



Systèmes d'exploitation

Système de fichiers



- Néerlandaise
- Université de Amsterdam
- Auteur de langage Python
- Dropbox

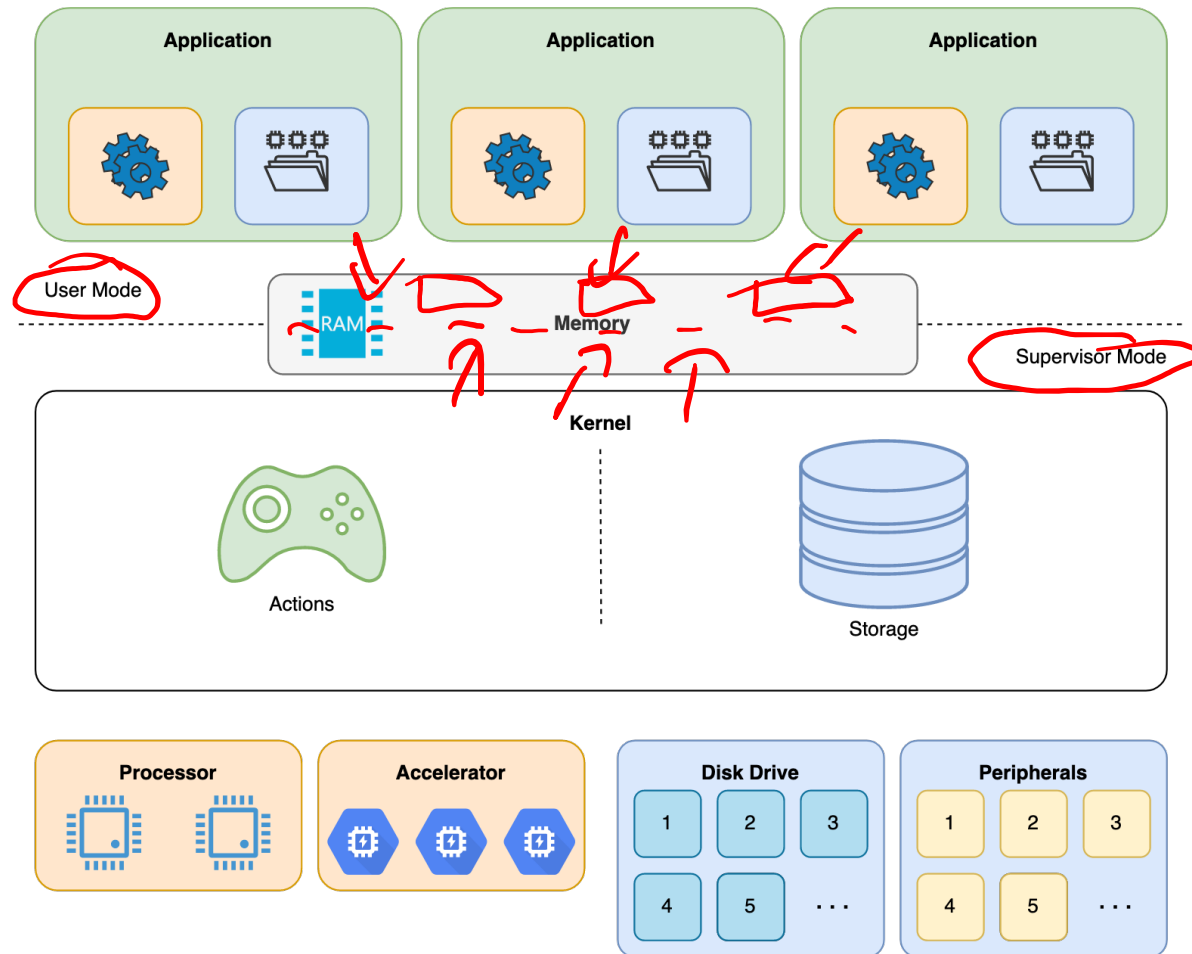
- Appel de système
- Partition
- Fichier
- Directeur
- Fichiers Spéciaux



Bibliographie pour aujourd'hui

- Modern Operating Systems
 - Chapitre 6
 - 6.1
 - 6.2
- Operating Systems Concepts
 - Chapitre 11

