

# Programmation Système

Département Informatique

IUT2 de Grenoble

BUT2 - Ressource 3.05

# Organisation du cours

1. Système d'exploitation
2. Processus
3. Partage des ressources
4. Système de Gestion de Fichiers
5. Entrées/Sorties

## 4. Système de Gestion de Fichiers

### 4.1. Révisions [R1.04]

- Mémoire secondaire
- Arborescence de fichiers et répertoires
- Attributs d'un fichier/répertoire (métadonnées)

### 4.2. SGF ext4

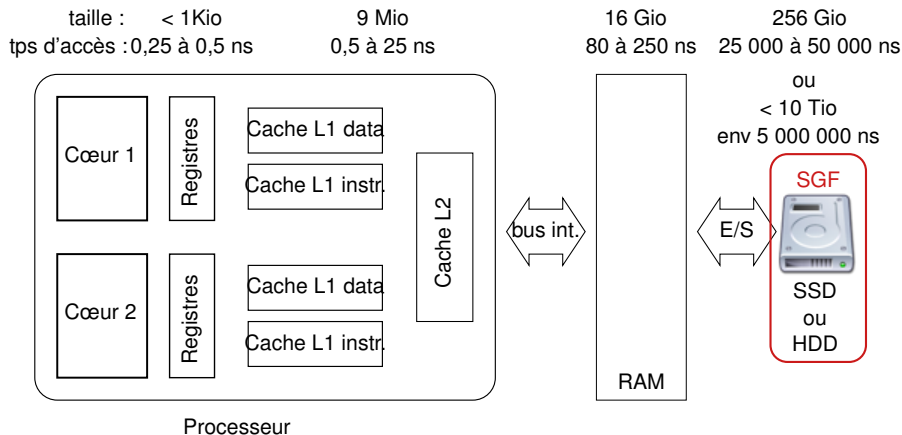
- Structures de données : i-nœud et datablocks
- Protection

### 4.3. Accès aux fichiers par les processus

### 4.4. Une arborescence mais des supports et SGF variés

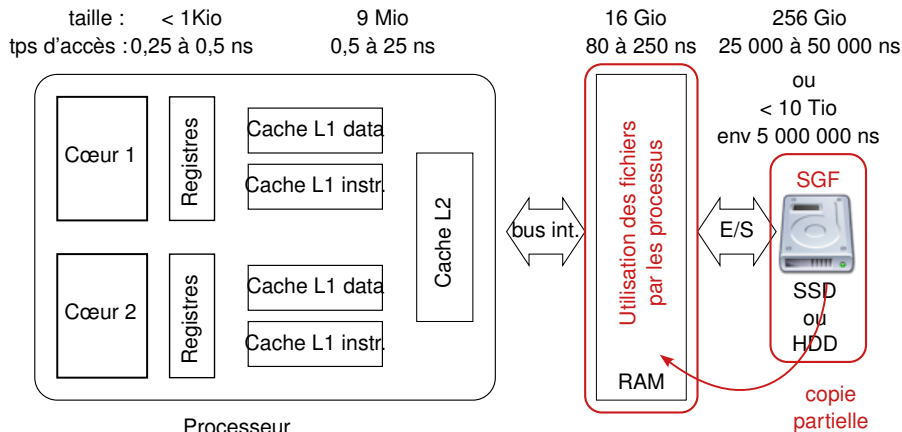
# Composants - Mémoires (Rappel)

## Exemple de hiérarchie



# Composants - Mémoires (Rappel)

## Exemple de hiérarchie



# Accès aux fichiers par les processus

## 1. Morceaux de fichiers *mappés* dans l'espace mémoire

- certains *Virtual Memory Area* issus du fichier exécutable (code etc.) ou de bibliothèques *mappées* au chargement
- pour en créer d'autres : `mmap()` (ex : pilotes).

## 2. Point d'accès vers un fichier

- le fichier n'est pas dans l'espace mémoire du processus ;
- chaque lecture/écriture se fait entre le fichier et une adresse ;
- pour un processus donné, chaque accès est identifié par un numéro : *File Descriptor*

## Bibliothèque POSIX

```
fd=open("nom_du_fichier",MODE_ACCES);  
taille=read/write(fd, adresse, nb_octets);  
res=close(fd);
```

# Accès aux fichiers par les processus (point d'accès)

## 3 premiers descripteurs de fichiers définis systématiquement

- 0 : entrée standard (terminal)
- 1 : sortie standard (terminal)
- 2 : erreur standard (terminal)

## Propriétés

- un fichier peut être ouvert plusieurs fois par un même processus ;
- le descripteur de processus (*Process Control Block*) contient une table des fichiers ouverts : ( fd + pointeur vers accès ) par accès
- un même accès peut être partagé par plusieurs processus (ex : père et fils) ;

## Accès aux fichiers par les processus (point d'accès)

table des descripteurs de fichiers  
du processus P1

| fd | nom_fichier | pointeur<br>vers accès |
|----|-------------|------------------------|
| 0  | entrée std  |                        |
| 1  | sortie std  |                        |
| 2  | erreur std  |                        |
| 3  | fichier.txt | _____                  |
|    |             |                        |

table des descripteurs de fichiers  
du processus P2

| fd | nom_fichier | pointeur<br>vers accès |
|----|-------------|------------------------|
| 0  | entrée std  |                        |
| 1  | sortie std  |                        |
| 2  | erreur std  |                        |
| 3  | fichier.txt | _____                  |
|    |             |                        |

| table des positions |     |
|---------------------|-----|
| 1                   | 2   |
| 3                   | 4   |
| 5                   | 6   |
| 7                   | 8   |
| 9                   | 10  |
| 11                  | 12  |
| 13                  | 14  |
| 15                  | 16  |
| 17                  | 18  |
| 19                  | 20  |
| 21                  | 22  |
| 23                  | 24  |
| 25                  | 26  |
| 27                  | 28  |
| 29                  | 30  |
| 31                  | 32  |
| 33                  | 34  |
| 35                  | 36  |
| 37                  | 38  |
| 39                  | 40  |
| 41                  | 42  |
| 43                  | 44  |
| 45                  | 46  |
| 47                  | 48  |
| 49                  | 50  |
| 51                  | 52  |
| 53                  | 54  |
| 55                  | 56  |
| 57                  | 58  |
| 59                  | 60  |
| 61                  | 62  |
| 63                  | 64  |
| 65                  | 66  |
| 67                  | 68  |
| 69                  | 70  |
| 71                  | 72  |
| 73                  | 74  |
| 75                  | 76  |
| 77                  | 78  |
| 79                  | 80  |
| 81                  | 82  |
| 83                  | 84  |
| 85                  | 86  |
| 87                  | 88  |
| 89                  | 90  |
| 91                  | 92  |
| 93                  | 94  |
| 95                  | 96  |
| 97                  | 98  |
| 99                  | 100 |

| pos.<br>courante | mode<br>accès | nb. proc. | point. vs.<br>i-nœuds |
|------------------|---------------|-----------|-----------------------|
| 6                | R             | 1         | —                     |
| 3                | R             | 1         | —                     |
|                  |               |           |                       |

