et nécessite une authentification
- ▶ Fichier (*file*) : structure logique qui délimite une zone de stockage de données et dotées de métadonnées (nom, attributs, etc.)
- ▶ Processus (*process*) : Un programme en cours d'exécution.

- ▶ De façon à garantir la sécurité, les processus s'exécutent :
  - ▶ soit en "mode noyau" (*kernel*) : utilisation sans restriction de toutes les fonctionnalités ;
  - ▶ soit en "mode utilisateur", avec des limitations.
- ▶ En mode utilisateur :
  - ▶ certaines fonctionnalités ne sont pas accessibles (accès direct E/S notamment)
  - ▶ la mémoire associée à chaque processus est **virtuelle** :
    - si on lui associe une plage de mémoire de \$00000 à \$0FFFF, ce bloc de mémoire pourra en réalité se trouver dans la mémoire physique entre \$20000 à \$2FFFF.

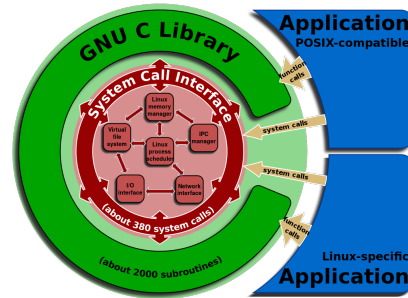


- ▶ Ce mode de fonctionnement à deux niveaux est pris en charge au niveau *hardware*, au niveau du processeur :
  - ▶ la plupart ont un mode d'exécution "privilegié" ;
  - ▶ un gestionnaire de mémoire virtuelle (*Memory management unit* : MMU) distribue aux processus des adresses virtuelles, qui sont "mappées" sur des adresses physiques.
- ▶ **Seul** le noyau de l'OS s'exécute en mode "kernel", toutes les applications utilisateurs s'exécutent en mode "user" :
  - Pour accéder aux périphériques, elles doivent faire des **appels système** au noyau.
- ▶ Le gestionnaire de processus (*task scheduler*) réalise la commutation entre tâches et provoque le basculement d'un mode dans l'autre.

## Appels systèmes : normalisation

- ▶ Les OS fournissent des appels systèmes (de 300 à 700) permettant d'effectuer des tâches bas-niveau.
  - ▶ Gestion des périphériques ;
  - ▶ Gestion de fichiers ;
  - ▶ Gestion mémoire, processus, ...

- ▶ POSIX : ensemble de standards permettant l'interopérabilité entre programme applicatifs pour les OS de type Unix / Linux.
- ▶ Windows : support partiel de cette norme.



## Interface Homme Machine

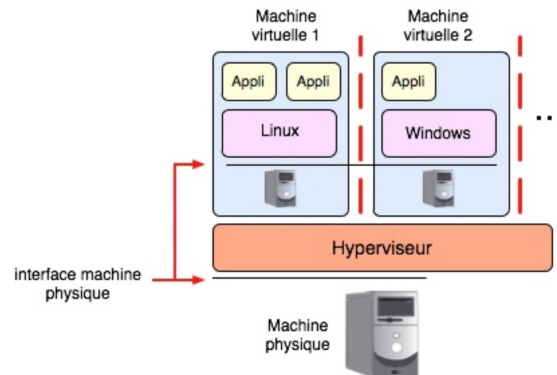
- ▶ Deux paradigmes coexistent : *Command-Line Interface* (CLI) et *Graphical User Interface* (GUI).
- ▶ Les deux ont des avantages et inconvénients.

