

Systeme



Département d'Informatique de Rouen
Université de Rouen
76821 Mont-Saint Aignan, Cédex

Saïd Abdedaïm et Pascal Caron

Table des matières

1	Introduction	2
1.1	Concepts fondamentaux	2
1.2	Objectifs et fonctions essentielles	2
1.3	Historique	4
2	Les processus	5
2.1	Introduction	5
2.2	les différents états d'un processus	5
2.3	Structure de données associée au processus	7
2.3.1	Le changement de contexte	7
2.3.2	Le noyau et les appels système	8
2.4	Création et destruction de processus : le cas d'Unix	8
2.4.1	Les appels-système Unix	9
2.5	Ordonnancement	11
2.5.1	Ordonnancement sans réquisition	12
2.5.2	L'ordonnancement avec réquisition	13
2.5.3	L'exemple d'Unix	16
2.6	La synchronisation des processus	17
2.6.1	Accès concurrents	17
2.6.2	Sections critiques	18
2.6.3	Différentes méthodes d'exclusion mutuelle	21
2.6.4	Un peut d'aide de la part du matériel	23
2.6.5	L'attente passive	26
2.6.6	Sémaphores	27
2.6.7	Les moniteurs	32
2.7	Communication inter-processus	34
2.7.1	Les messages	35
2.7.2	La gestion des signaux sous Unix	39
3	Les entrées-sorties	43
3.1	Introduction	43
3.2	Les principes généraux	43
3.2.1	Périphériques physiques et périphériques logiques	43

3.2.2	Périphériques et contrôleurs	44
3.3	Fonctionnement du logiciel d'E/S	44
3.3.1	Le traitement en couche	45
4	Le système de fichiers	47
4.1	Fichiers	47
4.1.1	Noms et types	47
4.1.2	Attributs	48
4.1.3	Attributs sous Unix	49
4.1.4	Opérations	50
4.1.5	Opérations sur les fichiers sous Unix	51
4.1.6	Fichiers et communication sous Unix	53
4.2	Répertoires	54
4.2.1	Système de répertoires hiérarchique	54
4.2.2	Opérations	54
4.2.3	Opérations sur les répertoires sous Unix	55
4.3	Mise en œuvre du système de fichiers	58
4.3.1	Mise en œuvre des fichiers	58
4.3.2	Mise en œuvre des répertoires	61
4.3.3	Gestion de l'espace disque	62
4.3.4	Fiabilité	64
4.3.5	Performances	65
4.3.6	Implantation du système de fichiers sous Linux	65
5	Gestion de la mémoire	70
5.1	Gestion mémoire basique	70
5.1.1	Gestion de la mémoire sans va-et-vient ni pagination	70
5.2	Va-et-vient (<i>swapping</i>)	72
5.2.1	Gestion de la mémoire par tables de bits	72
5.2.2	Gestion de la mémoire par listes chaînées	73
5.3	La mémoire virtuelle	73
5.3.1	Pagination	74
5.3.2	Les tables de pages inverses	75
5.4	Les algorithmes de remplacement de pages	76
5.4.1	Remplacement de page optimal	76
5.4.2	Remplacement de la page non récemment utilisée	77
5.4.3	Premier entré, premier sorti	77
5.4.4	Remplacement avec seconde chance	77
5.4.5	Remplacement avec horloge	77
5.4.6	Remplacement de la page la moins récemment utilisée	77
5.5	La segmentation	77
5.5.1	Implantation de segments purs	77
5.5.2	Segmentation avec pagination : MULTICS	79

5.5.3	Segmentation avec pagination : Le processeur Intel Pentium	80
5.6	Gestion de la mémoire sous Linux	82
5.6.1	Segmentation	82
5.6.2	Gestion des pages et va-et-vient	86
5.6.3	Allocation de mémoire pour le noyau	87
5.6.4	Appels systèmes Unix	88
6	Exercices corrigés	92
6.1	Vie et mort d'un processus	92
6.2	segment de mémoire partagée	93
6.3	Système de fichiers	99
6.4	Tubes et tubes nommées	110