| table des i-noeuds |     |
|--------------------|-----|
| 1                  | 2   |
| 3                  | 4   |
| 5                  | 6   |
| 7                  | 8   |
| 9                  | 10  |
| 11                 | 12  |
| 13                 | 14  |
| 15                 | 16  |
| 17                 | 18  |
| 19                 | 20  |
| 21                 | 22  |
| 23                 | 24  |
| 25                 | 26  |
| 27                 | 28  |
| 29                 | 30  |
| 31                 | 32  |
| 33                 | 34  |
| 35                 | 36  |
| 37                 | 38  |
| 39                 | 40  |
| 41                 | 42  |
| 43                 | 44  |
| 45                 | 46  |
| 47                 | 48  |
| 49                 | 50  |
| 51                 | 52  |
| 53                 | 54  |
| 55                 | 56  |
| 57                 | 58  |
| 59                 | 60  |
| 61                 | 62  |
| 63                 | 64  |
| 65                 | 66  |
| 67                 | 68  |
| 69                 | 70  |
| 71                 | 72  |
| 73                 | 74  |
| 75                 | 76  |
| 77                 | 78  |
| 79                 | 80  |
| 81                 | 82  |
| 83                 | 84  |
| 85                 | 86  |
| 87                 | 88  |
| 89                 | 90  |
| 91                 | 92  |
| 93                 | 94  |
| 95                 | 96  |
| 97                 | 98  |
| 99                 | 100 |

|  | copie<br>i-nœud | nb.<br>ouv. | num. ds<br>i-liste |
|--|-----------------|-------------|--------------------|
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
| → copie<br>i-nœud<br>de<br>fichier.txt | 2               | 6299        |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |
|  |                 |             |                    |



## Accès aux fichiers par les processus (point d'accès)

table des descripteurs de fichiers  
du processus P1

| fd | nom_fichier | pointeur<br>vers accès |
|----|-------------|------------------------|
| 0  | entrée std  |                        |
| 1  | sortie std  |                        |
| 2  | erreur std  |                        |
| 3  | fichier.txt |                        |
|    |             |                        |

table des descripteurs de fichiers  
du processus P2

| fd | nom_fichier | pointeur<br>vers accès |
|----|-------------|------------------------|
| 0  | entrée std  |                        |
| 1  | sortie std  |                        |
| 2  | erreur std  |                        |
| 3  | fichier.txt | _____                  |
|    |             |                        |

table des positions

| pos.<br>courante | mode<br>accès | nb. proc. | point. vs.<br>i-nœuds |
|------------------|---------------|-----------|-----------------------|
| 6                | R             | 2         | —                     |
|                  |               |           |                       |
|                  |               |           |                       |
|                  |               |           |                       |

table des i-noeuds

| copie<br>i-nœud                        | nb.<br>ouv. | num. ds<br>i-liste |
|--|-------------|--------------------|
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
| → copie<br>i-nœud<br>de<br>fichier.txt | 1           | 6299               |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |
|  |             |                    |

# Utiliser la RAM comme *cache* du disque dur

## Principe

- bien que n'étant pas *mappés* dans l'espace mémoire du processus, les données des fichiers accédés transitent en RAM.
- les y laisser dans l'hypothèse d'une future utilisation.

## Éléments copiés

- derniers datablocks et i-nœuds de fichiers / répertoires accédés.
- table contenant infos sur partition (*superblock*).

## 4. Système de Gestion de Fichiers

### 4.1. Révisions [R1.04]

- Mémoire secondaire
- Arborescence de fichiers et répertoires
- Attributs d'un fichier/répertoire (métadonnées)

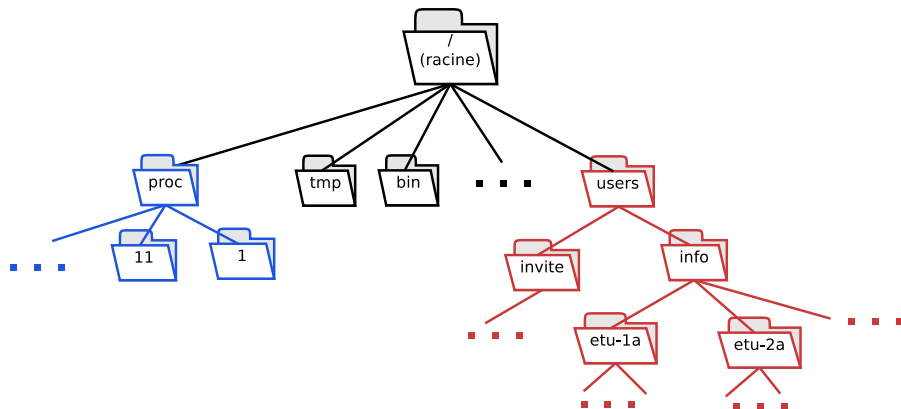
### 4.2. SGF ext4

- Structures de données : i-nœud et datablocks
- Protection

### 4.3. Accès aux fichiers par les processus

### 4.4. Une arborescence mais des supports et SGF variés

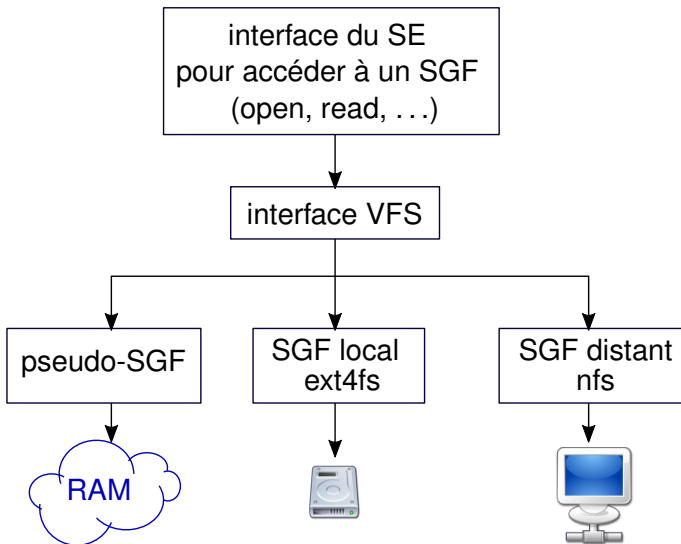
# Une arborescence mais des SGF variés : montage



ext4fs



# Une arborescence mais des SGF variés : VFS



## ...et des supports variés : SGF réseau

### Principe

- partage transparent entre SGF situés sur machines distantes
- relations client-serveur
- fichiers distants (sur serveur) apparaissent chez le client comme un sous-système de fichiers monté dans un répertoire

### communications

- interface commune (*Remote Procedure Calls*)
- protocole réseau (UDP/IP ou TCP/IP)

### exemples

- NFS (Solaris, IETF)
- SMB - CIFS (Windows)
- AFP (Apple)

# Organisation du cours

1. Système d'exploitation
2. Processus
3. Partage des ressources
4. Système de Gestion de Fichiers
5. Entrées/Sorties

# Introduction

- tâches d'un ordinateur :
  - exécuter des programmes
  - réaliser des entrées/sorties (*IO*)
- périphériques (*device*) de natures très différentes :
  - périph. de stockage
  - périph. de transmission
  - périph. d'interface homme-machine
- pilotes (*driver*) : entre chaque périph. et un système d'exploitation



# Introduction

- tâches d'un ordinateur :
  - exécuter des programmes
  - réaliser des entrées/sorties (IO)
- périphériques (*device*) de natures très différentes :
  - périph. de stockage
  - périph. de transmission
  - périph. d'interface homme-machine
- pilotes (*driver*) : entre chaque périph. et un système d'exploitation