Item	CLI	GUI
Prise en main	Difficile, nécessité de mémoriser les commandes <sup>1</sup>	Facile
Ressources système nécessaires	Faibles	Importantes
Puissance	Totale !	Limitée
Vitesse de travail	Élevée : 2 mains sur le clavier	Limitée : on doit alterner clavier & souris
Automatisation	Par nature	Complicquée
Tâches répétitives	Idéal	Laborieuses et risquées (oublis)

1. ou avoir un *cheatsheet*

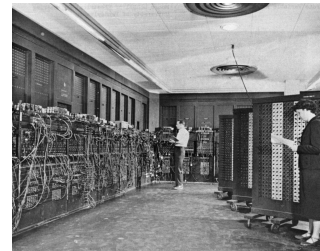
## Evolutions actuelles : virtualisation

- ▶ Tendance actuelle (serveurs) : séparer la partie Hardware de la partie software via une couche intermédiaire appelée **hyperviseur** → va intercepter toutes les interactions entre l'OS et le hardware
- ▶ Objectif : rentabiliser au mieux le Hardware (coût élevé) : une même machine physique va héberger plusieurs OS, qui seront complètement indépendants.



source : S. Krakowiak/SIF

## Historique des ordinateurs



ENIAC (1946)



Atlas (1963)



IBM 650 (1953)



DEC PDP-7 (1965)

**the 10-Megabyte Computer System**

**Only \$5995 COMPLETE**

*New From IMSAI®*

- 10-Megabyte Hard Disk
- 5 1/4" Dual-Density Floppy Disk Back-up
- 8-Bit Microprocessor
- Memory-Mapped Video Display Board
- Disk Controller
- Standard 64K RAM (Optional 256K RAM)
- 10-Slot S-100 Motherboard
- 28-Amp Power Supply
- 12" Monitor
- Standard Intelligent 62-Key ASCII Keyboard (Optional Intelligent 86-Key ASCII Extended Keyboard)
- 132-Column Dot-Matrix Printer
- CPM\* Operating System

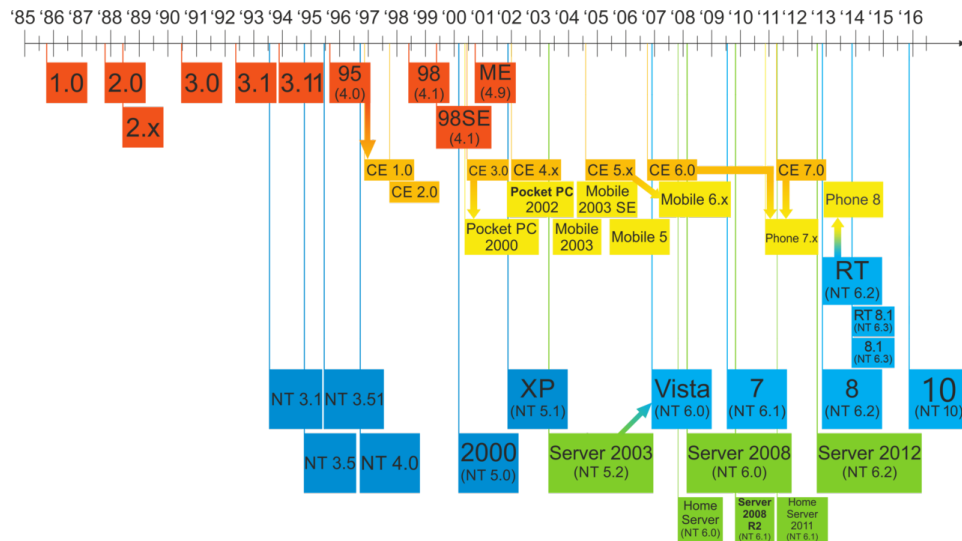
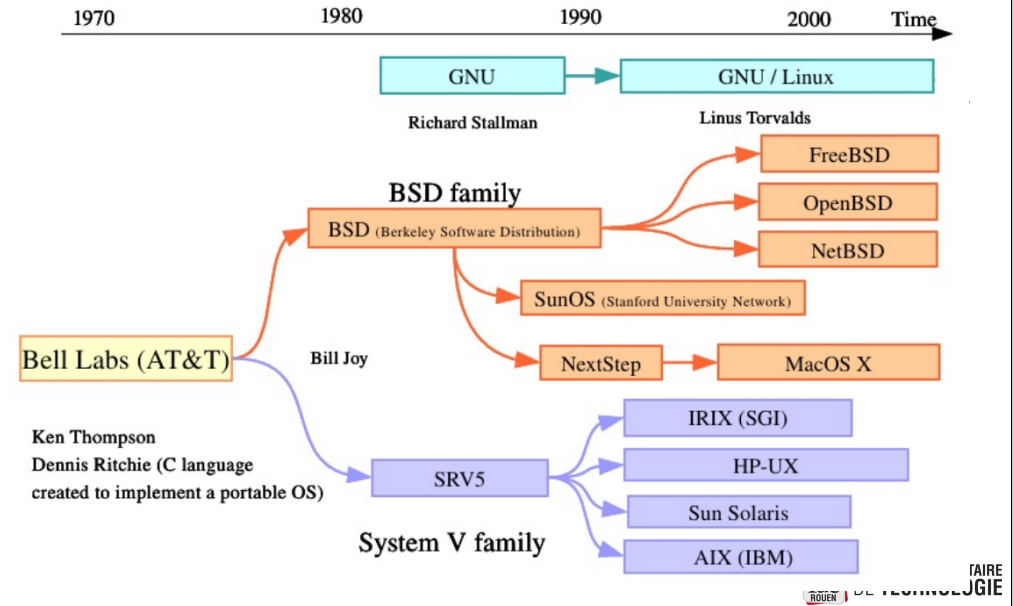
*You Read It Right ... All for \$5995!*

