

Révision finale



GIF-1001 Ordinateurs: Structure et Applications, Hiver 2015
Jean-François Lalonde

Logistique — examen final

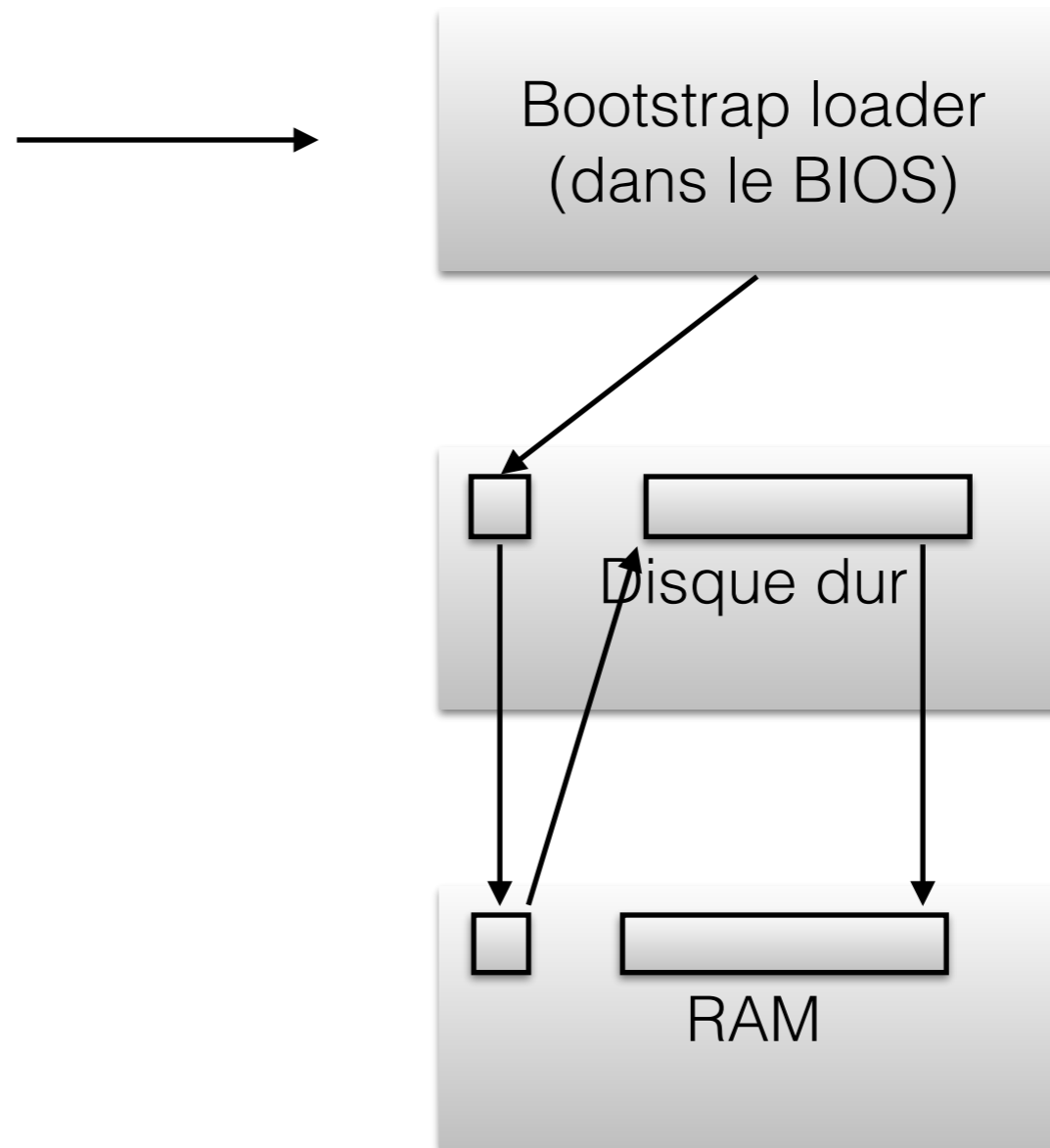
- 40% de la note totale
- PLT-1112 (amphithéâtre)
- Mardi 28 avril, 14h30 à 16h20
- Une feuille 8.5 x 11, recto-verso, écrite à *la main*

Ressources

- Exercices supplémentaires et examens des années antérieures disponibles sur le site web du cours
- <http://vision.gel.ulaval.ca/~jflalonde/cours/1001/h15/index.html#ressources>

Démarrer un ordinateur (“bootstrapping”)

