

UE NSY103

Les Systèmes d'Exploitation



Sommaire

I- Les Systèmes d'Exploitation

II- Le Noyau

III- Le Système Linux



I- Les Systèmes d'Exploitation

Le système d'exploitation (en anglais « Operating System ») est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications de l'utilisateur (traitement de texte, jeu vidéo...).

- Il assure le démarrage de l'ordinateur, et fournit aux programmes applicatifs des points d'entrée génériques pour les périphériques.
- Il a des objectifs principaux :
- la prise en charge, la gestion, et le partage des ressources d'un ordinateur,
- la construction au dessus de ces ressources d'une « interface standardisée » (au niveau des programmeurs, on parlera plus tard d'API : « Application Programming Interface », ou « Interface de programmation » en français) plus conviviale et plus facile d'emploi pour y accéder.



I- Les Systèmes d'Exploitation

Les fonctionnalités du système d'exploitation utilisent les mécanismes offerts par le matériel de la machine physique pour réaliser leurs opérations.

Notamment, le système d'exploitation s'interface avec la couche matérielle, par le biais du mécanisme des interruptions, qui lui permet de prendre connaissance des événements survenant au niveau matériel.

Par ailleurs, le système d'exploitation s'interface avec les applications du niveau utilisateur par le biais des fonctions prédéfinies que chacune de ses fonctionnalités offre. Ces fonctions que l'on qualifie de **primitives système** constituent les points d'entrées des fonctionnalités du système d'exploitation. Il est possible de les appeler depuis les applications de niveau utilisateur. Ces appels peuvent se faire à deux niveaux :

- dans le code d'un programme utilisateur à l'aide d'un **appel système**, qui n'est autre qu'une forme d'appel de procédure amenant à l'exécution d'une routine système ;
- depuis le « prompt » de l'interpréteur de commandes, à l'aide d'une commande. L'interpréteur de commandes est un outil de niveau utilisateur qui accepte les commandes de l'utilisateur, les analyse et lance l'exécution de la routine système associée ;
- plus récemment, dans les systèmes d'exploitations graphique, en utilisant les mécanismes d'icônes (double clic, menu contextuel, cliquer/déposer, etc.).



II- Le Noyau

Le noyau (kernel en anglais), est la partie fondamentale de certains systèmes d'exploitation. Il gère les ressources de l'ordinateur et permet aux différents composants - matériels et logiciels - de communiquer entre eux.

- En tant que partie du système d'exploitation, le noyau fournit des mécanismes d'abstraction du matériel, notamment de la mémoire, du (ou des) processeur(s), et des échanges d'informations entre logiciels et périphériques matériels. Le noyau autorise aussi diverses abstractions logicielles et facilite la communication entre les processus.
- Le noyau d'un système d'exploitation est lui-même un logiciel, mais ne peut cependant utiliser tous les mécanismes d'abstraction qu'il fournit aux autres logiciels. Son rôle central impose par ailleurs des performances élevées.
- La majorité des systèmes d'exploitation sont construits autour de la notion de noyau. L'existence d'un noyau, c'est-à-dire d'un programme unique responsable de la communication entre le matériel et le logiciel, résulte de compromis complexes portant sur des questions de performance, de sécurité et d'architecture des processeurs.



II- Le Noyau

L'existence d'un noyau présuppose une partition virtuelle de la mémoire vive physique en deux régions disjointes, l'une étant réservée au noyau (l'espace noyau) et l'autre aux applications (l'espace utilisateur).

- Cette division fondamentale de l'espace mémoire en un espace noyau et un espace utilisateur contribue beaucoup à donner la forme et le contenu actuels des systèmes généralistes (Linux, Windows, Mac OS X, etc.). Le noyau a de grands pouvoirs sur l'utilisation des ressources matérielles, en particulier de la mémoire. Elle structure également le travail des développeurs : le développement de code dans l'espace noyau est à priori plus délicat que dans l'espace utilisateur car la mémoire n'est pas protégée.
- Diverses abstractions de la notion d'application sont fournies par le noyau aux développeurs. La plus courante est celle de processus (ou tâche). Le noyau du système d'exploitation n'est en lui même pas une tâche, mais un ensemble de routines pouvant être appelées par les différents processus pour effectuer des opérations requérant un certain niveau de privilèges. Les flots d'exécution dans le noyau sont des continuations des flots d'exécution des processus utilisateurs bloqués lorsqu'ils effectuent des appels systèmes. En général, un processus bloqué ne consomme pas de temps processeur, il est réveillé par le processus système lorsque celui-ci se termine.