**IMSAI®** ... Thinking ahead for the 80's

Computer Division of the Fischer-Freitas Corporation  
415/635-7615 910 81st Avenue, Bldg. 14 • Oakland, CA 94621

\*CPM is a trademark of Digital Research. Imsai is a trademark of the Fischer-Freitas Corporation

1969 : Ken Thompson écrit le premier OS : Unix, sur un DEC PDP-7



► MS-DOS-based versions of Windows

- 1981-1995 : MS-DOS : système d'exploitation avec IHM de type CLI
- 1986 : Windows 1 : surcouche graphique à MS-DOS
- 1992 : Windows 3.1 : première version réellement fonctionnelle... mais mono-utilisateur
- 1993 : Windows NT : ne repose plus sur MS-DOS, nouveau noyau, multi-utilisateurs, multi-tâches préemptif
- 2001 : Windows XP : réunification des games grand public et professionnel en une seule famille.  
Ventes arrêtées en 2010, fin du support étendu (=payant) en 2014... mais toujours utilisé aujourd'hui sur certains systèmes critiques !

## Défis actuels

- ▶ Aujourd'hui : relative stabilité des OS
- ▶ nombreux problèmes encore ouverts :
  - ▶ Sécurité : complexité exponentielle des systèmes et absence de modèle théorique ;
  - ▶ Certification : obligatoire dans certains domaines applicatifs (aéronautique) mais très difficile ;
  - ▶ Arrivée massive de la mobilité ("Systèmes embarqués") : ressources limitées et connectivité incertaine ;
  - ▶ Parallélisme : absence de cadre théorique.

## OS pour mobiles

- ▶ Avec l'explosion du mobile, de nombreux OS dédiés sont apparus.
- ▶ Aujourd'hui, seuls quelques grands acteurs sont significatifs :
  - ▶ Microsoft → Windows 10 Mobile
  - ▶ Google → Android
  - ▶ Apple → iOS



## Android

- ▶ Fondé en Californie en 2003, racheté par Google en 2005.
- ▶ Supporte aujourd'hui de nombreux processeurs (architectures) : ARMv7 et ARMv8, x86, MIPS, ...
- ▶ S'appuie sur le **Noyau Linux**.
- ▶ Utilisé sur smartphone, tablette, mais aussi sur ...