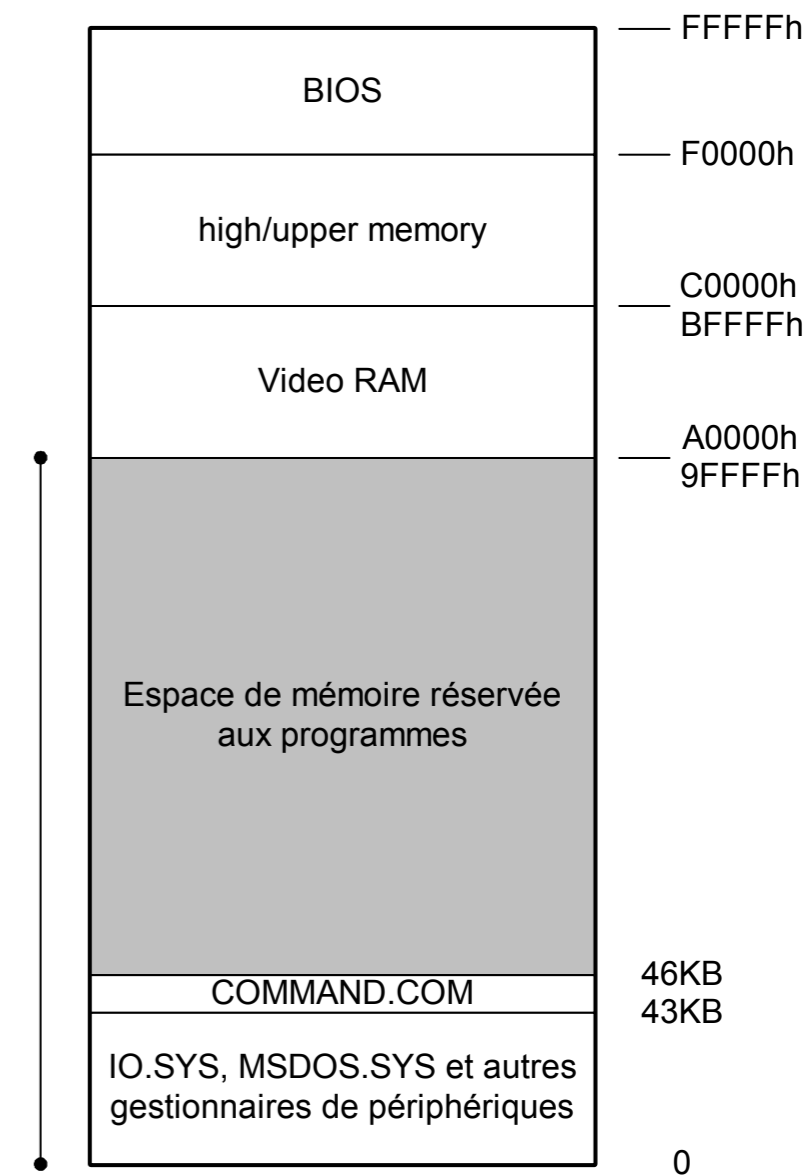
1. Exécution du “bootstrap loader”, qui est stocké dans une ROM
2. Le bootstrap “trouve” l’emplacement du système d’exploitation sur le disque dur (habituellement toujours au même endroit)
3. Le bootstrap copie le **programme de chargement (“loader”) du SE** en RAM
4. Branchement vers la RAM pour que le **“loader” du SE** puisse s’exécuter
5. Le **“loader” du SE** copie ensuite le **SE en RAM**



Carte mémoire typique de MS-DOS

- Après le démarrage, le bas de la mémoire (46Ko) contient le système d'exploitation.
- Les programmes sont chargés dans un espace restreint de 594Ko.
- La RAM VIDEO sert à l'affichage.
- L'espace mémoire entre 768Ko et 960Ko est la mémoire haute. Cette mémoire peut servir pour augmenter la taille des programmes.
- Le BIOS, dans le haut de la mémoire, est en ROM.
- Le système d'exploitation est résident en mémoire. Il restera là alors que les applications peuvent être retirées de la mémoire (non-résident).

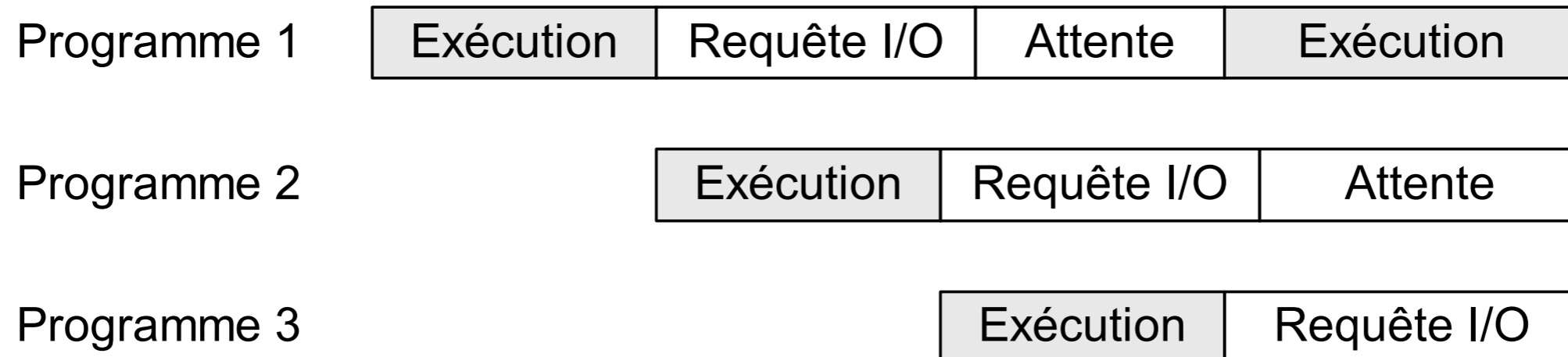
ROM
64K
ROM ou RAM
192K
RAM
128K
RAM
640K