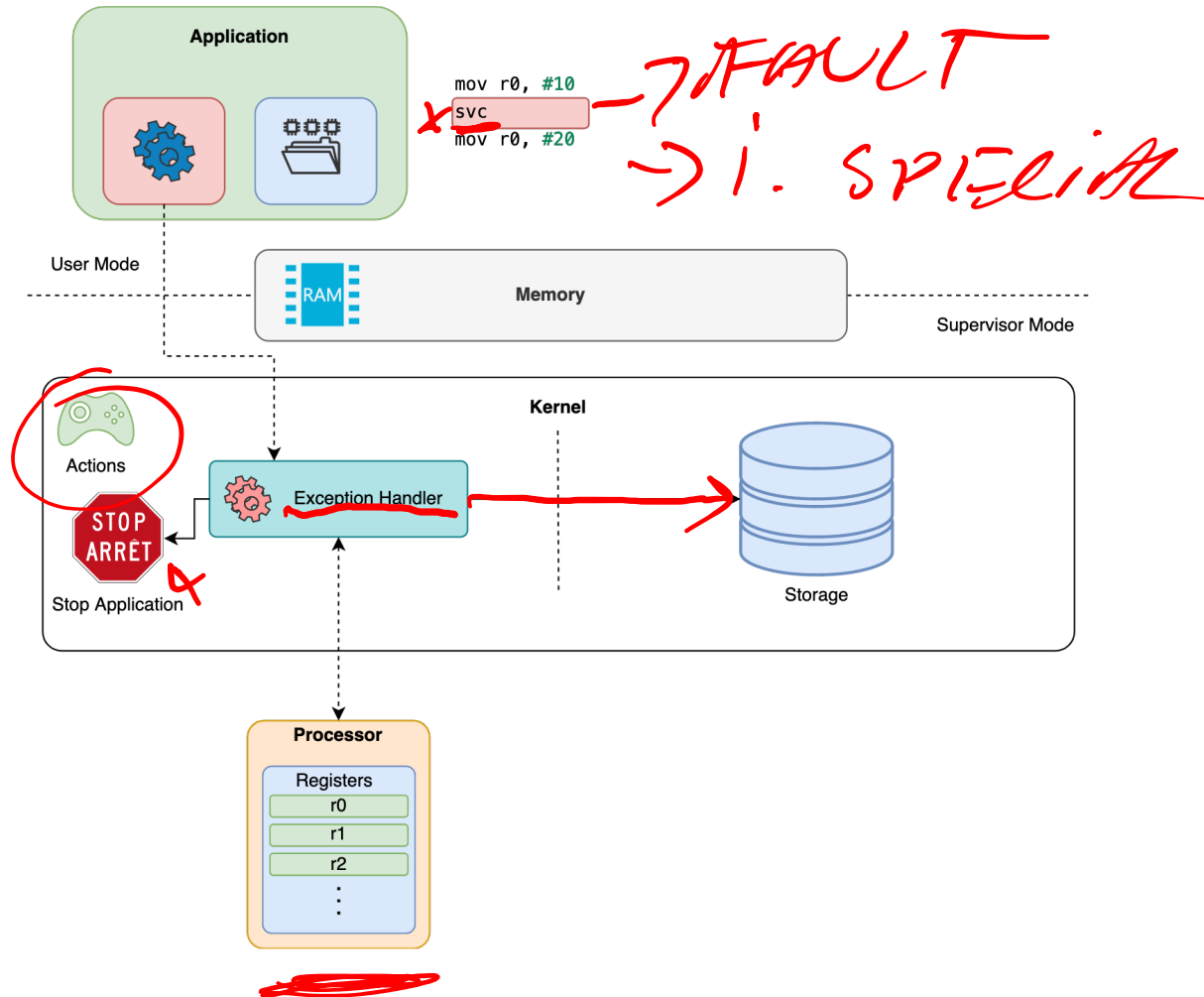
Idée General



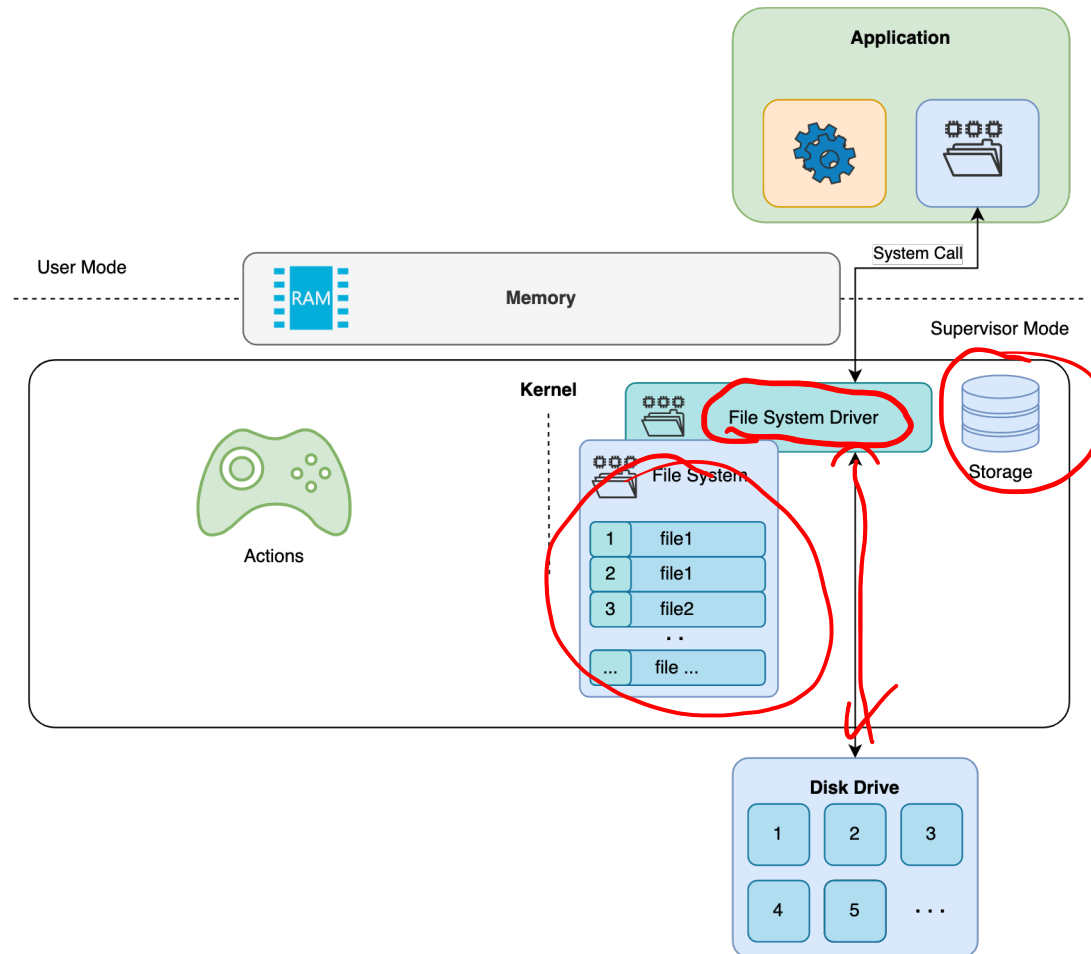
Modes d'exécution

- Le processeur a deux modes d'exécution
- **Mode superviseur**
 - Actions privilégiées
 - C'est le mode utilisé par le SE pour s'exécuter
- **Mode utilisateur**
 - L'accès direct au matériel n'est pas autorisé
 - L'espace adresse mémoire ne peut pas être modifié
 - C'est le mode utilisé pour les applications
- Le noyau est le intermédiaire pour:
 - l'accès des processus au matériel
 - l'accès des processus aux ressources
- **Appel système**
 - la transition du mode utilisateur au mode noyau

Appel du système



System de fichiers



System de fichiers

- transforme le tableau de blocs en fichiers et dossiers