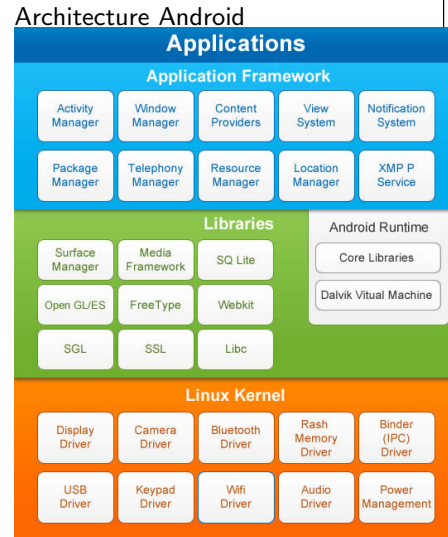


## Ventes smartphones aujourd'hui (2016)

Period	Android	iOS	Windows Phone	Others
2015Q3	84.3%	13.4%	1.8%	0.5%
2015Q4	79.6%	18.6%	1.2%	0.5%
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%

Source: IDC, Aug 2016

- ▶ Android n'utilise que le noyau Linux
- ▶ Pas de Serveur X11 ni de bibliothèque C classique
- ▶ Gestionnaire de paquet : *Android application package (APK)*
- ▶ Les applications s'exécutent dans une machine virtuelle
  - ▶ avant 5.0 : Dalvik
  - ▶ 5.0 → : *Android Runtime (ART)*



1. Par architecture supportée
2. Par type de licence :
  - ▶ propriétaire ;
  - ▶ libre ou assimilée.
3. Par services rendus :
  - ▶ Mono ou Multi-utilisateur implique une **gestion des droits** : qui a le droit de faire quoi avec les ressources de la machines (fichiers, programmes, ressources diverses, etc.)
  - ▶ Mono-tâche ou Multi-tâche implique un **ordonnanceur** qui va distribuer du temps-CPU aux différents processus.
4. Par type :
  - ▶ généraliste
  - ▶ dédié à un type d'application donné

## 1 - Architecture supportée

Suivant la cible matérielle (hardware)

- ▶ Monoprocasseur : Mise en place d'un Pseudo-parallélisme l'OS gère la commutation de tâches, via un **ordonnanceur de tâches**.
- ▶ Multiprocasseur et/ou multicœurs : l'OS doit (en plus) répartir l'exécution sur les différents CPU de façon transparente et efficace.

## 2 - Licence logicielle

- ▶ A chaque logiciel est rattachée une licence : contrat par lequel le titulaire des droits sur ce logiciel définit les conditions dans lesquelles ce programme peut être utilisé, diffusé ou modifié.
- ▶ On distingue les licences **libres** et licences **propriétaires**.
- ▶ Indépendant de la notion de coût :
  - ▶ un logiciel sous licence libre peut être **vendu**
  - ▶ un logiciel sous licence propriétaire peut être **gratuit**



## Licence libre

- ▶ Beaucoup de licences différentes, avec des principes similaires mais quelques différences  
GPLv2, GPLv3, BSD, MIT, Apache, etc.
- ▶ Très liées au concept de "copyright" anglo-saxon, il existe des versions plus adaptées au droit français (Licence CeCILL).



R. Stallman

### ▶ Accorde **quatre droits fondamentaux** aux utilisateurs :

1. Liberté d'utiliser le logiciel ;
2. Liberté d'étudier le logiciel pour en comprendre le fonctionnement ou l'adapter à ses besoins ;
3. Liberté de modifier (améliorer, étendre, etc.) ou de l'incorporer à un autre logiciel ;
4. Liberté de redistribuer : diffusion à d'autres usagers, y compris commercialement.

- ▶ Principale différence : notion de **copyleft** (ou principe de **viralité**) : Tout logiciel utilisant une brique sous une telle licence passe **automatiquement** sous cette licence.

