

INTRODUCTION AUX SYSTÈMES INFORMATIQUES

INPUT & OUTPUT

Victor Ostromoukhov

Département d'Informatique et de Recherche Opérationnelle

Http : [//www.iro.umontreal.ca/~ostrom/](http://www.iro.umontreal.ca/~ostrom/)

E-mail : ostrom@iro.umontreal.ca

INPUT & OUTPUT SOMMAIRE

Introduction	2
I/O Programmé	3
Interruption de Programme	5
Accès Direct à la Mémoire	9
I/O - Interruption DMA	10
Bus	11
Bus - Port - Driver	12
Bus d'interface SCSI	13
Bus d'interface FireWire	14
Bus d'interface USB	15
Exemple de périphérique : Le Disque Dur	17

INPUT & OUTPUT INTRODUCTION

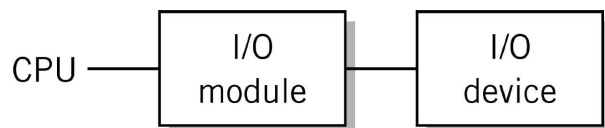
Introduction

La fonction d'un ordinateur est le **traitement de l'information** (fonction réalisée au niveau de la mémoire et l'UC). L'ordinateur acquiert cette information et restitue cette information au moyen d'E/S ([I/O])

Dans le modèle LMC

▷ E/S modélisées par des paniers IN/OUT

Dans un ordinateur réel ces derniers sont remplacé par un contrôleur [I/O module] dont le but est d'adapter la diversité des périphériques (débit, tps de réponse, format des données, forme des signaux de commande, etc.) à une interface commune, obéissant aux normes adoptés par le constructeur



Méthodologies de communications

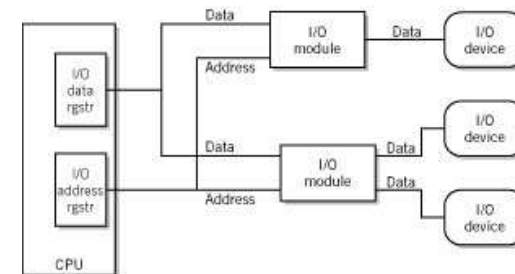
1. I/O programmé [*Programmed I/O*]
2. Interruptions [*Interrupt Driven I/O*]
3. Accès direct à la mémoire [*Direct Memory Access*]

INPUT & OUTPUT I/O PROGRAMMÉ

I/O Programmé

Modèle le plus simple dans lequel le module I/O est connecté à une paire de registre I/O (donné & adresse) dans le CPU *via* un bus

▷ analogies entre les registres et les paniers IN/OUT du modèle LMC

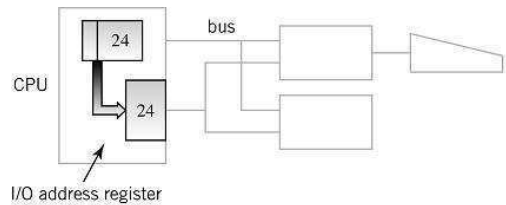


- Transfert d'un mot à la fois et le CPU reste bloqué durant toute la durée de l'échange!
- En pratique, il y a plusieurs périphériques [devices] connecté au CPU (on utilise le champs d'adresse de l'instruction I/O)
- En pratique, il est courant pour un module I/O d'avoir plusieurs adresses, chacune représentant une commande de contrôle différente

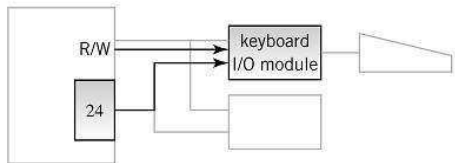
▽

Méthodologie de communication très lente
Exemple : clavier d'ordinateur

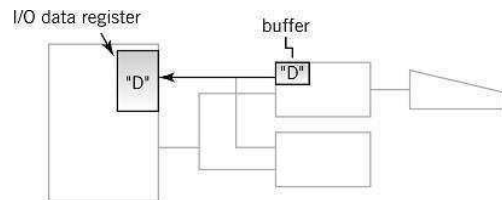
INPUT & OUTPUT I/O PROGRAMMÉ : EXEMPLE



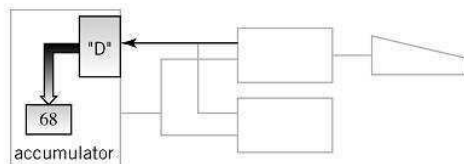
1. CPU executes INPUT 24 instruction. Address 24 is copied to the I/O address register.