- structures de données utilisées par le système d'exploitation



FICHER

Types de fichiers

- fichiers simple ✓ – regular files
- directeurs ✓ – directories
- lien (symbolique) ✓ – hard/soft links
- dispositifs de type caractère – char devices
- dispositifs de type bloc – block devices
- pipes/FIFOS
- sockets UNIX
- Comment trouvez-vous le type d'un fichier sous UNIX?

/dev/...

PERMISSION POUR UN FICHIER

- 3 types des entités
 - Utilisateur (user) U
 - Group (Group) G
 - Autres (Others) O (OTHERS) } A (ALL)
- 3 types des permissions
 - Lire (Read)
 - Ecrire (Write)
 - Executer (Execute)

- Permissions = chaîne de 9 caracteres

- 3 pour l'utilisateur
- 3 pour le group
- 3 pour les autres

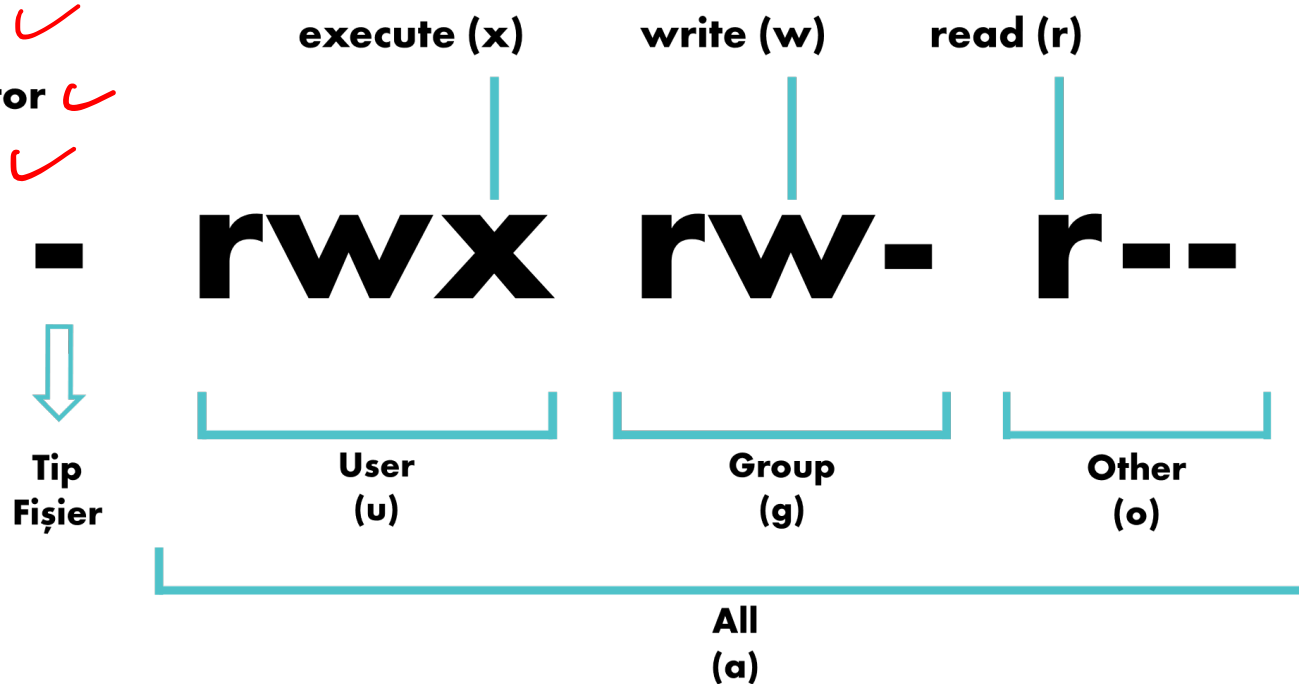


ls -l

/etc

Tip fişier:

