Table des matières

0. Introduction

La gestion de système d'entrées/sorties (E/S) est une partie essentielle des systèmes informatiques modernes. Elle a des origines lointaines, qui remontent aux premiers ordinateurs mécaniques et électromécaniques, mais a connu une évolution rapide et continue depuis les années 1950 avec l'introduction des premiers ordinateurs électroniques.

Les premiers ordinateurs mécaniques, tels que la machine analytique de Charles Babbage, n'avait pas de systèmes d'E/S sophistiqués. Les données étaient souvent entrées dans la machine via des cartes perforées et les résultats étaient imprimés sur du papier. Les ordinateurs électromécaniques, tels que le Harvard Mark I, ont également utilisé des cartes perforées pour l'entrée des données et des lumières pour afficher les résultats.

Cependant, avec l'arrivée des premiers ordinateurs électroniques dans les années 1950, la gestion de système d'E/S a commencé à se développer rapidement. Les premiers ordinateurs électroniques tels que l'ENIAC et l'EDVAC utilisaient des bandes perforées comme support de stockage pour les données d'entrée/sortie.

Au fil des ans, les systèmes d'E/S ont été améliorés pour gérer un plus grand nombre de périphériques et pour gérer des volumes de données plus importants. L'un des premiers systèmes d'E/S modernes a été introduit dans le système d'exploitation OS/360 d'IBM dans les années 1960. Ce système a permis une gestion centralisée des périphériques d'E/S à partir d'un emplacement unique dans le système.

La gestion de système d'entrées/sorties (E/S) est une fonctionnalité clé des systèmes informatiques modernes. Elle est utilisée pour permettre la communication entre le processeur central et les périphériques externes. Cette communication est nécessaire pour que les données puissent être entrées dans le système ou extraites de celui-ci, pour le stockage, le traitement ou l'affichage. Les dispositifs d'E/S peuvent inclure des claviers, des souris, des moniteurs, des imprimantes, des scanners, des appareils photo, des disques durs externes et des dispositifs de réseau.

Dans le cadre de notre travail, nous essayerons, dans la mesure du possible, d'expliciter ces notions par le biais du système d'exploitation Linux, nous exhiberons les concepts importants et les cas d'utilisation.

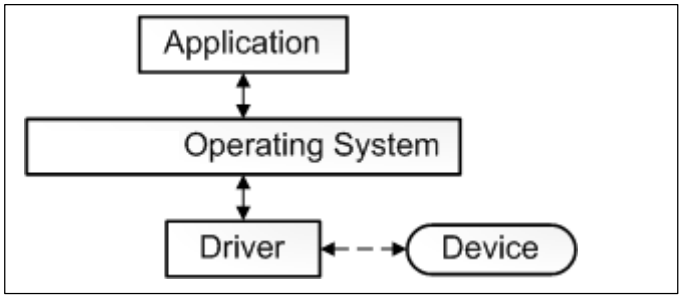
I. Explication des concepts

1. Pilotes

a. Définition

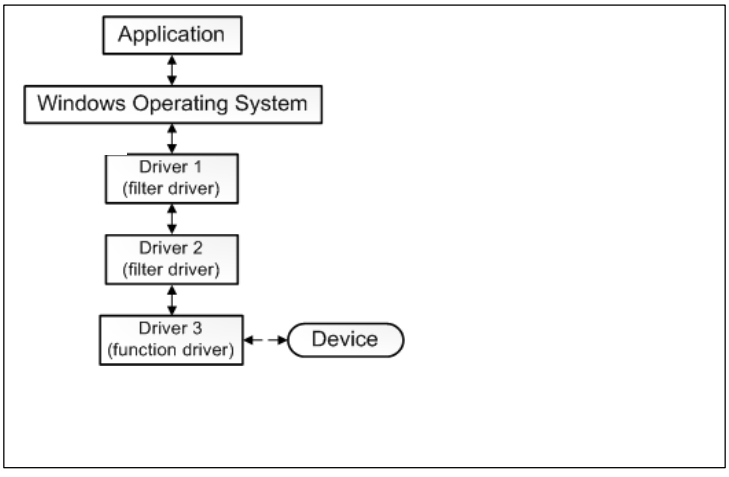
Il est difficile de donner une définition précise unique pour le terme *pilote*. Dans le sens le plus fondamental, un pilote est un composant logiciel qui permet au système d'exploitation et à un appareil de communiquer entre eux.