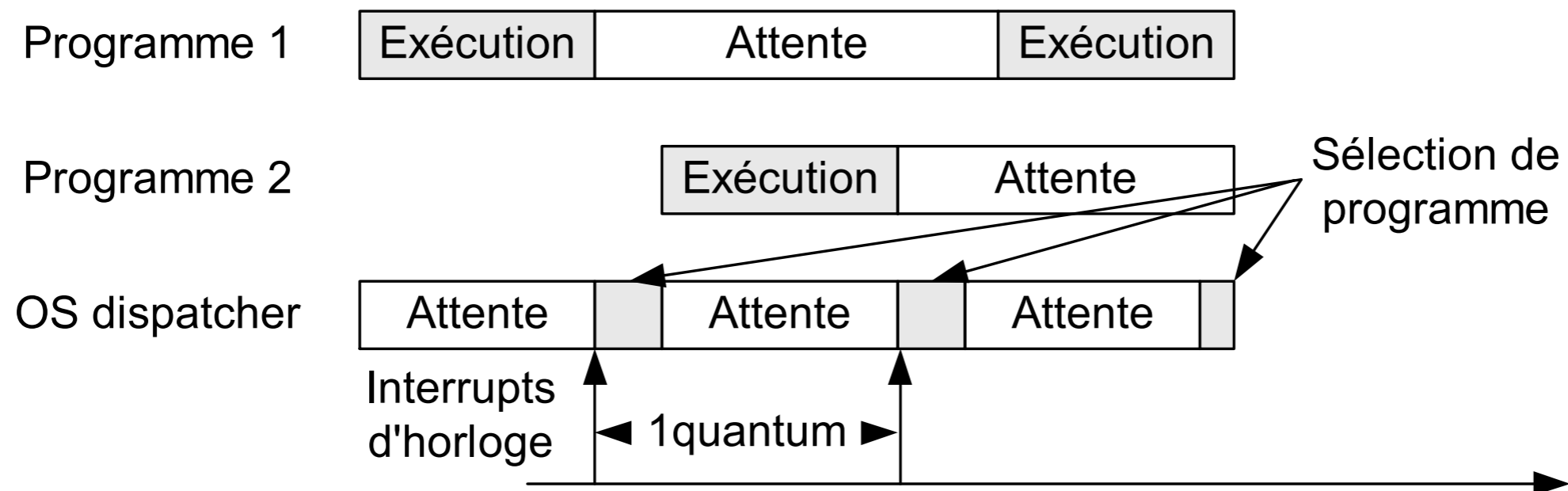
Note: 64K = $2^{(6+10)} = \text{FFFF}+1$

Exécution multi-tâches

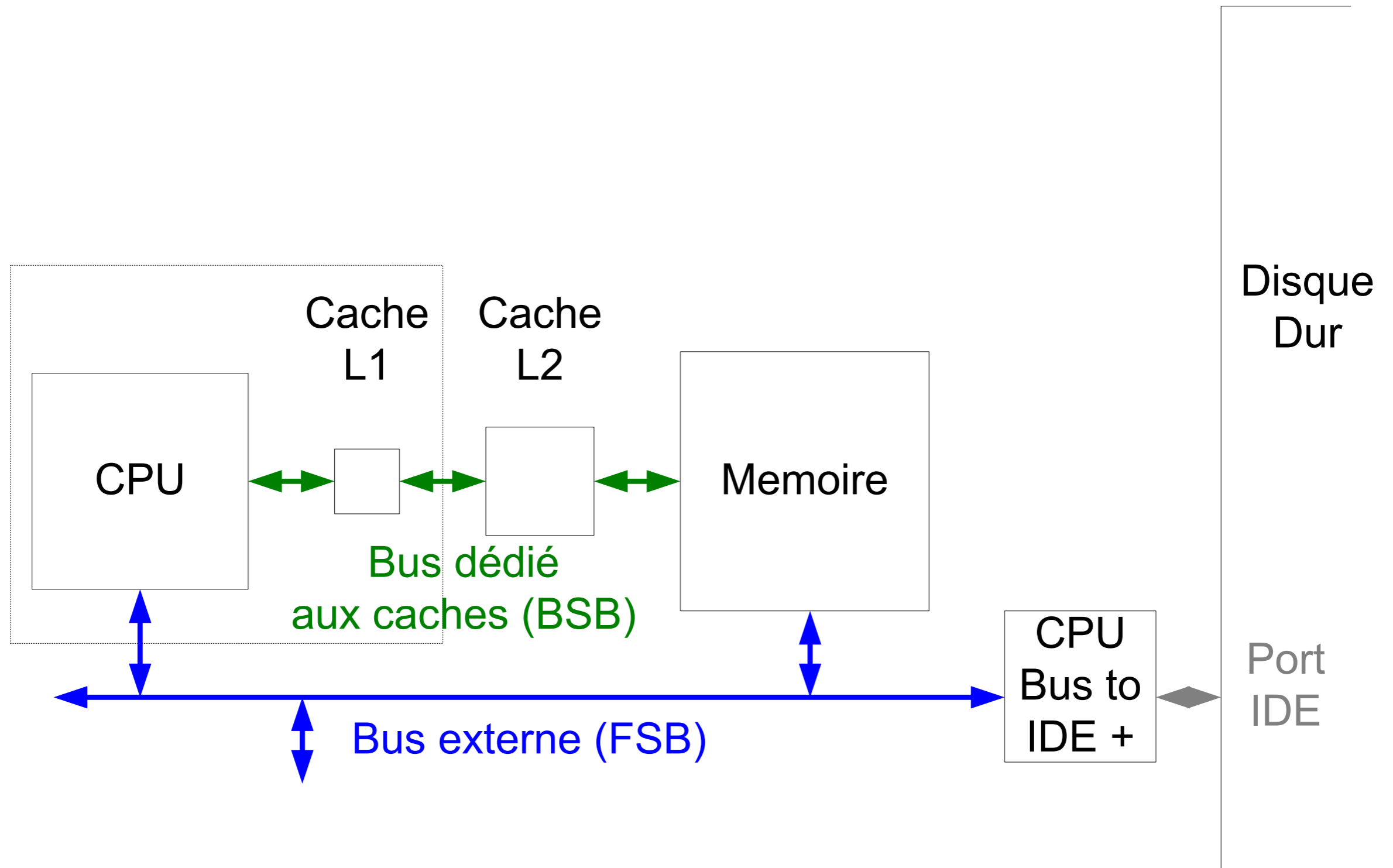
- Partage du CPU lors d'attente après les I/Os