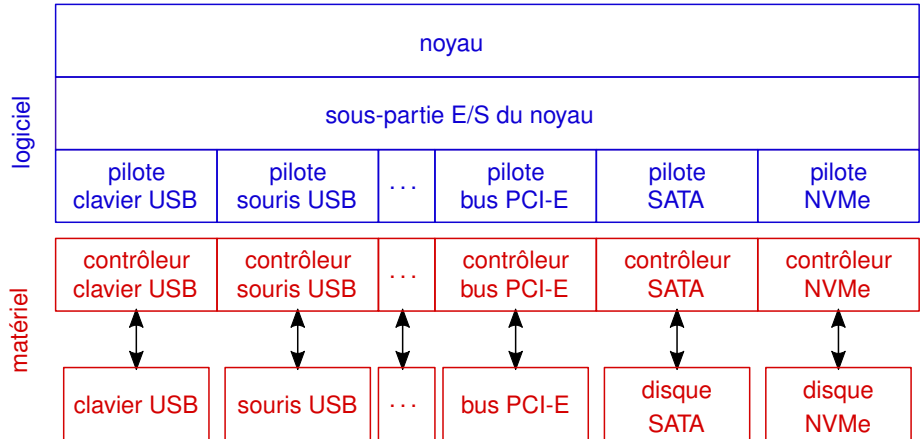
# Introduction

- tâches d'un ordinateur :
  - exécuter des programmes
  - réaliser des entrées/sorties (IO)
- périphériques (*device*) de natures très différentes :
  - périph. de stockage
  - périph. de transmission
  - périph. d'interface homme-machine
- pilotes (*driver*) : entre chaque périph. et un système d'exploitation

# Introduction

- tâches d'un ordinateur :
  - exécuter des programmes
  - réaliser des entrées/sorties (IO)
- périphériques (*device*) de natures très différentes :
  - périph. de stockage
  - périph. de transmission
  - périph. d'interface homme-machine
- pilotes (*driver*) : entre chaque périph. et un système d'exploitation

# Introduction



## 5. Entrées/Sorties

### 5.1. Entrées/Sorties : aspects matériels

- Bus : véhicule de l'information
- Contrôleur : interface matérielle
- Interruption : communication asynchrone
- DMA : Direct Memory Access

### 5.2. Entrées/Sorties : aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Bus

## Définition

- véhicule l'information entre composants
- suivant un protocole

## Trois niveaux

- **bus système** :
  - entre CPU et mémoire centrale
  - TT rapides
- **bus interne** :
  - entre CPU+RAM et composants internes à l'ordi.
  - de rapide à TT rapides
- **bus d'extension** :
  - entre l'ordi et les composants externes à l'ordi.
  - plus lent

# Bus

## Définition

- véhicule l'information entre composants
- suivant un protocole

## Trois niveaux

- **bus système** :
  - entre CPU et mémoire centrale
  - TT rapides
- **bus interne** :
  - entre CPU+RAM et composants internes à l'ordi.
  - de rapide à TT rapides
- **bus d'extension** :
  - entre l'ordi et les composants externes à l'ordi.
  - plus lent

# Bus et Architecture PC (*Chipset*)

## CPU

- contient contrôleurs pour gérer la communication avec
  - mémoire centrale via des bus *Double Data Rate* (contrôleur mémoire)
  - carte graphique via bus interne très rapide (PCIe)
- est connecté au *Chipset* (via *Direct Media Interface*)

## Chipset

- gère la communication avec les périphériques via divers bus internes (PCIe, USB, SATA...)

## Exemples de bus série

- **PCI-Express** : remplace PCI et AGP (*Accelerated Graphic Port*)
- **SATA** (*Serial Advanced Technology Attachment*) : disque dur
- **USB** (*Universal Serial Bus*) : port externe classique



# Bus et Architecture PC (*Chipset*)

## CPU

- contient contrôleurs pour gérer la communication avec
  - mémoire centrale via des bus *Double Data Rate* (contrôleur mémoire)
  - carte graphique via bus interne très rapide (PCIe)
- est connecté au *Chipset* (via *Direct Media Interface*)

## Chipset

- gère la communication avec les périphériques via divers bus internes (PCIe, USB, SATA...)

## Exemples de bus série

- **PCI-Express** : remplace PCI et AGP (*Accelerated Graphic Port*)
- **SATA** (*Serial Advanced Technology Attachment*) : disque dur
- **USB** (*Universal Serial Bus*) : port externe classique

# Bus et Architecture PC (*Chipset*)

## CPU

- contient contrôleurs pour gérer la communication avec
  - mémoire centrale via des bus *Double Data Rate* (contrôleur mémoire)
  - carte graphique via bus interne très rapide (PCIe)
- est connecté au *Chipset* (via *Direct Media Interface*)

## Chipset

- gère la communication avec les périphériques via divers bus internes (PCIe, USB, SATA...)

## Exemples de bus série

- **PCI-Express** : remplace PCI et AGP (*Accelerated Graphic Port*)
- **SATA** (*Serial Advanced Technology Attachment*) : disque dur
- **USB** (*Universal Serial Bus*) : port externe classique

# Bus et Architecture PC (*Chipset*)

## CPU

- contient contrôleurs pour gérer la communication avec
  - mémoire centrale via des bus *Double Data Rate* (contrôleur mémoire)
  - carte graphique via bus interne très rapide (PCIe)
- est connecté au *Chipset* (via *Direct Media Interface*)

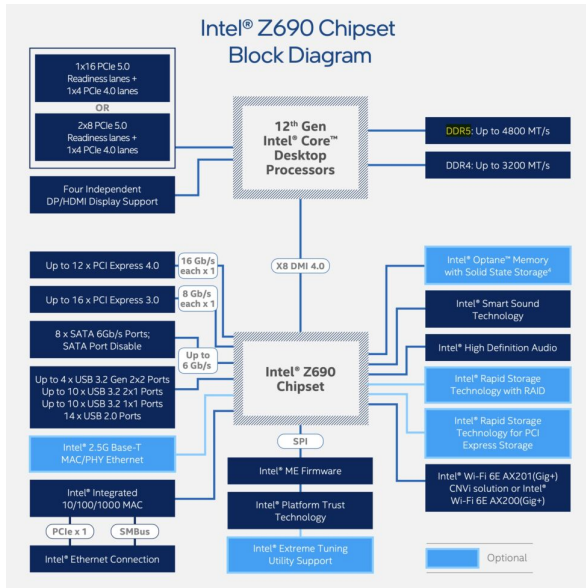
## Chipset

- gère la communication avec les périphériques via divers bus internes (PCIe, USB, SATA...)

