

# Systemes d'exploitation

---

# Plan

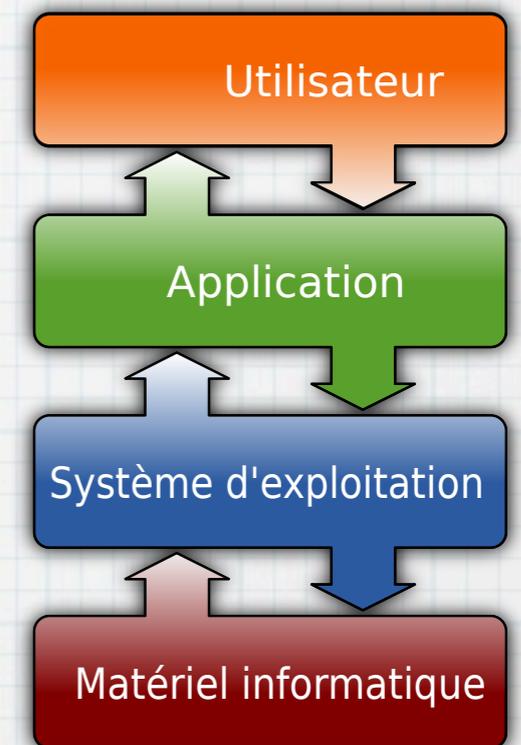
- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

# Systeme d'exploitation

- \* Propose une interface utilisateur/machine
  - \* Gestion des ressources materielles
  - \* Gestion du (des) utilisateur(s)
  - \* Gestion des processus
  - \* Sécurité (processus / mémoire)
  - \* Abstraction -> unification
  - \* Bibliothèque de fonction
- 
- \* Exemples : Windows, MacOS X, Linux, Solaris, \*BSD, VMS, Symbian OS ...

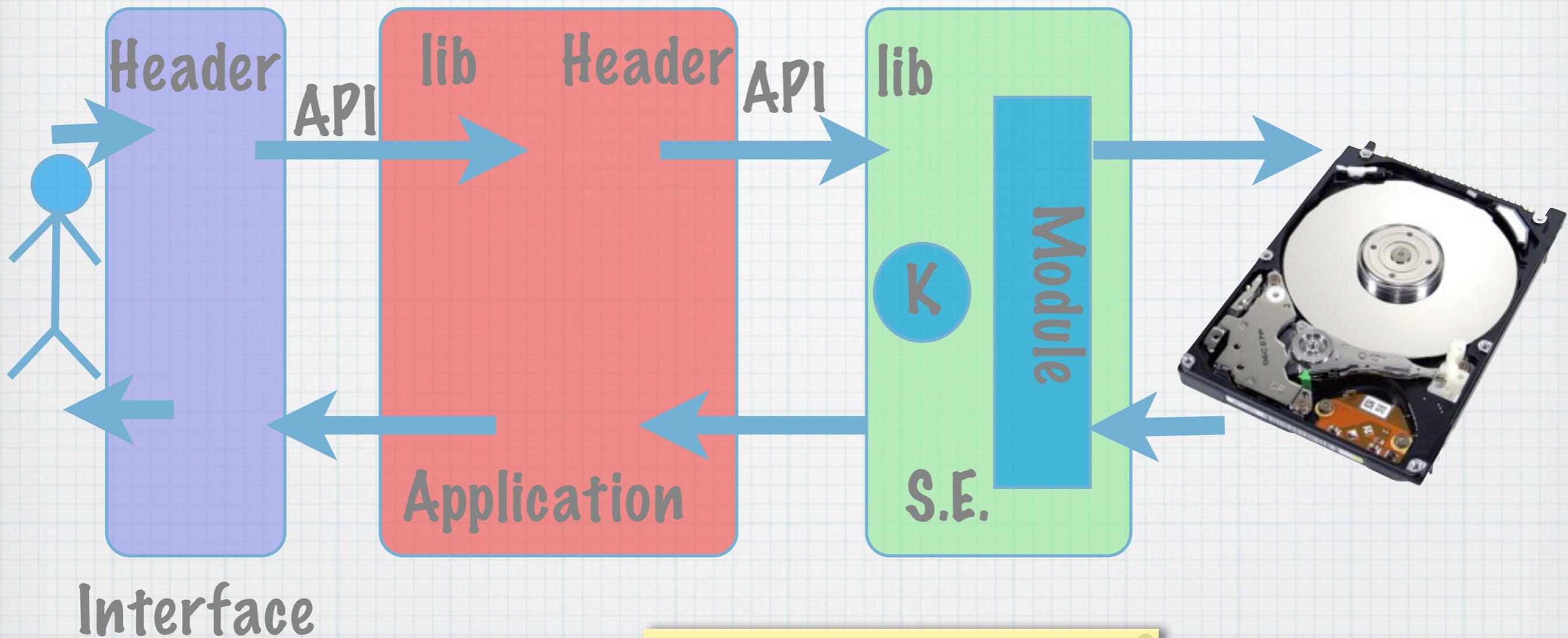
# Modélisation en couche

- \* L'utilisateur demande à l'application
- \* L'application demande au noyau
- \* Noyau se charge de la base
  - \* Mémoire ((des)allocation, pas de débordement)
  - \* CPU (gestion multi-tache, IPC)
  - \* Gestion des fichiers (hiérarchie)
  - \* Gestion des périphériques (clavier, souris, vidéo, réseau, etc.)
- \* Appels systèmes (demande de logiciels)



IPC= inter-process communication

# Exemple d'interaction



API = Application Programming Interface

# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

# Interface Shell

- \* Une interface en ligne de commande (CLI)
  - \* Windows : Command.com / cmd.exe
  - \* \*nix : Bash, ksh, sch, tcsh, zsh, ...
- \* Ancien, mais toujours puissante
- \* Utilise peu de ressources
- \* Ex : **Commande arg<sub>1</sub> arg<sub>2</sub> ... arg<sub>n</sub>**
- \* La commande qui sauve la vie dans un \*nix  
**man <commande>**

On écrit dans la doc on écrit les arguments de façon suivante :  
<arg> comme obligatoire  
[arg] comme optionnel

# Interface graphique

- \* GUI (Graphical User Interface)
- \* User Friendly (ergonomique)
- \* Les pictogrammes remplacent le texte
- \* Inventé par Xerox 1973 Alto.
- \* Généralement des fenêtres avec une barre de processus.
- \* Dirigé avec une souris ou un écran tactile.



# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

2K	BlueEyedOS	Data General	Exopc	Hurd
86-DOS	BOS	DC/OSx	ExOS	Hurricane
A/UX	BOS1810	DCP	Express	HydrixOS
Acados	BoxOS	Degenerate OS	Famos	i5/OS
ACP (Airlift Control Program)	BPMK	Delitalk	FDOS	IBM PC-DOS
AdaOS	bpink	Deming OS	Fiasco	IBSYS
ADMIRAL	BP	DELOS	FarmPro	Image Desktop
Adrenaline	BS/0	Desktop/D	FlashOS	Image Unix
aerolithOS	BS/200	DISKWORKS	FlexOS	Image Unix
Aimos	BSDi	DG/UX	FLP-80 DOS	Inferno
AIOS	BugOS	DIGITAL UNIX	Flux	INMOS
AIX	Calmira	dingOS	Flux-Fluke-Flask	INTEGRITY RTOS
AIX/370	CCP (Computer Control Program)	DK/DOS	FMS	Iridium OS
AIX/ESA	CDOS	DLD	Forth	IRIX
Aleris Operating System	Cefarix	DNIX	FreeBSD	iRMX
Allegro	C Executive	Domain OS	FreeDOS	IRTS
AllianceOS	Chaos	DOS	FreeDOWS	ISC (Interactive)
Alto OS	Chimera	DOS2	FreeVMS	ISIS
Amiga OS	Chippewa OS	DOS 50	Frenzy	ISSL
Amoeba	Choices	Dosket	FullPliant	ITRON
Amstrad *	Chorus	dr'ex	FunatixOS	ITS
AMX RTOS	Cinder OS	DFDOS	FxOS	JAMB
AngelOS	Cisco IOS	Drops	GazOS	JavaOS
Antarctica	Clicker32	Drywell OS	GCOS	Jbed
AOS/VS	CMW+ (SCO)	DS-OS	GECOS	JeniOS
Aperios *	COBRA	DTOS	Geek OS	Jeo-OS
Apollo Domain/OS	Coherent	DVIX	Genri Nucleus	Jibbed
ApolloOS	CONSENSYS	DYNIX Unix (Sequent)	Genera	JOS
Apostle	ConvexOS	ECL-121	GEORGE	JTMOS
Archimedes OS	Cos	elch Statb	GEOS	JUNOS
AROS	Cosy	eCos	GM OS	JxOS
ARTOS	Counterpoise	EduOS	GNU Hurd	KAOS
Asbestos	CP/K	EGOS	GNUstep	Katix
Athena *	CP/M	ekl BSD	Go	Kea
AtheOS	CPNET	elit	Goah	Kerberos
AtomsNet	CP/Z	ELKS	Gould OS	KeyKOS
Atomthreads	CPF (Control Program Facility)	Elysium	Grasshopper	KolibriOS
AuroraOS	Cromix	EOS	GUIDE	KOS
B-Free	Cronus	EP/IX	HA-MSP	KRONOS
Bada	CSOC	EPOC	Hactar	KROS
BAL	CTOS	ERaMS	Harmony	KRUD
Banyan VINES	CTSS	ERIKA	Haiku	Kylin
Basic Executive System	CX/SX	EROS	Helios	L4
BeIA	Cyber (CDC)	ESER	HES	L13Plus
BeOS	Cygnus	ESIX	Hive	LainOS
Beowulf	DAC	ESKO	HOPE	LAN Manager
BKY	Darwin	Eumel	HP-87 OS	LDOS
BlueEyedOS	Data General	EuNIX	HP-UX	LegOS

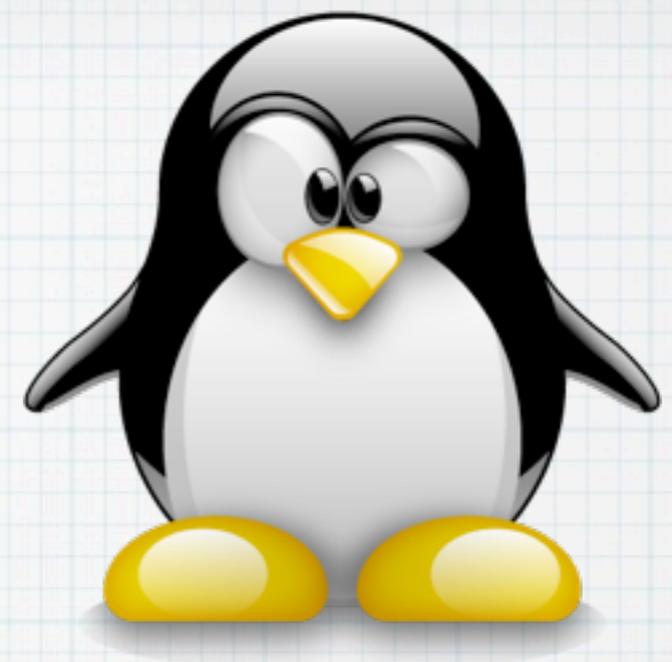
# Systemes d'exploitation

\* Il en existe beaucoup.

\* On se concentrera sur les 3 les plus connus du grand public

\* Windows, Linux, Mac OS

# GNU/Linux



- \* \*nix
- \* Open source (gratuit - adaptable partout)
- \* Dans le standard
- \* Des participants dans le monde entier
- \* Pas de timing à respecter
- \* Tout le monde peut contribuer
- \* Stable et robuste
- \* Modulable

