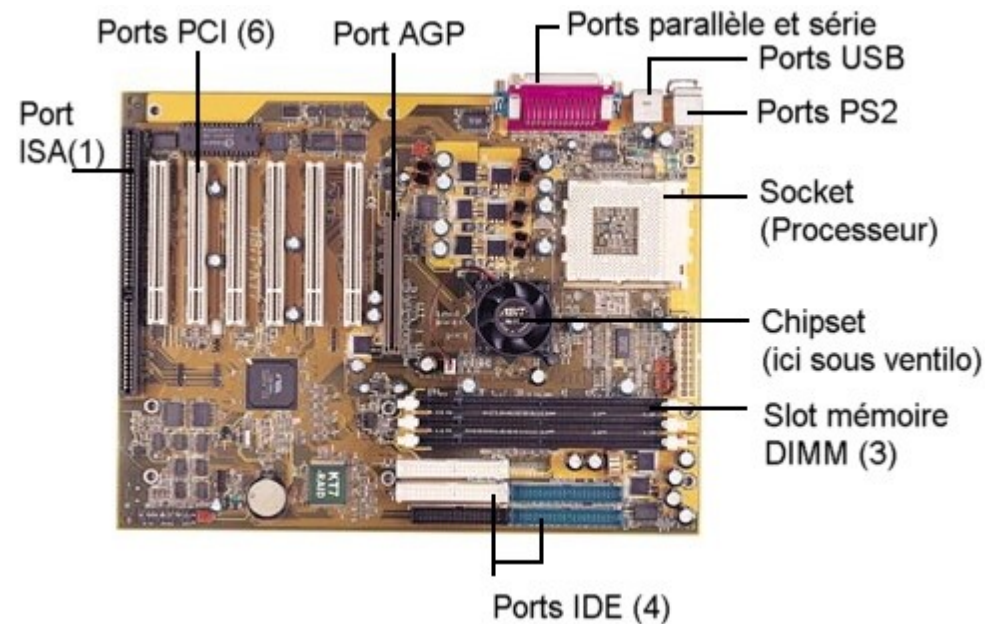


Le groupe est :

<https://groups.google.com/group/SE1LMD>

Systemes d'exploitation



Carte mère d'un ordinateur

Rôle des SE



- SE = ensemble des composantes logicielles qui font le pont entre les **applications** et le **matériel**
- Offre des **services abstraits** aux applications (p.e. ouvrir le fichier "abc") et demande des **services concrets** du matériel (p.e. lire le secteur N du disque)
- Responsable de l'exécution des applications et la gestion des ressources (matérielles et logiques)

Attraits des SE

- **Facilite la réalisation** des applications (services prédéfinis)
- Applications plus portables (abstraction du matériel); le SE est une **machine virtuelle**
- Uniformise l'interface usager des applications (qu'il soit visuel, ou autre) ce qui contribue à la **convivialité**
- Centralise les politiques de gestion des ressources ce qui permet d'en améliorer l'**efficacité** et de **protéger** l'accès aux ressources privées ou critiques

Objectifs des SE

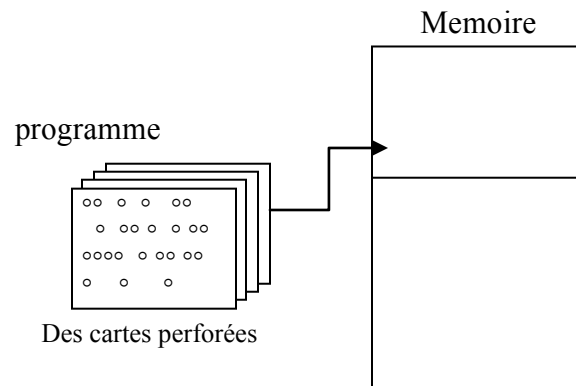
- Rendre l'ordinateur **plus convivial** (plus facile à utiliser)
- Accroître l'**efficacité** de l'ordinateur (meilleure exploitation des ressources)
- Ce sont des **objectifs contradictoires** qui mènent à des compromis dans le SE (p.e.: interfaces usagers graphiques, multi-usagers)