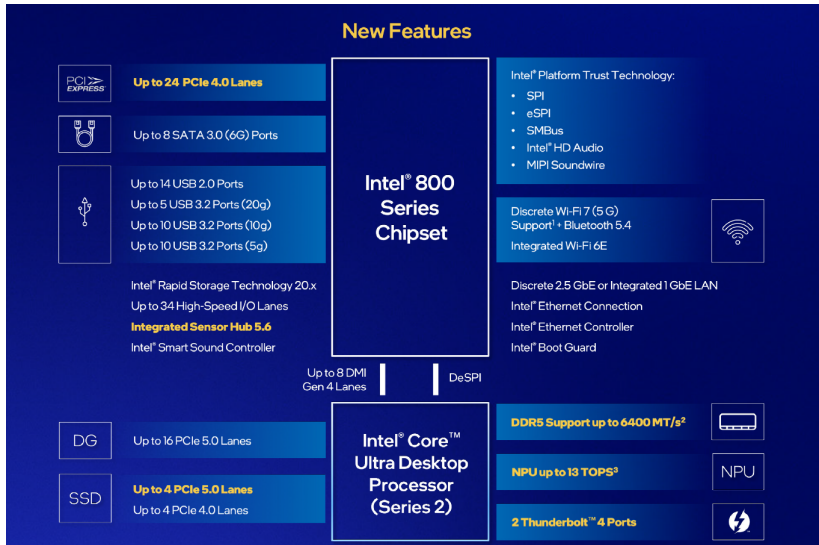
## Exemples de bus série

- **PCI-Express** : remplace PCI et AGP (*Accelerated Graphic Port*)
- **SATA** (*Serial Advanced Technology Attachment*) : disque dur
- **USB** (*Universal Serial Bus*) : port externe classique

# Chipset Intel Z690



# Chipset Intel Z890



# 5. Entrées/Sorties

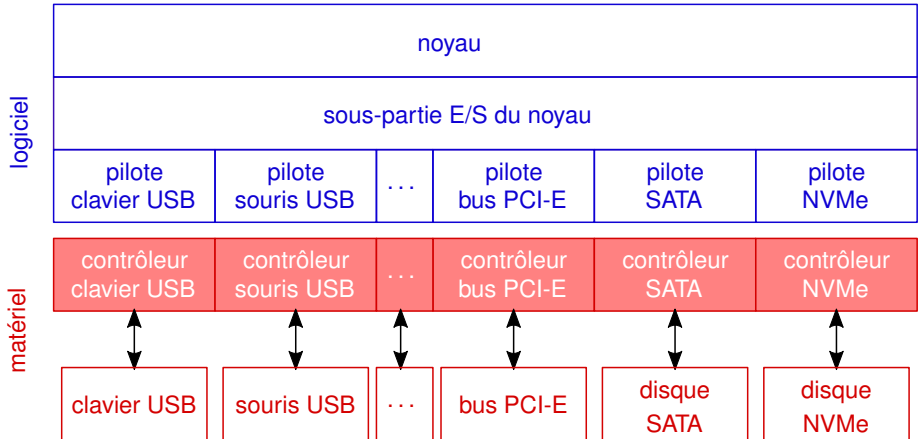
## 5.1. Entrées/Sorties : aspects matériels

- Bus : véhicule de l'information
- **Contrôleur : interface matérielle**
- Interruption : communication asynchrone
- DMA : Direct Memory Access

## 5.2. Entrées/Sorties : aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Contrôleur



# Contrôleur

## Définition

circuit électronique qui

- interprète/envoie les messages depuis/sur les bus
- agit sur le matériel qu'il interface

## Propriétés

- d'une petite puce jusqu'à une carte (processeurs + mémoire)
- sur la carte mère ou sur le périphérique même (disque dur)



# Contrôleur

## Définition

circuit électronique qui

- interprète/envoie les messages depuis/sur les bus
- agit sur le matériel qu'il interface

## Propriétés

- d'une petite puce jusqu'à une carte (processeurs + mémoire)
- sur la carte mère ou sur le périphérique même (disque dur)

## 5. Entrées/Sorties

### 5.1. Entrées/Sorties : aspects matériels

- Bus : véhicule de l'information
- Contrôleur : interface matérielle
- **Interruption : communication asynchrone**
- DMA : Direct Memory Access

### 5.2. Entrées/Sorties : aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Interruption

## Principe

changement de tension sur pattes d'interruption du processeur => interruption de l'exécution en cours :

- sauvegarde de l'état courant
- exécution de la routine de gestion (*handler*) adéquate
- reprise du processus interrompu

## 3 types d'interruption

- **erreur** matérielle (*segfault*) ou logicielle (division par zéro)
- **interruption matérielle** (*irq*)
- **interruption logicielle** (appel système)

table des vecteurs d'interruption

une interruption <-> un vecteur = adresse de *handler*

# Interruption

## Principe

changement de tension sur pattes d'interruption du processeur => interruption de l'exécution en cours :

- sauvegarde de l'état courant
- exécution de la routine de gestion (*handler*) adéquate
- reprise du processus interrompu

## 3 types d'interruption

- **erreur** matérielle (*segfault*) ou logicielle (division par zéro)
- **interruption matérielle** (*irq*)
- **interruption logicielle** (appel système)

## table des vecteurs d'interruption

une interruption <-> un vecteur = adresse de *handler*

# Interruption

## Vecteurs d'interruption pour Intel Pentium

| num. | description             |
|------|-------------------------|
| 0    | divide error            |
| 1    | debug exception         |
| 2    | null interrupt          |
| 3    | breakpoint              |
| 4    | INTO-detect overflow    |
| 5    | bound-range exception   |
| 6    | invalid opcode          |
| 7    | device not available    |
| 8    | double fault            |
| 9    | (reserved)              |
| 10   | invalid task state seg. |

| num.   | description          |
|--------|----------------------|
| 11     | segment not present  |
| 12     | stack fault          |
| 13     | general protection   |
| 14     | page fault           |
| 15     | (intel reserved)     |
| 16     | floating-point error |
| 17     | alignment check      |
| 18     | machine check        |
| 19-31  | (intel reserved)     |
| 32-255 | maskable interrupts  |

# Interruption

## Contrôleur d'interruption

- plusieurs interruptions  $\leftrightarrow$  une patte d'interruption
- registre d'état : identifie l'interruption
- registre de masque : permet d'ignorer ou non les interruptions

## Remarque

certaines interruptions primordiales sont non masquables

## 5. Entrées/Sorties

### 5.1. Entrées/Sorties : aspects matériels

- Bus : véhicule de l'information
- Contrôleur : interface matérielle
- Interruption : communication asynchrone
- **DMA : Direct Memory Access**

### 5.2. Entrées/Sorties : aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Direct Memory Access

## Contrôleur DMA

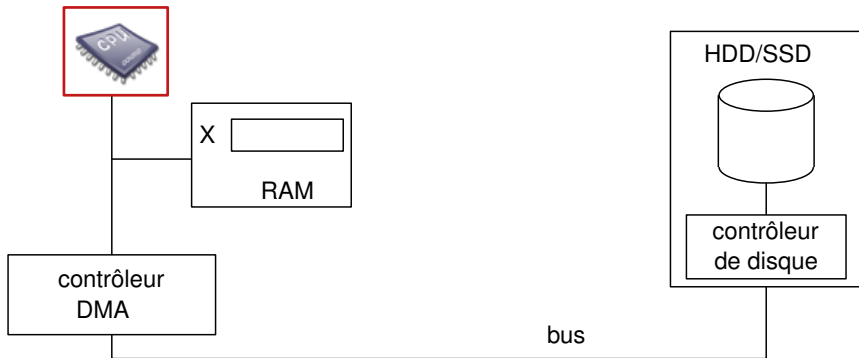
- fait partie du *Chipset*
- se charge du transfert de l'information entre périph. et mémoire sans intervention du processeur

## Communication CPU - DMA

- CPU écrit dans un bloc de sa mémoire
  - pointeur vers la source
  - pointeur vers la destination
  - nb d'octets à transférer
- CPU écrit l'adresse de ce bloc dans un registre du DMA
- DMA lit son registre, effectue le transfert, puis interrompt le CPU

