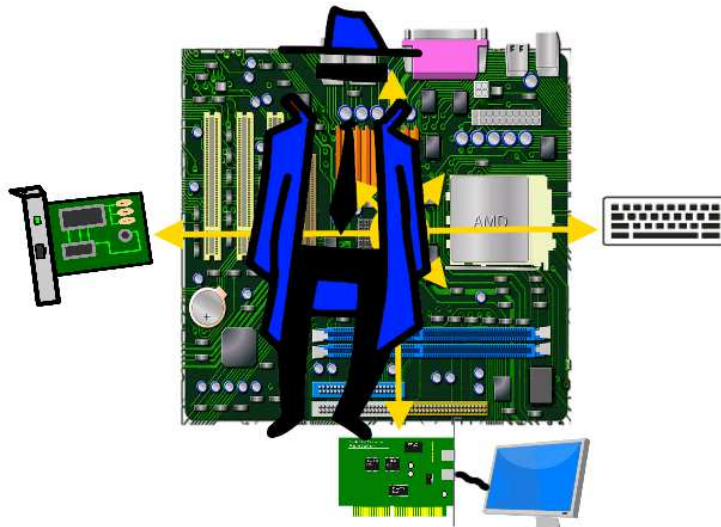


# SYSTEME D'EXPLOITATION



## Baccalauréat STI2D - SIN

- SIN 2.1 : Traitement des informations numériques
- SIN 3.1 : Implémentation d'un programme dans un composant programmable
- SIN 3.2 : Mise à jour d'un système d'information

## Objectifs

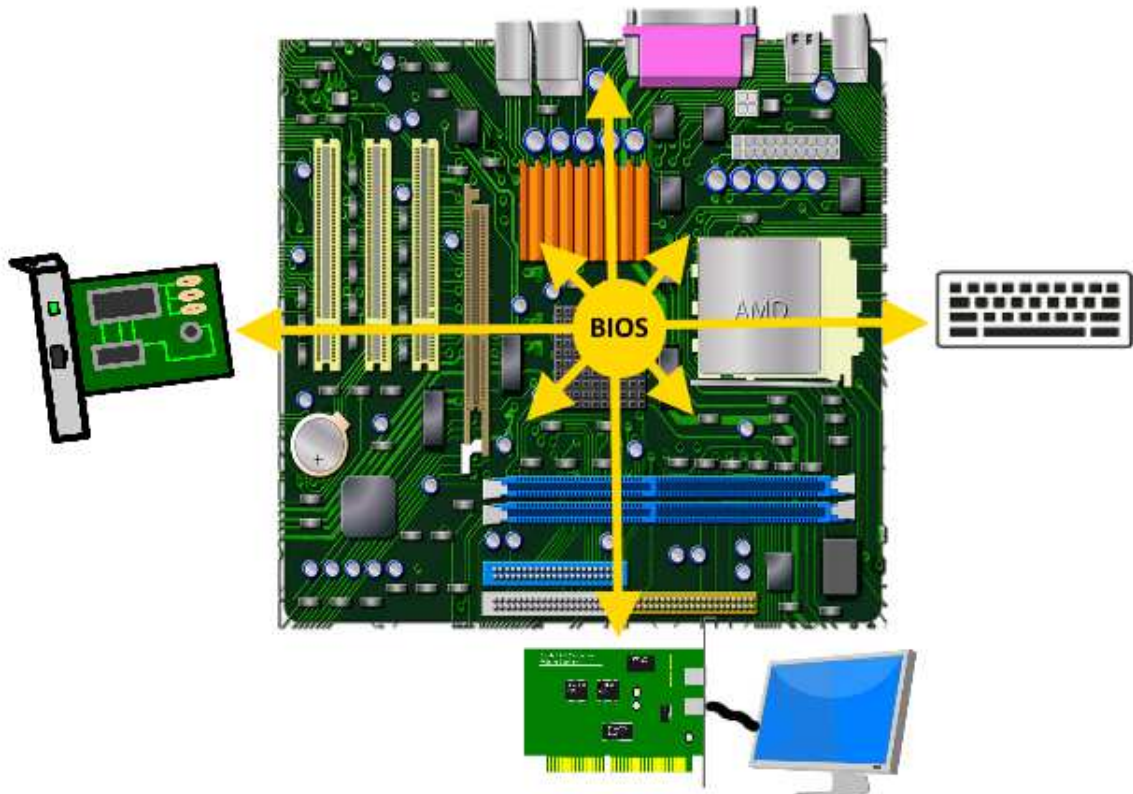
L'apprenant doit être capable de :

