

Utilisation des systèmes UNIX

Benjamin Negrevergne

PSL University – Paris Dauphine – Équipes *MILES*



Outline

- 1 Les processus Unix

Programme/Processus

Programme/Processus

Programme:

- Fichier binaire contenant des instructions pour le CPU
- Résultat de la compilation d'un code source
- Objet inerte sur le système de fichier

Programme/Processus

Programme:

- Fichier binaire contenant des instructions pour le CPU
- Résultat de la compilation d'un code source
- Objet inerte sur le système de fichier

Processus

- Programme en cours d'exécution
- Dispose d'un état qui évolue dans le temps
- Peut interagir avec l'utilisateur
- Peut être non-deterministe
- Plusieurs processus peuvent exécuter le même programme (et obtenir des résultats différents)

Programme/Processus

Programme:

- Fichier binaire contenant des instructions pour le CPU
- Résultat de la compilation d'un code source
- Objet inerte sur le système de fichier

Processus

- Programme en cours d'exécution
 - Dispose d'un état qui évolue dans le temps
 - Peut interagir avec l'utilisateur
 - Peut être non-deterministe
 - Plusieurs processus peuvent exécuter le même programme (et obtenir des résultats différents)
- Nombreuses analogies avec les être vivants dans les systèmes UNIX

La commande ps

Permet de voir les processus en cours sur le système

```
1 $ ps # processus en cours dans ma session
2   PID TTY          TIME CMD
3   370529 pts/41    00:00:00 bash
4   371534 pts/41    00:00:00 ps
5 $ ps -a # tous les processus en cours
6 ...
```

```
1 $ ps -o PID,TTY,TIME,CMD
2   PID TTY          TIME CMD
3   ...
4 $ ps -a -o pid,user
5   USER PID
6   ...
```

La commande ps

Permet de voir les processus en cours sur le système

```
1 $ ps # processus en cours dans ma session
2   PID TTY          TIME CMD
3   370529 pts/41    00:00:00 bash
4   371534 pts/41    00:00:00 ps
5 $ ps -a # tous les processus en cours
6 ...
```

```
1 $ ps -o PID,TTY,TIME,CMD
2   PID TTY          TIME CMD
3   ...
4 $ ps -a -o pid,user
5   USER PID
6   ...
```

Informations disponibles:

- UID utilisateur propriétaire du processus
- PID numéro du process
- PPID numéro du process parent
- C priorité
- STIME correspond à l'heure de lancement du processus
- TTY nom du terminal rattaché au processus
- TIME temps d'exécution
- COMMAND utilisée pour lancer le processus
- ...

Descripteur de processus

- Structure du noyau qui permet au noyau de manipuler les différents processus (e.g. `struct task_struct` dans le noyau Linux)
- Un par processus qui existe sur le système

Descripteur de processus

- Structure du noyau qui permet au noyau de manipuler les différents processus (e.g. `struct task_struct` dans le noyau Linux)
- Un par processus qui existe sur le système

Cette structure contient:

- le PID (*process id*), PPID (*parent process id*)
- l'utilisateur propriétaire du processus
- l'état du processus (D/R/S/T/Z)
- l'espace d'adressage mémoire alloué au processus (code/pile/tas)
- la table des fichiers ouverts
 - En particulier: `stdin (0)`, `stdout(1)`, `stdout(2)`
- Un espace alloué pour sauvegarder le contexte d'exécution
Permet de sauvegarder/restaurer l'exécution d'un processus
Registres PC, EAX, EBX, ...

Démarrage, et cycle de vie d'un processus

Démarrage

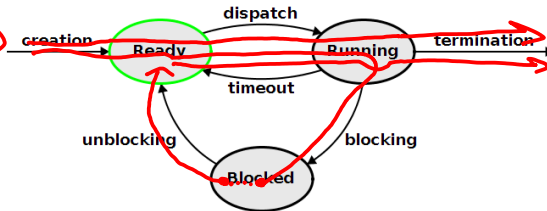
- 1 Le noyau attribue un PID et un espace d'adressage mémoire
- 2 Le noyau charge le programme en mémoire et initialise les variables
- 3 Le noyau donne la main au processus pour qu'il s'exécute

Démarrage, et cycle de vie d'un processus

Démarrage

- 1 Le noyau attribue un PID et un espace d'adressage mémoire
- 2 Le noyau charge le programme en mémoire et initialise les variables
- 3 Le noyau donne la main au processus pour qu'il s'exécute