- Partage du CPU dans le temps



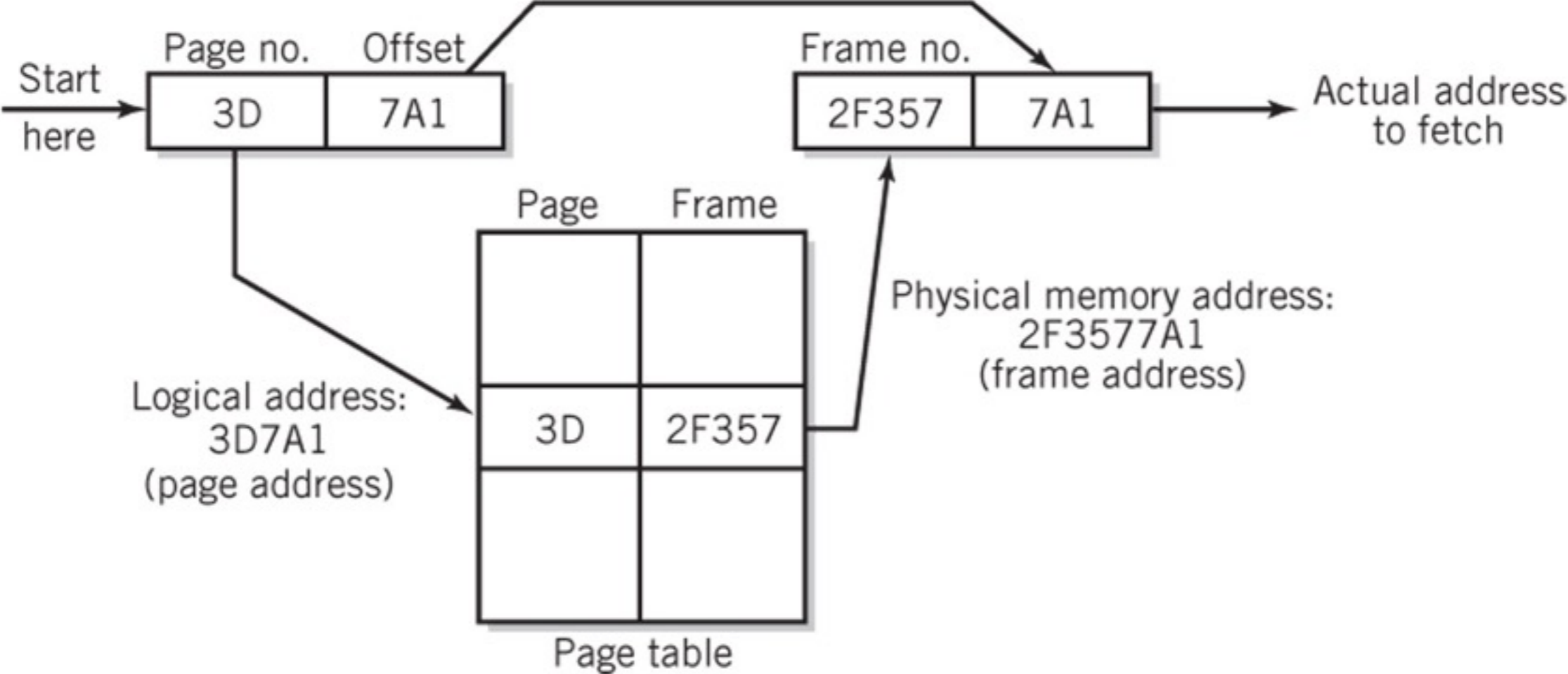
La mémoire d'un ordinateur



Récapitulation

1. Un nouveau programme est copié dans un emplacement disponible en mémoire, de façon contigüe.
2. On peut créer des partitions de taille:
 - **fixe**: la première partition disponible est choisie quand un nouveau processus doit être alloué
 - **variable**: on doit déterminer où créer la partition, nécessite le choix d'un algorithme d'allocation mémoire plus compliqué
3. Le programme utilise des adresses "virtuelles"
4. Le MMU traduit les adresse virtuelles en adresses physiques