■ ➡ fişier ✓
d ➡ director ✓
l ➡ link ✓



- Sur bits:

- **r w - r - - r - -** corresponde a **110100100**
- **r w x r - x - - -** corresponde a **111101000**
- **r w - rw - - - -** corresponde a **110110000**

- Octal:

- **r w - r - - r - -** corresponde a **644**
- **r w x r - x - - -** corresponde a **750**
- **r w - rw - - - -** corresponde a **660**

rw x
6 = 110
4 = 100
4 = 100

4 = 100 (r)
2 = 010 (w)
1 = 001 (x)

OPERATIONS AVEC LES FICHIERS

Operations avec les fichiers

- création
- ouverture ?
- lecture
- écriture
- positionnement dans le fichier ?
- tronquer ?
- fermeture ?
- suppression

Creation

- **shell**

touch /path/to/file

toute commande qui écrit dans un fichier le crée

- **Python**

file = open ("path/to/file", "rw")

- **POSIX** ~ *unix*

int fd = open ("path/to/file", O_CREAT | O_EXCL,
0644);

Ouverture

- **Python**

```
file = open ("/path/to/file", "rw")
```

- **POSIX**

```
int fd = open("/path/to/file",  
O_RDONLY);
```


Fichier Ouvert

- identification
 - Windows: poignée **HANDLE**
 - POSIX: descripteur
 - pourquoi pas un nom?
- position
 - Windows: pointeur de fichier
 - Linux: curseur de fichier
- droits d'ouverture (mode)
- compteur d'utilisation (file-open count)

Descripteur de fichier

- plus de descripteur peut correspondre aux même fichier
- chaque processus a une table de descripteurs de fichiers

- **Python**

- **File**

PYTHON – File
P. Renferme – int.
SE – int.

- **POSIX**

- un entier identifiant une instance de fichier ouvert dans un processus
 - descripteurs spéciaux – int
 - 0 (stdin), 1 (stdout), 2 (stderr)

