

Système partie 1

Pastille 6 - Identité des processus

Vincent Colotte - N. de Rugy-Altherre

Identité d'un processus

PID

Primitives pour récupérer l'identité du processus :

```
#include <unistd.h>
#include <sys/types.h>
prototypes :
    pid_t getpid(); // PID du processus courant
    pid_t getppid(); // PID de son processus pa
```

à ne pas confondre avec *getuid()*, *getgid()*... (identité de l'utilisateur)

Identité d'un processus

PID

Primitives pour récupérer l'identité du processus :

```
#include <unistd.h>
#include <sys/types.h>
prototypes :
    pid_t getpid(); // PID du processus courant
    pid_t getppid(); // PID de son processus parent
```

à ne pas confondre avec *getuid()*, *getgid()*... (identité de l'utilisateur)

Exemples :

- Le Père fait *getpid()* \Rightarrow on obtient PPPPP
- Le Fils fait *getpid()* \Rightarrow on obtient FFFFF
- Le Fils fait *getppid()* \Rightarrow on obtient
- Le Père fait *getppid()* \Rightarrow on obtient

Identité d'un processus

PID

Primitives pour récupérer l'identité du processus :

```
#include <unistd.h>
#include <sys/types.h>
prototypes :
    pid_t getpid(); // PID du processus courant
    pid_t getppid(); // PID de son processus parent
```

à ne pas confondre avec *getuid()*, *getgid()*... (identité de l'utilisateur)

Exemples :

- Le Père fait *getpid()* \Rightarrow on obtient PPPPP
- Le Fils fait *getpid()* \Rightarrow on obtient FFFFF
- Le Fils fait *getppid()* \Rightarrow on obtient
- Le Père fait *getppid()* \Rightarrow on obtient

Identité d'un processus

PID

Primitives pour récupérer l'identité du processus :

```
#include <unistd.h>
#include <sys/types.h>
prototypes :
    pid_t getpid(); // PID du processus courant
    pid_t getppid(); // PID de son processus parent
```

à ne pas confondre avec *getuid()*, *getgid()*... (identité de l'utilisateur)

Exemples :

- Le Père fait `getpid()` \Rightarrow on obtient PFFFF
- Le Fils fait `getpid()` \Rightarrow on obtient FFFFF
- Le Fils fait `getppid()` \Rightarrow on obtient PFFFF
- Le Père fait `getppid()` \Rightarrow on obtient 0

Identité d'un processus

PID

Primitives pour récupérer l'identité du processus :

```
#include <unistd.h>
#include <sys/types.h>
prototypes :
    pid_t getpid(); // PID du processus courant
    pid_t getppid(); // PID de son processus pa
```

à ne pas confondre avec *getuid()*, *getgid()*... (identité de l'utilisateur)

Exemples :

- Le Père fait `getpid()` \Rightarrow on obtient PPPPP
- Le Fils fait `getpid()` \Rightarrow on obtient FFFFF
- Le Fils fait `getppid()` \Rightarrow on obtient PPPPP
- Le Père fait `getppid()` \Rightarrow on obtient GGGGG (le Grand-Père)

Session et processus initial

PID

Le processus `init` est le processus originel :

- créé par le noyau au démarrage (`pid=1`)
- lance les autres processus système. Notamment le processus `login-shell`.

Session et processus initial

PID

Le processus `init` est le processus originel :

- créé par le noyau au démarrage (`pid=1`)
- lance les autres processus système. Notamment le processus `login-shell`.

Session

- Le processus `login-shell` crée une session associée à un utilisateur (un shell + terminal).
- notion de processus **leader** : le premier de la session (shell) ou du groupe.

Session et processus initial

PID

Le processus `init` est le processus originel :

- créé par le noyau au démarrage (`pid=1`)
- lance les autres processus système. Notamment le processus `login-shell`.

Session

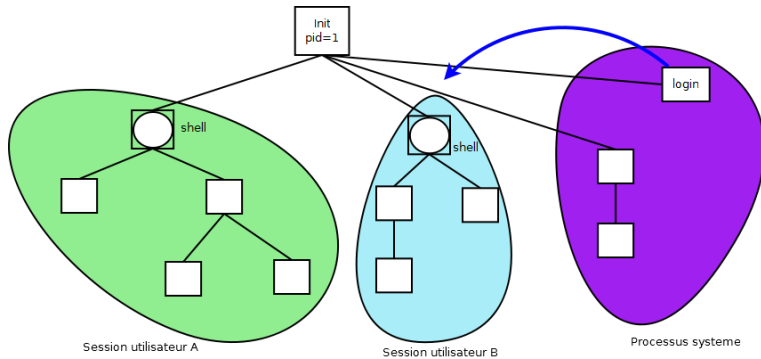
- Le processus `login-shell` crée une session associée à un utilisateur (un shell + terminal).
- notion de processus **leader** : le premier de la session (shell) ou du groupe.

Un groupe (*job*) est constitué du processus lancé par le shell et de de ses descendants.

- groupe interactif : avant-plan
- groupe arrière-plan

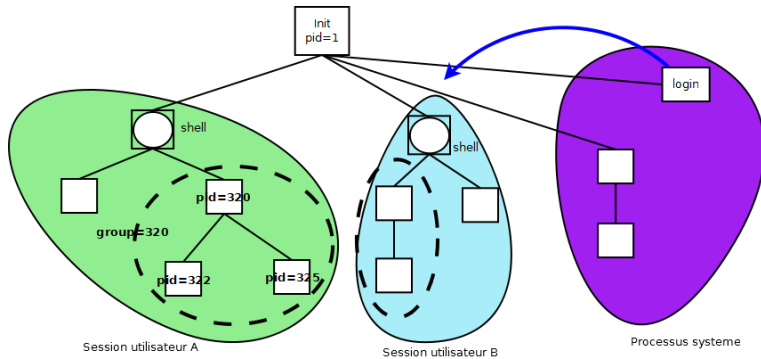
Session et processus initial

PID



Session et processus initial

PID



Exercice

Écrivez un programme avec un fork. Pour le père et le fils, affichez getpid et getppid ainsi que la valeur de retour du fork (ret).