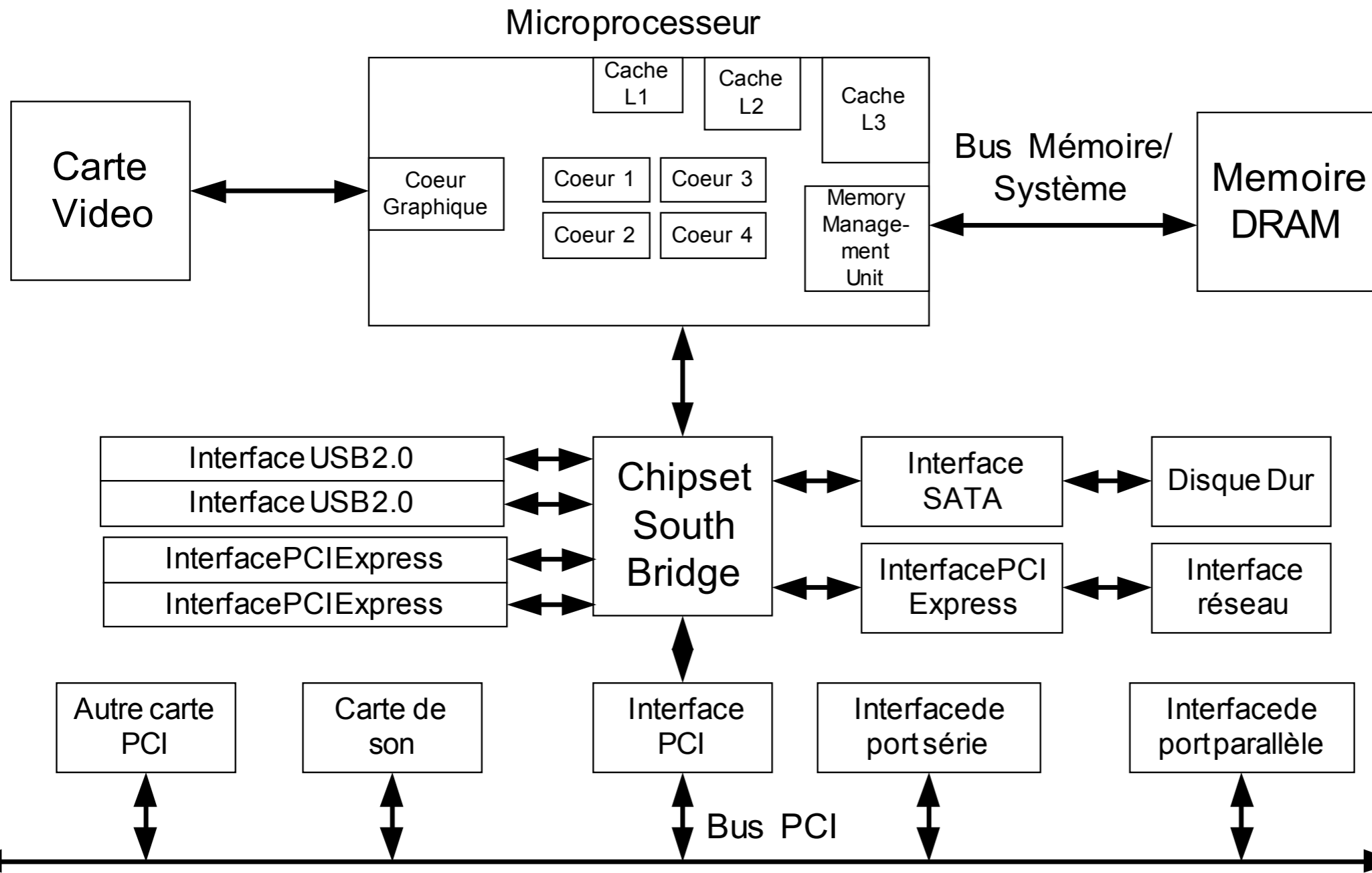
Table des pages



Entrées/Sorties

- Il existe trois techniques principales pour communiquer à partir du CPU/Mémoire vers un périphérique à travers un module de I/O:
 - E/S programmées
 - E/S avec interruptions
 - le DMA

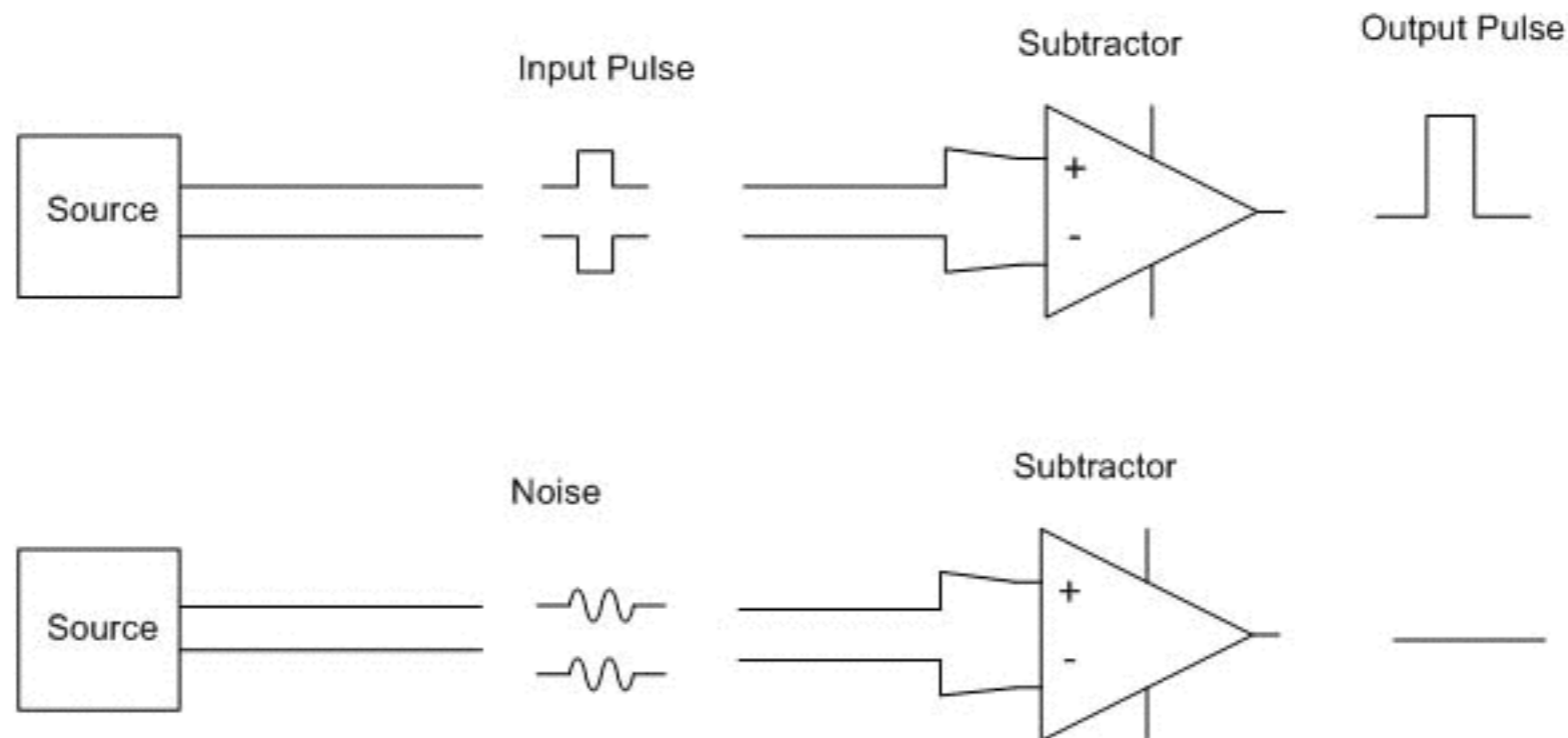
Les Bus de votre Ordinateur (3/4)