Par exemple, supposons qu'une application doit lire certaines données d'un appareil. L'application appelle une fonction implémentée par le système d'exploitation, et le système d'exploitation appelle une fonction implémentée par le pilote. Le pilote, qui a été écrit par la même société qui a conçu et fabriqué l'appareil, sait comment communiquer avec le matériel de l'appareil pour obtenir les données. Une fois que le pilote obtient les données de l'appareil, il retourne les données au système d'exploitation, qui les renvoie à l'application.



NB : Tous les pilotes ne communiquent pas directement avec un appareil.

Pour une demande d'E/S donnée (comme la lecture de données à partir d'un appareil), il existe souvent plusieurs pilotes superposés dans une pile de pilotes qui participent à la demande. La façon conventionnelle de visualiser la pile consiste à utiliser le premier participant en haut et le dernier participant en bas, comme illustré dans ce diagramme. Certains pilotes de la pile peuvent participer en transformant la demande d'un format à un autre. Ces pilotes ne communiquent pas directement avec l'appareil ; ils manipulent simplement la requête et la transmettent aux pilotes qui sont plus bas dans la pile.



b. Types de pilotes :

À cause de la diversité des matériels modernes et des systèmes d'exploitation, il existe une multitude de pilotes. Ils gèrent l'interface entre le système d'exploitation et le matériel :

- Des imprimantes ;
- Des cartes vidéo ;
- Des cartes réseau ;
- Des cartes son (par exemple : pilotes Realtek AC'97 Audio, rvlkl.exe) ;
- Des bus locaux de divers types, en particulier pour gérer les bus sur les systèmes modernes ;
- Des bus d'entrée/sortie de plusieurs types (par exemple pour les souris, claviers, l'Universal Serial Bus (USB), etc.) ;
- Des disques durs (ATA, Serial ATA, SCSI). En revanche les systèmes de fichiers (NTFS, ReiserFS, ext3fs) ne sont pas considérés comme des pilotes car ils s'adressent non au matériel lui-même, mais déjà à une abstraction de celui-ci ;
- Des scanners, appareils photo numériques, téléphone et caméscopes.

Les niveaux d'abstraction pour les pilotes sont fréquemment :

- Du côté matériel :
    - Interfaçage direct ;
    - Utilisation d'une interface de plus haut niveau (par exemple : vidéo BIOS) ;
    - Utilisation d'un autre pilote de plus bas niveau (par exemple : les pilotes de systèmes de fichiers) ;
    - Simulation du fonctionnement avec un matériel, alors qu'il fait complètement autre chose ;
- Du côté logiciel :
    - Permettre au système d'exploitation l'accès direct aux ressources matérielles ;
    - Mettre en œuvre uniquement des primitives ;
    - Mettre en œuvre une interface pour logiciel sans pilote (par exemple : TWAIN) ;
    - Mettre en œuvre un langage, parfois de haut niveau (par exemple : PostScript).

Au contraire de la plupart des logiciels de niveau utilisateur, qui peuvent être arrêtés sans affecter le reste du système, un bug dans un pilote peut mener à des dysfonctionnements du système, et dans de plus rares cas sévèrement endommager les données voire le matériel lui-même.

c. Importance :

Un matériel sans pilote ne peut pas fonctionner, les deux sont indissociables. C’est pour cela qu’il existe des pilotes pour chaque matériel utilisé en informatique dans

le monde : écran, clavier, souris, carte graphique, disque dur…Sans ces pilotes, rappelons-le, un matériel ne peut pas fonctionner.
Le pilote va permettre à l'ordinateur de dialoguer avec le matériel en question, il va connaitre ses fonctions, son utilité, et ses limites. Le pilote est en quelques sorte le manuel d'utilisation d'un matériel. Sans ce manuel, l'ordinateur ne comprend pas comment l'utiliser.

2. Les périphériques

a. Définition :

Un périphérique informatique est un dispositif connecté à un système de traitement de l'information central (ordinateur, console de jeu, etc…) et qui ajoute à ce dernier des fonctionnalités.

b. Types de périphériques :

On peut classer généralement les périphériques en trois types : les périphériques d'entrée (permettant de fournir des données à un système de traitement de l'information tel qu'un ordinateur.), ceux de sortie et ceux qui agissent dans les deux sens autrement dit périphériques de stockage ou d'entrée-sortie (permettant d'échanger des données entre le processeur et les périphériques qui lui sont associés).