→ Dans le produit final, tous les composants utilisés passent sous cette licence



## Copyleft des principales licences libres

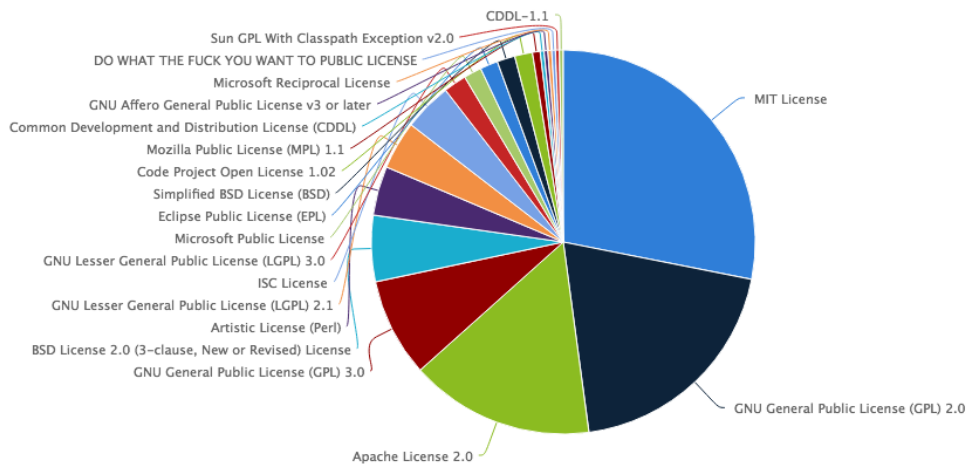
- ▶ On distingue copyleft fort ou faible
- ▶ Copyleft faible : de nouveaux composants ayant une autre licence peuvent être ajoutés au projet.

Nom	Copyleft
GPL	copyleft fort
GNU Library or Lesser (LGPL)	copyleft faible
Apache licence	sans copyleft
New BSD Licence	sans copyleft
MIT Licence	sans copyleft
Mozilla public licence	copyleft faible



## Les 10 licences libres les plus utilisées

### Top 20 Most Commonly Used Licenses in Open Source Projects



source : <https://www.blackducksoftware.com>



## Licences propriétaires

- ▶ Ces licences sont souvent très restrictives.
- ▶ Chaque éditeur a sa propre politique de licence.
- ▶ Exemple : Windows a 3 types de licence, avec des prix différents (indépendamment de la version) :
  1. Licence *Retail* : achetée séparément ("boxed copy", ou achat en ligne)  
Appelée aussi *Full Packaged Product* (FPP)
  2. Licence OEM (*Original Equipment Manufacturer*) : la moins chère mais la plus restrictive.  
Destinée aux assembleurs : utilisée dans le cas d'un achat matériel avec le logiciel préinstallé  
Inconvénient : liée à la machine, impossible de déplacer la licence sur une autre.
  3. Licence VLK (*Volume Licensing Key*), pour les organisations avec un parc important.



## Exemple : licences Windows 7

	OEM (top PC makers)	OEM System Builder	Windows Anytime Upgrade	Retail upgrade	Retail (Full Package Product)	Enterprise (Volume License)
Requires previous Windows license	No	No	Yes	Yes	No	Yes
Can be transferred to new PC	No	No	No	Yes	Yes	No
Can switch between 32/64 bit	No <sup>1</sup>	No <sup>1</sup>	No	Yes	Yes	Yes
Support provided by Microsoft	No	No	Yes	Yes	Yes	Yes <sup>2</sup>
Support provided by PC maker	Yes	Yes	No	No	No	No
Microsoft activation required	No	Yes	Yes	Yes	Yes	No <sup>3</sup>
OEM preinstallation required	Yes	Yes	No	No	No	No
Clean install permitted	Yes <sup>4</sup>	Yes <sup>4</sup>	No <sup>5</sup>	No <sup>6</sup>	Yes	Yes