Architecture possible des bus d'un PC acheté en 2011

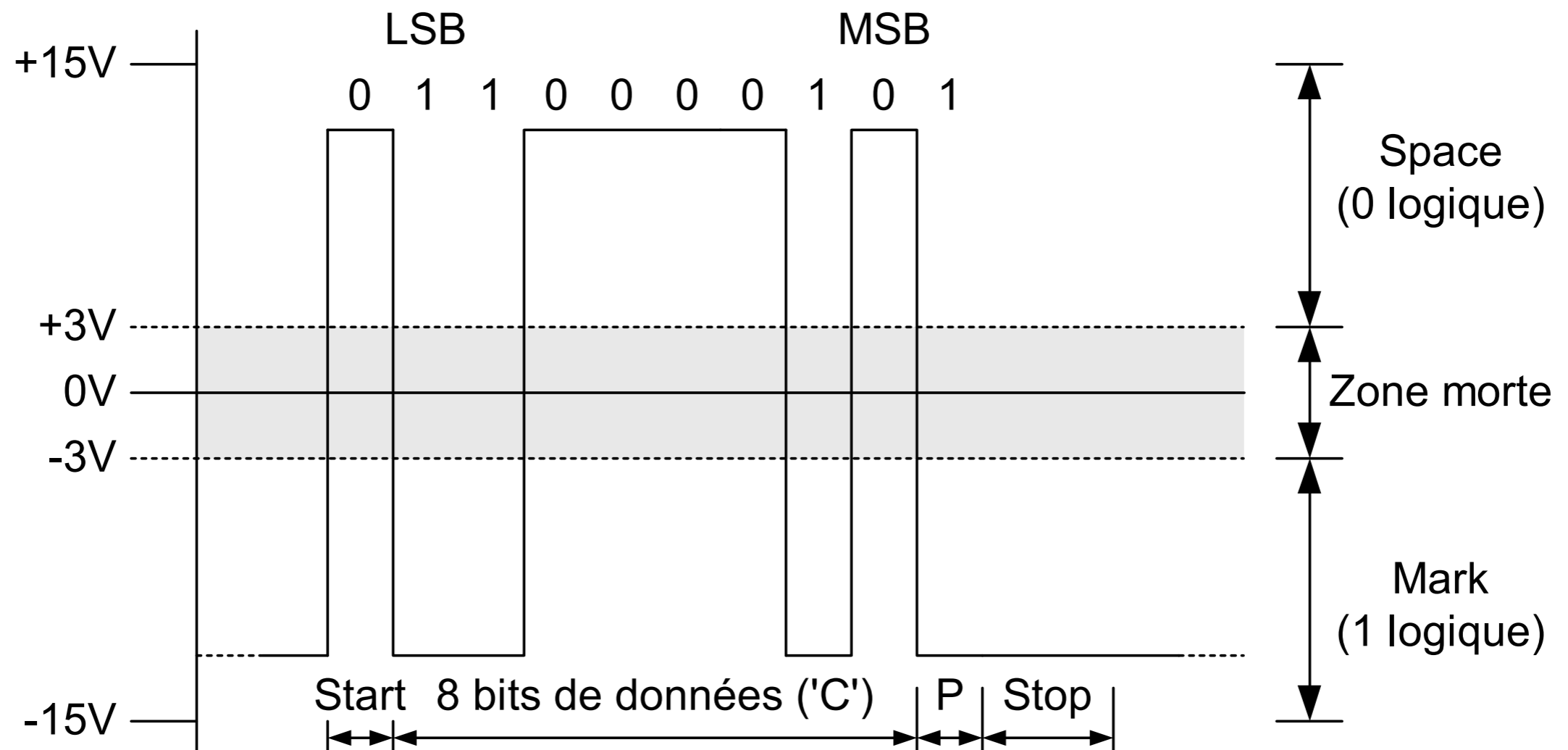
Améliorations des bus

- Les bits transmis sont encodés en mode différentiel. Pourquoi?
 - La différence de tension entre deux signaux propagés sur deux lignes différentes détermine la valeur d'un bit transmis. Des symboles différents sont transmis si la différence est positive ou négative.
 - Le bruit commun sur les deux lignes propageant le signal est éliminé lorsque la différence est effectuée. Très robuste.
 - Lorsque la différence est nulle, le bit est invalide ou une autre information peut être transmise.