III- Le Système Linux

Le système d'exploitation Linux est un système multiprogrammé, compatible avec la norme pour les systèmes d'exploitation IEEE-POSIX 1003.1, appartenant à la grande famille des systèmes de type Unix. C'est un système de type interactif qui présente également des aspects compatibles avec la problématique du temps réel faiblement constraint.

- Le système Linux est né en 1991 du travail de développement de Linus Torvalds. Initialement créé pour s'exécuter sur des plates-formes matérielles de type IBM/PC/Intel 386, le système est à présent disponible sur des architectures matérielles très diverses telles que SPARC, Alpha, IBMSystem/390, Motorola, etc.
- Une des caractéristiques principales de Linux est le libre accès à son code source, celui-ci ayant été déposé sous Licence Publique GNU (GPL). Le code source peut ainsi être téléchargé (<http://kernel.org/>), étudié et modifié par toute personne désireuse de le faire.



III- Le Système Linux

Le système Linux est structuré comme un noyau monolithique modulaire qui peut être découpé en quatre grandes fonctions :

- 1) une fonctionnalité de gestion des **processus** qui administre les exécutions de programmes, le **processus** étant l'image dynamique de l'exécution d'un programme ;
 - 2) une fonctionnalité de gestion de la mémoire ;
 - 3) une fonctionnalité de gestion des fichiers de haut niveau : le **VFS (Virtual File System)**, qui s'interface avec des gestionnaires de fichiers plus spécifiques de type Unix, DOS, etc., lesquels s'interfacent eux-mêmes avec les contrôleurs de disques, disquettes, CD-Rom, clés USB, etc.;
 - 4) une fonctionnalité de gestion du réseau qui s'interface avec le gestionnaire de protocole puis le contrôleur de la carte réseau.
- L'architecture du système Linux peut être ajustée autour du noyau grâce au concept de « *modules chargeables* » (Cf. précédent chapitre).
 - Lors du paramétrage de la compilation du noyau, il est possible de définir pour chaque module s'il doit être liée de façon statique au noyau (il fait alors partie du noyau monolithique), où s'il doit être lié de façon dynamique (lors de l'insertion d'un périphérique « *plug & play* » par exemple).



III- Le Système Linux

Quelques principes de base

Comme nous l'avons déjà évoqué précédemment, le système d'exploitation s'interface avec les applications du niveau utilisateur par le biais des fonctions prédefinies qualifiées de primitives systèmes

- A ce titre, le noyau Linux ne doit pas être appréhendé comme étant un processus, mais un gestionnaire de processus, qui offre des services à ceux-ci.
- L'exécution des primitives systèmes s'effectue sous un mode privilégié, appelé « *mode noyau* » (« **kernel mode** »). Lorsqu'un programme est exécuté en mode, aucune restriction ne s'applique à lui. Il peut accéder à toute la mémoire, dialoguer directement avec les périphériques, etc.
- Un processus utilisateur s'exécute par défaut selon un mode qualifié de « *mode utilisateur* » (« *User Mode* »): ce mode d'exécution est un mode pour lequel les actions pouvant être entreprises par le programme sont volontairement restreintes afin de protéger la machine des actions parfois malencontreuses du programmeur. Notamment, le jeu d'instructions utilisables par le programme en mode utilisateur est réduit..



III- Le Système Linux

La commutation de contexte

Lorsqu'un processus utilisateur demande l'exécution d'une routine du système d'exploitation par le biais d'un appel système, ce processus quitte son mode courant d'exécution (le mode utilisateur) pour passer en mode d'exécution du système, soit le mode noyau.

- Ce passage du mode utilisateur au mode superviseur constitue une **commutation de contexte** : elle s'accompagne d'une opération de sauvegarde du contexte utilisateur, c'est-à-dire principalement de la valeur des registres du processeur (compteur ordinal, registre d'état), sur la pile noyau. Un contexte noyau est chargé constitué d'une valeur de compteur ordinal correspondant à l'adresse de la fonction à exécuter dans le noyau, et d'un registre d'état en mode superviseur. Lorsque l'exécution de la **fonction système** (on parle aussi de **primitive**) est achevée, le processus repasse du mode superviseur au mode utilisateur. Il y a de nouveau une opération de commutation de contexte avec restauration du contexte utilisateur sauvegardé lors de l'appel système, ce qui permet de reprendre l'exécution du programme utilisateur juste après l'appel.



