

TP SYSTÈME M1 RSD 2020-2021

Processus et signaux

La création de processus sous linux

La fonction fork()

- Image mémoire d'un processus

Processus père (le programme main)

Identification processus :
PID=**1242** , PPID=? , ETAT

Segment données: **pid= 1243**
stack, heap , etc

Code :

`int pid=fork();
if(pid==0)`

VALEURS REGISTRES
PROCESSEURS: CO, registres de
données

Processus fils

Identification processus fils :
PID=**1243** , PPID=PID du père,
ETAT

Segment données: **pid=0**
stack, heap , etc

Code :

`int pid=fork();
if(pid==0)`

VALEURS REGISTRES **hérités du
père : CO, registres de données**

Creation d'un nouveau
processus par le noyau

Nouvelle valeur du CO :

CO:

Les appels systèmes fork, getpid, getppid

- Utiliser le manuel linux pour en apprendre plus
- Les pages du man sont structurées en plusieurs sections comme suit:
 - Man 1 [?] ou man tout court, section des commandes shell et programmes exécutables installés
 - **Man 2** [?] section des appels systèmes
 - Man 3 [?] fonctions de la bibliothèque installées
 - Man 4 [?] fichiers spéciaux situés généralement dans dev
 - Man 5 [?] format de fichiers et conventions (exp /etc/passwd)
 - Man 6 [?] section des jeux ...

Utilisation man 2 sur le shell linux : **man 2 fork , man 2 getpid, man2 getppid**

fork(), getpid(), getppid()

fork(), getpid(), getppid() sont des appels système définis dans unistd.h

pid_t pid=fork(): créer un processus fils au processus appelant, l'image mémoire du père est dupliquée pour créer celle du fils ...

Retourne le pid du fils crée pour le père (car c'est lui qui a fait l'appel) et 0 pour le fils (normal il n'a pas exécuté l'instruction fork !)

- **pid_t getpid() :** retourne pid du processus appelant
- **pid_t getppid() :** retourne le pid du processus père

pid_t est un type défini dans sys/types.h, c'est un int donc on peut aussi écrire `int pid=fork()` ...

La commande ps : visualiser les processus du système

- **ps aux** (plus de détails? Voir man ps)

Les options **aux**:

a = show processes for all users

u = display the process's user/owner

x = also show processes not attached to a terminal (colonne Tty=?)

Autres options? Voir le man

ps -ef

e: affiche l'environnement des processus (usage mémoire, cpu etc)

f: affiche aussi l'arbre généalogique des processus (montrer le ppid)

Commande ps

```
sanjay@dev1: ~  
File Edit View Search Terminal Help  
sanjay@dev1:~$ ps  
  PID TTY          TIME CMD  
 2687 pts/0        00:00:00 bash  
 3043 pts/0        00:00:00 ps  
sanjay@dev1:~$
```

```
sanjay@dev1: ~  
File Edit View Search Terminal Help  
sanjay@dev1:~$ ps aux  
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root           1   0.0  0.2 220192  5144 ?        Ss   15:22   0:06 /sbin/in  
root           2   0.0  0.0      0      0 ?        S    15:22   0:00 [kthread  
root           4   0.0  0.0      0      0 ?        S<   15:22   0:00 [kworker  
root           6   0.0  0.0      0      0 ?        S<   15:22   0:00 [mm_perc  
root           7   0.0  0.0      0      0 ?        S    15:22   0:01 [ksoftir  
root           8   0.0  0.0      0      0 ?        S    15:22   0:01 [rcu_sch  
root           9   0.0  0.0      0      0 ?        S    15:22   0:00 [rcu_bh]  
root          10   0.0  0.0      0      0 ?        S    15:22   0:00 [migrati  
root          11   0.0  0.0      0      0 ?        S    15:22   0:00 [watchdo  
root          12   0.0  0.0      0      0 ?        S    15:22   0:00 [cpuhp/0
```

Commande ps

- Ps -ef

```
sanjay@dev1: ~  
File Edit View Search Terminal Help  
sanjay@dev1:~$ ps -ef  
UID          PID     PPID  C  STIME TTY          TIME CMD  
root           1         0  0  15:22 ?           00:00:06 /sbin/init splash  
root           2         0  0  15:22 ?           00:00:00 [kthreadd]  
root           4         2  0  15:22 ?           00:00:00 [kworker/0:0H]  
root           6         2  0  15:22 ?           00:00:00 [mm_percpu_wq]  
root           7         2  0  15:22 ?           00:00:01 [ksoftirqd/0]  
root           8         2  0  15:22 ?           00:00:01 [rcu_sched]  
  
sanjay@dev1: ~  
File Edit View Search Terminal Help  
sanjay@dev1:~$ ps -eF  
UID          PID     PPID  C    SZ    RSS  PSR  STIME TTY          TIME CMD  
root           1         0  0  55048  5204    0  15:22 ?           00:00:06 /sbin/in  
root           2         0  0      0      0    0  15:22 ?           00:00:00 [kthread  
root           4         2  0      0      0    0  15:22 ?           00:00:00 [kworker  
root           6         2  0      0      0    0  15:22 ?           00:00:00 [mm_perc  
root           7         2  0      0      0    0  15:22 ?           00:00:01 [ksoftir  
root           8         2  0      0      0    0  15:22 ?           00:00:01 [rcu_sch  
root           9         2  0      0      0    0  15:22 ?           00:00:00 [rcu_bh]
```