3. Les bus

a. Définition

Un bus est un système de transfert de données entre plusieurs unités fonctionnelles de traitement de données par l'intermédiaire d'une voie de transmission commune, dans lequel les composants ne prennent aucune part à la transmission des données entre les autres participants.

En plus de l'aspect physique permettant de faire transiter les informations, un bus informatique est constitué des circuits d'interface, et du protocole qui définit la manière dont les signaux doivent se comporter pour réaliser ce transfert. Les caractéristiques du matériel conditionnent en partie le type de communication et le protocole peut parfois imposer le type de matériel.

b. Types de bus

Les bus se décomposent en trois sous-ensembles logiques :

- Les *données*, soit le message proprement dit ;

- Les *adresses*, qui permettent d'identifier les composants qui partagent les données ;
- Le *contrôle*, un ensemble de signaux identifiant le type d'action : lecture ou écriture, taille du message, etc…

NB : La majorité de périphériques sont plus lents que le processeur alors pour ne pas ralentir le système, celui-ci propose différents modes de dialogue :

- Dialogue par le flux d'octets (la console et le clavier) ;
- Dialogue par lecture/copie de blocs (disque dur) ;
- Par socket ;
- Par le lien symbolique ;

4. Autres notions nécessaires :

Dans la gestion des entrées-sorties, il y a quelques autres notions nécessaires :

1) Redirection :

Les redirections sont en fait un mécanisme de communication inter-processus fourni par Linux pour permettre aux programmes (processus) et aux fichiers de communiquer entre eux, en transférant la sortie d'un programme/fichier vers l'entrée d'un autre programme/fichier.

Dans Linux, les redirections sont matérialisées par des opérateurs < >. L'opérateur de redirection est essentiellement un tampon ou un bloc de données qui ont deux descripteurs de fichiers ; l'un est utilisé pour la lecture, et l'autre pour l'écriture.
Comme mentionné dans le paragraphe précédent, les symboles <> sont les opérateurs de redirection ; le symbole supérieur à > est utilisé pour envoyer la sortie d'un programme (processus) vers un fichier en entrée.
De l'autre, le symbole inférieur à < est utilisé pour envoyer la sortie d'un fichier à un programme (processus) en entrée.

- **La sortie standard utilisant le symbole >** : la sortie de l'opérande de gauche est envoyée à l'opérande de droite, donc l'opérande de droite prend l'entrée à partir de l'extrémité de lecture du symbole >.
- **Sortie standard utilisant le symbole <** : La sortie de l'opérande de droite est envoyée à l'opérande de gauche, donc l'opérande de gauche prend l'entrée de l'extrémité d'écriture du symbole <.

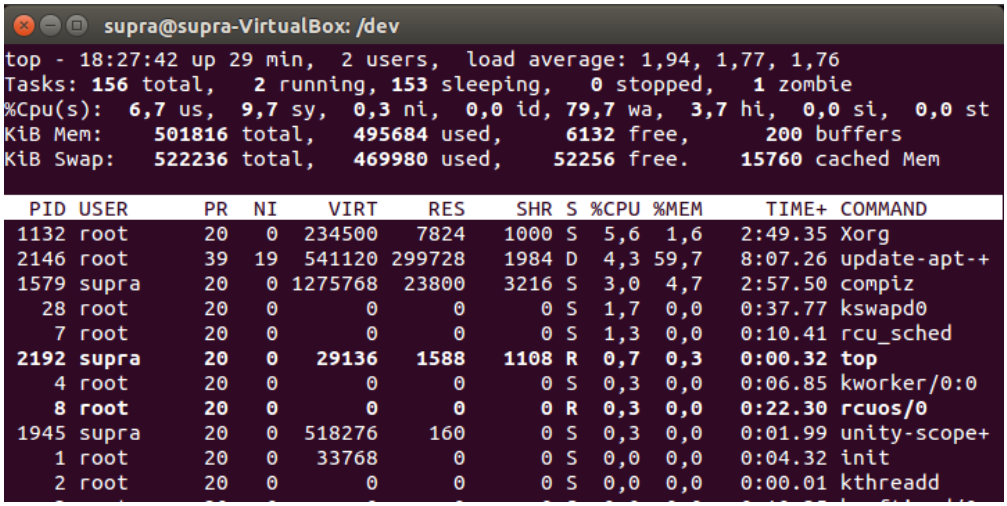
2) Fonctions de l'interface :