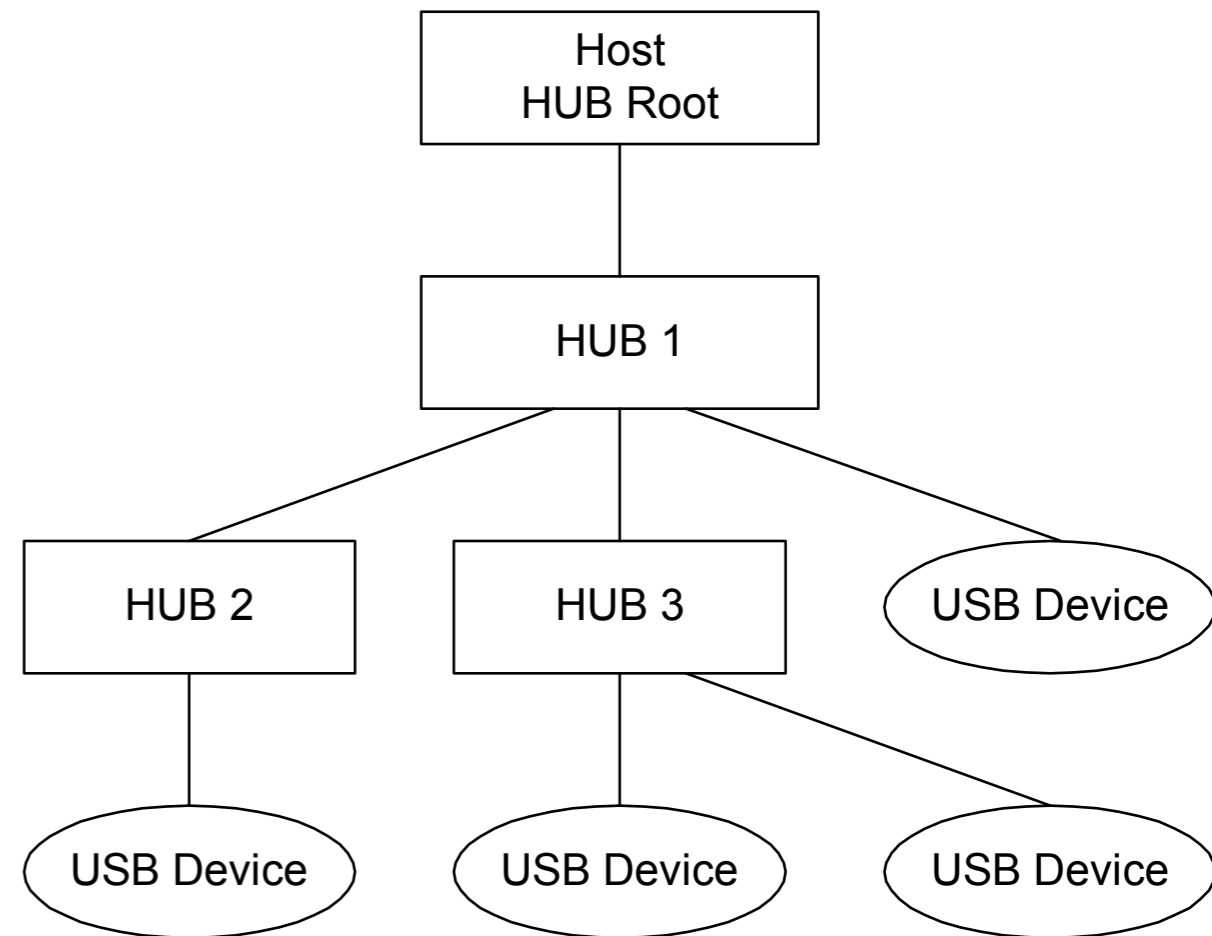
Signaux

- Le signal transmis sur les pins RD et TD va de +15V à -15V:
 - entre +3V et +15V, il est interprété comme un 0 logique;
 - entre -3V et 15V, il est interprété comme un 1 logique;
 - entre -3V et 3V, le signal est considéré invalide.
- Des bits de départs et de fins servent à délimiter les bits de données.
- Il peut y avoir un bit de parité servant à détecter les erreurs. Ce bit est décrit plus loin.



Topologie d'un réseau USB

- Un réseau USB a une topologie en étoile.
- Le port USB est contrôlé entièrement par un contrôleur unique appelé hôte ("host"). Souvent le PC, il initie toutes les communications, et est le maître absolu du bus.
- Les "hubs" permettent de relier plusieurs appareils à un seul port USB.
 - Le rôle principal des hubs est de transférer les données de l'hôte aux périphériques.
 - Chaque hub contrôle ses ports afin de savoir si un appareil s'y connecte
 - Il peut y avoir 5 niveaux de hub en plus du hub racine.
- Il y a 127 appareils maximum dans un réseau USB. Chaque appareil a son adresse.



Matériel — mode différentiel

- USB utilise le mode différentiel:
 - les données sont transmises par la différence entre D+ et D-.
 - lorsque D+ et D- sont forcés près de la masse par l'hôte, cela signifie une demande de reset.

Délais d'accès

- **Temps de déplacement (“seek time”)**: la tête de lecture se déplace radialement vers la piste contenant le bloc de données voulu
 - Ce temps est donné par le fabricant
- **Temps de recherche (“latency time”)**: un moteur tourne le plateau à Vitesse Angulaire Constante (“Constant Angular Velocity”) afin de trouver le bloc de donnée et le secteur voulu
 - $1/(2 \times \text{vitesse de rotation})$
- **Temps de transfert (“transfer time”)**: une fois le bloc trouvé, le moteur tourne le plateau pour lire les données
 - $1/(\text{nombre de secteurs} \times \text{vitesse de rotation})$