Tableau de descripteur de fichier

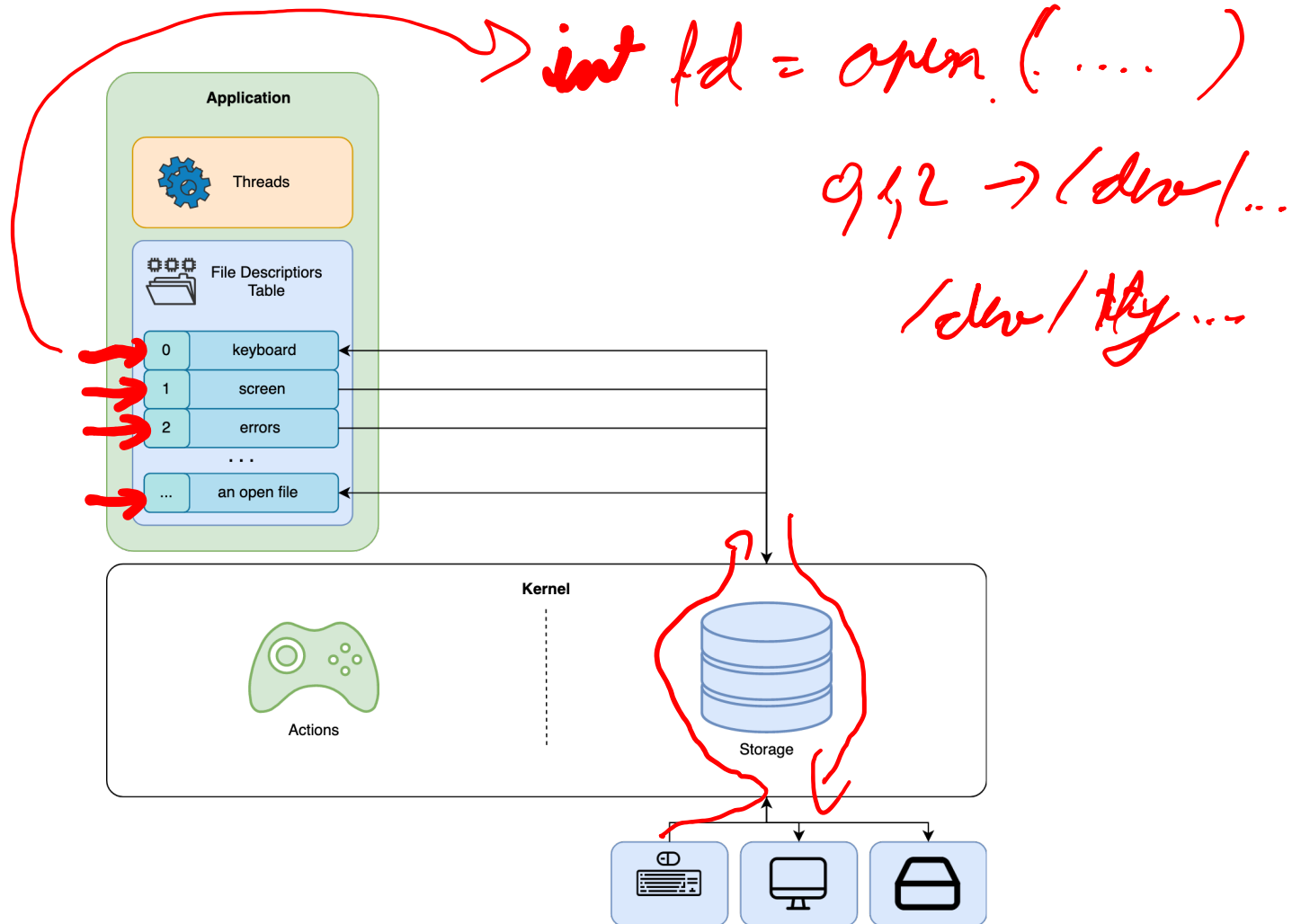
