

<http://lphweb3.in2p3.fr/spip.php?article1227>

# contrôle des processus fils créés par fork()

- Le LPNHE - Pages du personnel permanent - Chercheurs - Beau Tristan - Divers - Vielleries -



Date de mise en ligne : mercredi 7 juin 2017

---

Copyright © LPNHE - UMR 7585 - Tous droits réservés

---

Lors de l'utilisation de `fork` (voir aussi [article 32](#)), le parent connaît le `pid` du processus fils créé.

Avec la fonction `wait(int *status)`, il est très facile de contrôler la fin des fils. En allant faire un tour dans le [man 2 wait](#), on trouvera même des macros permettant de tester le retour `status` et identifier la raison de la fin des processus fils.

Petit exemple trivial...

```
#include <stdio.h> #include <stdlib.h> #include <unistd.h> #include <sys/types.h> #include <sys/wait.h>
#define NB_CHILDS 5

void do_something(int k) {

    useconds_t s;

    sranddev();
    s = 20*((float)rand())/RAND_MAX;

    printf("CHILD : do %d; sleep %d seconds\n",k,s);
    usleep(s*1e6);
    printf("CHILD : end %d\n",k);

    exit(0);
}

// --- main -----
int main () {

    int i,status;
    pid_t p,pt[NB_CHILDS];

    for(i=0;i<NB_CHILDS;i++) {
        p = fork();
        if ( ! p ) do_something(i);
        else
            printf("PARENT : %d -> pid = %d\n",i,(pt[i]=p));
    }

    while( (p=wait(&status)) != -1) {
        printf("PARENT : pid %d -> ended\n",p);
    }

    printf("\nPARENT : normal end.\n");
    return 0;
}
```