- Justifier la nécessité d'un système d'exploitation en citant ses fonctions principales
- Citer quelques systèmes d'exploitations courants
- Installer et mettre à jour un système d'exploitation

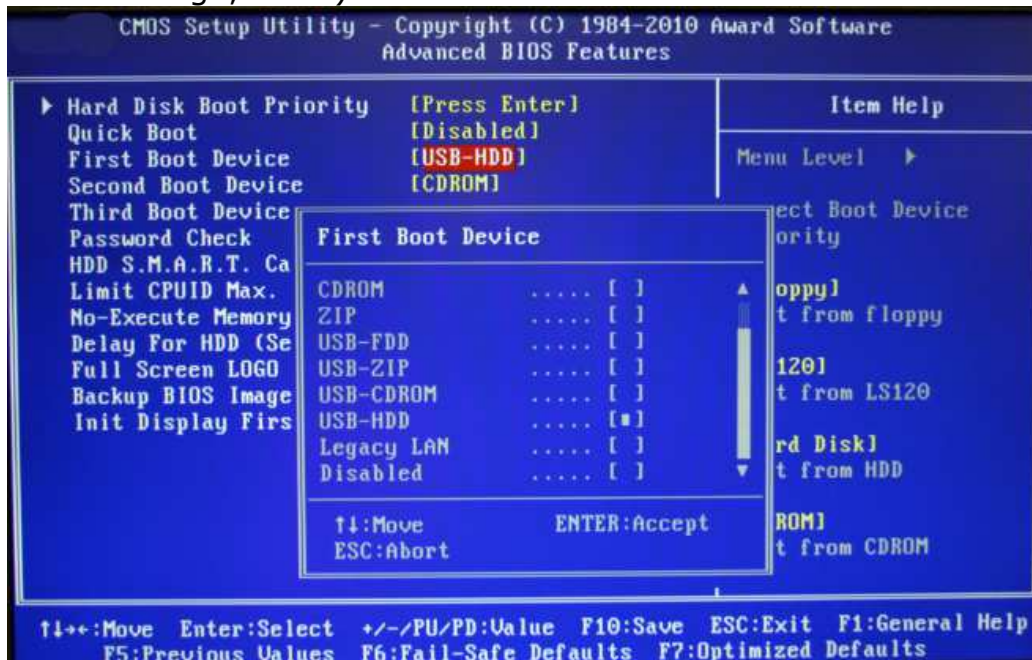
## Fonctionnalités

### Basic Input Output System

Pour qu'un microprocesseur, puisse gérer les périphériques d'entrée sortie et certaines données du système, un système de traitement minimal, (une carte mère par exemple) doit posséder un programme intégré non volatile appelé BIOS. Ce microprogramme démarre dès la mise sous tension du système, appartient au matériel et est donc développé par le concepteur de la carte matérielle.



Souvent le BIOS dispose d'une interface utilisateur qui permet de définir certains paramètres (mot de passe de démarrage, heure et date, média de démarrage, etc...)



Interface utilisateur d'un BIOS Award

Le BIOS ne permet pas d'exécuter des programmes personnels de l'utilisateur, il établit juste les liaisons logicielles entre les différents éléments matériels du système.

En particulier, il donne l'accès au média qui comporte le système logiciel capable de gérer le système de traitement - l'ordinateur.

## Démarrage d'un système

Lors du démarrage d'un ordinateur, différentes actions sont réalisées

- copie du BIOS dans la mémoire
- exécution du POST (Power-On Self Test) :
  - Vérification du processeur et de l'alimentation
  - Vérification de l'intégrité du BIOS (checksum)
  - Vérification du fonctionnement correct de la mémoire
  - Détection et vérification des périphériques
  - initialisation des périphériques (clavier, video,...)
  - initialisation de l'ordre de priorité des médias d'entrée ce qui détermine où se trouve le bootloader
- démarrage du système d'exploitation

## Le système d'exploitation

Le système d'exploitation (**Operating System** en anglais) met à la disposition de l'utilisateur un interface qui lui permet d'utiliser le système de traitement de données. Il n'y a pas que les ordinateurs qui ont des OS; les switch, routeurs, machines outils, smartphones, etc....

