

FORK(2)

Manuel du programmeur Linux

FORK(2)

NOM

fork - Créer un processus fils (child).

SYNOPSIS

```
#include <sys/types.h>
#include <unistd.h>

pid_t fork(void);
```

DESCRIPTION

fork crée un processus fils qui diffère du processus parent uniquement par ses valeurs PID et PPID et par le fait que toutes les statistiques d'utilisation des ressources sont remises à zéro. Les verrouillages de fichiers, et les signaux en attente ne sont pas hérités.

Sous Linux, fork est implementé en utilisant une méthode de copie à l'écriture. Ceci consiste à ne faire la véritable duplication d'une page mémoire que lorsqu'un processus en modifie une instance. Tant qu'aucun des deux processus n'écrit dans une page donnée, celle-ci n'est pas vraiment dupliquée. Ainsi les seules pénalisations induites par fork sont le temps et la mémoire nécessaires à la copie de la table des pages du parent ainsi que la création d'une structure de tâche pour le fils.

VALEUR RENVOYÉE

En cas de succès, le PID du fils est renvoyé au processus parent, et 0 est renvoyé au processus fils. En cas d'échec -1 est renvoyé dans le contexte du parent, aucun processus fils n'est créé, et errno contient le code d'erreur.

ERREURS

ENOMEM Impossible d'allouer assez de mémoire pour copier la table des pages du père et d'allouer une structure de tâche pour le fils.

EAGAIN Impossible de trouver un emplacement vide dans la table des processus.

EXEMPLE

```
pid_t  pid;

pid = fork ();

if (pid > 0) {
    /* Processus père      */
} else if (pid == 0) {
    /* Processus fils     */
} else {
    /* Traitement d'erreur */
}
```

CONFORMITÉ

Svr4, SVID, POSIX, X/OPEN, BSD 4.3

VOIR AUSSI

clone(2), execve(2), vfork(2), wait(2).

TRADUCTION

Christophe Blaess, 1997.

Linux

30 mai 2001

FORK(2)