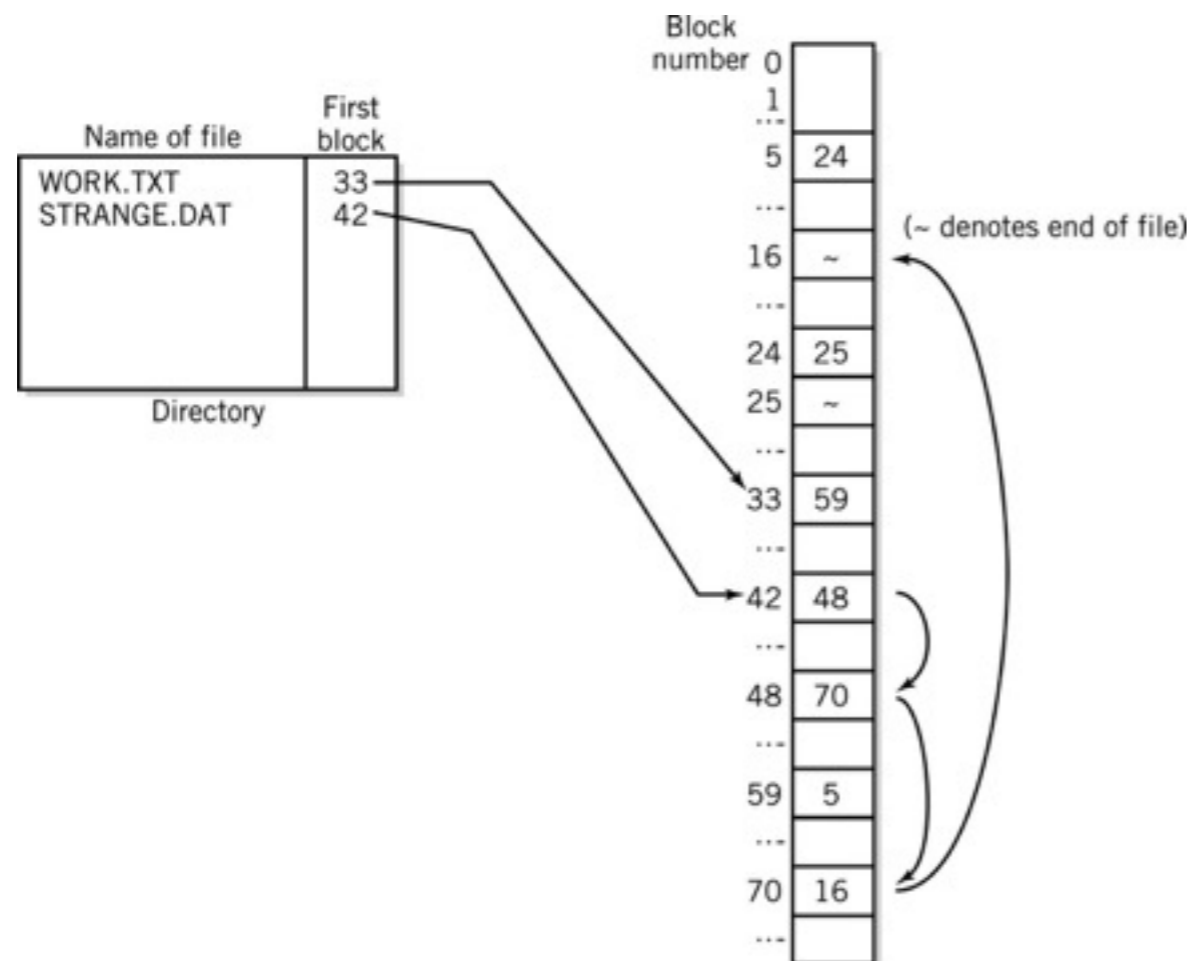
Allocation du disque

- Problème similaire à l'allocation mémoire
- Différents types d'allocation:
 - contigüe
 - chaînée
 - indexée

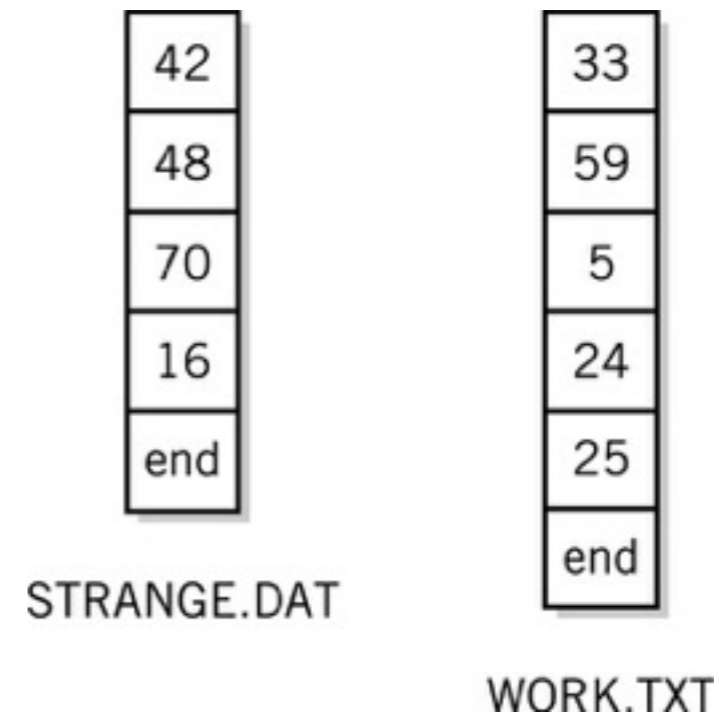
Allocation indexée

- Allocation non-contigüe
- Au lieu d'utiliser une seule table (la FAT) pour tous les fichiers, on utilise une table *par* fichier (appelée la table d'index)

Allocation chaînée



Allocation indexée



Résumé: architectures

- Dans toutes les architectures parallèles, de la mémoire est partagée afin d'échanger des informations entre les différents processeurs.
- Afin d'exécuter des instructions en parallèle et de maximiser l'utilisation des ressources, certaines ressources sont partagées.
- Le niveau de la mémoire partagée et la nature des ressources partagées définissent souvent le nom de l'architecture parallèle:
 - **Pipeline d'instruction**: Les différentes sections de l'unité d'exécution du processeur sont partagées pour exécuter plusieurs instructions en parallèle.
 - **Multithreading**: Les unités d'exécution des pipeline d'instructions sont partagées entre les instructions de différents threads.
 - **Symmetric Multi Processors** (SMP): La mémoire de l'ordinateur est partagée par plusieurs microprocesseurs afin d'exécuter plusieurs threads en parallèle.
 - **Non-Uniform Memory Access** (NUMA): Les entrées/sorties de l'ordinateur, certains bus et, habituellement, les espaces d'adresse, sont partagés afin d'exécuter plusieurs threads en parallèle. En d'autres mots, le système d'exploitation est partagé afin de permettre l'exécution de tâches en parallèle.
 - **Grappe d'ordinateur ou super-ordinateur**: Une application est partagée afin de gérer les threads à exécuter.