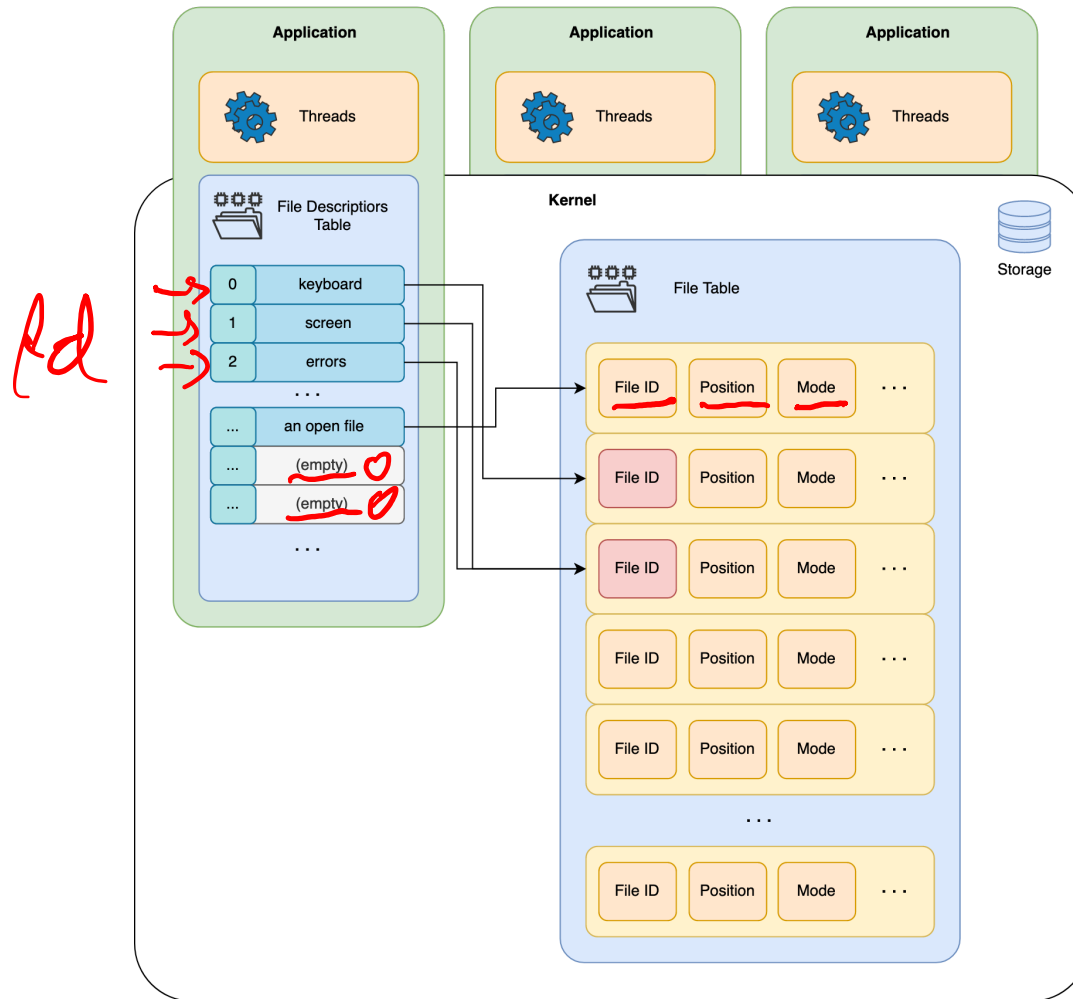
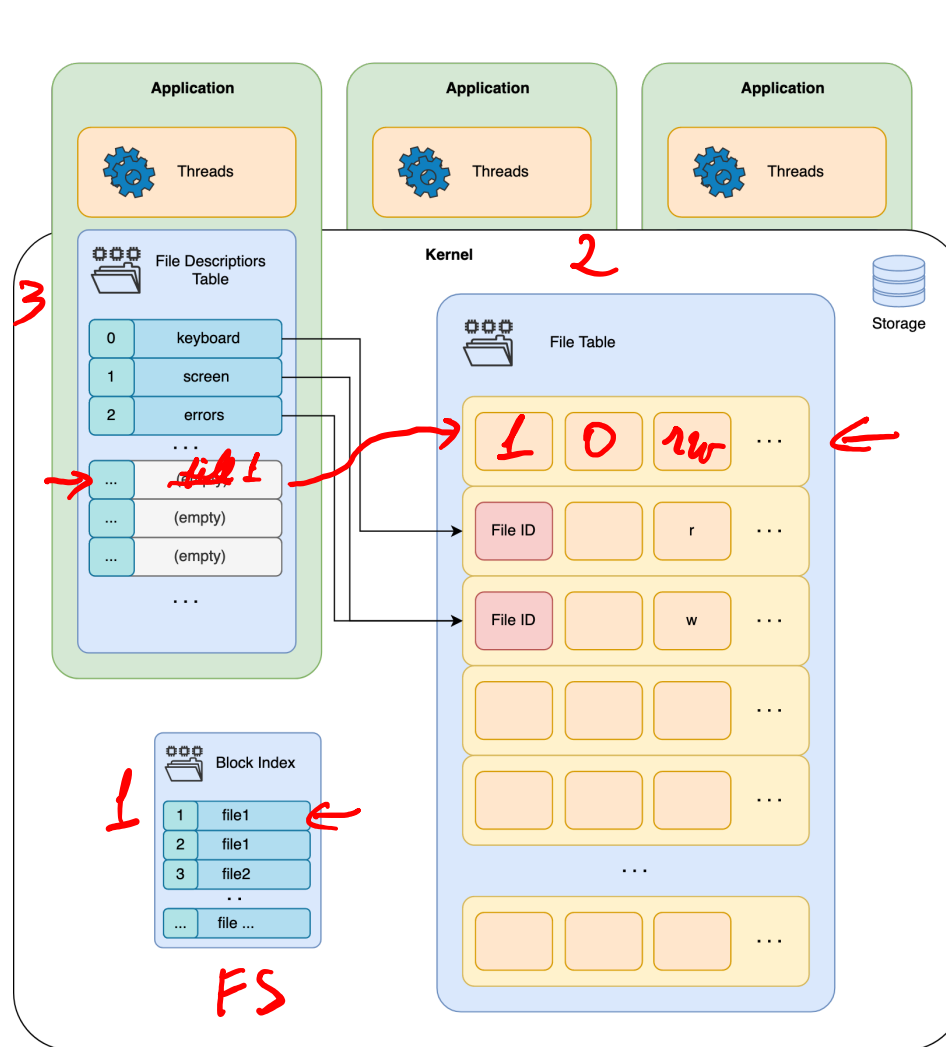