# Familles UNIX (\*nix)

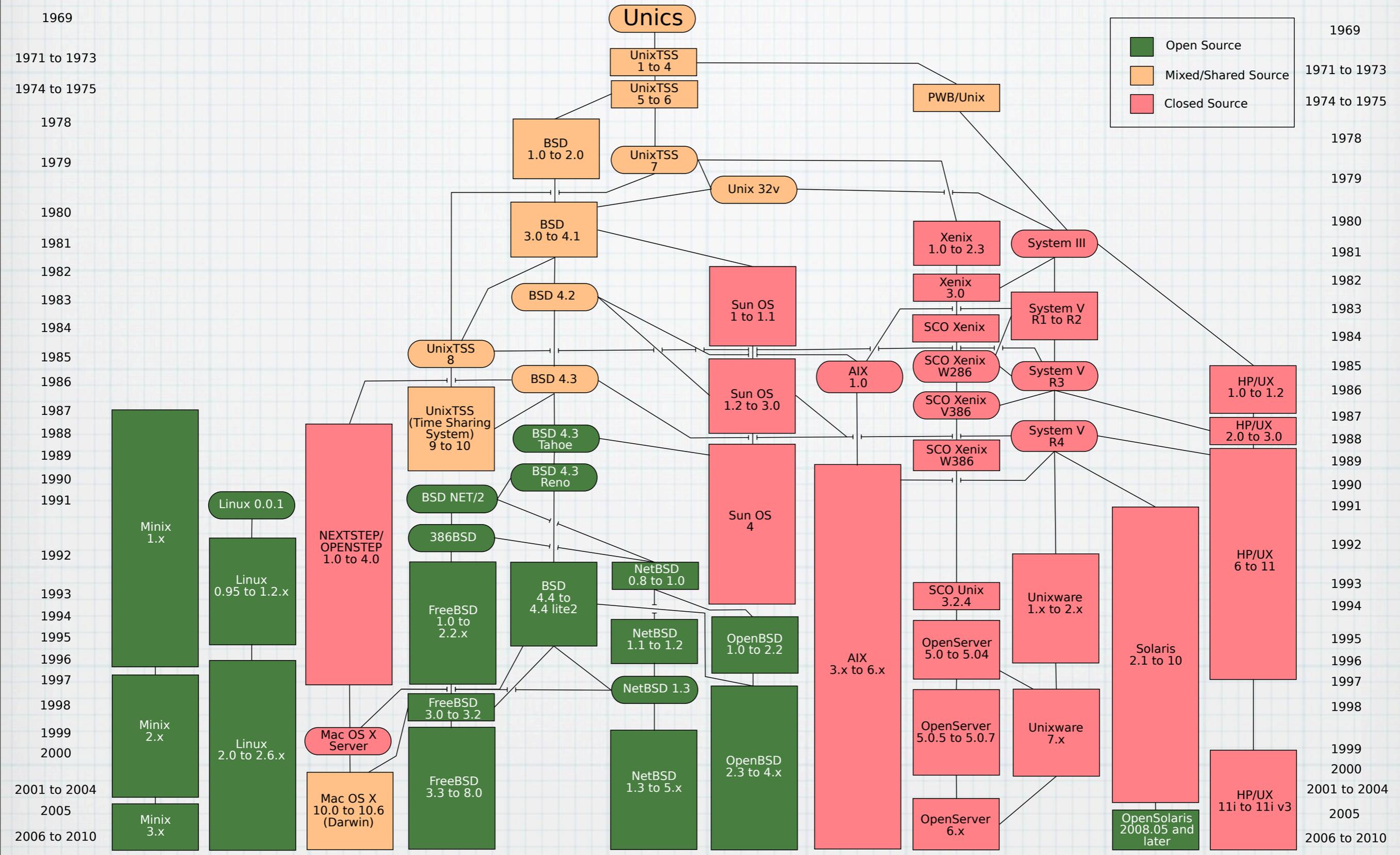
- \* Concept : petits programmes faisant bien leur tâche
- \* Modulaire
- \* Il n'y a que des fichiers et des répertoires.
- \* Multi-tâche
- \* Multi-Utilisateurs  
(avec droits - nom d'utilisateur et mot de passe)
- \* Il existe souvent un super utilisateur (root) ou un groupe d'administrateurs

# Encore de l'histoire

- \* 1969 Unix (descendant de multics) est de licence fermé AT&T (Unics des laboratoires Bell).
- \* 1983 GNU project (Richard Stallman) GNU's Not Unix - Free Software Foundation
- \* 1987 Minix de Tanenbaum uniquement pour l'enseignement
- \* 1991 Linux, Linus Torvalds

# Libre

- \* 1983 : Richard Stallman dévoile son projet de développer un système d'exploitation compatible Unix appelé GNU
- \* 1987 : Minix développé par Tanenbaum
- \* 1990 : Debut de Hurd
- \* 1991 : Linus Torvalds développe un noyau.
- \* 1998 : Mandrake linux pour grand public.



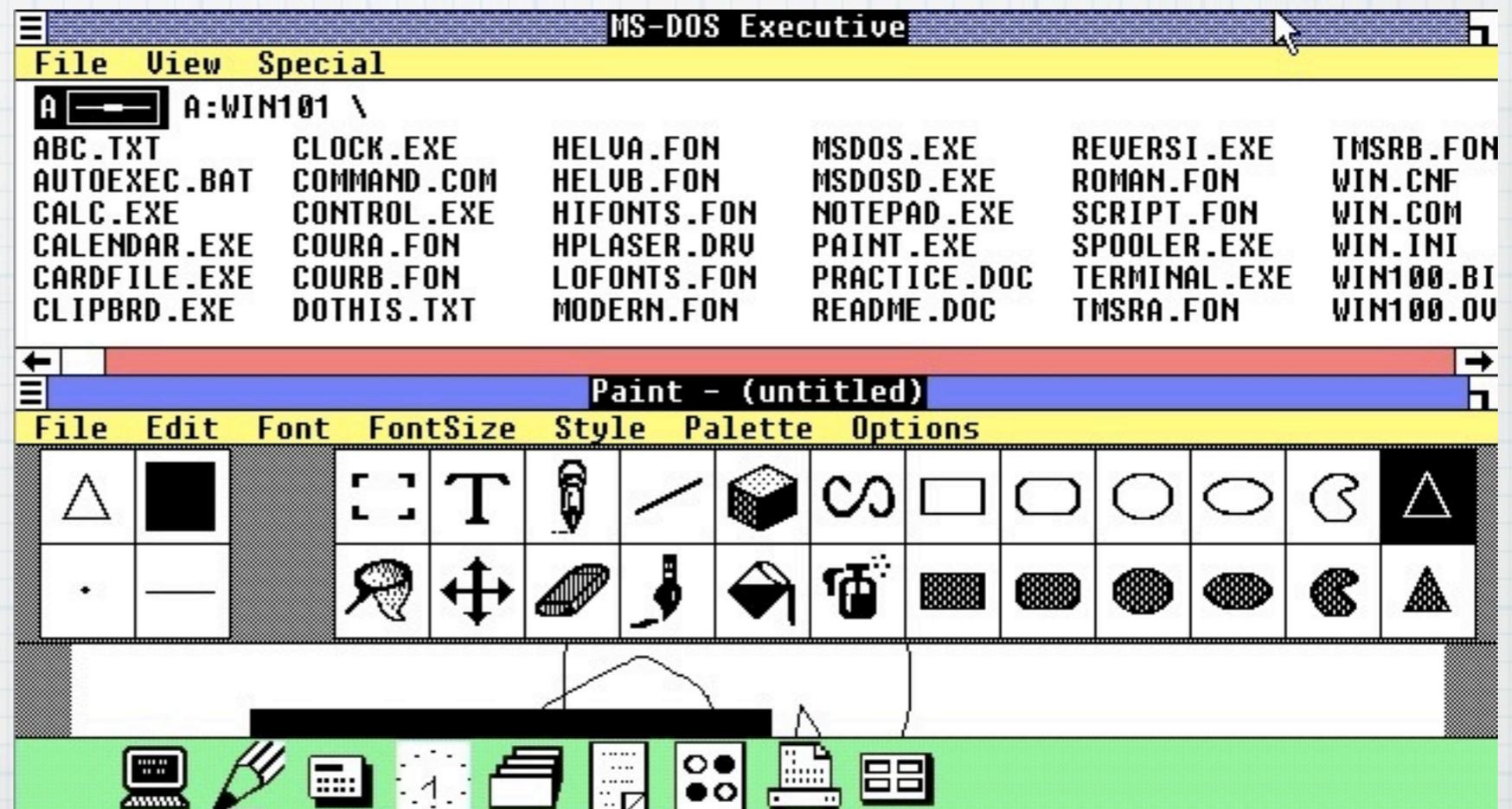
# Windows



- \* Système d'exploitation propriétaire de Microsoft.
- \* Part de marché +89% => pleins de logiciels=> plus de marché
- \* Vente lié avec PC grand public
- \* Fonctionne bien dans la génération NT.
- \* Le seul OS non Toujours pas Unix. Mais Implémente les standard de programmation POSIX.

