

Fork - Creazione di processi

Leonardo Bizzoni

January 26, 2024

Solo un processo può creare un altro processo, all'avvio l'OS crea un processo *primordiale* dal quale tutti i processi utente e di sistema vengono progressivamente creati. Questo va a creare un albero gerarchico di processi *padre e figlio*. L'API POSIX che permette di creare un nuovo processo da uno pre-esistente è la *fork*.

```
#include <stdio.h>
#include <unistd.h>

int main(void) {
    int child_pid = fork();

    if (child_pid > 0) {
        printf("[%d] Sono il processo padre, il child PID è %d.\n", getpid(),
            child_pid);
        printf("\tIl processo padre è %d.\n", getppid());
    } else if (child_pid == 0) {
        printf("[%d] Sono il processo figlio, il parent PID è %d.\n", getpid(),
            getppid());
    } else {
        perror("fork error");
    }
}
```

Nel momento in cui viene chiamata la funzione `fork()` si genera un nuovo processo identico al processo padre. Al processo padre `fork()` restituisce il PID del figlio, mentre al processo figlio viene restituito 0.

Inoltre utilizzando la funzione `getpid()` è possibile ottenere il PID del processo in esecuzione e con `getppid()` quello del padre.

Non è garantito che termini sempre prima il processo padre e poi quello figlio o che il processo figlio terminini insieme al padre.

In un programma multithreaded effettuare una chiamata *fork()* duplica **solo** la thread chiamante.