2. Address 24 is recognized by the keyboard I/O module. A read/write control line indicates that the instruction is an INPUT.



3. A buffer in the I/O module holds a keystroke, in this case ASCII 68, the letter "D". The data is transferred to the I/O data register.



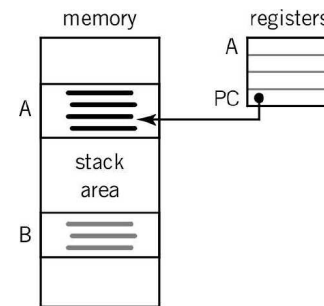
4. From there it is copied to the appropriate accumulator or general-purpose register, completing the operation.

INPUT & OUTPUT INTERRUPTION DE PROGRAMME

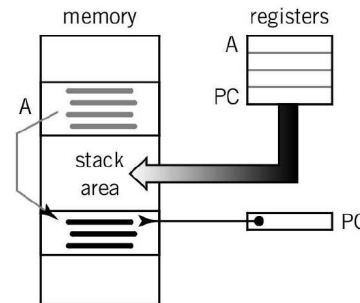
Interruptions de programme

Introduction d'un signal d'interruption [*interrupt*] envoyé au CPU par le périphérique concerné à l'instant où il est prêt à effectuer un transfert

Ce signal provoque l'interruption du programme en cours d'exécution. Le CPU s'occupe de l'échange (prog. de service de l'interruption). Après le transfert le CPU reprend l'exécution du programme interrompu.

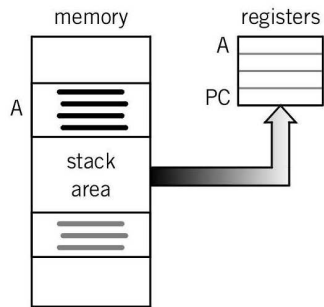


1. Before interrupt arrives, program A is executing. The program counter points to the current instruction.



2. When the interrupt is received by the CPU, the current instruction is completed, all the registers are saved in the stack area (or in a special area known as a process control block). The PC is loaded with the starting location of program B, the interrupt handler program. This causes a jump to program B, which becomes the executing program.

INPUT & OUTPUT INTERRUPTION DE PROGRAMME



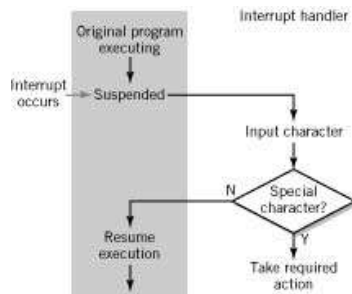
3. When the interrupt routine is complete, the registers are restored, including the program counter, and the original program resumes exactly where it left off.

Gain en temps considérable par rapport à une liaison directe (I/O programmé)

Néanmoins (bcp) de temps est perdu dans l'exécution du prog. de service de l'interruption

Exemple

clavier : combinaison I/O programmé / interruption



INPUT & OUTPUT INTERRUPTION DE PROGRAMME

Gestions des interruptions

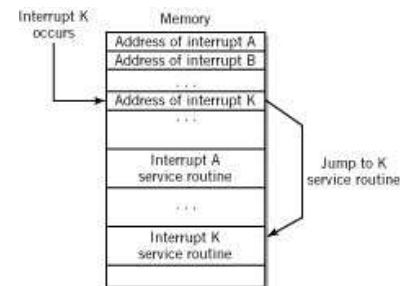
1- Comment l'ordinateur identifie le périphérique demandant une interruption ?

Priorité ?