Cycle de vie



<https://www.d.umn.edu/~gshute/os/images/ready.png>

États des processus

```
1 #define TASK_RUNNING          0 // Running/runnable
2 #define TASK_INTERRUPTIBLE    1 // S en attente
3 #define TASK_UNINTERRUPTIBLE  2 // D en sommeil ininterruptible
4 #define TASK_ZOMBIE           4 // Z en attente de terminaison
5 #define TASK_STOPPED          8 // T Interrompu
6 ..
```

États des processus

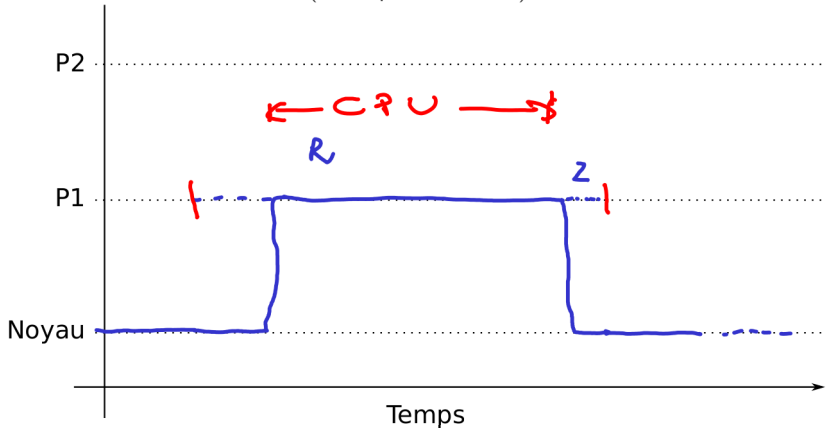
```
1 #define TASK_RUNNING          0 // Running/runnable
2 #define TASK_INTERRUPTIBLE    1 // S en attente
3 #define TASK_UNINTERRUPTIBLE  2 // D en sommeil ininterrompible
4 #define TASK_ZOMBIE           4 // Z en attente de terminaison
5 #define TASK_STOPPED          8 // T Interrompu
6 ..
```

La commande `ps` permet de connaître l'état des processus existants

```
1 $ ps -a -o state | sort | uniq -c
2     2 R
3    245 S
4     4 Z
```

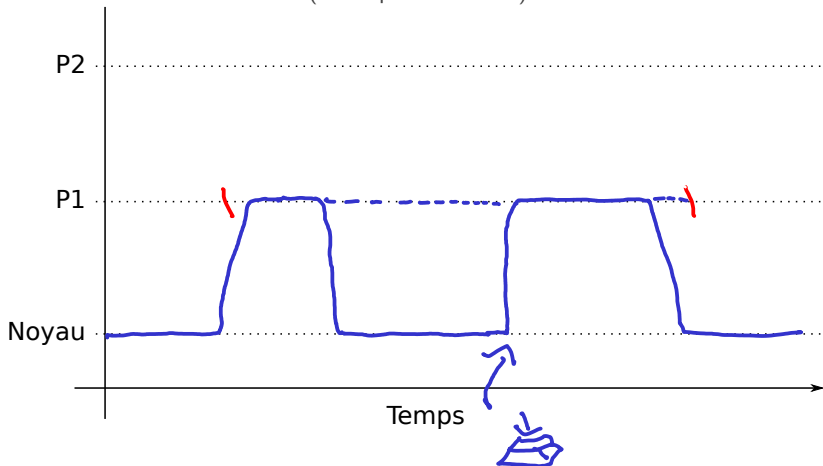
Scénario 1: exécution de P1 ininterrompue

(À compléter en cours)



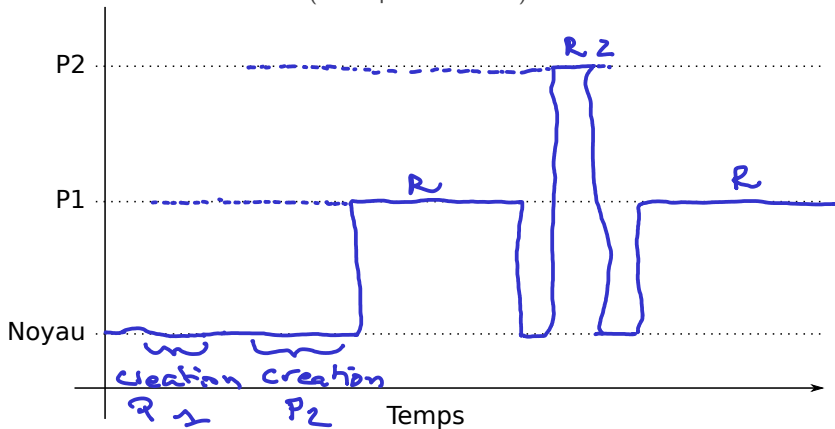
Scénario 2: mise en attente de P1

(À compléter en cours)



