

NSI Terminale - Architecture

Système d'exploitation : rappels et compléments

qkzk

2021/06/25

Rappels de première

Les *systèmes d'exploitation* (OS) sont des logiciels spécialisés qui servent d'intermédiaire entre le matériel et l'utilisateur. Ils sécurisent l'utilisation du matériel.

Il existe deux grandes familles d'OS : UNIX (Linux, Android, BSD, OSX, iOS etc.) et Windows (95, 98, NT, 10 etc.)

UNIX désigne *une famille* d'OS, pas un OS en particulier.

La mémoire

On peut simplifier le matériel à trois grandes catégories :

- la mémoire (les informations)
- le processeur (qui calcule)
- les entrées/sorties (clavier, souris, écran, réseau etc.)

Chaque programme en cours d'exécution (*un processus*) requiert de lire et d'écrire dans la mémoire. Afin de sécuriser le système on empêche une application d'interférer avec les instructions d'une autres application.

Les processus n'ont donc aucun moyen de savoir s'ils sont plusieurs à tenter d'écrire à la même adresse.

Mémoire virtuelle

Chaque processus se voit allouer de la mémoire dans laquelle il peut écrire et lire.

Lorsque l'instruction "écrire à l'adresse 3" du processus P1 est exécutée, l'OS écrit réellement dans la mémoire, mais à une adresse qu'il a réservé pour le processus. Par exemple 53. Il retient la correspondance :

Processus	adresse virtuelle	adresse physique
P1	3	53

Lorsque l'instruction "écrire à l'adresse 3" du processus P2 est exécutée l'OS utilise alors une *autre adresse physique* pour enregistrer.

Processus	adresse virtuelle	adresse physique
P1	3	53
P2	3	142

Ainsi, le contenu de l'adresse 3 pour le processus P1 n'est pas écrasé et lorsqu'il voudra lire cette adresse il retrouvera les informations qu'il y avait écrites.

Ordonnancement

Les processeurs n'exécutent qu'une instruction à la fois. Un processus ayant beaucoup d'instructions à réaliser va paralyser le système durant son exécution rendra l'ordinateur inutilisable.

L'ordonnanceur

Le rôle de l'ordonnanceur est de donner la parole à chaque processus selon des règles préétablies afin de donner l'illusion de la simultanéité.

Plusieurs algorithmes existent :

- Tourniquet (*Round Robin*) : une instruction de chaque processus à la fois, jusqu'à épuisement.
- Plus court d'abord (*shortest first*) : en premier les instructions de l'application qui en a demandé le moins,
- Premier arrivé premier servi (*First in First out, FIFO*) le premier arrivé est exécuté jusqu'à épuisement. C'est ce qu'on souhaite éviter !

En pratique les algorithmes sont beaucoup plus complexes.

Les interruptions

Un même type de périphérique (un microphone par exemple) peut disposer d'un millier de variante qui fonctionnent toutes différemment.

Comment un programme qui souhaite utiliser le microphone (l'application "Téléphoner" de votre smartphone) pourrait-il être capable de les utiliser toutes ?

Il faudrait que son code contienne les pilotes de *chaque variante*... et il faudrait constamment le mettre à jour !

Et ceci pour chaque programme de l'ordinateur.

Les appels systèmes

Lorsqu'une application souhaite utiliser le matériel, l'OS vérifie d'abord qu'il en a le droit.

Si c'est le cas, l'OS attend qu'un événement venant du microphone se produise, (une *interruption*) et transmet alors ce signal à l'application.

Ces appels sont formalisés et sont indépendants du matériel.

Les pilotes

Lorsqu'un fabricant crée un nouveau matériel il fournit un programme pour le système d'exploitation : un *pilote*.

Celui-ci permet de répondre aux appels systèmes standardisés définis par l'API de l'OS.

Cela permet à du matériel très varié de communiquer de la même manière avec les applications, à travers leur pilote et l'API commune.

Autres rôles

Citons :

- la gestion des *droits* : l'OS est chargé de la sécurité liée à l'exécution des programmes en garantissant que les ressources ne sont utilisées que par les programmes et utilisateurs possédant les droits adéquats.
- la gestion des *fichiers* et de leur arborescence. L'OS gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.

Multitude d'OS

Pourquoi tant d'OS différents s'ils utilisent le même principe ?

La spécialisation

Certains OS répondent à un cahier des charges précis et sont spécialisés.

Android n'est pas adapté à un super ordinateur. Mais les super ordinateurs utilisent le même noyau qu'android : Linux.