L'OS fournit :

- l'interface utilisateur
- la gestion des entrées/sorties
- l'exécution de programme
- la gestion des fichiers
- la communication
- la gestion des erreurs



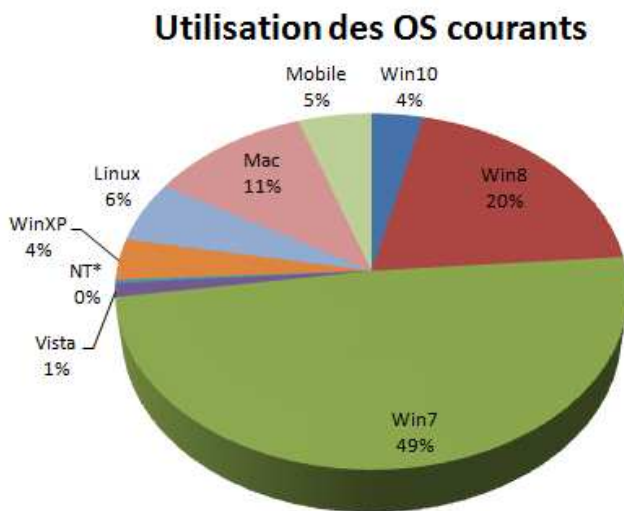
Il est généralement composé de trois parties (3 anneaux de privilèges) :

- **le noyau (kernel)** : Toujours présent dans un OS, il gère les

ressources et permet la communication entre le matériel et le logiciel. Il fonctionne dans le mode de privilège le plus élevé (*mode privilégié*).

- **l'interface système (shell)** permet soit de manière textuel (CLI) soit graphique de saisir des commandes pour le système de traitement de données. Souvent présent.
- **l'interface graphique (GUI)** : il s'agit de l'interface permettant le dialogue homme-machine par manipulation de pictogrammes qui représentent les ressources du système. Pas nécessairement présent.

## Principaux systèmes d'exploitations Utilisation en Aout 2015



Sources [http://www.w3schools.com/browsers/browsers\\_os.asp](http://www.w3schools.com/browsers/browsers_os.asp)

### Quelques caractéristiques

Nom	Architecture supportée	Systèmes de fichiers	Type de noyau	Nbre de lignes de codes	Interface graphique	API natif
AmigaOS 4	PowerPC	Propriétaire (OFS, Microkernel, FFS, SFS, PFS), JXFS, FAT, ISO 9660, UDF, et plus avec l'ajout de drivers			Oui	Proprietary
FreeBSD	x86, x86-64, ARM, MIPS, PowerPC, others	UFS2, ZFS, ext2, ext3, FAT, ISO 9660, UDF, NFS, ZFS, others	Monolithic with modules	6.25 million[16]	No	BSD/POSIX

Nom	Architecture supportée	Systèmes de fichiers	Type de noyau	Nbre de lignes de codes	Interface graphique	API natif
Linux	x86, x86-64, ARM, PowerPC, SPARC, autres	ext2, ext3, ext4, btrfs, ReiserFS, FAT, ISO 9660, UDF, NFS, and others	Monolithic avec modules	~15 million (kernel)	Dépend de la distribution	Linux/POSIX
iOS	ARM	HFS+, FTP	Hybrid	~80 million	Oui	Cocoa, BSD-POSIX
OS X	PowerPC, x86, x86-64, iOS for ARM	HFS+ (default), HFS, UFS, AFP, ISO 9660, FAT, UDF, NFS, SMBFS, NTFS (read only), FTP, WebD AV	Hybrid	~86 million[18]	Oui	Carbon, Cocoa, Java, BSD-POSIX
OS/2	x86	HPFS, JFS, FAT, ISO 9660, UDF, NFS	Monolithic avec modules		Oui	Proprietary, DOS API, Win16
Windows Server (NT family)	x86, x86-64, IA-64	NTFS, FAT, ISO 9660, UDF; 3rd-party drivers support ext2, ext3, ReiserFS,[t 10]and HFS	Hybrid	~45 million	Oui	Win32, NT API
Windows(NT family)	x86, x86-64, ARM	NTFS, FAT, UDF; avec driver supplémentaire ext2, ext3, ReiserFS, HFS+, FATX, et HFS	Hybrid	~40(XP)/64(Vis ta et après) million	Oui	Win32, NT API

## Vocabulaire

**ACPI** : Advanced Configuration Power Interface : norme industrielle qui définit les les fonctionnalités de gestion de l'alimentation (économie d'énergie en particulier) et autres informations de configuration

**BIOS** : Basic Input Output Système. Microprogramme qui réalise les liaisons logiques entre le matériel et permet le démarrage du système d'exploitation

**Boot Loader** : il s'agit du code permettant de lancer le système d'exploitation.

**CLI** : **Command Line Interface** : interface logiciel permettant la saisie de commande par du texte (CMD.EXE ou POWERSHEL.EXE sous windows)

**Flasher le BIOS** : opération qui consiste à changer la version du BIOS.

**Mode privilégié** : Plus le niveau de privilège est élevé et plus les opérations au niveau du système sont autorisées. Certaines opérations du mode privilégié (mode kernel) ne sont pas possibles en mode normal (GUI par exemple)