

## Chapitre 5

# LES PERIPHERIQUES

## 1. PERIPHERIQUES VIRTUELS D'ENTREE-SORTIE

### 1.1 Mécanisme des périphériques virtuels

L'objectif est de rendre tout programme utilisateur indépendant des types de périphériques, nombreux et en évolution constante, et des contrôles qui en découlent. D'où la création de **périphériques virtuels** ou **flots d'E/S** ou **fichiers d'E/S**. Sous UNIX, par exemple, une console, un disque ou un fichier ont la même interface d'accès.

Les opérations possibles sur un flot sont :

- ouverture et association physique (open en C)
- fermeture et dissociation (close en C)
- lecture (read en C)
- écriture (write en C)
- paramétrage et contrôle (ioctl , seek en C)

Le système gère une table des flots par processus dans le BCP.

La concordance entre un périphérique virtuel et un périphérique physique associé est assurée par un **pilote de périphérique** (device driver) : soit un type de pilote par type de périphérique, soit un seul type de pilotes configurable par des **descripteurs de périphériques**. Un pilote est la seule partie du SE à connaître les registres du contrôleur, les secteurs et leur facteur d'entrelacement, les pistes, les cylindres, les têtes, les déplacements des bras, les moteurs, le temps de positionnement des têtes, etc..., ainsi que le traitement des erreurs.

### 1.2 Fonctionnement d'un périphérique virtuel

Un processus P demande une E/S avec un périphérique. La correspondance périphérique virtuel-périphérique physique est ainsi résolue :

- **appel système** : les paramètres de l'appel sont empilés et un appel système (Supervisor Call = requête SVC) est généré
- **action du superviseur** : il identifie la nature de l'appel et récupère les paramètres (nom du flot, nombre d'octets, adresse de transfert); puis il identifie le pilote concerné et le périphérique physique. Enfin, il appelle le pilote
- **le pilote** : le contexte du processus actif est commuté après sauvegarde. La procédure de traitement des E/S est exécutée. Ensuite, le pilote envoie une interruption pour le signaler. Le processus initial est rechargé et réactivé.

## 2. LES DISQUES

### 2.1 Caractéristiques physiques

Un disque est logiquement organisé en cylindres. Un cylindre contient autant de pistes qu'il y a de têtes. Chaque piste est divisée en secteurs (un nombre constant et de taille constante quelque soit le rayon de la piste). Un contrôleur de disques peut effectuer des recherches simultanées sur plusieurs disques, ou bien lire ou écrire sur un disque et attendre des données d'un ou plusieurs autres disques.

Pour le système, une adresse sur disque est formée du triplet : (n° de cylindre, n° de piste, n° de secteur).

Le temps moyen de lecture/écriture sur un secteur est égal à la somme de trois durées élémentaires :

- le temps de recherche : positionnement de la tête sur le bon cylindre
- le temps de latence ou délai rotationnel : pour atteindre le bon secteur (en moyenne 1/2 tour de rotation)
- le temps de transfert réel de l'information

## **2.2 Ordonnement des accès**

**Algorithme FCFS (FIFO)** : la gestion la plus simple du mouvement du bras du disque consiste à satisfaire les requêtes d'accès dans l'ordre où elles surviennent. Bien entendu, cet algorithme donne généralement de mauvais temps de réponse. Ainsi, si une suite de requêtes concerne les pistes : 1, 36, 16, 34, 9, 12, alors que les têtes sont sur le cylindre 11, il faudra se déplacer au total de 111 cylindres. Il s'agit de l'algorithme First Come First Served.

**Algorithme SSTF (ou PCD)** : ici, le pilote choisit la requête la plus proche de sa position actuelle (Shortest Seek Time First, plus courte d'abord. Avec l'exemple ci-dessus, les cylindres seront traités dans l'ordre : 12, 9, 16, 1, 34, 36. On se sera déplacé de 61 cylindres seulement. Mais les requêtes portant sur des cylindres éloignés peuvent être durablement différées, si d'autres requêtes surviennent pendant le traitement de la liste : l'équité peut souffrir de la réduction du temps de réponse.

**Algorithme de l'ascenseur (scan)** : pour éviter cet inconvénient, on déplace la tête dans une direction donnée en traitant toutes les requêtes rencontrées, puis le sens du balayage s'inverse et on traite les requêtes rencontrées, etc... Dans l'exemple traité, on obtiendrait l'ordre : 12, 16, 34, 36, 9, 1 et un parcours total de 60 cylindres.

## **2.3 Mémoire cache pour les pistes du disque**

En général, le temps de recherche est très supérieur au temps de transfert. Il importe donc peu de lire un secteur ou bien une piste complète pour simplifier le fonctionnement du contrôleur. On utilise une **mémoire cache** pour stocker une piste : mémoire à accès rapide, à accès direct par le contrôleur et par l'UC.

## **2.4 Disque virtuel**

Un disque virtuel utilise une portion de la mémoire centrale pour sauvegarder des secteurs. Il convient bien pour stocker programmes et données fréquemment utilisées, qui seront ainsi accessibles très rapidement.