Composantes Fondamentales d'un SE

- Le logiciel central du SE est le noyau (“**kernel**”); celui-ci est normalement toujours chargé en mémoire
- Le “**bootstrap loader**” s’occupe de charger le kernel en mémoire lorsque l’ordinateur est mis en marche (à la mise sous tension ou “reset”)
- Pour chaque type de périphérique (disque dur, carte réseau, carte graphique, carte de son, etc) il y a une composante qui est responsable de son contrôle; les “**device drivers**”
- Pour chaque type de ressource (CPU, mémoire, imprimantes, lien réseau, etc) il y a une composante qui est responsable de sa gestion; les **allocateurs de ressources**

Au Tout Début

- Les premiers ordinateurs n'avaient **pas de SE résidant**
- Le programme était **chargé en mémoire** (d'un ruban perforé, cartes, ou autre) à un emplacement préétabli et puis son exécution lancée à son **point d'entrée**



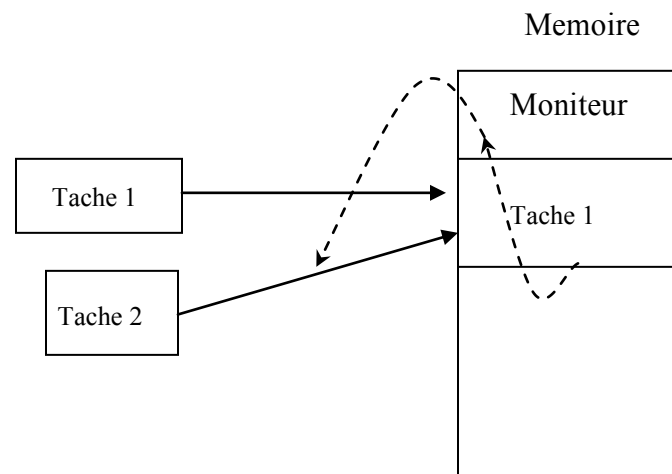
- Des **librairies** de routines (principalement pour diverses opérations d'E/S) ont vu le jour pour éviter à les écrire à nouveau pour chaque nouveau programme

Traitement en Lot

- Plus tard l'utilisation d'ordinateurs pour traiter des lots de tâches (“**batch**” processing) devint la norme
- Déf: une **tâche** est un ensemble de calculs à effectuer; une **sous-tâche** c'est une partie d'une tâche; la **taille** d'une tâche c'est la quantité d'opérations à effectuer
- Typiquement, chaque tâche correspondait à la lecture de données d'un ou plusieurs rubans magnétique et l'écriture de données sur ruban après un traitement relativement simple
- Pour **minimiser le temps d'inactivité de l'ordinateur** lorsque l'opérateur humain montait les rubans requis par le programme, il était profitable de grouper dans un même lot des tâches qui utilisaient les mêmes rubans
- De nos jours **les E/S sont toujours relativement lentes** et demandent une considération spéciale

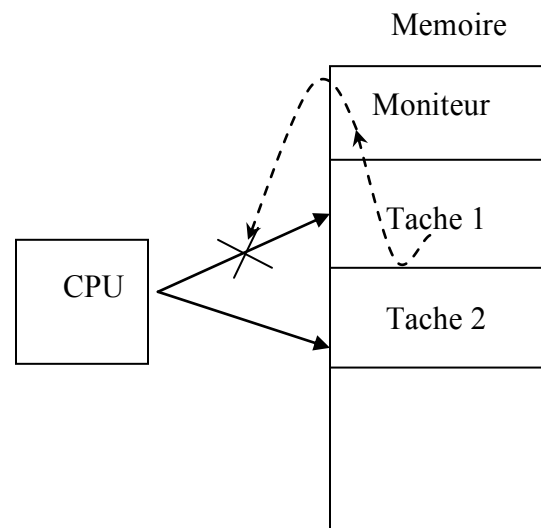
Moniteur

- Les premiers SE (“**moniteur**” résidant) automatisaient cette procédure en enchaînant simplement l'exécution de chaque tâche
 - le moniteur charge la première tâche puis y transfère l'exécution
 - à la fin de cette tâche le contrôle est redonné au moniteur
 - le moniteur charge la deuxième tâche puis y transfère l'exécution, etc