Chapitre 1

Introduction

1.1 Concepts fondamentaux

Pour pouvoir parler de Système d'exploitation, il faut tout d'abord définir quelques termes techniques. La première chose à savoir est : qu'est-ce qu'un ordinateur ? Il est évident que nous ne rentrerons pas dans les détails ici. Un cours d'architecture des ordinateurs vous permettrait d'appréhender cette notion en détails, mais nous pouvons en donner ici une définition (simpliste et) simplifiée.

Ordinateur : Machine permettant de stocker des informations et d'effectuer des calculs (beaucoup plus rapide que l'humain).

Cet ordinateur regroupe un certain nombre de composants que nous appellerons **matériel** (hardware). Parmi ceux-ci, nous trouvons clavier, souris, écran, mémoire, disques, . . .

Ce matériel, sans **logiciels** (software) n'est qu'un amas de ferraille et de composants électroniques.

Les **logiciels** sont des séquences d'instructions que l'on appelle souvent programmes. Un **programme** est l'implantation d'un ensemble de méthodes de résolution de problèmes dans un langage compris par l'ordinateur. La compréhension de l'ordinateur n'est en fait que la séquence d'actions résultant du programme (écrire sur l'imprimante, sur le disque, à l'écran, . . .).

La méthode de résolution est ce que l'on appelle l'**algorithme**.

Le système d'exploitation d'un ordinateur est un programme servant d'interface entre le matériel et les utilisateurs. Son but est de rendre le maniement de l'ordinateur facile et de proposer une utilisation efficace de celui-ci (il n'est par exemple, pas question pour l'utilisateur de savoir quelle va être l'organisation du fichier sur le disque lorsqu'il le sauve).

1.2 Objectifs et fonctions essentielles

On peut faire une décomposition de l'ordinateur selon les quatre grandes classes suivantes :

- Les utilisateurs (et leurs programmes),
- les programmes d'application,
- le système d'exploitation,

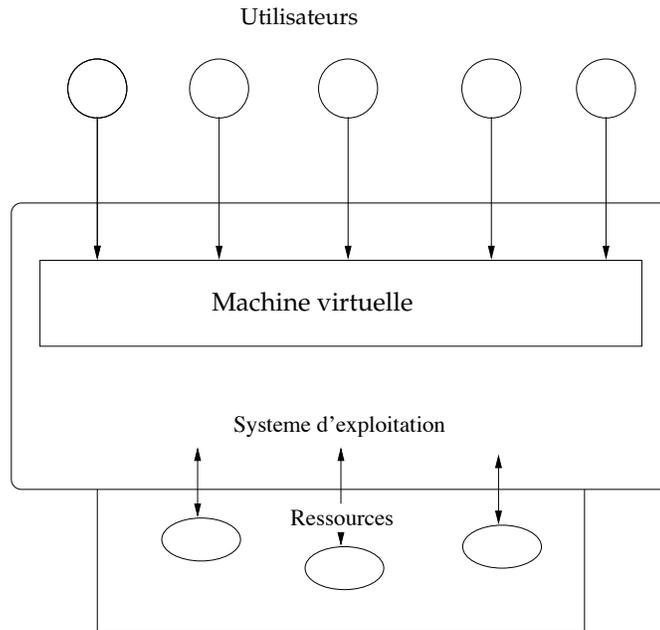


FIG. 1.1 – Les composantes d'un système informatique

– le matériel.

Il est plus facile de définir le rôle d'un système d'exploitation que d'en donner une définition exacte. Le premier des objectifs d'un système d'exploitation est d'offrir une machine virtuelle aux utilisateurs. Cette machine devant être à la fois facile d'emploi et conviviale. L'utilisateur ne doit pas avoir à se soucier de la programmation des lecteurs de disquettes (ou tout autre périphérique) mais doit avoir, grâce au système, une interface simple pour son utilisation. Le système d'exploitation masque beaucoup d'aspects fastidieux qui concernent la programmation du matériel, la gestion de la mémoire, des interruptions, des fichiers, des processus, ...

