

Systèmes d'exploitation - Introduction

Pierre Gançarski
DUT Informatique - S31

ATTENTION

Ces transparents ne sont qu'un guide du cours : de nombreuses explications et illustrations manquent.

De nombreuses précisions seront données au tableau et à l'oral pendant le cours.



Plan

① Introduction

② Structure et concepts de base

③ Plan du cours

Pourquoi un S.Ex?

Matériel

- Structure classique d'un ordinateur actuel :
 - un ou plusieurs processeurs
 - des mémoires : centrale, caches, ...
 - des périphériques : clavier/souris, écran, imprimante ...
 - des disques durs,
 - un accès réseau ...

reliés par un bus.

Usage

- Les utilisateurs :
 - sont rarement des informaticiens
 - peuvent être plusieurs sur une même machine
 - veulent pouvoir faire chacun plusieurs choses en même temps
 - sont imprévisibles
 - peuvent être “malfaisants”
 - ...

Pourquoi un S.Ex ?

Rôle d'un Système d'exploitation

- Faire l'interface entre ce matériel et les utilisateurs
 - Gérer le matériel et les ressources associées :
 - * masquer des éléments fastidieux liés au matériel, comme les interruptions, les horloges, la gestion de la mémoire, la gestion des périphériques (déplacement du bras du lecteur de disque) ...
 - * assurer l'ordonnancement et le contrôle de l'allocation des processeurs, des mémoires et des périphériques d'E/S
 - * stocker de façon pérenne les informations : fichiers ...
 - Rendre les choses simples et uniformes pour l'utilisateur, le programmeur et les applications : API génériques

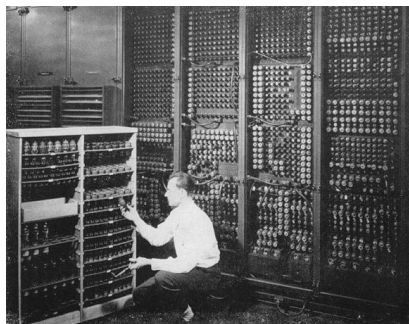
Pourquoi étudier les S.Ex ?

Pourquoi ?

- Permet de mieux comprendre sa machine
- Un des logiciels des plus complexes → Illustration de bcp de problèmes techniques et théoriques (ordonnancement, partage de ressources, ...), de génie logiciel ...
- Aspects intéressants :
 - Optimisation : performances du système sont cruciales
 - Fiabilité : évident !
 - Compromis : tout faire à peu près bien ou se focaliser sur certains cas
 - Heuristiques : deviner ce que veut l'utilisateur, s'y adapter, s'y préparer à l'avance

1945-1955 : tubes à vide

- Pas de système d'exploitation : programmation par interrupteurs.
- L'équipe conçoit, construit, administre et maintient la machine.
- Langages de programmation inexistant (y compris l'assembleur).
- Le programmeur s'inscrit pour une durée de temps dans la salle machine (et prie pour qu'aucun des 20000 tubes ne grille...).



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

- Essentiellement des calculs numériques

1955-1965 : transistors et traitements par lots

Cartes imprimées

- Le transistor remplace le tube à vide : les ordinateurs sont plus fiables.
- Programmation sur papier → codage sur cartes perforées → soumission à l'opérateur en salle machine (si le compilateur nécessaire : chargement préalable) → résultat sur une imprimante.



Mainframe IBM 7094



Carte perforée FORTRAN



Appareil de perforation de cartes

Apparition des premiers systèmes

- FMS (Fortran Monitor System) et IBSYS (SE de l'IBM 7094).

1965-1980 : circuits intégrés et multiprogrammation

Des ordinateurs génériques

- Applications scientifiques, commerciales, ...
- Système d'exploitation OS/360 :
 - Unique pour toute la gamme 360.
 - Millions de lignes d'assembleur. Milliers de bugs.