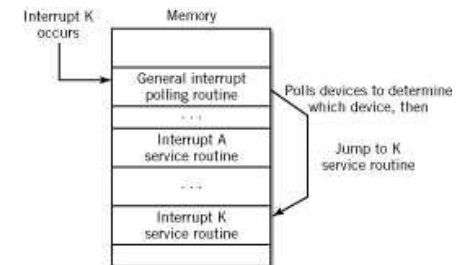
2a- Que se passe t-il si une interruption arrive alors que l'ordinateur en traite déjà une autre ?

2b- Que se passe t-il si plusieurs interruptions arrivent en même temps ?

-1-



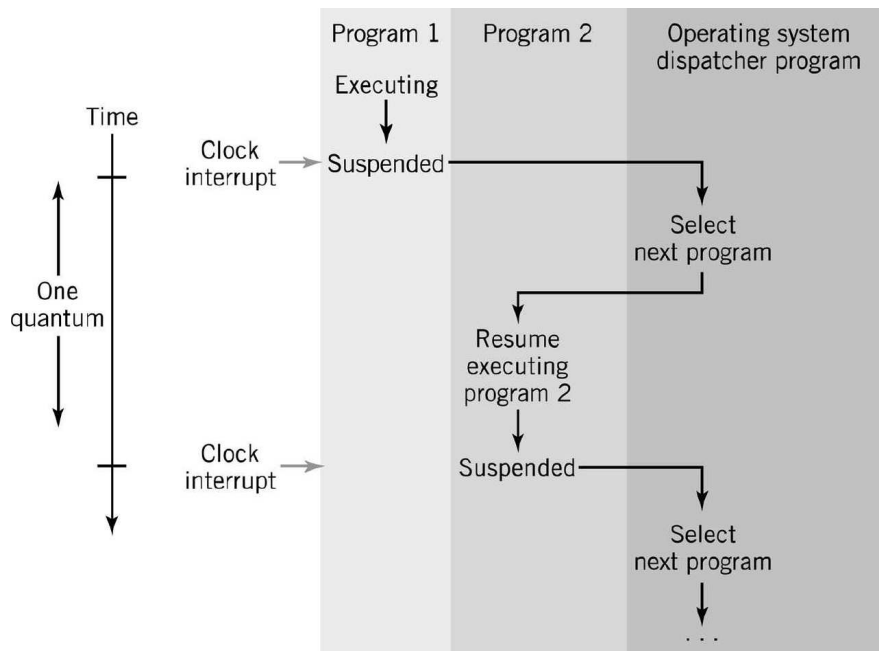
Vectored Interrupt



Polling

INPUT & OUTPUT INTERRUPTION DE PROGRAMME

-2-

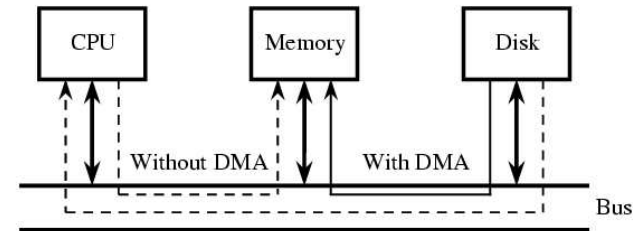


Traitement de multiples interruptions

INPUT & OUTPUT ACCÈS DIRECT A LA MÉMOIRE

Accès direct à la mémoire

Permet au périphérique d'accéder à la mémoire sans passer par le CPU. Si le CPU et le DMA demandent simultanément un accès à la mémoire, le DMA passe en priorité



- Le DMA se charge entièrement du transfert d'un **bloc** de données
- Le CPU initialise l'échange en lui donnant l'identification du périphérique concerné, le sens du transfert, l'adresse en mémoire centrale du premier mot à transférer et le nombre de mot concernés par l'échange
- Lorsque l'échange est terminée, le DMA signale au CPU que l'opération est terminée **par interruption**