III- Le Système Linux

La gestion des interruptions matérielles et logicielles

Dans le système Linux, chaque interruption, qu'elle soit matérielle ou logicielle, est identifiée par un entier de 8 bits appelé « vecteur d'interruption », dont la valeur varie de 0 à 255 :

- les valeurs de 0 à 31 correspondent aux interruptions « *non masquables* » (nous reviendrons sur ce terme) et aux exceptions. Correspond aux différentes interruptions générées nativement par le CPU ;
- les valeurs de 32 à 47 sont affectées aux interruptions « *masquables* » levées par les périphériques (IRQ) ;
- les valeurs de 48 à 255 peuvent être utilisées pour identifier d'autres types de trappes que celles admises par le processeur (qui correspondent aux valeurs de 0 à 31). Comme nous l'avons déjà vu, l'entrée 128 (0x80 en hexadécimal) est réservée aux appels système.
- Ce numéro permet d'adresser une table comportant 256 entrées, appelée « *table des vecteurs d'interruptions* » (idt_table), placée en mémoire centrale lors du démarrage de l'OS.



III- Le Système Linux

La gestion des interruptions matérielles et logicielles (suite)

L'unité de contrôle du processeur, avant de commencer l'exécution d'une nouvelle instruction machine, vérifie si une interruption ne lui a pas été délivrée.

Si tel est le cas, les étapes suivantes sont mises en œuvre :

- l'interruption i a été délivrée au processeur. L'unité de contrôle accède à l'entrée n° i de la table des vecteurs d'interruptions, dont l'adresse est conservée dans un registre du processeur, et récupère l'adresse en mémoire centrale du gestionnaire de l'interruption levée ;
- l'unité de contrôle vérifie que l'interruption a été émise par une source autorisée ;
- l'unité de contrôle effectue un changement de niveau d'exécution (passage en mode superviseur) si cela est nécessaire et commute de pile d'exécution. En effet, comme nous l'avons déjà vu, les exécutions des gestionnaires d'interruptions pouvant être imbriquées, la prise en compte d'une interruption par l'unité de contrôle peut être faite alors que le mode d'exécution du processeur est déjà le mode superviseur (niveau d'imbrication > 1) ;
- l'unité de contrôle sauvegarde dans la pile noyau le contenu du registre d'état et du compteur ordinal;
- l'unité de contrôle charge le compteur ordinal avec l'adresse du **gestionnaire d'interruption**.



III- Le Système Linux

La gestion des interruptions matérielles et logicielles (suite)

Le gestionnaire d'interruption s'exécute. Cette exécution achevée, l'unité de contrôle restaure le contexte sauvegardé au moment de la prise en compte de l'interruption, c'est-à-dire :

- l'unité de contrôle restaure les registres d'état et le compteur ordinal à partir des valeurs sauvegardées dans la pile noyau ;
- l'unité de contrôle commute de pile d'exécution pour revenir à la pile utilisateur si le niveau d'imbrication des exécutions du noyau est égal à 1.



III- Le Système Linux

Imbrication de la prise en compte des interruptions

Le noyau Linux est un noyau réentrant, c'est à dire qu'à un instant t donné, plusieurs exécutions en mode noyau peuvent exister, une seule étant active.

- En effet, le noyau autorise un gestionnaire d'interruption à interrompre l'exécution d'un autre gestionnaire d'interruption. L'exécution du gestionnaire est suspendue, son contexte sauvegardé sur la pile noyau, et l'exécution est reprise lorsque l'exécution du gestionnaire survenue entre-temps est achevée. De ce fait, les exécutions au sein du noyau peuvent être arbitrairement imbriquées.
- Le noyau maintient une valeur de niveau d'imbrication, qui indique la profondeur d'imbrication courante dans les exécutions du noyau. Le retour en mode utilisateur n'est effectué que lorsque ce niveau d'imbrication est à 1. Il est précédé de l'exécution des parties basses activées par l'ensemble des routines de services exécutées. Plus précisément :
 - un gestionnaire d'interruption peut interrompre un gestionnaire de trappe ou un autre gestionnaire d'interruption ;
 - par contre, un gestionnaire de trappe ne peut jamais interrompre un gestionnaire d'interruption.



EXERCICES

Vous pensez avoir bien assimilé les concepts présentés dans ce cours.

Vous devez alors passer aux exercices, ce sont eux qui vous permettront de valider vos connaissances .