# Histoire Windows

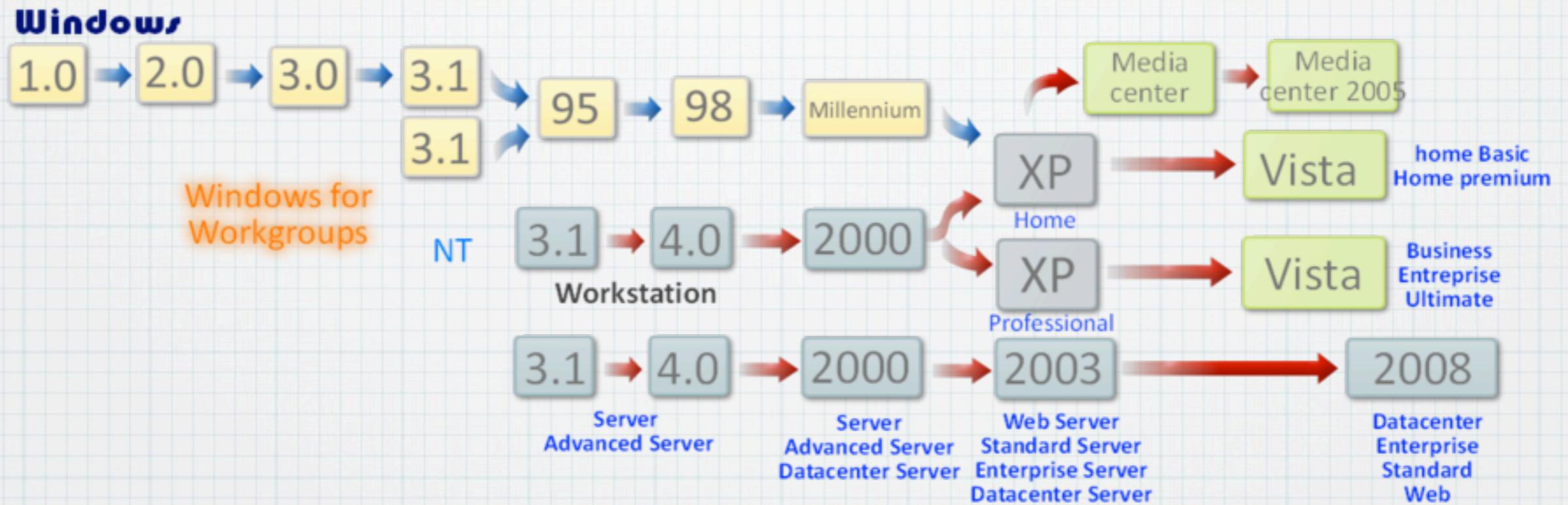
- \* 1981 : DOS (Disk Operating System) ligne de commandes Mono utilisateur et monotache  
Limite de 640k de ram
- \* 1985 - Windows 1.0
- \* Interface graphique sur DOS
- \* 1987 : Windows 2 : icon clickable et superposition fenêtres multi-tache cooperatif



# Histoire Windows

- \* 1990 : Windows 3 dépasse les 640k
- \* 1993 : Windows 3.1 1 police true type + Réseaux
- \* 1991 : Windows NT 3.1 Basé sur OS/2 de IBM II devient multi-tâches préemptif. Uniquement pour les professionnels écrit en C.
- \* 1995 : Windows 95, 32bits, commence à gérer les périphériques PnP, Dos reste un handicap.
- \* 1998 : Windows NT 4 Ecrit en 32bit cloisonnement de la mémoire.
- \* 2000: W2K NT Pour le grand publique débarque.
- \* W7 : Nt 6.1

# Schéma Windows



# Mac Os

- \* Systèmes d'exploitation d'Apple pour les machines apple.
- \* Fournis avec le matériel.
- \* La version X est basé sur une BSD et est entré dans la famille Unix.
- \* Interface soignée.
- \* Destiné à peu de configuration donc mieux testé
- \* Système conçu pour être simple.

# Chez Apple

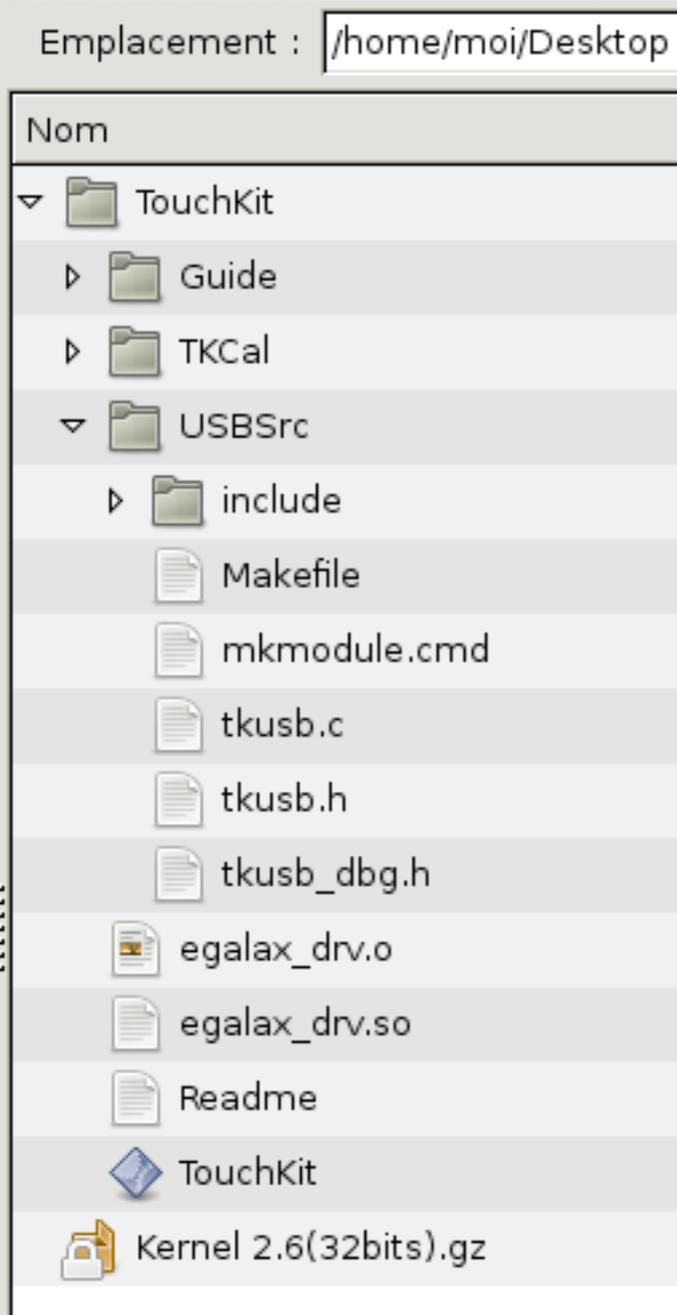
- \* 1984-1987 : Mac Os v1-4, Mono-tâches puis couleurs. Sur des Motorola 68020
- \* 1984-1987 : Mac os 5-6 Multi-tâche coopératif, Réseau et Pnp.
- \* 1991 : System 7 : Couleur ttf. 32bits.
- \* 1997 : System 8 : Multi-tâche pré-emptif, multi-threaded.
- \* 2000 : Mac Os X, Basé sur une net BSD



# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

# Gestion des fichiers



\* Fichier = Feuille avec des papier

\* Répertoire = Boite contenant des fichiers et/ou des répertoires

\* Chemin relatif = chemin à partir du répertoire courant

\* Chemin absolu = chemin à partir de la racine

\* / répertoire racine

\* ./ répertoire courant

\* ../ répertoire parent

# Répertoires Windows

- \* Différents lecteurs A: B: C: D: E: ...
- \* C:\Windows : Répertoire windows
- \* C:\Windows\System : Librairies
- \* C:\Windows\System32 : Librairies 32bits
- \* C:\Windows\inf : Information des pilotes
- \* C:\Windows\System32\drivers : lib des pilotes
- \* C:\Program Files : Programmes installés
- \* C:\Users ou c:\Document and settings Documents et paramètres des utilisateurs.

# Répertoires Unix

- \* /bin : Fichiers exécutable (rm, ls, ...)
- \* /sbin : Fichiers exécutable pour root
- \* /dev : Périphérique accès directe
- \* /etc : Configuration du système
- \* /home ou /Users: Répertoire utilisateurs
- \* ~/ : Répertoire de l'utilisateur connecté
- \* /lib /include : bibliothèques
- \* /mnt : De base mount (/etc/fstab)
- \* /media : Automount, gnome-mount etc ...
- \* /proc : processus et configuration noyau
- \* /usr : Application installé
- \* /tmp : fichier temporaires
- \* /var : fichier variables (mail, impression, log ...)

# Droit et utilisateurs

```
[smartin@daumier]$ ls -lh
drwxr-xr-x 10 smartin  ens   4,0K 2010-08-31 14:29 Bureau
-rw-r----- 1 smartin  ens  4,0M 2010-09-08 12:30 cd100627.iso
drwxr-xr-x  2 smartin  ens   8,0K 2009-11-21 11:44 dossier photo
drwx----- 3 smartin  ens   4,0K 2010-01-27 19:22 Downloads
```

\* d rwx rwx rwx

\* d=directory=répertoire ou dossier

\* User Group Other

\* Read Write eXecution

\* Read=lire

\* Write=écrire

\* eXecution=exécuter le  
fichier

\* Read=lister le répertoire

\* Write=créer un fichier

\* eXecution=traverser le  
répertoire

# Systeme de fichiers

- \* Fat 16 : < w95, très vieux système de fichier nom de fichiers 8.3, limité à 2Go, 65 518 fichiers
- \* Fat 32 : Depuis w95, Système plus récent, utilisé pour les clef usb. Nom 255 caractères ascii. Limité à 2To, 4Go /fichiers.
- \* exFat ou fat 64,  $2^{64}=16\text{Eio}$  et d'autres avantages.
- \* NTFS : Depuis Windows NT, Limite 16 Tio fichier et volume. Supporte l'utf16.

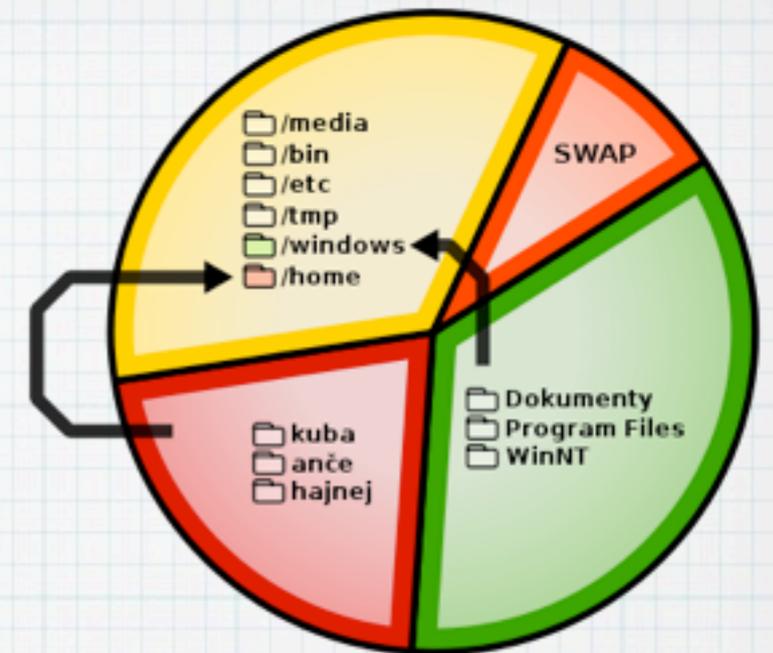
# Systeme de fichiers

- \* Ext2/3/4 : GNU/Linux, Limite la fragmentation, Limite 2 à 32 Tio, 16Gio - 2Tio (1Eio pour la v4) par fichiers. 255 caractères en Unicode.

Sensible à la casse. Utilisation d'Inode.

- \* HFS+ : Systeme de fichier Mac OS X. Limite 8Eio Volume et fichier. 255 caractères en UTF16.