- Ouvrir : fournit un descripteur utilisé dans la suite des opérations
- Fermer : invalide le descripteur
- Lire : transfère en mémoire une suite de caractères depuis le périphérique

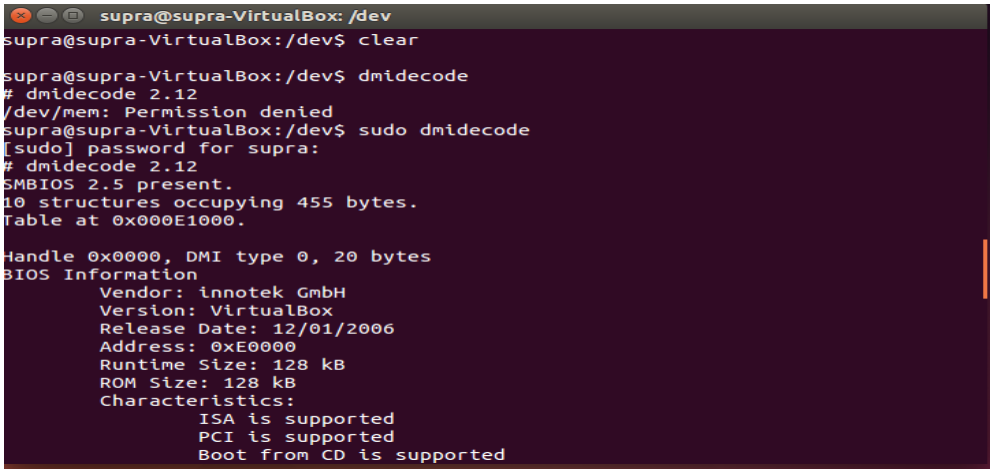
- Écrire : transfère depuis la mémoire une suite de caractères sur le périphérique
- Se-déplacer : lorsque le périphérique le permet, se déplacer dans le flot (périphérique adressable)
  Il sied de signaler que les fonctions primitives lire et écrire manipulent un flot d'octets elles n'effectuent donc aucune transformation du format interne vers le format externe des données.

II. Quelques exemples de cas d'utilisation

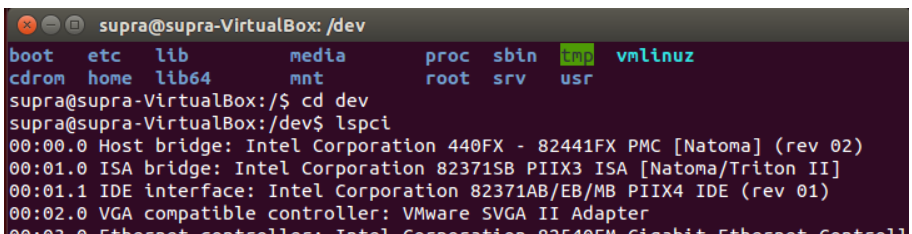
a. La commande « Top »



b. La commande « dmidecode »



c. La commande « lspci »



Précisons que les commandes exécutées ci-dessus sont loin d'être les seules existantes, nous les avons mises à titre d'aperçu de la multiplicité et les possibilités infinies qu'offre cette notion de gestion système d'entrée-sortie sois linux.

III. Conclusion

En somme, la gestion de système d'entrées/sorties est une fonctionnalité essentielle pour tous les systèmes informatiques modernes. Elle permet la communication efficace entre les périphériques externes et le système informatique, permettant ainsi le stockage, le traitement et l'affichage des données. Elle permet également aux différents processus de partager les périphériques d'E/S sans perturber les autres processus en cours d'exécution.

Enfin, la gestion des E/S est importante pour garantir une utilisation efficace des ressources matérielles, en évitant les conflits de ressources, en minimisant les temps d'attente et en optimisant les transferts de données.

Bibliographie

KASENGEDIA, P. (2021 - 2022). *Cours de système d'exploitation L2LMD.* Kinshasa: UNIKIN.

Kozierok, C. (2011). *The TCP/ IP Guide: A Comprehensive, Illustrated Internet Protocols Reference .* No Starch Press.

R., L. (2010). *Linux Network Administrator's Guide .* O'Reilly Media .

Silberschartz, A. G. (2012). *Operating System Concepts .* John Willey & sons.

Tanenbaum, A. S. (2015). *Operating Systems : Design and Implementation .* Prentice Hall.