INPUT & OUTPUT

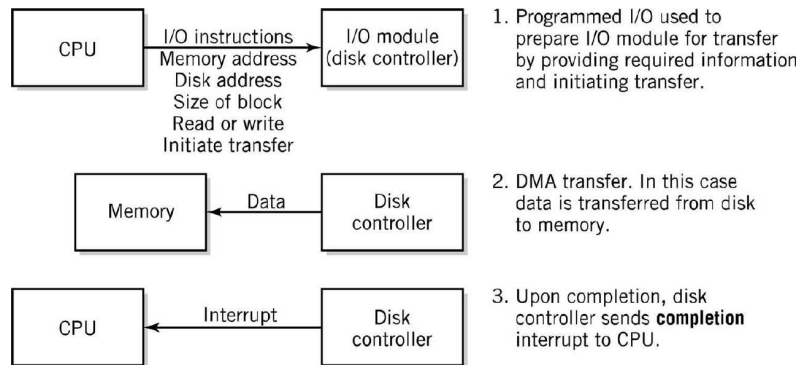
I/O - INTERRUPTION - DMA

I/O - Interruption - DMA

Pour les petits transferts, il est préférable d'utiliser la méthodologie de communication I/O programmé, c'est aussi le cas pour l'initialisation et le contrôle d'un transfert DMA

Exemple

Écriture/Lecture d'un bloc mémoire sur un disque



INPUT & OUTPUT

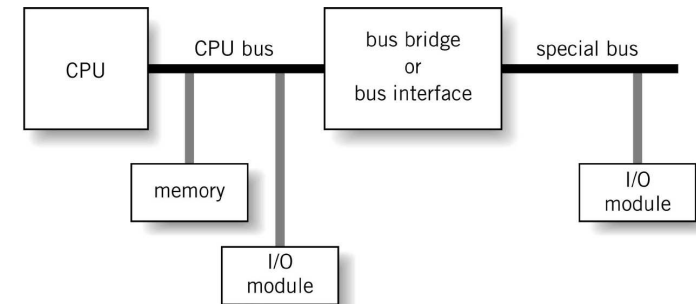
BUS

Bus

Bus : Lignes associées aux données, adresses, contrôle (horloge, interruptions, lecture/écriture, exceptions, etc.)

Un PC actuel est organisé autour d'un (non exhaustif)

- Bus CPU
- Bus PCI (32/64-bits de largeur pour les donnée [partagé avec les adresses])
- (qqfois) un bus ISA (16 bits pour les donnée, adresses et données ≠ lignes)



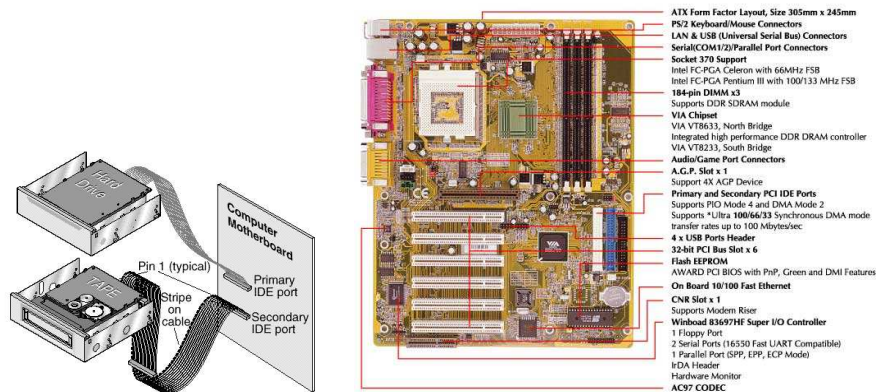
- Les interface de bus étendent la flexibilité de l'architecture en convertissant le signal provenant d'un bus à un autre

Bus série ▷ données transférées séquentiellement
 Bus parallèle ▷ 1 ligne individ. /bit de donnée

INPUT & OUTPUT BUS - PORT - DRIVER

Rappel : Le bus fournit un moyen pour le périphérique de communiquer avec le CPU

Un **port** [*port*] est simplement un connecteur à l'extrémité d'un bus dans lequel est connecté [*plugged*] un périphérique



- Le contrôle générale de ce port est fait par le **contrôleur de port** (*port controller*) qui est connecté à un bus de l'ordinateur (ISA, PCI, AGP, etc.)
- Le logiciel qui contrôle les I/O à partir de ces appareils électroniques (périphérique, contrôleur de port, bus de l'ordi) est le **pilote de périphérique** [*device driver*]
- Les [*drivers*] sont soit dans l'OS ou soit doivent être installé dans l'OS par l'opérateur
- Il s'occupe, en plus du transfert, de la gestion des incidents, détection des erreurs, conversion de format, etc.

