

INF4033 – Programmation système et temps réel

Cours 1 - Introduction

Alexandre BRIÈRE



6 séances de cours/TP (18h) :

- ① Rappels sur les OS et l'environnement Linux
- ② Interactions avec l'OS en C et en Shell
- ③ La gestion des fichiers
- ④ Les processus
- ⑤ Les threads
- ⑥ Les communications inter-processus (IPC)

⇒ Évaluation via les compte-rendus de TP

Qu'est-ce qu'un système d'exploitation ?

Interface entre le logiciel et le matériel :

- séparer les applications des spécificités du matériel :
 - abstraction logicielle
 - portabilité
- masquer aux applications les limitations du matériel :
 - ressources matérielles limitées
 - ressources logiques en théorie infinies
- protéger le matériel des applications

Pourquoi un système d'exploitation ?

- Les détails techniques sont complexes
 - Chaque programmeur doit-il connaître le fonctionnement de l'ensemble des périphériques qu'il utilise ?
- Les ressources doivent être équitablement partagées entre les applications
 - Utilisation d'une imprimante
 - Ne pas mélanger les flots de caractères
 - Ne pas faire attendre inutilement le programme ayant demandé l'impression

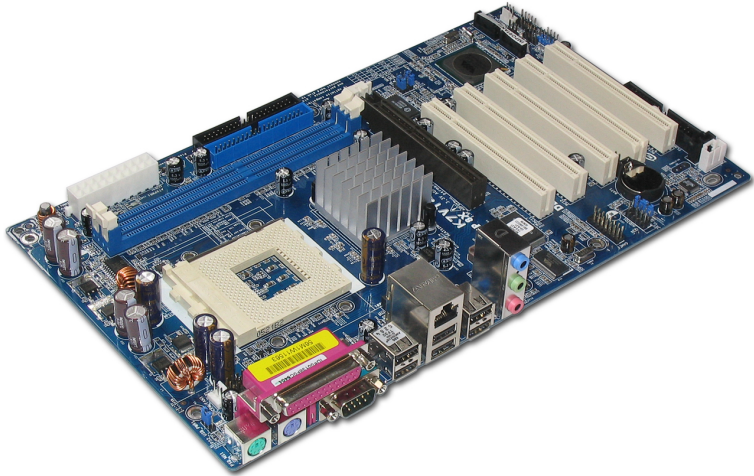
Pourquoi étudier les systèmes d'exploitation ?

- Pour les utilisateurs
 - Meilleure maîtrise
- Pour les programmeurs
 - Meilleure efficacité
 - Ne pas réinventer la roue : on rencontre souvent les mêmes problèmes

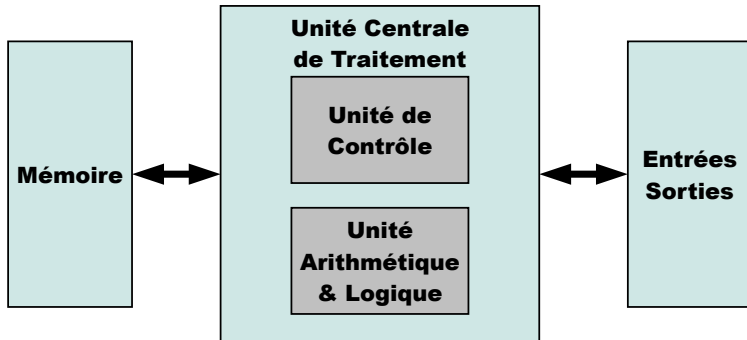
Différents OS pour différentes utilisations

- Systèmes pour supercomputer
 - Machines massivement parallèles
 - Quantité importante de ressources de calcul/communication à gérer
- Systèmes pour serveurs
 - Beaucoup d'utilisateurs
 - Forte utilisation du réseau
- Systèmes personnels
 - Interface conviviale/ergonomique
 - Utilisation « basique »
- Systèmes temps réels
 - Respect des contraintes temporelles
 - Temps réel « dur » vs. temps réel « mou »
- Systèmes embarqués
 - Interface conviviale/ergonomique, ou pas...
 - Contraintes matérielles

Qu'est-ce qu'un ordinateur ?



Architecture de von Neuman



Gestion des différents niveaux de mémoire, temps d'accès et capacité :

- Registre du processeur (~ 1 ns et ~ 1 kio)
 - Le compilateur
- Caches (~ 5 ns et ~ 1 Mio)
 - Composant matériel lié au processeur
- Mémoire centrale (~ 10 ns et ~ 1 Gio)
 - Le système d'exploitation
- Disque (~ 10 ms et ~ 1 Tio)
 - Le système de fichier (OS)

- Avec qui ?
 - Tout ce qui entre/sort de l'ordinateur (disque, clavier, réseau, etc.)
- Est-ce simple ?
 - L'OS doit gérer les flux de données entre le CPU et les périphériques
 - Les périphériques sont beaucoup plus lent que le CPU
 - Les périphériques peuvent être asynchrones
- En pratique, trois solutions :
 - Attente active
 - Interruptions
 - Accès direct à la mémoire

Attente active (polling)

- Principe
 - attendre la fin de l'opération en interrogeant régulièrement la ressource afin de déterminer si elle a effectivement fini
- Efficacité :
 - Assez faible
 - Pendant toute la durée de l'attente, l'UC ne fait rien d'autre

- Principe
 - Réception d'un message lorsque la ressource est effectivement disponible
- Efficacité :
 - Assez bonne
 - Pendant toute la durée de l'attente, l'UC peut travailler sur autre chose
- Déroulement :
 - Sauvegarde du contexte d'exécution du programme en cours
 - Identification de l'interruption en question et lecture du programme destiné à gérer l'interruption
 - Exécution la routine d'interruption
 - Restauration du contexte d'exécution du programme interrompu
 - Reprise de son exécution
- Attention à ne pas abuser des interruptions !

Accès direct à la mémoire (DMA)

- Technique fournissant un chemin direct entre le bus d'E/S et la mémoire sans intervention du processeur
- Soulage l'UC du contrôle des entrées/sorties
- L'exécution du programme sur l'UC continue pendant le transfert DMA
- Si conflit d'accès à la mémoire, le DMA est prioritaire

Rôle d'un système d'exploitation

- Interface entre les applications et le matériel
- Gestion des processus/applications
- Gestion des ressources pour le compte de processus/applications
 - CPU, mémoire, fichiers, etc.
- Protection vis-à-vis des processus :
 - Protège les ressources matérielles des processus
 - Protège les processus les uns des autres
- Décomposition du système d'exploitation en sous-systèmes

Les différents sous-systèmes d'un OS

- Gestion des fichiers
 - robustesse et sécurité
- Gestion de la mémoire
 - affectation et protection
- Gestion des processus
 - création, ordonnancement et destruction
- Gestion des threads (processus légers)
 - création, ordonnancement et destruction
- Gestion des communications
 - entre processus et entre machines
- Gestion des entrées/sorties
 - contrôle des périphériques et unification des interfaces