Multiprogrammation

- Le prochain pas fut la **multiprogrammation**: plusieurs tâches sont chargées en mémoire; le CPU s'occupe seulement d'une tâche à la fois; lors d'une opération d'E/S qui ne peut être complétée immédiatement le moniteur fait passer le CPU à la prochaine tâche (**multiplexage du CPU**):



- Ceci permet au CPU de **faire des calculs utiles** plutôt que de d'être inactif pendant les opérations d'E/S

Multi-tâche (“Multitasking”)

- Le **multi-tâche** (“multitasking” ou “time sharing”) est une extension de la **multiprogrammation** pour permettre à **plusieurs usagers** d'utiliser simultanément un même ordinateur (au moyen d'un terminal)
- Le CPU passe rapidement d'une tâche à la suivante à intervalles réguliers (“**quantum**”)
- Chaque usager (tâche), reçoit une fraction de la puissance de l'ordinateur, a l'illusion d'être le seul sur la machine
- Le SE peut permettre un nombre de tâche qui **dépasse la quantité de mémoire physique**
- L'idée de la **mémoire virtuelle** c'est que seules les données qui sont présentement traitées ont besoin d'être en mémoire physique (pour un accès rapide)
- Le SE présente à chaque tâche l'illusion d'une mémoire qui lui est propre (**mémoire logique**)

SE Temps-Réel

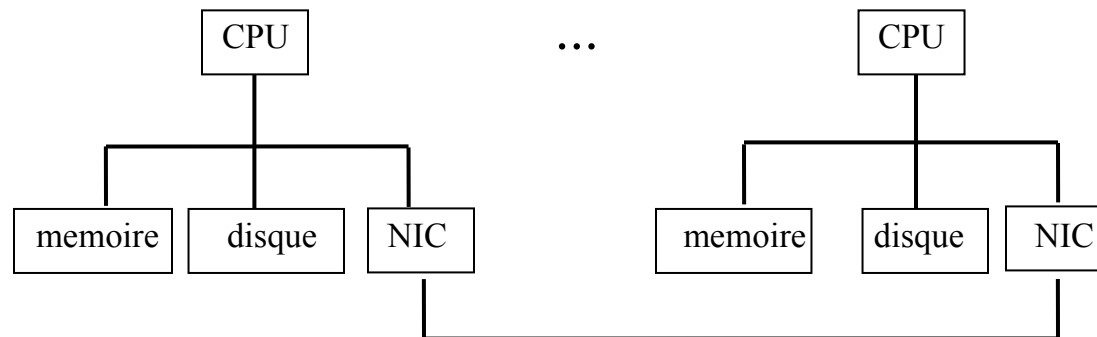
- Dans un **SE temps-réel** le SE place des garanties sur le temps de réponse (le temps entre l'occurrence d'un événement et la réaction par le système)
 - **“Hard real-time”**: temps de réponse maximal garanti
 - **“Soft real-time”**: le temps de réponse est court statistiquement
- Domaines: contrôle (robotique, freins ABS, machinerie), jeux vidéo, multimédia, télécommunication, serveurs
- Pour atteindre des contraintes de temps serrées, il faut normalement abandonner les aspects plus avancés (système de fichier, mémoire virtuelle, multi-tâche)

SE Parallèle

- Les **SE parallèles** gèrent les ressources partagées d'un multiprocesseur (CPUs, mémoire, périphériques)
- Avantages des multiprocesseurs:
 - **débit plus élevé**: en théorie il est possible d'accélérer le traitement jusqu'à un facteur de N avec N processeurs (cela dépend du **degré de parallélisme** de la tâche)
 - **économies d'échelle**: un CPU à très haute vitesse est plus cher que deux de moitié vitesse
 - **tolérance aux pannes**: en théorie, la panne d'un CPU ne fait que réduire légèrement le débit si le SE peut tolérer à la panne

SE Distribué

- Dans un **SE distribué** plusieurs processeurs sont interconnectés en réseau (de type LAN ou WAN) et la gestion des ressources se fait par **échange de messages**



- La **sécurité** et **efficacité** sont des considérations importantes dans les SE distribués.