Tableau des fichiers



Overture



*open("file1",
"rw");*

Lecture

- stocker des informations dans un tampon
- avance du curseur de fichier

- **Python**

`data = file.read(n)`

pos?

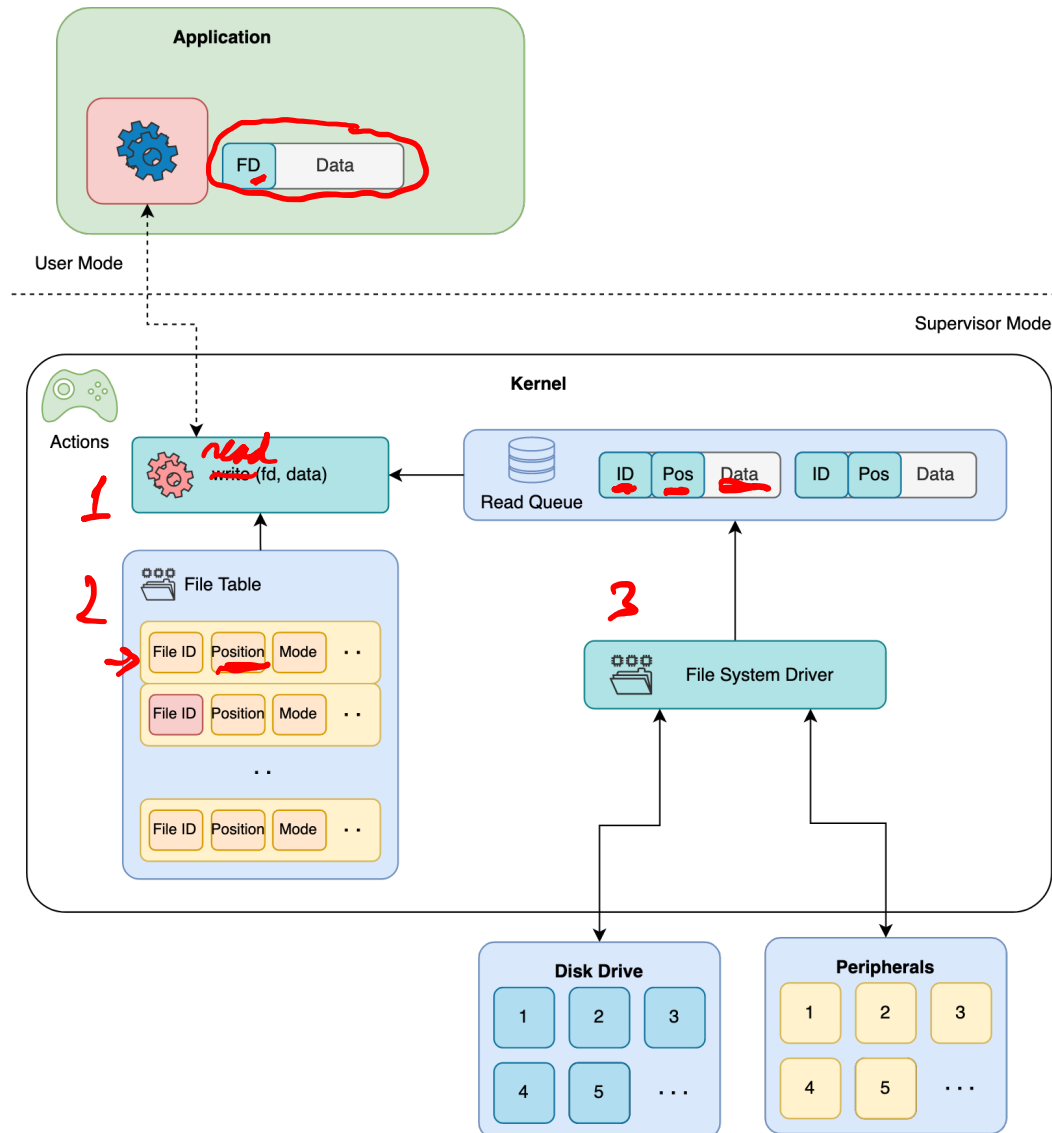
- **Unix**

`// reads maximum size bytes`

`// use only within a loop`

`n_read = read(fd, buffer, size);`

Flux de lecture



Écriture

- écrire des informations à partir d'un tampon
- avance du curseur de fichier

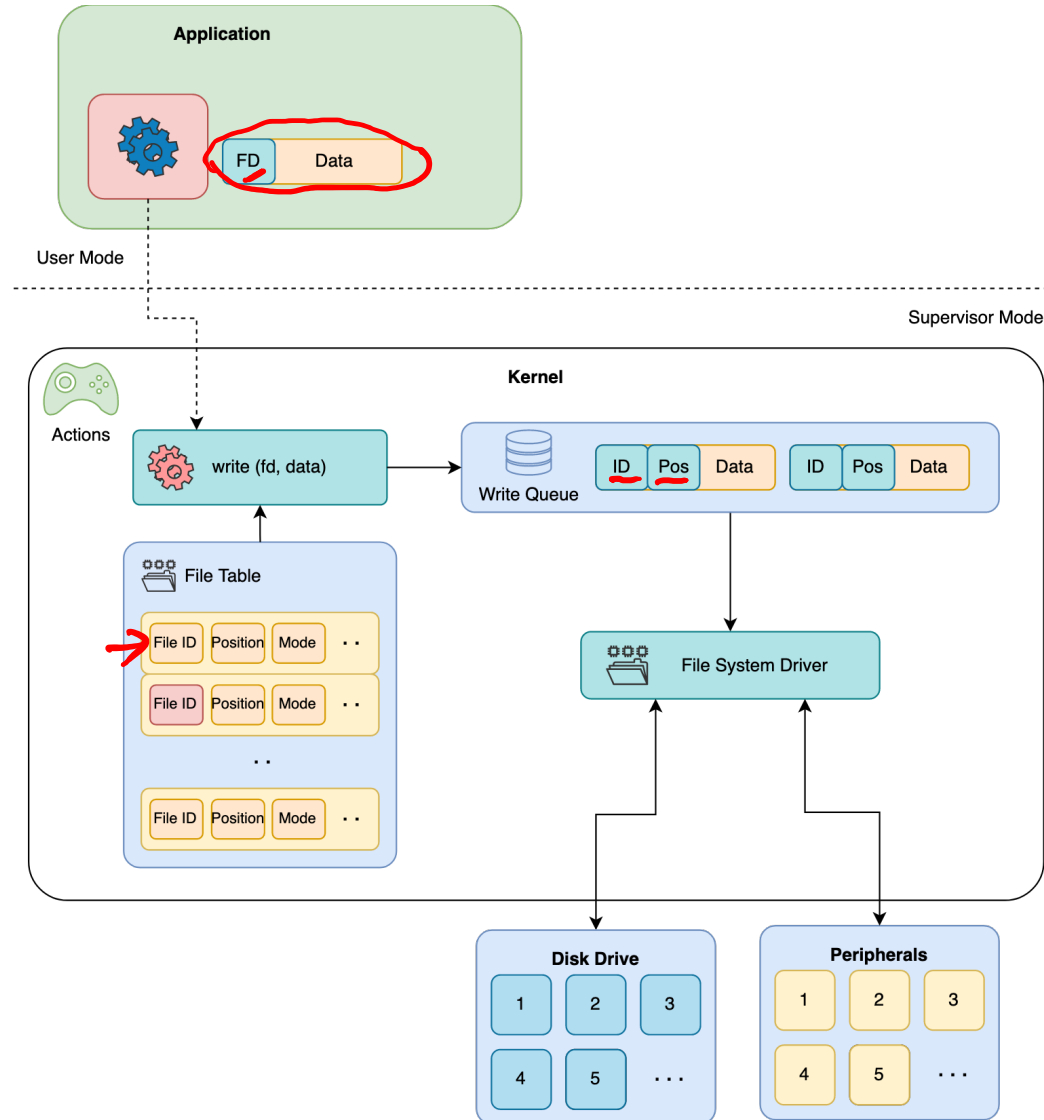
- **Python**

pos?
file.write (data)

- **Unix**

```
// writes maximum size bytes  
// use only within a loop  
n_write = write(fd, buffer, size);
```