Scénario 3: exécution concurrente de P1 et P2

(À compléter en cours)



Mémoire d'un processus

```
1 int f(){
2     int a = 10;
3     printf("PID %d\n", getpid());
4     printf("adresse de a : %p\n", &a);
5     printf("adresse de f : %p\n", &f);
6     sleep(10000);
7     printf("%d\n", a); //to prevent compiler optimization
8 }
9 int main(){
10    f();
11 }
```

```
1 bnegreve@neb:~$ gcc test.c
2 bnegreve@neb:~$ ./a.out
3 PID 373015
4 adresse de a : 0x7ffd058d06ec
5 adresse de f : 0x56192b3ea155
```

Mémoire d'un processus

```
1 int f(){
2   int a = 10;
3   printf ("PID %d\n", getpid());
4   printf ("adresse de a : %p\n", &a);
5   printf ("adresse de f : %p\n", &f);
6   sleep(10000);
7   printf ("%d\n", a); //to prevent compiler optimization
8 }
9 int main(){
10  f();
11 }
```

```
1 bnegreve@neb:~$ gcc test.c
2 bnegreve@neb:~$ ./a.out
3 PID 373015
4 adresse de a : 0x7ffd058d06ec
5 adresse de f : 0x56192b3ea155
```

```
1 bnegreve@neb:~$ cat /proc/373015/maps
2 56192b3e9000-56192b3ea000 r--p 00000000 fd:03 24791179 /home/bnegreve/a.out
3 56192b3ea000-56192b3eb000 r-xp 00001000 fd:03 24791179 /home/bnegreve/a.out <---- function f()!
4 56192b3eb000-56192b3ec000 r--p 00002000 fd:03 24791179 /home/bnegreve/a.out
5 56192b3ec000-56192b3ed000 r--p 00002000 fd:03 24791179 /home/bnegreve/a.out
6 56192b3ed000-56192b3ee000 rw-p 00003000 fd:03 24791179 /home/bnegreve/a.out
7 56192c68b000-56192c6ac000 rw-p 00000000 00:00 0 [heap]
8 7fd68447c000-7fd6844a1000 r--p 00000000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
9 7fd6844a1000-7fd6845ec000 r-xp 00025000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
10 7fd6845ec000-7fd684636000 r--p 00170000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
11 7fd684636000-7fd684637000 ----p 001ba000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
12 7fd684637000-7fd68463a000 r--p 001ba000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
13 7fd68463a000-7fd68463d000 rw-p 001bd000 fd:01 1443743 /lib/x86_64-linux-gnu/libc-2.31.so
14 7fd68463d000-7fd684643000 rw-p 00000000 00:00 0
15 7fd684677000-7fd684678000 r--p 00000000 fd:01 1441694 /lib/x86_64-linux-gnu/ld-2.31.so
16 7fd684678000-7fd684698000 r-xp 00001000 fd:01 1441694 /lib/x86_64-linux-gnu/ld-2.31.so
17 7fd684698000-7fd6846a0000 r--p 00021000 fd:01 1441694 /lib/x86_64-linux-gnu/ld-2.31.so
18 7fd6846a1000-7fd6846a2000 r--p 00029000 fd:01 1441694 /lib/x86_64-linux-gnu/ld-2.31.so
19 7fd6846a2000-7fd6846a3000 rw-p 0002a000 fd:01 1441694 /lib/x86_64-linux-gnu/ld-2.31.so
20 7fd6846a3000-7fd6846a4000 rw-p 00000000 00:00 0
21 7ffd058b1000-7ffd058d2000 rw-p 00000000 00:00 0 [stack] <---- var a !
22 7ffd05989000-7ffd0598d000 r--p 00000000 00:00 0 [vvar]
23 7ffd0598d000-7ffd0598f000 r-xp 00000000 00:00 0 [vdso]
```

