[SYS1] Systèmes d’exploitation (4)

Par Rhaeven, relecture par Find3r

Contenu d’un process

- pid
- uid / gid
- Address space
- fdtable
- signal : signaux à serveurs | handlers

Signaux

- SIGTERM : Demande au process de stop
- SIGKILL : Détruit un processes et tous ses enfants

Gestion des erreurs

```
rc = write(...);
perror();


int handler() {
    open(...);
}
```

> Valeur d’errno globale

Allocation de fd

```
struct file *fdtable[]; // fd est l'index dans fdtable
```

- On prend le plus petit fd libre

Redirections

Exemple : cat > file

Façon un peu nulle mais ça marche :

```
fork()
    close(1) //stdout
    open('file')
execvp
```

Meilleure facon :

- On utilise dup
- **dup(oldfd)** : Duplique un fd existant
- **dup2(oldfd, newfd)** : Close newfd (Avec O_CLOEXEC) et le remplace par oldfd
  - Équivalent à faire “close(newfd); open(oldfd);”

> **fcntl** permet de set des propriétés sur un fichier (dont O_CLOEXEC)

```
fork()
    dup(open("file"))
    close(1)
execvp
```

Redirection stdout

- cat > toto :

```
open("toto")
dup2(fd, 1)
```

Redirection stderr

- cat 2> toto :

```
open("toto")
dup2(fd, 2)
```

Pipe

- cat | cat2
  pipe(int fd[2]);

> On a un buffer, fd[0] écrit et fd[1] lit

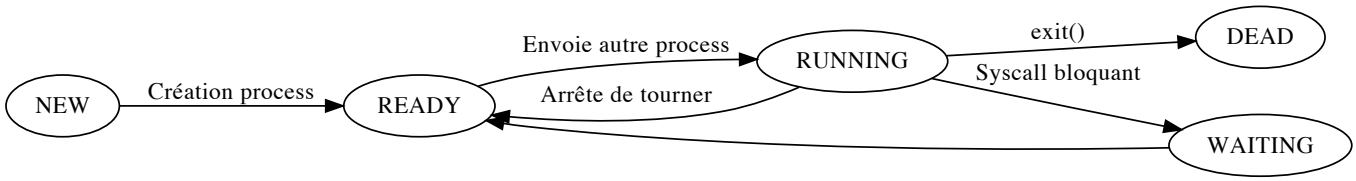
```
fork()
// p1
    fd[0]
    close(fd[1])
    dup2(fd[0], 1)
    fork()
        execvp('cat')

// p2
close(fd[0])
fd[1]
dup2(fd[1], 0)
execvp('cat2')//?
```

États des process

Le process peut être dans 2 états :

- RUNNING
- READY



Syscall bloquants : sleep, wait, read, write, etc
État **DEAD** : Process termine mais un autre process a appelé wait() dessus

init() (pid = 1) : crée un programme P1 à la racine de tous les autres

Process P1 et son child P2

- P2 exit
- P2 envoie un signal SIGCHILD à P1

Process orphelin : rattaché à init

Uid / Gid d’un process

0777 : permission user / group / other
Bit setuid : Premier bit des permissions

- Quand on lance le programme, il n’est pas lancé avec nos permissions mais avec celles de l’owner du programme

Exemple :
17777 root : Lance le programme avec les permissions de root
sudo v $user prg -> /etc/sudoers