<sup>1</sup> Choice must be made at time of purchase/installation  
<sup>2</sup> Support options vary by contract  
<sup>3</sup> VL copies may require activation through local Key Management servers  
<sup>4</sup> Reinstallation procedures vary by manufacturer  
<sup>5</sup> Must reinstall original Windows 7 edition, then re-run Anytime Upgrade  
<sup>6</sup> Requires telephone activation

source : Ed Bott, zdnet.com, 2009

## Tarifs Windows 10 "Retail" (octobre 2016)

### Windows 10 Famille

Profitez de démarrages plus rapides, d'un menu Démarrer familier mais étendu, et d'excellentes nouvelles façons d'accomplir des tâches même sur des appareils multiples.

135,00 € TTC

★★★★☆ 3.4 (103)

### Windows 10 Pro

Toutes les caractéristiques de Windows 10 Famille auxquelles s'ajoutent des fonctionnalités professionnelles pour le chiffrement, la connexion à distance, la création de machines virtuelles, et plus encore.

279,00 € TTC

★★★★☆ 4.1 (26)

## 4.1 - Autre types d'OS

### OS embarqués

- ▶ OS prévus pour fonctionner sur des machines de petite taille ou à faible ressources.
- ▶ Evolution actuelle : coût élevés, tendent à disparaître au profit d'OS basé sur un noyau Linux "customisé".

freebox mini 4K



## 4.2 - Autre types d'OS

### OS temps réel

- ▶ Objectifs : garantit les temps de réponse
- ▶ Utilisé pour des process avec des contraintes temporelles fortes : une tâche **doit** être terminée dans le laps de temps imparti
- ▶ Deux types :
  - ▶ Temp-réel "dur" : gestion stricte du temps  
Exemple : automobile, biomédical (pacemaker), etc.
  - ▶ Temps-réel "mou" : systèmes acceptant des variations minimales  
Exemple : streaming video, multimédia
- ▶ Le plus connu : **VxWorks**, utilisé dans Curiosity Mars Rover, B787, SpaceX, robotique, etc.





## Composants d'une distribution Linux

- ▶ "Linux" ≠ "Distribution Linux" (ou aussi "système GNU/Linux")  
= Linux (noyau) + ensemble de programmes permettant d'avoir un ordinateur fonctionnel.
- ▶ Une distribution est composée des éléments suivants :
  - ▶ noyau (kernel)
  - ▶ bootloader (Grub)
  - ▶ Shell et utilitaires associés
  - ▶ serveur graphique ("serveur X" en général)
  - ▶ environnement de bureau
  - ▶ gestionnaire de paquets
  - ▶ utilitaires, sous forme de bibliothèques binaires
  - ▶ applications "desktop" classique (bureautique, graphisme, vidéo, ...)
  - ▶ etc.

⚠ Certaines distributions pour serveur n'ont pas de GUI!

## Linux

- ▶ Création : 1991, par **Linus Torvalds**, étudiant finlandais.
- ▶ S'inspire d'un clone d'UNIX (Minix).
- ▶ Ne concerne que les fonctions essentielles d'un OS, mais +50% du code concerne les pilotes de périphériques.
- ▶ Aujourd'hui :
  - ▶ développé principalement par un ensemble d'entreprises (Red Hat, Novell, IBM, Intel, ...)
  - ▶ release 3.10 / 2013 : 15 MLOC (*Lines Of Code*)
  - ▶ communauté de "kernel developers" : 5000 à 6000 personnes.
- ▶ Cycle de développement :
  - ▶ un contributeur soumet un **patch** : ensemble de modifications ou d'ajouts au code source ;
  - ▶ le patch est "reviewé" par des experts, puis est accepté (ou non) dans le code du noyau.