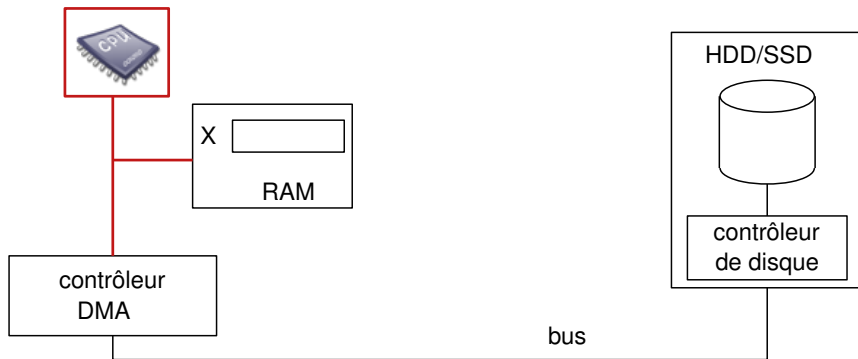


# Direct Memory Access



1. processus utilisateur  
demande au pilote de transférer  
des données à @X

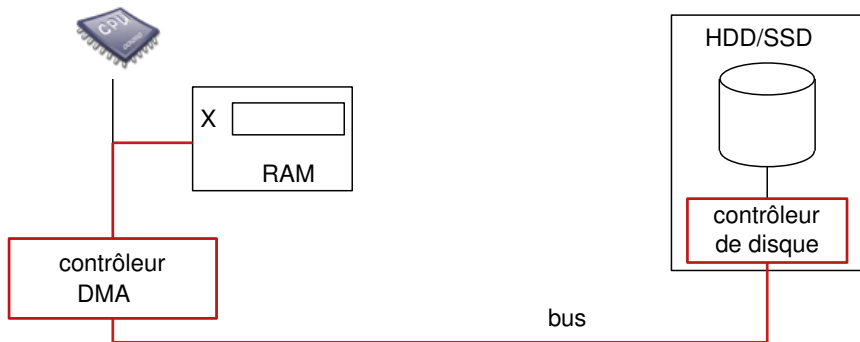
# Direct Memory Access



1. processus utilisateur  
demande au pilote de transférer  
des données à @X

2. le pilote range les instructions  
en mémoire et avertit DMA

# Direct Memory Access

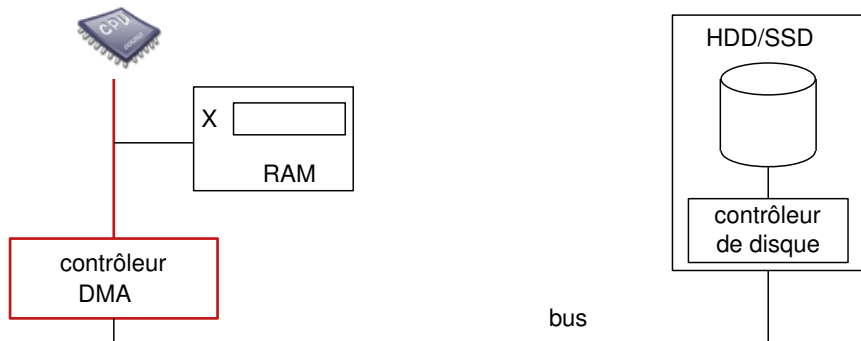


1. processus utilisateur  
demande au pilote de transférer  
des données à @X

2. le pilote range les instructions  
en mémoire et avertit DMA

3. DMA transfère les données  
du disque en mémoire  
octet après octet, sans le CPU

# Direct Memory Access



1. processus utilisateur demande au pilote de transférer des données à @X

2. le pilote range les instructions en mémoire et avertit DMA

3. DMA transfère les données du disque en mémoire octet après octet, sans le CPU

4. DMA interrompt le CPU pour lui signaler la fin du transfert

## 5. Entrées/Sorties

### 5.1. Entrées/Sorties : aspects matériels

- Bus : véhicule de l'information
- Contrôleur : interface matérielle
- Interruption : communication asynchrone
- DMA : Direct Memory Access

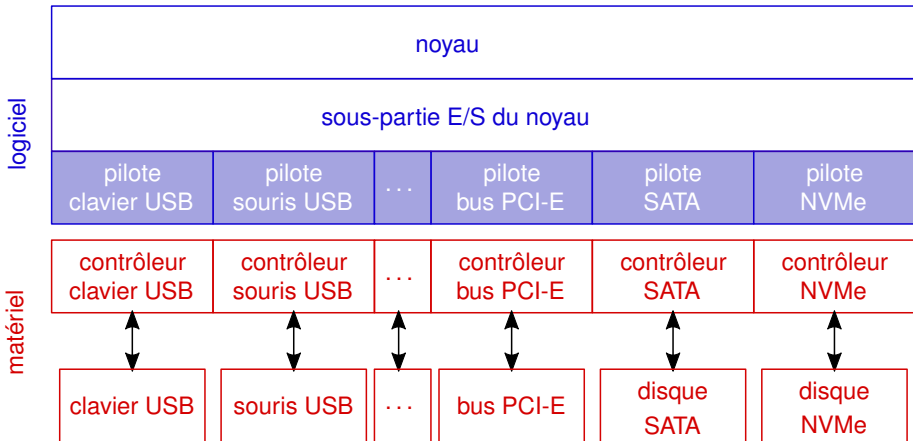
### 5.2. Entrées/Sorties : aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Pilote (*driver*)

## Définition

programme faisant l'interface entre un périphérique particulier et le SE



# Pilote (*driver*)

## Propriétés

- gestion homogène de différents périphériques via pilotes dédiés
- simplifie l'écriture des SE et la tâche des constructeurs matériel

## Propose des fonctions permettant des actions simples

- `open()`
- `close()` ou `release()`
- `read()`
- `write()`
- `ioctl()` pour configurer le périphérique (*Input Output ConTroL*)

# Pilote (*driver*)

## Propriétés

- gestion homogène de différents périphériques via pilotes dédiés
- simplifie l'écriture des SE et la tâche des constructeurs matériel

## Propose des fonctions permettant des actions simples

- `open()`
- `close()` ou `release()`
- `read()`
- `write()`
- `ioctl()` pour configurer le périphérique (*Input Output ConTroL*)



# Pilote adapté au type de périphérique

## Périphérique de blocs

- périphérique de stockage (disque dur)
- données manipulées par blocs de 512, 1024... octets
- données amenées à être utilisées plusieurs fois (caches)

## Périphérique de caractères

- flux d'info imprévisibles (souris, clavier)
- données manipulées par flux d'octets de taille variable
- données consommées une fois (tampon)

## Périphérique réseau

- contrôleur réseau (carte ethernet)
- **n'est pas vu** comme un fichier spécial dans `/dev/`
- est identifié par une interface nommée `eth0`, `eth1...`

# Pilote adapté au type de périphérique

## Périphérique de blocs

- périphérique de stockage (disque dur)
- données manipulées par blocs de 512, 1024... octets
- données amenées à être utilisées plusieurs fois (caches)

## Périphérique de caractères

- flux d'info imprévisibles (souris, clavier)
- données manipulées par flux d'octets de taille variable
- données consommées une fois (tampon)

## Périphérique réseau

- contrôleur réseau (carte ethernet)
- **n'est pas vu** comme un fichier spécial dans `/dev/`
- est identifié par une interface nommée `eth0`, `eth1...`

# Pilote adapté au type de périphérique

## Périphérique de blocs

- périphérique de stockage (disque dur)
- données manipulées par blocs de 512, 1024... octets
- données amenées à être utilisées plusieurs fois (caches)

## Périphérique de caractères

- flux d'info imprévisibles (souris, clavier)
- données manipulées par flux d'octets de taille variable
- données consommées une fois (tampon)

## Périphérique réseau

- contrôleur réseau (carte ethernet)
- **n'est pas vu** comme un fichier spécial dans `/dev/`
- est identifié par une interface nommée **eth0**, **eth1**...