Nous expliquerons dans ce cours, la manière dont le système d'exploitation s'acquitte de ces différentes tâches.

Le second rôle essentiel d'un système d'exploitation est de prendre en charge la gestion complexe des ressources de l'ordinateur et de les répartir entre les différents programmes qui peuvent appartenir à plusieurs utilisateurs. Un système informatique complexe peut être composé d'une multitude de périphériques. Imaginez ce qui se passerait si trois programmes s'exécutant sur la même machine tentaient d'imprimer leur résultat simultanément. Aurait-on une ligne provenant de chacun des programmes ? un caractère ? On voit ici que le rôle du système d'exploitation dans la répartition des périphériques aux différents utilisateurs est essentiel. C'est lui qui stocke les données allant vers l'imprimante et qui s'occupe de gérer une file d'attente.

1.3 Historique

Les toutes premières machines (1945-55) étaient des machines nues, c'est-à-dire, sans aucun logiciel. Ces machines étaient composées de milliers de tubes à vide et prenaient une place énorme. Leur programmation se faisait en binaire et les programmes étaient entrés et mis au point à l'aide d'un pupitre de commande. La procédure s'est améliorée avec l'introduction des cartes perforées que pouvaient lire l'ordinateur.

Avec l'introduction du transistor (1955-65), la situation se trouve modifiée. A cette époque, les ordinateurs devinrent assez fiable pour être commercialisés mais seules les grandes sociétés pouvaient s'en offrir un. Nous sommes alors à l'apogée du traitement par lots. Pour lancer un travail, un programmeur devait en écrire le code (par exemple en FORTRAN), puis le mettre sur carte et le donner aux opérateurs pour qu'il soit soumis. L'opérateur devait alors effectuer les tâches suivantes :

- Chargement du compilateur FORTRAN,
- chargement de la carte pour compilation du programme,
- chargement du programme en mémoire,
- exécution et sortie des résultats sur imprimantes.

La solution du traitement par lots permettait de regrouper des tâches de même nature (plusieurs compilations FORTRAN). De même plutôt que d'effectuer toutes ces tâches successives, un ordinateur a été dédié à la sauvegarde des cartes sur bande et à la sortie des résultats d'après bande. Ainsi le calculateur n'avait plus qu'à lire une bande sur laquelle il y avait un enchaînement de travaux et il sauvegardait de même les sortie sur bande. Ce périphérique étant plus rapide que l'imprimante et le lecteur de carte, une première optimisation a été effectuée. Cependant les phases où le CPU reste inactifs sont longues (problème des Entrée/Sortie).

Pour palier à ce problème est apparu la notion de multiprogrammation (1965-80). Plutôt que d'attendre après une Entrée/Sortie, on peut basculer les tâches de manière à ce que le CPU reste le plus souvent actif. Cette notion permet de passer d'une tâche à une autre mais pose quelques problèmes.

- problème de protection : comment protéger les utilisateurs entre eux, le système contre les utilisateurs ?
- problème de partage : comment répartir les ressources (temps de CPU, mémoire centrale)

Ce sera au système d'exploitation de gérer ces différents problème. Unix fut l'un des premiers système d'exploitation multiprogrammé. Nous donnerons dans ce cours quelques exemple sous Unix.

Dans les années 1980-90 on vit l'avènement du micro-ordinateur. Avec son système d'exploitation MS-DOS puis Windows-NT, il domine actuellement le marché. Un intéressant concurrent à ces deux systèmes a pris énormément d'ampleur depuis 1995. Il s'agit de **Linux**, un Unix revu et corrigé, ouvert, développé initialement pour la famille des processeurs Intel 80xxx, mais qui s'étend aujourd'hui vers d'autres plateformes.