Mascotte du projet :  
manchot Tux



(Larry Ewing, 1996)

## Kernel

- ▶ [www.kernel.org](http://www.kernel.org) : fourni sous forme de sources, qu'il faut **compiler**.
- ▶ Le noyau s'occupe de toutes les fonctions essentielles : gestion des processus, des E/S, de la sécurité, etc.
- ▶ Portable sur de très nombreuses plateformes matérielles (CPU) : depuis le Raspberry Pi (30\$) jusqu'aux super-calculateurs du Top 500<sup>2</sup>
- ▶ Écrit presque entièrement en C
- ▶ Conçu de façon **modulaire** : les différentes fonctionnalités peuvent être intégrées ou non :
  - ▶ soit à la compilation, via une étape de configuration ;
  - ▶ soit sous la forme de **modules**, qu'on peut charger de façon dynamique pendant l'exécution.

## The Linux Kernel Archives



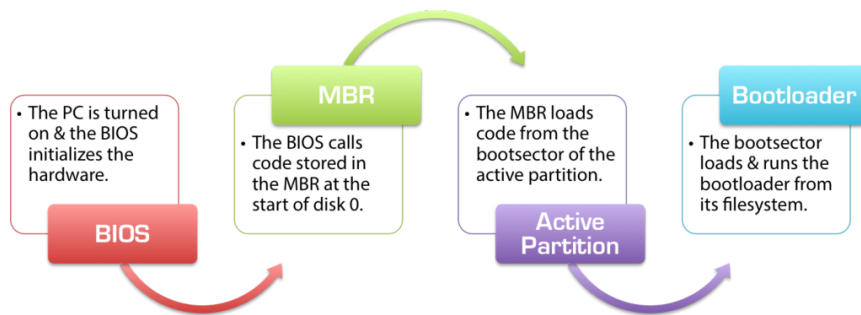
## Utilitaires / projet GNU

- ▶ Projet initial, lancé en 1983 par Richard Stallman : système d'exploitation entièrement libre.
- ▶ Aujourd'hui : ensemble de composants utilisés de concert avec le noyau Linux.
- ▶ Soutenu par la *Free Software Foundation* (FSF), organisation américaine à but non lucratif fondée en 1985 par RS
- ▶ Principaux outils :  
compilateur GCC, le débogueur GDB, le shell Bash, la bibliothèque C GNU, les outils de base GNU (*coreutils* : ensemble de commandes externes), les outils binaires GNU (*binutils* : manipulation de fichiers binaires), etc.



## Bootloader

- ▶ Rôle : permettre la sélection et le démarrage de l'OS "à froid".
- ▶ Exécuté juste après le BIOS, après lecture dans la MBR du disque de sa localisation sur le DD.
- ▶ Processus de boot :