# Partition HDD



- \* Découper un disque dur en plusieurs sous parties.
- \* Intérêts :
  - \* Avoir plusieurs systèmes d'exploitations
  - \* Séparation d'une zone du SE par rapport aux données.
  - \* Utiliser un disque plus gros que celui supporté par la système de fichiers.
- \* Outils : Partition Magic, Gparted, fdisk

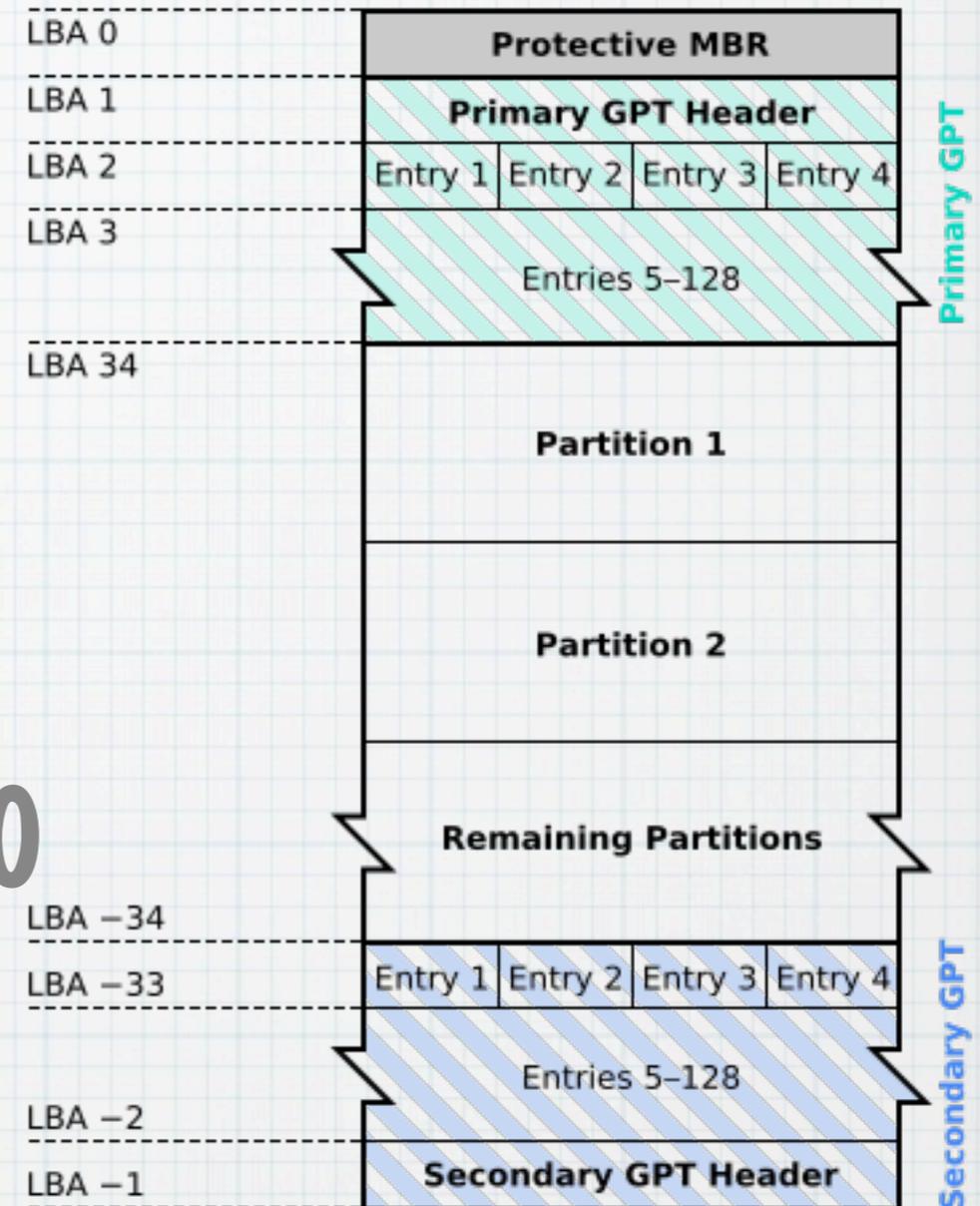
# Table partition type MBR

- \* Contient un boot strap Master Boot Record
- \* 4 partition principales 2,2 Tio /partition (Extented Boot Record).
- \* Partition Extented une partition pouvant contenir plusieurs partitions logiques non bootable.

# GPT

- \* Nouveau table de partitionnement
- \* GUID Table Partitionnement
- \* Globally  
{3F2504E0-4F89-11D3-9A0C-005E82C3301}
- \* Supporte jusqu'à 8Zio.
- \* Compatible BIOS.
- \* Pas de limite du nombre de partitions, 128 pour MS.