# 5. Entrées/Sorties

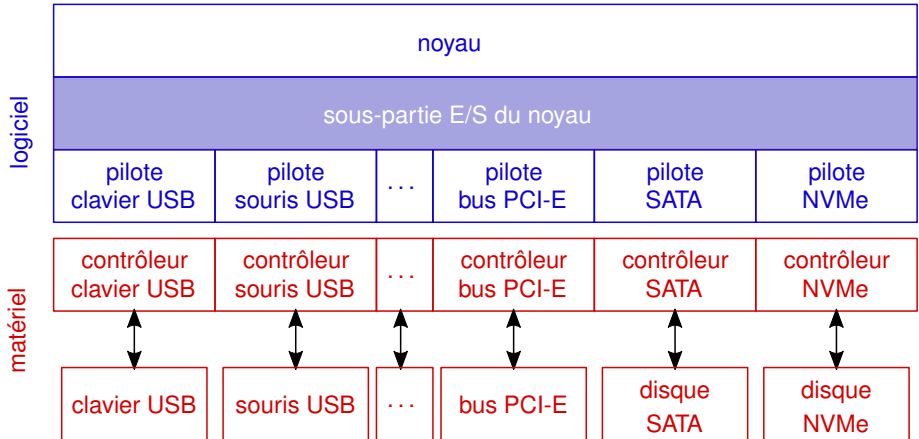
## 5.1. Aspects matériels

- Bus : véhicule de l'information
- Contrôleur : interface matérielle
- Interruption : communication asynchrone
- DMA : Direct Memory Access

## 5.2. Aspects logiciels

- Pilote
- Sous-partie E/S du noyau

# Sous-partie E/S du noyau



# Sous-partie E/S du noyau

- Ordonnancement des entrées/sorties
  - ordonner les requêtes par périphérique (HDD : C-SCAN)
- *Caching*
- Protection des entrées/sorties
- Gestion des erreurs

# Sous-partie E/S du noyau

- Ordonnancement des entrées/sorties
  - ordonner les requêtes par périphérique (HDD : C-SCAN)
- *Caching*
- Protection des entrées/sorties
- Gestion des erreurs

# Sous-partie E/S du noyau

- Ordonnancement des entrées/sorties
  - ordonner les requêtes par périphérique (HDD : C-SCAN)
- *Caching*
  - accélérer les E/S
- Protection des entrées/sorties
- Gestion des erreurs



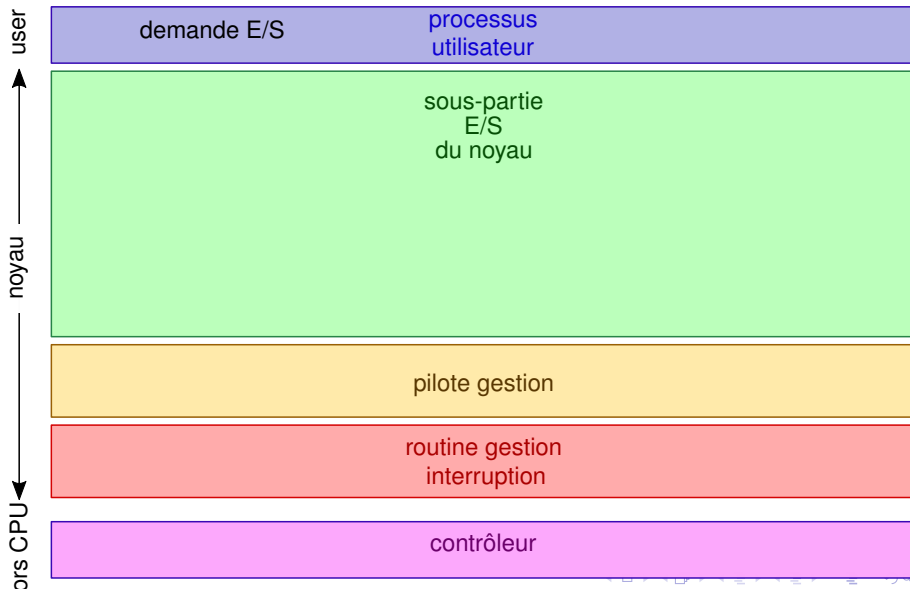
# Sous-partie E/S du noyau

- Ordonnancement des entrées/sorties
  - ordonner les requêtes par périphérique (HDD : C-SCAN)
- *Caching*
  - accélérer les E/S
- Protection des entrées/sorties
  - appels systèmes : interface pour toute E/S
  - isolation entre utilisateurs
- Gestion des erreurs

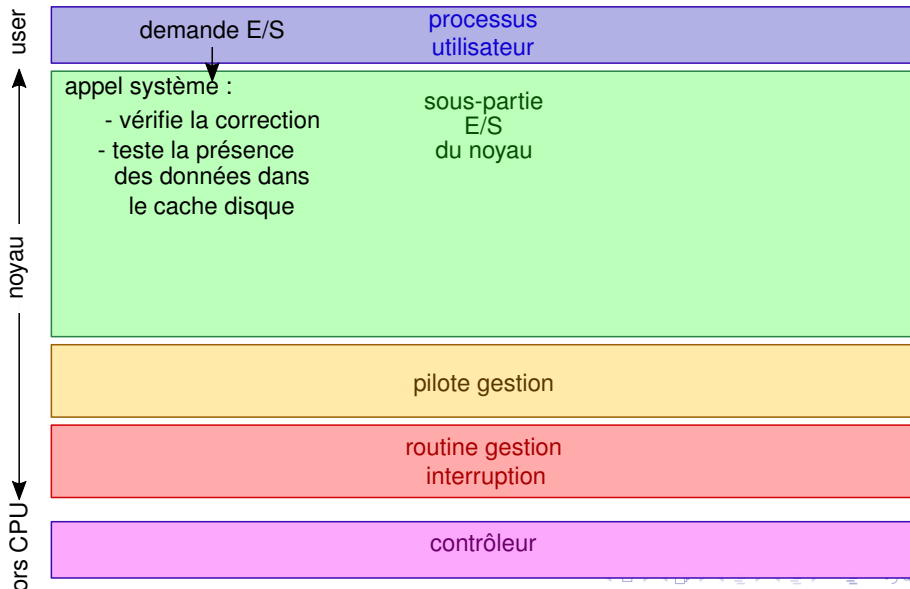
## Sous-partie E/S du noyau

- Ordonnancement des entrées/sorties
  - ordonner les requêtes par périphérique (HDD : C-SCAN)
- *Caching*
  - accélérer les E/S
- Protection des entrées/sorties
  - appels systèmes : interface pour toute E/S
  - isolation entre utilisateurs
- Gestion des erreurs
  - diagnostiquer erreur permanente avant d'informer l'utilisateur (`errno` ou exception)