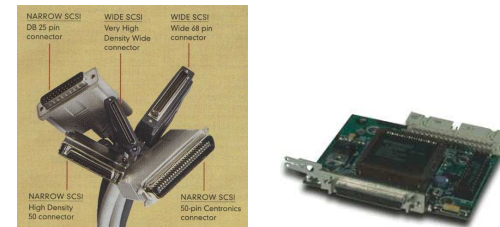
INPUT & OUTPUT BUS D'INTERFACE SCSI

Bus d'interface

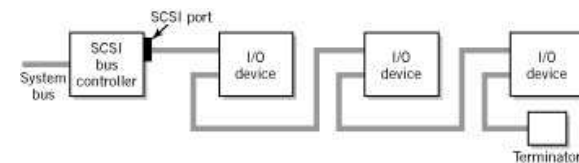
Les périphériques connecté à l'un des bus de l'ordinateur utilisent un bus externe, comme le bus SCSI, USB, IEEE 1394, etc.

SCSI

[*Small Computer System Interface*], Bus d'I/O parallèles, référence en matière d'interface pour les disques dur, CD-ROM, lecteur DVD, scanners, etc.



- Permet de connecter en parallèle près de 8 unités



- SCSI-2 : Bus de données 16 ou 32 bits, jusqu'à 40 MBytes/s sur un bus 32 bits
- SCSI-3 (Ultra-SCSI) : vitesse doublée pour la même taille de Bus

▷ On arrive au limite des interfaces parallèles ◁

INPUT & OUTPUT BUS D'INTERFACE FIREWIRE

IEEE 1394



[FireWire] baptisé [IEEE 1394-1395 standard for High Performance Serial Bus]

Bus série externe très rapide jusqu'à 50/150/400 MBytes/s utilisé pour connecter des périphériques nécessitant des transferts de grandes qtés de données en tps réel

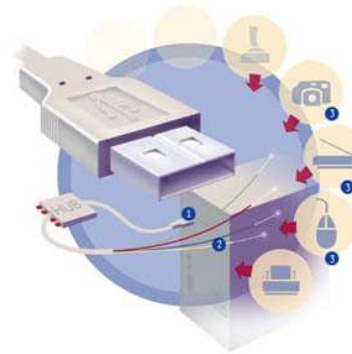
- Bus multipoint, cuivre ou fibre optique, Mode de transfert isochrone
- On peut y connecter jusqu'à 63 périphériques



- Il supporte le [Hot plug and play]

INPUT & OUTPUT BUS D'INTERFACE USB

USB

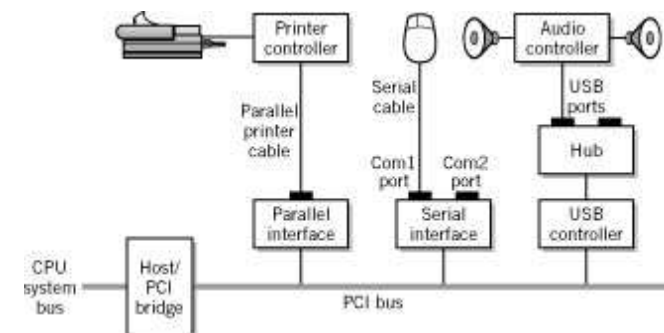


[Universal Serial Bus]

Développé par 7 compagnies : Compaq, Digital, IBM, Intel, Microsoft, NEC, Northern Telecom