Organisation de la mémoire d'un processus

(de bas en haut)

```
1 7ffd0598d000-7ffd0598f000 r-xp [vdso]
2 7ffd05989000-7ffd0598d000 r--p [vvar]
3 7ffd058b1000-7ffd058d2000 rw-p [stack]
4 7fd6846a3000-7fd6846a4000 rw-p
5 7fd6846a2000-7fd6846a3000 rw-p /lib/.../ld-2.31.so
6 7fd6846a1000-7fd6846a2000 r--p /lib/.../ld-2.31.so
7 7fd684698000-7fd6846a0000 r--p /lib/.../ld-2.31.so
8 7fd684678000-7fd684698000 r-xp /lib/.../ld-2.31.so
9 7fd684677000-7fd684678000 r--p /lib/.../ld-2.31.so
10 7fd68463d000-7fd684643000 rw-p
11 7fd68463a000-7fd68463d000 rw-p /lib/.../libc-2.31.so
12 7fd684637000-7fd68463a000 r--p /lib/.../libc-2.31.so
13 7fd684636000-7fd684637000 ---p /lib/.../libc-2.31.so
14 7fd6845ec000-7fd684636000 r--p /lib/.../libc-2.31.so
15 7fd6844a1000-7fd6845ec000 r-xp /lib/.../libc-2.31.so
16 7fd68447c000-7fd6844a1000 r--p /lib/.../libc-2.31.so
17 56192c68b000-56192c6ac000 rw-p [heap]
18 56192b3ed000-56192b3ee000 rw-p /home/bnegreve/a.out
19 56192b3ec000-56192b3ed000 r--p /home/bnegreve/a.out
20 56192b3eb000-56192b3ec000 r--p /home/bnegreve/a.out
21 56192b3ea000-56192b3eb000 r-xp /home/bnegreve/a.out
22 56192b3e9000-56192b3ea000 r--p /home/bnegreve/a.out
```

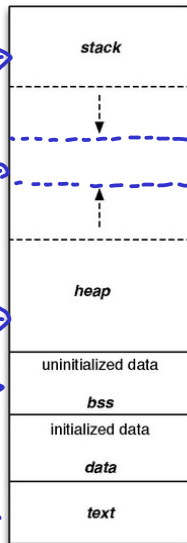
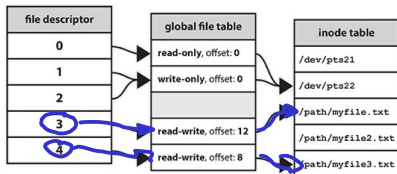


Table des fichiers

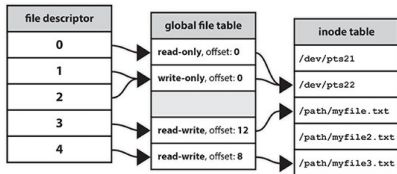
```
1 file1=open("/path/myfile.txt", "rw");  
2 file2=open("/path/myfile3.txt", "rw");  
3 ...  
4 close(file1)  
5 close(file12)
```



ComputerHope.com

Table des fichiers

```
1 file1=open("/path/myfile.txt", "rw");
2 file2=open("/path/myfile3.txt", "rw");
3 ...
4 close(file1)
5 close(file12)
```



ComputerHope.com

Descripteurs réservés:

- 0 (*stdin*) : Entrée standard
- 1 (*stdout*) : Sortie standard
- 2 (*stderr*) : Sortie erreur

write(3, "salut");
printf("salut");
↳ write(1, "salut");

Table des fichiers

```
1 bnegreve@neb:~$ ./a.out
2 PID 373015
3 ...
4 bnegreve@neb:~$ ls -l /proc/373015/fd/
5 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 0 -> /dev/pts/41
6 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 1 -> /dev/pts/41
7 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 2 -> /dev/pts/41
```

Table des fichiers

• $> \text{out}$ \rightarrow redirige 1 vers le fichier out

```
1 bnegreve@neb:~$ ./a.out
2 PID 373015
3 ...
4 bnegreve@neb:~$ ls -l /proc/373015/fd/
5 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 0 -> /dev/pts/41
6 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 1 -> /dev/pts/41
7 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:39 2 -> /dev/pts/41
```

• $< \text{in}$ $\leftarrow \rightarrow$ redirige 0 vers le fichier in

```
1 bnegreve@neb:~$ ./a.out > out
2 bnegreve@neb:~$ ps aux | grep a.out
3 bnegreve 373752 0.0 0.0 2304 500 pts/41 S 15:44 0:00 ./a.out
4 bnegreve@neb:~$ ls -l /proc/373752/fd
5 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:44 0 -> /dev/pts/41
6 l-wx----- 1 bnegreve bnegreve 64 Feb 8 15:44 1 -> /home/bnegreve/out
7 lrwx----- 1 bnegreve bnegreve 64 Feb 8 15:44 2 -> /dev/pts/41
```

a | b \rightarrow redirige 1 de a vers 0 de b.

• $2 > \text{out}$ \rightarrow redirige 2 vers le fichier out

kill

kill - send a signal to a process

- 18 SIGCONT Demande de reprise du processus
- 19 SIGSTOP Demande de suspension imbloquentable
- 9 SIGKILL Instruction (KILL) - termine le processus immédiatement
- 15 SIGTERM Signal de terminaison
- 11 SIGSEGV Violation de mémoire