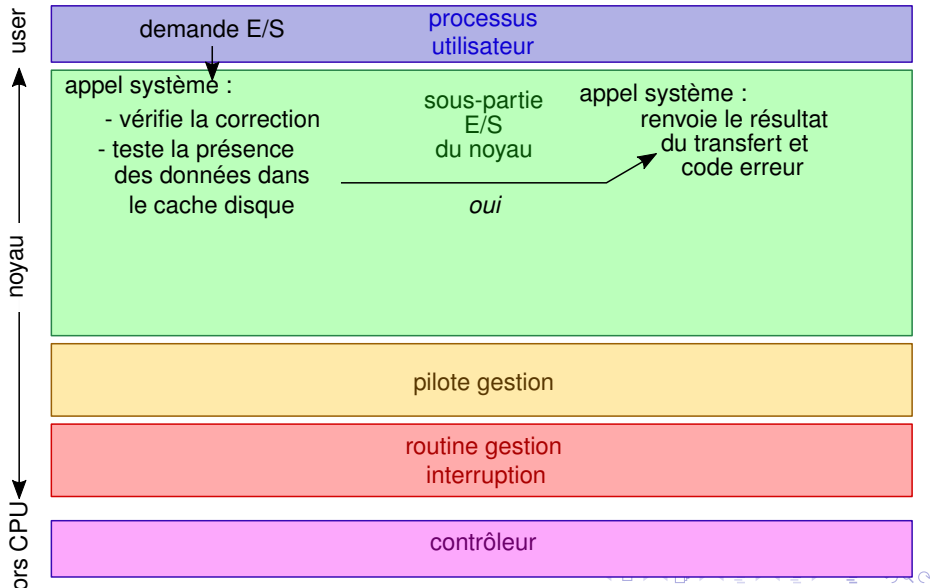
# Bilan par un exemple récapitulatif : un *read*



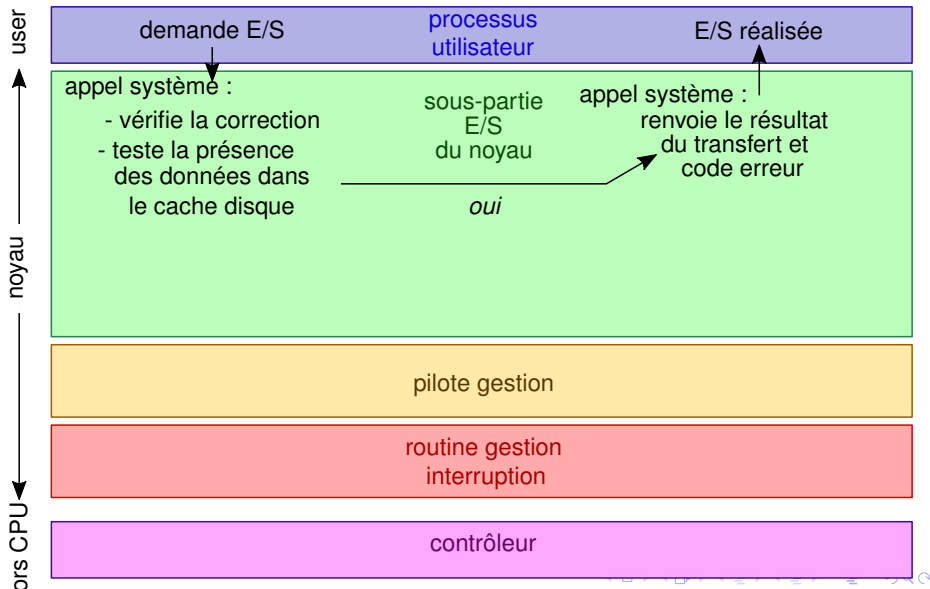
# Bilan par un exemple récapitulatif : un *read*



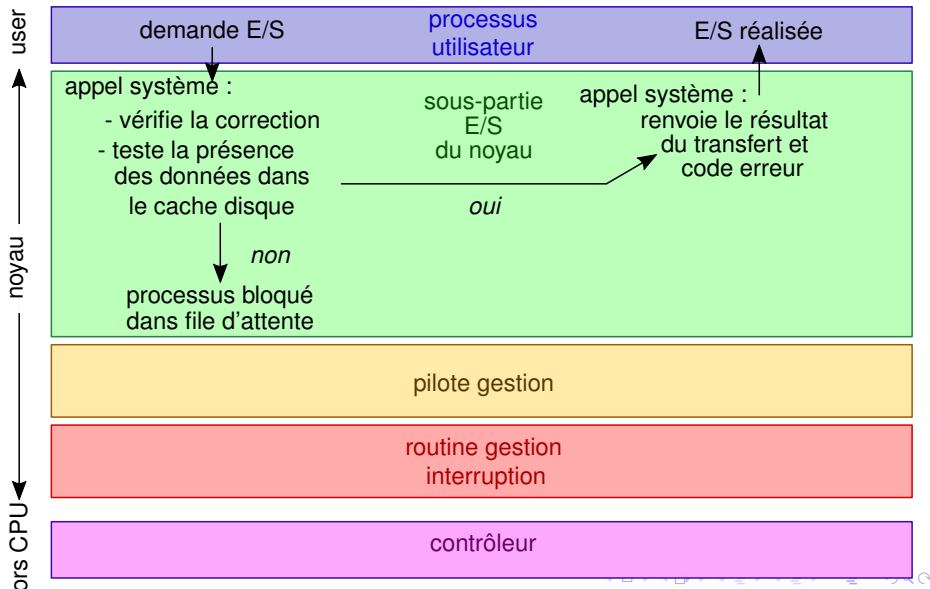
# Bilan par un exemple récapitulatif : un *read*



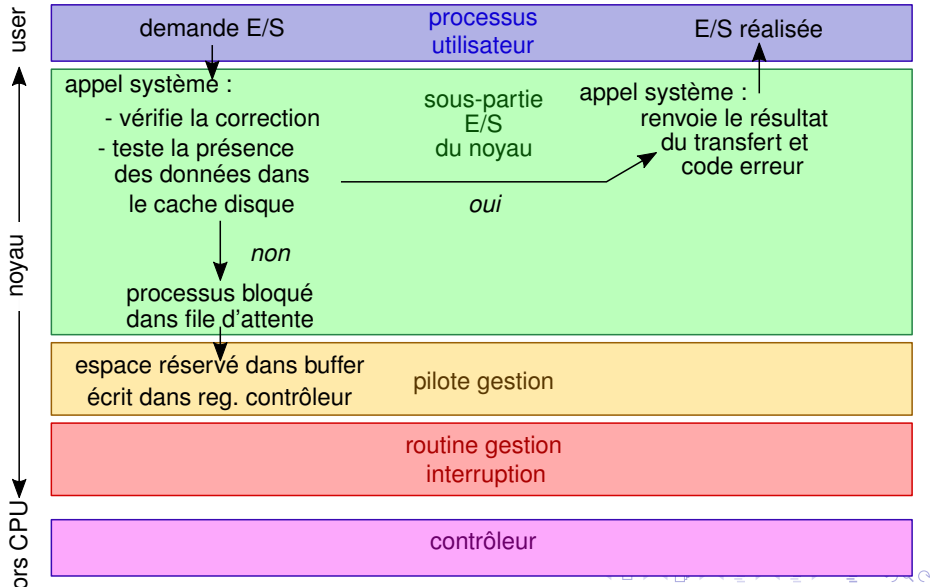
# Bilan par un exemple récapitulatif : un *read*