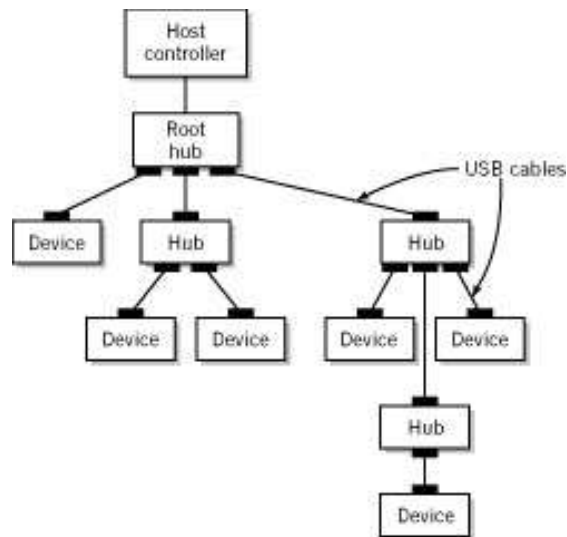
Bus série supportant des taux de transfert de 1.5 MBytes/s

- USB supporte le [Hot Plug & Play]
- Ce bus, à bas et moyen débit, est complémentaire avec le FireWire et de plus en plus utilisé



INPUT & OUTPUT BUS D'INTERFACE USB

Sur un même port USB, on peut connecter jusqu'à 127 périphériques, sur le port du bus ou connecté à des concentrateurs d'extension [*hub*].



Le rôle d'un [*Hub*] est de contrôler les périphériques qui lui sont reliés, détecte les modif. de de raccordement, gère l'alimentation électrique, répète les données transmises, etc. supporte 127 périphériques

INPUT & OUTPUT EXEMPLE DE PÉRIPHÉRIQUE : LE DISQUE DUR

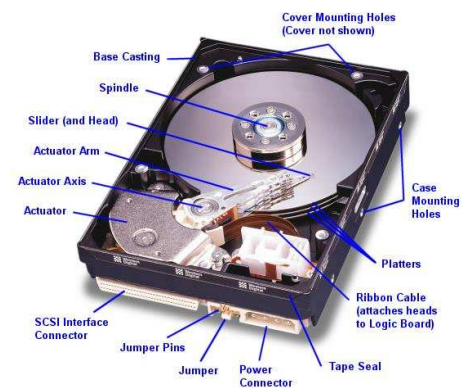
Disque dur

Mémoire de stockage secondaire utilisé dans le modèle de mémoire virtuelle

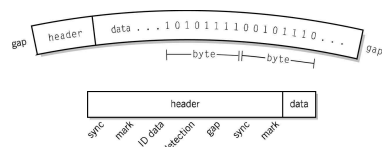
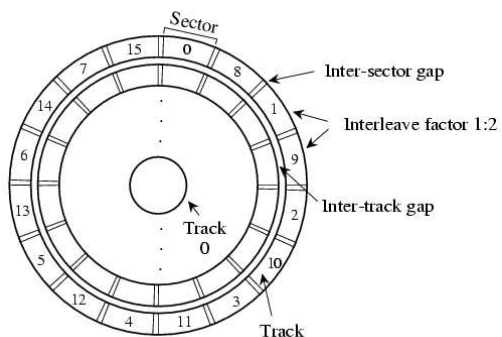
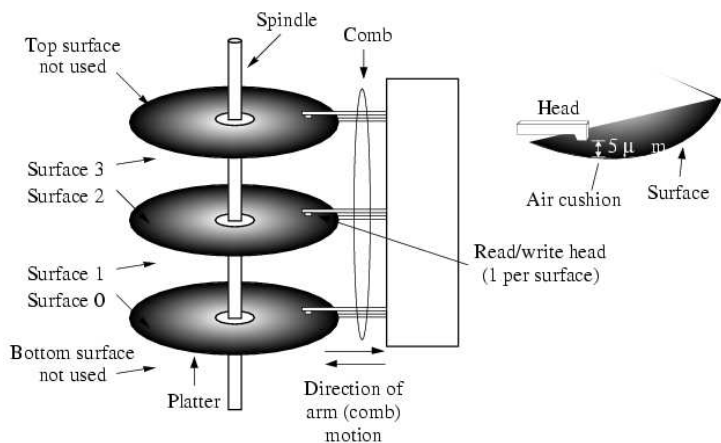
Storage medium	Typical average access time	Data throughput rate
CPU registers		
Cache memory	2-10 nsec	
Conventional memory	10-50 nsec	
Expanded storage	75-500 nsec	
Flash memory	10.8 msec	3.5-8 MB/sec
Hard disk	10-50 msec	600-6000 KB/sec
Floppy disk	95 msec	100-200 KB/sec
CD-ROM	100-600 msec	500-4000 KB/sec
Tape	.5 sec. and up	2000 KB/sec (cartridge)

