

## 7 – ARCHITECTURE – SYSTEMES D'EXPLOITATION

### Compléments sur le cours de Bonnefoy & Petit

#### Points d'histoire

1945 : **premier ordinateur** : l'ENIAC

1981 : IBM sort le **premier PC** – MS-DOS est le **premier système d'exploitation grand public** pour PC

1985 : Microsoft lance **Windows 1.0**

2018 : record de vitesse d'un microprocesseur : **7,6 GHz**

#### Définitions

**Système sur puce** (« **SoC** » : System on a Chip) : ensemble d'éléments différents (processeurs, mémoires, interfaces,...) intégrés sur une même puce, et propre à une application.

Exemple : le SoC d'un smartphone en assure les principales fonctions (système d'exploitation, communications, WiFi, Bluetooth, affichage, son, vidéos, ...).

Contre-exemple : un microprocesseur seul n'est pas un SoC : c'est un élément standard que l'on peut trouver sur de nombreux appareils. Il n'est pas spécifique à une application particulière.

**Processus** : programme en cours d'exécution. Un même programme peut donner lieu à plusieurs processus.

**Tâche** ou **fil d'exécution** ou "**thread**" : séquence d'instructions d'un processus. Un processeur ne peut exécuter qu'une seule tâche à la fois (sauf s'il est multicœurs).

**PID** (Process Identification) : numéro attribué à un processus par le système d'exploitation.

**PPID** : numéro du processus père.

**Interblocage** (« deadlock ») : situation dans laquelle deux ou plusieurs processus s'attendent mutuellement. Sans intervention extérieure, cette situation est définitivement bloquée.

#### Divers

**Avantages d'un « SoC » :**

- Optimisation (circuit fabriqué « sur mesure » et répondant exactement aux besoins)
- Rapidité (grâce à la réduction des distances)
- Gain de place
- Economie d'énergie (moins de sources d'alimentation qu'avec des circuits intégrés multiples)
- Economie de matière

**Inconvénients :**

- Longue durée de mise au point
- Coût élevé d'investissement (rémunération des concepteurs, masques, nouvelle chaîne de fabrication)

.../

### Fonctions d'un système d'exploitation :

- fournir une **interface** entre l'humain et le processeur,
- **gérer les ressources** de l'ordinateur : mémoires, périphériques, énergie,...
- gérer les utilisateurs et leurs **droits d'accès**, les fichiers, les processus.

**NB :**

### 1. Ordonnancement

Un **processus** est un programme dont on demande l'exécution. On repère essentiellement 3 états d'un processus : **prêt, élu, bloqué**. Une fois le processus terminé, il disparaît.

Afin de gérer plusieurs processus en même temps, le système d'exploitation possède un outil spécifique : **l'ordonnanceur**.

**Algorithmes d'ordonnancement** des processus (cf manuel p.172-173) :

- en « **tourniquet** »
- par **priorité**
- en **FIFO**
- par le **plus court d'abord**
- etc

### 2. Commandes linux :

- gestion de dossiers : **mkdir, rmdir, cd, ../(dossier courant), ../(dossier parent)**
- gestion dossiers & fichiers : **mv** (déplace et/ou change le nom), **ls** (liste contenu), **cp** (copie), **touch** (crée fichier), **rm** (supprime fichier), **cat** (crée du texte dans un fichier)
- gestion des droits : **chmod, u**(user), **g**(group), **o**(others), **a**(all), **r, w, x**

Exemples : « **cd ../** » , « **touch rép1/fich3.txt** » , “**chmod go+x fich1.txt**” , etc.