Multiprogrammation (Sera étudiée en cours et TD)

- Dans les applications commerciales, les temps d'E/S peuvent atteindre 80 à 90% du temps total.
- Solution :
 - Partage de la mémoire entre plusieurs tâches.
 - Commutation de processus : Quand une tâche attend une E/S, une autre tâche peut utiliser l'UC.
- Apparition de *MULTICS* en 1964. Opérationnel en 1971 (UNIX réécrit en C)

1980-aujourd'hui : ordinateurs personnels

Développement des microprocesseurs

- Ordinateurs personnels : IBM PC + MS-DOS
- Xerox PARC : première interface homme-machine graphique.
- Premiers systèmes d'exploitation basés sur des IHM graphiques : MacOs
- Microsoft Window - surcouche de MS-DOS

1990 - ... : ordinateurs en réseau

1990- ... : les ordinateurs connectés

- Systèmes d'exploitation partageant des ressources
- Telnet, SSH, NFS, ...
- Attaques et virus

2000 - ... : les ordinateurs de poche

- Système d'exploitation de type "micronoyau" modulaires (un module par fonction)
- ils peuvent être configurés avec plus ou moins de modules et donc adaptables à des très petites machines (PDA et PIC).

2010 - ... : les objets connectés

- Système d'exploitation minimal
- Les efforts portent sur la communication et le partage d'informations.

Plan

1 Introduction

2 Structure et concepts de base

3 Plan du cours

Quelques concepts généraux aux systèmes d'exploitation

Processus et fichiers

- Un processus est une entité qui permet au système d'exécuter les instructions données par un programme :
 - Programme : une recette de cuisine
 - Processeur : cuisinière, four, ...
 - Processus : un cuisinier qui traduit chaque ligne de la recette en une suite d'actions qu'il exécute
 - Système : le chef → donne les tâches aux cuisiniers, ordonne leur travail, surveille le bon déroulement, affecte les outils aux cuisiniers ...
- Les fichiers
 - Système de fichiers : abstraction permettant de représenter le contenu d'un disque comme un ensemble de fichiers indépendants généralement structurés au sein de répertoires.

Structure d'un système d'exploitation

Coeur de tout système : Le noyau

- Regroupe l'ensemble des mécanismes de base : gestion de la mémoire, des processus, de la communication interprocessus ...
- Chargé en mémoire au lancement du système
- Intègre les pilotes des périphériques
- Accessible via des **appels système** protégés par le matériel

Lien application - noyau : la librairie standard libC

- Lien entre les applications et le noyau
- Ensemble des fonctions permettant les appels système : allocation de mémoire (`malloc`), contrôle de processus (`fork`), signaux (`kill`)..
- Fournit les capacités d'entrée/sortie du langage C, comme la fonction `printf`.

Plan

1 Introduction

2 Structure et concepts de base

3 Plan du cours

Les prochains cours

- Processus et temps partagé et ordonnancement
- Gestion de la mémoire
- Gestion de fichiers

Les TD/TP

- TD : Approfondissement des notions vues en cours
- TP : Mise en œuvre en C de ces notions : programmation système sous Linux.

Contrôles

- Un projet bimestriel au fil des TD/TP représentant entre 30 et 35% de la note finale
- 3 contrôles (intégrant un projet bimestriel)

⇒ voir détails sur Moodle

Bibliographie

- Systèmes d'exploitation. Andrew Tanenbaum, Pearson Education, 3e ed. 2008.
- Operating system concepts. Silberschatz, Galvin, Gagne, Wiley, 8e ed. 2008.
- La programmation sous Unix. Jean-Marie Rifflet, Ediscience, 3e ed. 1991.
- Understanding the Linux kernel. Bovet, Cesati, O'Reilly, 3e ed. 2005.
- Windows Internals. Russinovich, Solomon, Microsoft Press, 5e ed, 2009.
- Développement système sous Linux : Ordonnancement multi-tâche, gestion mémoire, communications, programmation réseau, Christophe Blaess, 3ème édition, Eyrolles, 2011.