# Bilan par un exemple récapitulatif : un *read*

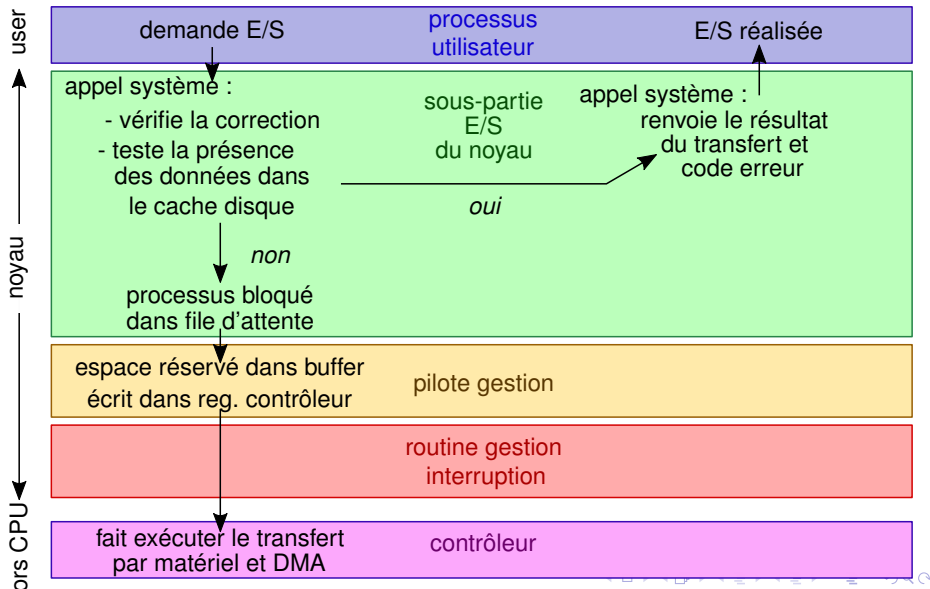


# Bilan par un exemple récapitulatif : un *read*

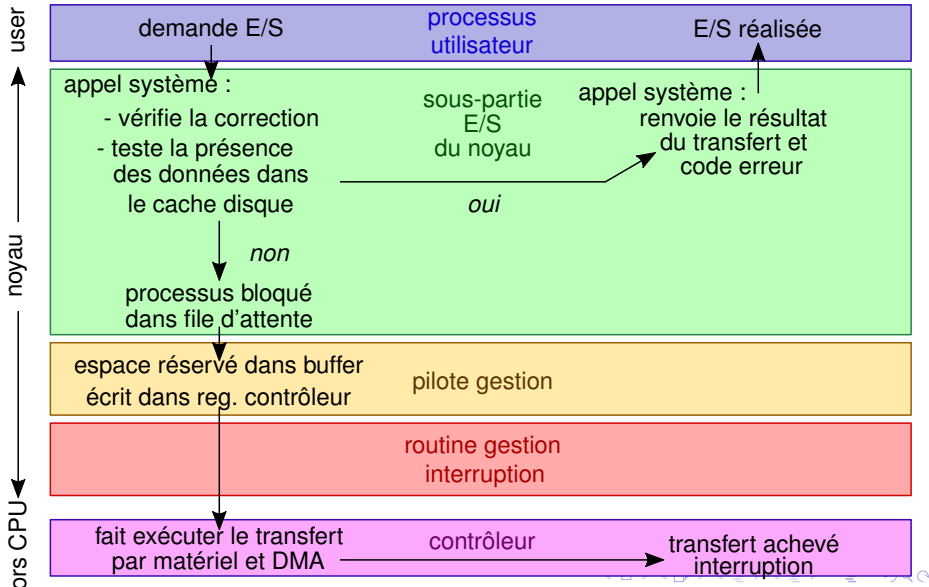




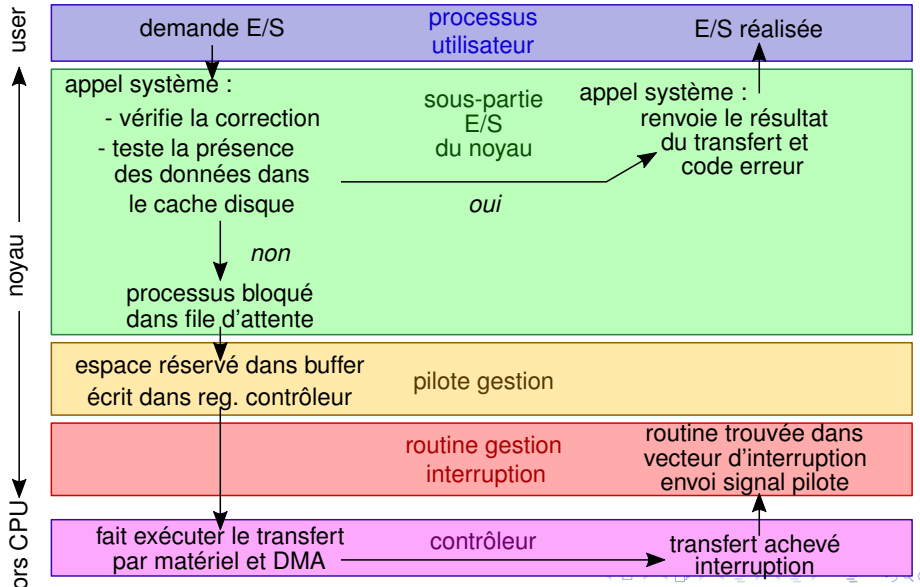
# Bilan par un exemple récapitulatif : un *read*



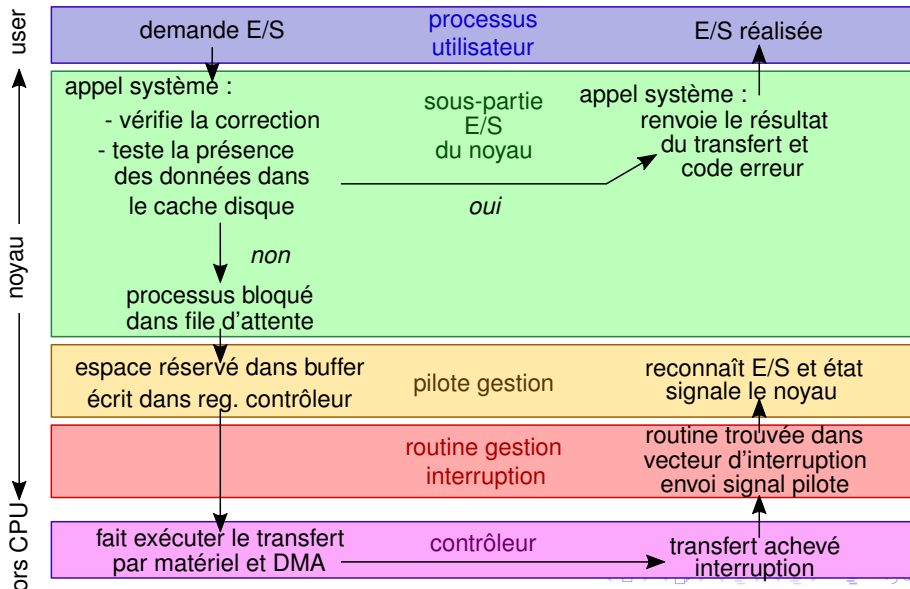
# Bilan par un exemple récapitulatif : un *read*



# Bilan par un exemple récapitulatif : un *read*



# Bilan par un exemple récapitulatif : un *read*



# Bilan par un exemple récapitulatif : un *read*

