

Système partie 1

Pastille 3 - fork

Vincent Colotte - N. de Rugy-Altherre

Création

Fork

```
#include <unistd.h>
#include <sys/types.h>
prototype : pid_t fork();
```

La fonction *fork* :

- Provoque la création d'un nouveau processus (le fils)

```
#include <unistd.h>
#include <sys/types.h>
prototype : pid_t fork();
```

La fonction *fork* :

- Provoque la création d'un nouveau processus (le fils)
- En dupliquant l'image mémoire du processus père :
 - texte exécutable (le code)
 - les variables
 - *fichiers ouverts**

```
#include <unistd.h>
#include <sys/types.h>
prototype : pid_t fork();
```

La fonction *fork* :

- Provoque la création d'un nouveau processus (le fils)
- En dupliquant l'image mémoire du processus père :
 - texte exécutable (le code)
 - les variables
 - *fichiers ouverts**

Seule la variable de retour change

- $= 0 \rightarrow$ on est dans le (nouveau) processus fils
- $> 0 \rightarrow$ *pid* du fils (mais on est dans le processus père)
- $-1 \rightarrow$ échec

Création

Fork

```
...
switch ( ret = fork() ){
case (pid_t) -1 :    perror("creation impossible");
                    exit(1);
case (pid_t) 0 :
    /* on est dans le fils */
    ...
    break; (ou exit(0));
default :
    /* on est dans le pere*/
    ...
}
```

Exercice

Écrivez un programme avec un fork. Faites en sorte que le père affiche "Luc je suis ton père" et le fils affiche "NOOOOOON". Le fils ne doit pas faire d'exit.

Affichez avant le switch : "Bienvenu dans un film"

Affichez après la fin du switch : "C'est la fin !"

La dernière phrase s'affiche deux fois. Pourquoi ?

Création

Fork

Processus père

pile kernel

pile user

données

texte executable

•
•
•
ret=fork()
•
•
•

Création

Fork

Processus père

pile kernel

pile user

données

texte executable



•
•
•
•

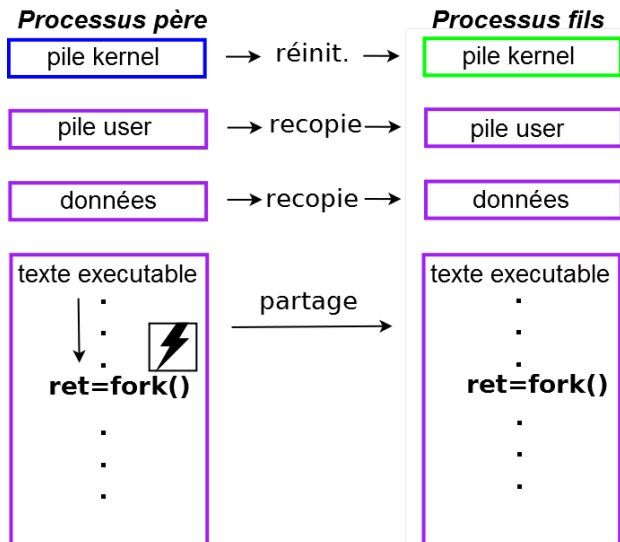


ret=fork()

•
•
•

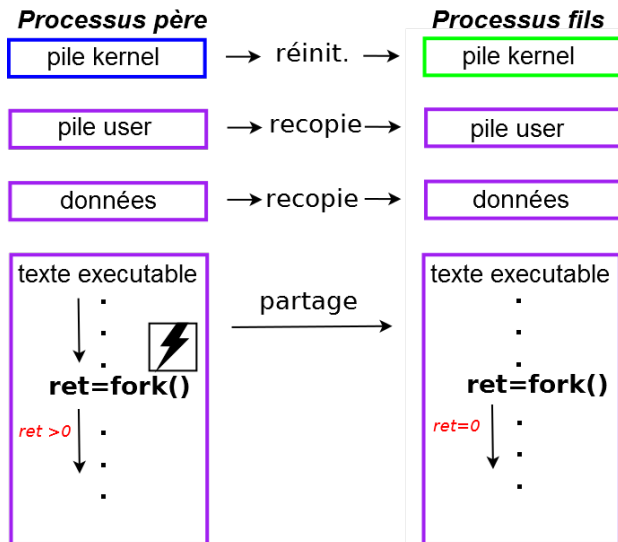
Création

Fork



Création

Fork



Exercice

Exercice

Écrivez un programme avec un fork. Le fils ne fera pas de exit.
Déclarez une variable `int i = 0;` avant le fork.

Dans le père, ajoutez 10 à `i`.

Dans le fils, ajoutez 50 à `i`

À la fin du switch, ajoutez 4 à `i` puis affichez-le. Avez-vous deviné ce qu'il vaut ?



Fork =

- 1 même programme
- 2 processus

→ "clonage"