- Taux de transfert 100000 fois plus lent que le transfert impliquant la plus lente des mémoires (+ Tps d'accès) mais mémoire permanente

Fonctionnement



INPUT & OUTPUT EXEMPLE DE PÉRIPHÉRIQUE : LE DISQUE DUR

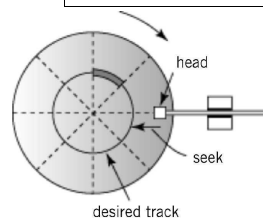


1 Bloc = 512 Octets + entête [footer]

INPUT & OUTPUT EXEMPLE DE PÉRIPHÉRIQUE : LE DISQUE DUR

Opérations nécessaires pour localiser/transférer

Tps de recherche Moyen

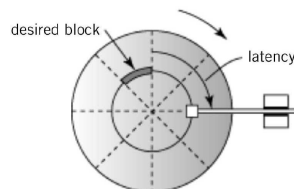


[Seek Time]

Tps nécessaire pour que la tête bouge d'une piste à une autre

▷ [Average Seek Time]

Tps de latence

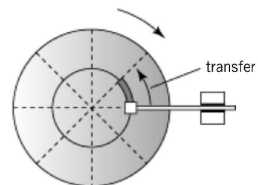


[Latency Time]

Tps nécessaire pour que la disque tourne jusqu'au bon secteur

▷ [Average Latency Time]

Tps de transfert



[Transfert Time]

Tps nécessaire pour transférer un bloc de donné au contrôleur de disque

INPUT & OUTPUT

EXEMPLE DE PÉRIPHÉRIQUE : LE DISQUE DUR

- Tps Latence Moyen = $\frac{1}{2} \times \frac{1}{\text{vitesse rotation}}$

Exemple

Pour un disque tournant à 3600 rpm (révolutions par minute) (ou 60 rps)

Tps Latence Moyen = $\frac{1}{2} \times \frac{1}{60} = 8.33 \text{ msec}$

- Tps Transf. Moyen = $\frac{1}{\text{nb de secteur/pistes} \times \text{vit. rotation}}$

Exemple

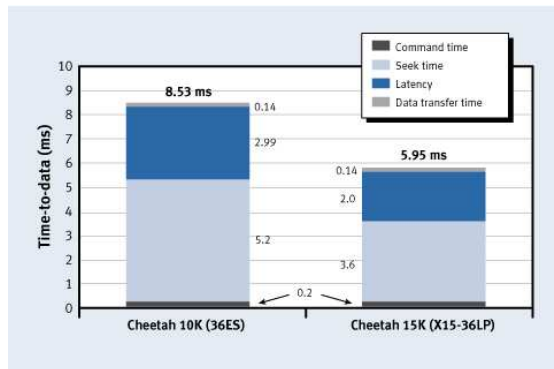
Pour le même disque

Tps Transfert = $\frac{1}{30 \times 60} = 0.55 \text{ msec}$

Tps total et moyen d'accès à un bloc

=

Tps de rech. moy. + Tps latence moy. + Tps transfert moy.



INPUT & OUTPUT

EXEMPLE DE PÉRIPHÉRIQUE : LE DISQUE DUR

Remarques

Le CPU est capable d'exécuter des *millions* d'instructions dans le temps requis pour un simple accès disque !!

Le Temps de Transfert de plusieurs blocs est plus important dans le cas où le disque est très fragmenté



Disque Dur vs CD-ROM