## GUID Partition Table Scheme



# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

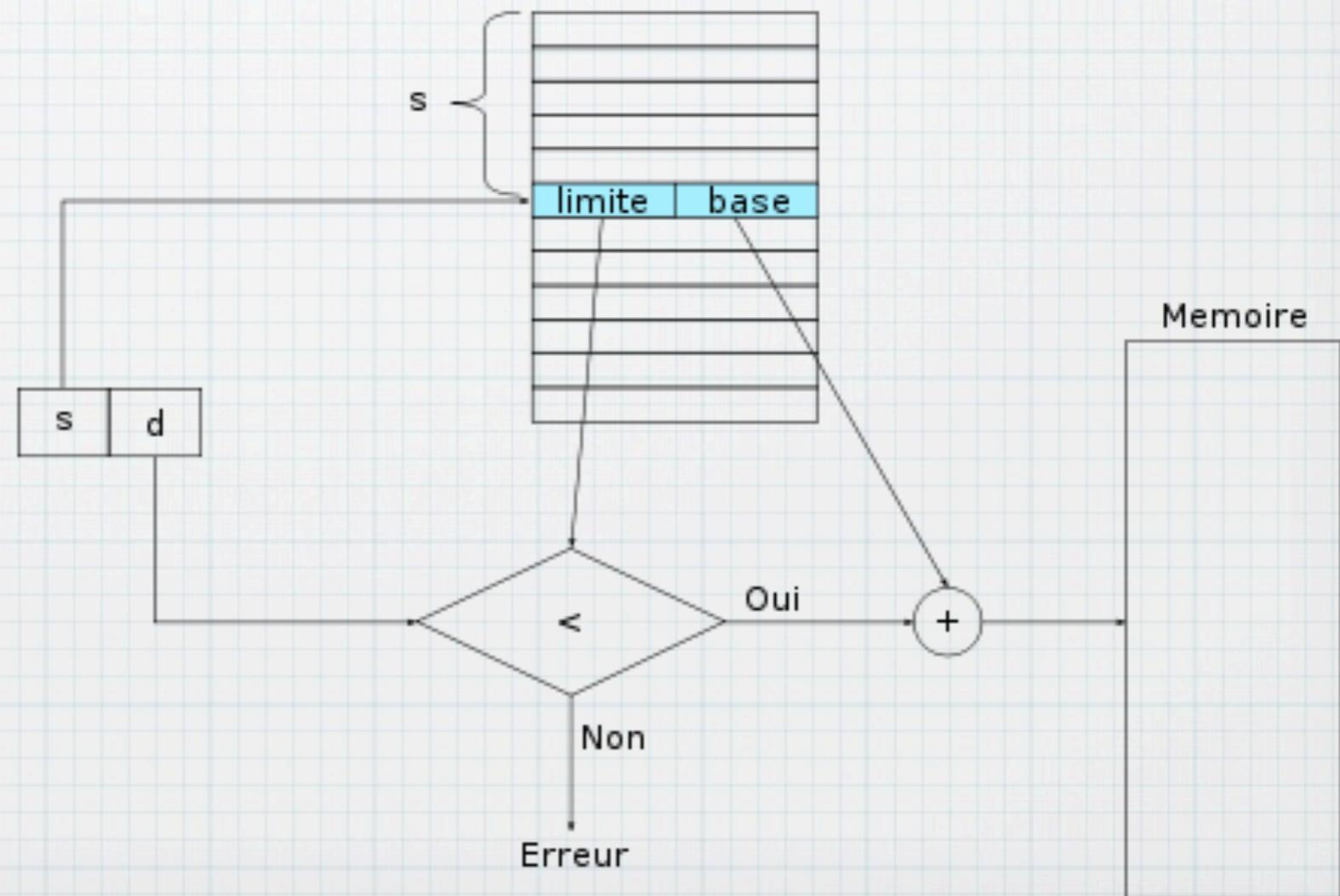
# Gestion de la mémoire



- \* Le noyau avec l'aide de la Memory Management Unit (matériel) gère la mémoire.
- \* Partage de la mémoire entre les différents processus.
- \* Cloisonne la mémoire.
- \* Sécurise l'accès.

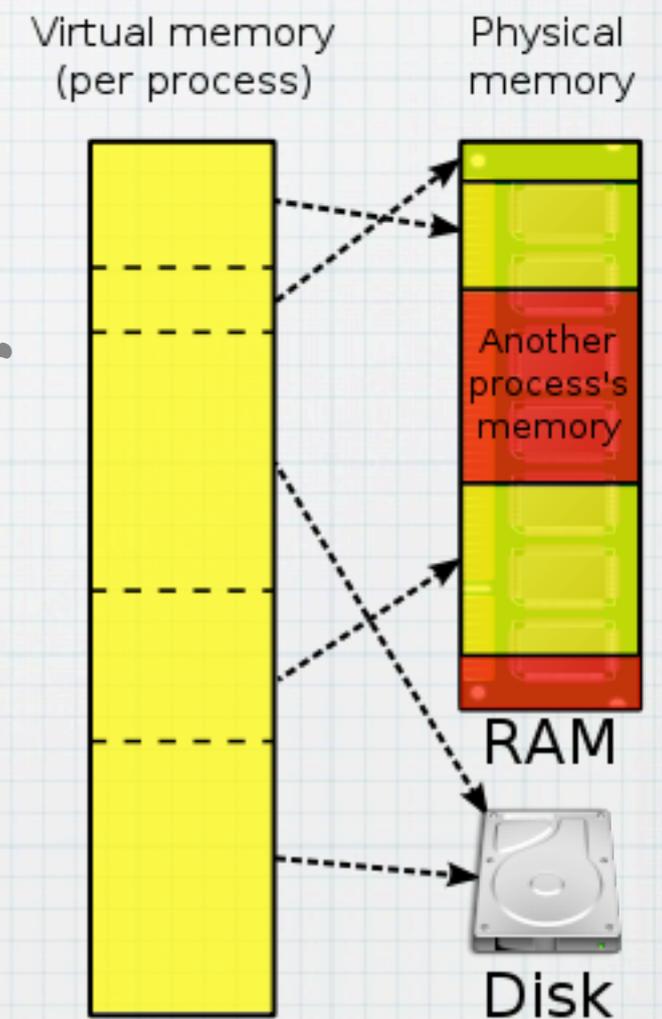
# Segmentation

- \* Permet d'allouer de dés-allouer de la mémoire dynamiquement par processus.
- \* Permet aussi de cloisonner la mémoire.



# Mémoire Virtuelle

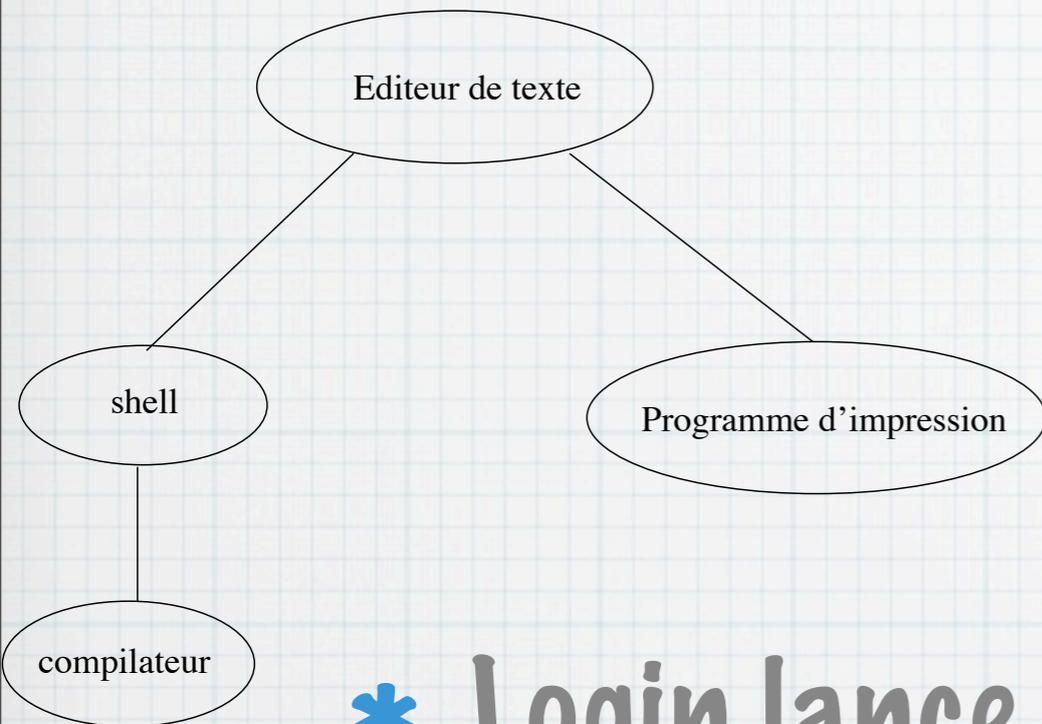
- \* Permet d'augmenter artificiellement la capacité mémoire.
- \* Fonctionne par page (pagination)
- \* Echange des pages entre la mémoire et le fichier/partition d'échange sur le disque dur.
- \* Gérer par la MMU qui envoie une interruption lorsqu'il faut charger la page.



# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

# Processus et arbres

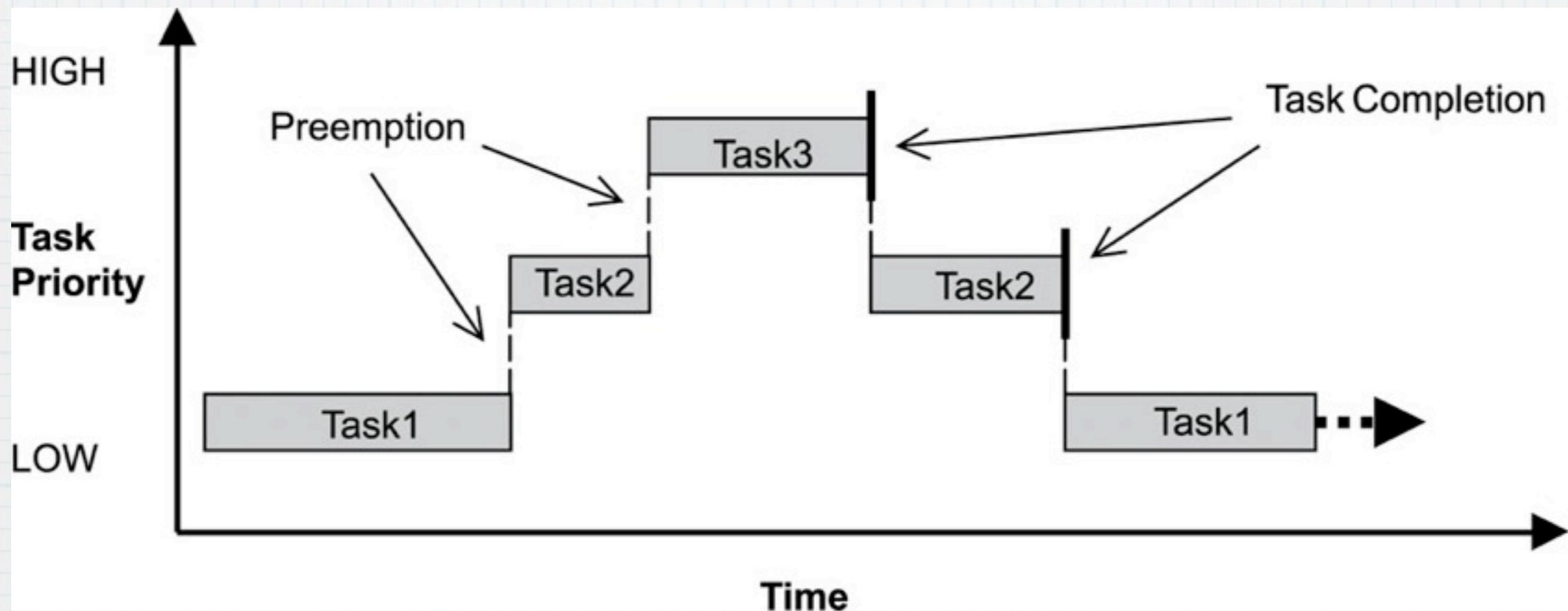


- \* Login lance le premier processus de l'utilisateur (shell, server X, explorer, Finder,...)
- \* À partir de là on lance les prochains
- \* Ainsi de suite

# Gestion des tâches

- \* Tâches gérées par le noyau
- \* Liste de tâche enregistrée
- \* SE sécurisé et stable = Système maître et peut interrompre toute tâche à tout moment
- \* Il peut interdire l'accès à certaines ressources (celle d'autrui)

# Comment faire du multi-tâches ? (préemptif)



Beaucoup plus sur que le multi-tache coopératif

# En résumé :

- \* Programme chargé en mémoire
- \* Chaque lancement de programme une zone mémoire à ne pas dépasser est allouée (segmentation)
- \* En cas de tentative de dépassement, le noyau arrête le programme
- \* Le temps d'exécution est partagé entre différents processus.

# Le noyau ne fait pas tout

- \* Services : Application lancée au démarrage sans E/S standard ou fenêtres. Ex : RPC, network Manager, Serveur Web, Maj, etc.)
- \* L'interface utilisateur n'est pas dans le noyau.
- \* Quelques milliers d'utilitaires aidant à utiliser sont OS. (grep, sed, explorer, Dock, etc.)
- \*

# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités

# Démarrage OS

- \* Amorçage (bootstrap bios ou EFI)
- \* Lancement du Chargeur de noyau (kernel loader)
- \* Montage de la partition principale ou d'une pseudo partition
- \* Chargement du noyau
- \* Initialisation des pilotes et modules
- \* Montage des partitions
- \* Lancement des services (finissant par l'interface graphique)
- \* == demande de login
- \* Lancement des programmes de démarrage.

# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités



# Installation OS

- \* Généralement, il y a un cd ou usb démarrable lançant un système d'installation.
- \* Mettre le média, démarrer dessus et se laisser guider.

# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités



# Configuration



- \* La partie la plus délicate.
- \* Généralement installer les pilotes de tout les périphérique. Si l'os ne les a pas déjà.
- \* Configurer certains de ces pilotes ou périphériques. (Résolution de conflits, port, etc.)
- \* Quelques réglage de l'OS (Réseau, Démarrage etc.)
- \* Installer, des logiciels. Peut être long.
- \* Les configurer. (Entre logiciel, ou différent réglages).

# Plan

- \* Introduction
- \* Interface
- \* Différents OS
- \* Systèmes de fichiers
- \* Mémoire
- \* Processus
- \* Démarrage
- \* Installation
- \* Configuration
- \* Particularités



# Pilotes - Drivers

- \* Pilote ou en anglais driver.
- \* Mini logiciel permettant au Système d'exploitation d'interagir avec le matériel.
- \* Fournis par le fabricant, car lui seul a la documentation technique complète. (au fond du carton)
- \* Tout Systèmes d'exploitation en ont besoins. Même si certains en ont un bon nombre.
- \* Peut-être utilisé avec différents droits.
- \* .drv et .inf windows, module noyau pour les unix (fichier ko). Sous mac il sont fournis avec un installeur.
- \* Théoriquement, ces programmes pourraient avoir des droits restreints. Mais ce n'est pas le cas. Attention !

# Particularités windows

- \* F8 au démarrage propose certaines options
- \* Utilitaire MSConfig donne beaucoup d'information sur les services et programmes lancés au démarrage.  
Utile pour éliminer certains vers.
- \* La configuration est dans le registre windows (regedit) pour la partie non accessible du panneau de configuration.

# Particularité Linux

- \* Grub configurable. Possibilité de changer les paramètres du noyau avant de lancer l'os
- \* Services sont dans /etc/init.d
- \* Les répertoires rc1.d à rc6.d contiennent des raccourcis de services à lancer ou à arrêter.  
Utilitaire : update.rc dépend de la distribution.
- \* Configuration : différents répertoire de /etc/

# Particularité Mac OS X

- \* Démarrages : Pomme+s : Single user, pomme+v : version verbeux
- \* 9/10 de configuration se fait par une interface graphique.
- \* 1/10 Se fait comme un système unix (édition de fichiers et syntaxe pas toujours claire)

Des questions ?