Flux de écriture



Curseur de fichier

- changements à lire et à écrire
- initialisation à l'ouverture
- mouvement de curseur de fichier

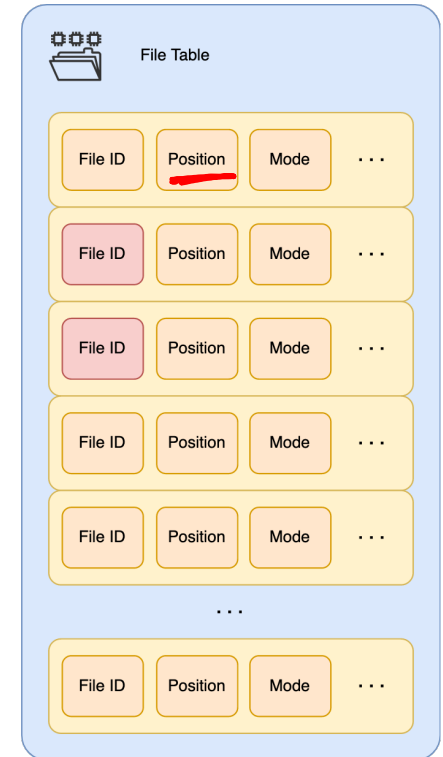
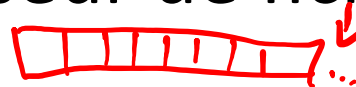
- **Python**

file.seek (offset, whence)

- **Unix**

lseek (fd, offset, SEEK_SET);

Fin Fis: (1, 2)



0 - DEBUT
1 - pos CURRENT 2 - FIN

SEEK - CUR
SEEK - END

Tronquer

- supprimer le contenu du fichier
- le curseur de fichier est mis à *size*

- **Python**

```
f.truncate(size)
```

- **POSIX**

```
truncate("/path/to/file", size);
```

Fermer

- l'entrée de la table de descripteur est supprimée

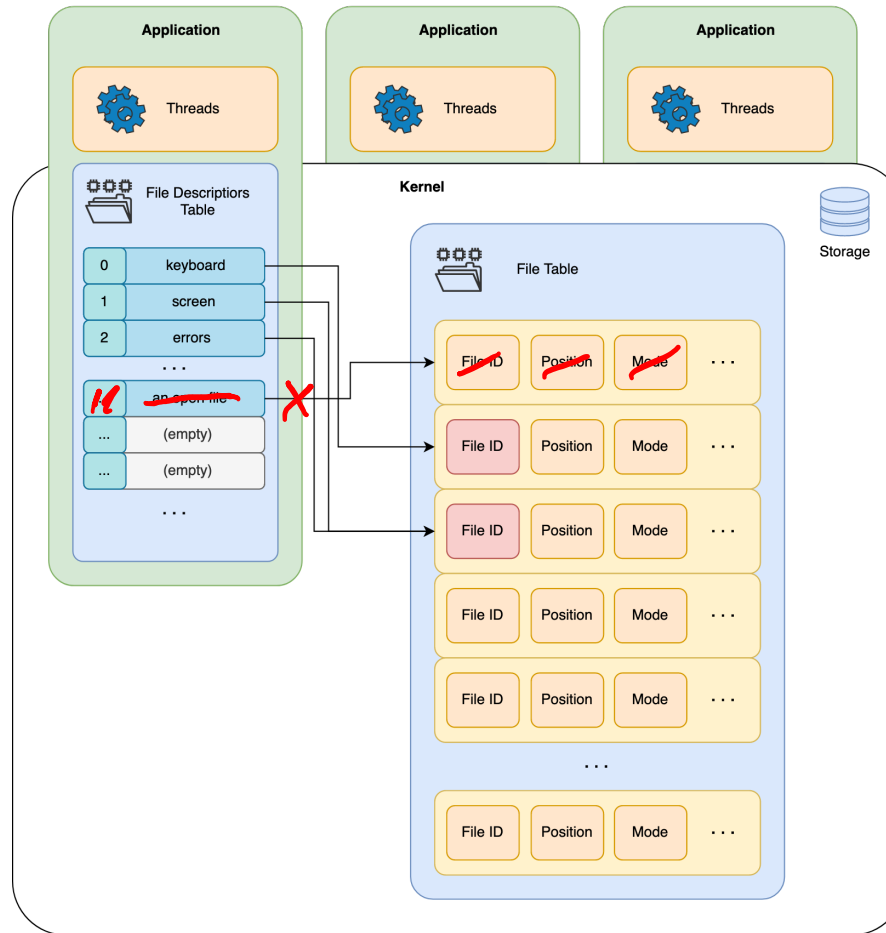
- **Python**

```
f.close (f);
```

- **POSIX**

```
close(fd);
```

Fermer



file.close();
↓
//
flush

Types des données

- **Utilisateur**
 - Nom de fichier
- **Application**
 - Descripteur de fichier
- **Système d'Exploitation**
 - table de descripteur
 - structure de données pour le fichier ouverte
 - structure de données pour le fichier sur de disc

- Partition
- Montage
- System de fichiers
- Fichier
- Directeur
- Lien
- Dispositif caractère
- Dispositif bloc
- Socket UNIX
- FIFO
- Permissions
- Chemin de fichier
- Absolue
- Relatif
- MBR
- GPT

Questions