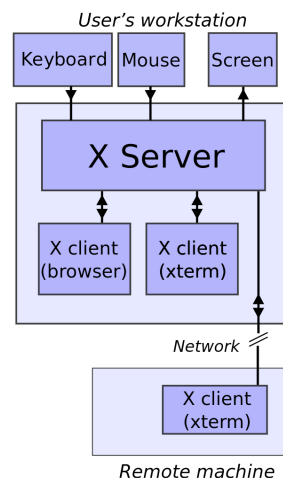
## Bootloader : GRUB

- ▶ Le plus universel : GRUB
- ▶ Gère le double boot Windows - Linux



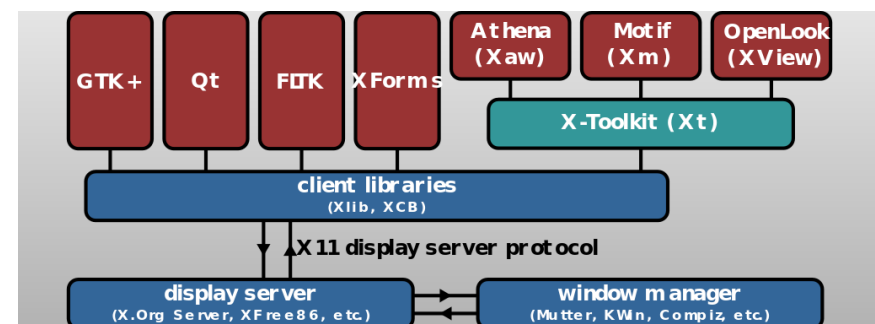
## Gestionnaire graphique

- ▶ Le plus utilisé : X11 (appelé aussi "X server" ou "X window")
- ▶ Fonctionne en architecture client-serveur :
  - ▶ serveur : réside en mémoire, gère les périphériques d'entrée (clavier, souris) et l'écran via des appels systèmes au pilotes de périphériques.
  - ▶ client : application qui souhaite accéder aux ressources E/S GUI (clavier, souris, écran) : envoie des requetes au serveur.
- ▶ Avantages :
  - ▶ Indépendant du canal de communication : serveur et client peuvent être sur des machines distinctes.
  - ▶ Indépendant de la carte graphique de la machine.



## Serveur X

- ▶ Communication : via un protocole dédié, ou via des appels à une bibliothèque de fonctions.
- ▶ En pratique, sera utilisé à travers des **bibliothèques applicatives** (GTK+, Qt, ...)



- ▶ Evolution en cours : évolution vers une approche plus intégrée au noyau, et disparition de X : projet Wayland

## Desktop Manager

- ▶ "Environnement de bureau" : outil permettant l'utilisation de la machine via une GUI
- ▶ Constitué de : bureau et icônes de lancement, gestionnaire de fenêtres, barres de menus et panneaux associés, gestionnaire de sessions, etc.
- ▶ Les distributions en proposent souvent plusieurs au choix :
  - ▶ Ubuntu → Unity
  - ▶ Kubuntu → KDE
  - ▶ Xubuntu → Xfce
  - ▶ Lubuntu → Lxde



## Gestionnaire de paquets - 1

- ▶ Définition : outil automatisant le processus d'installation, désinstallation, mise à jour de logiciels installés.
- ▶ Concerne les logiciels systèmes (noyau, pilotes, etc.) mais aussi les logiciels applicatifs.
- ▶ Un **paquet** est une archive comprenant les exécutables, fichiers de configuration et tout ce qui est nécessaire au logiciel.
- ▶ Le gestionnaire de paquets gère :
  - ▶ Le téléchargement du paquet pour la version de la distribution installée ;
  - ▶ l'installation, mise à jour, et désinstallation ;
  - ▶ la vérification des **dépendances logicielles** : autres composants logiciels nécessaires.

## Gestionnaire de paquets - 2

- ▶ Les fournisseurs de distributions mettent à disposition des **dépôts** : serveurs permettant d'accéder à des dizaines de milliers de paquets, adaptés à chaque **version** de la distribution.
- ▶ Utilisable en GUI ou en CLI

```
sk@PORT-LITIS-SKRAMM:~$ sudo apt-get install libreoffice
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libreoffice-base libreoffice-base-core libreoffice-base-drivers
  libreoffice-calc libreoffice-common libreoffice-core libreoffice-draw
  libreoffice-gnome libreoffice-gtk libreoffice-gtk2 libreoffice-impress
  libreoffice-java-common libreoffice-math libreoffice-writer python3-uno
Suggested packages:
  libreoffice-grammarcheck openclipart2-libreoffice openclipart-libreoffice
  nstoeedit libreoffice-officebean libreoffice-gci libreoffice-report-builder
```

## Principales distributions

- ▶ Très nombreuses (centaines!), mais des "familles"
- ▶ Différences : format de paquets, gestionnaire de bureau, rythme de mise à jour, orientation générale, public visé, etc.
- ▶ Autre critère de choix : ancienneté, communauté, support (commercial ou non)
- ▶ Principales familles :
  - ▶ format RPM : Red Hat, openSUSE, Fedora, Mandriva, ...
  - ▶ format Deb : Debian, sur laquelle est basée Ubuntu, sur laquelle est basée Mint, ...
  - ▶ de nombreuses autres.
- ▶ Beaucoup de fragmentation...

