

Create the process using fork 0 system cgl Child Process creation Parent Process creation PPID and PID

a

```
GNU nano 6.2 orphan.c *
#include <stdio.h>
#include <unistd.h>

int main() {
    if (fork() == 0) {
        sleep(5);
        printf("Child running PID=%d PPID=%d\n", getpid(), getppid());
    } else {
        printf("Parent exiting\n");
    }
    return 0;
}
```

Orphan Process

```
GNU nano 6.2
#include <stdio.h>
#include <unistd.h>

int main() {
    if (fork() == 0) {
        printf("Child exiting\n");
    } else {
        sleep(10);
        printf("Parent sleeping\n");
    }
    return 0;
}
```

Zombie Process

```
GNU nano 6.2 fork.c *
#include <stdio.h>
#include <unistd.h>

int main() {
    int pid = fork();

    if (pid == 0)
        printf("Child PID=%d PPID=%d\n", getpid(), getppid());
    else
        printf("Parent PID=%d ChildPID=%d\n", getpid(), pid);

    return 0;
}
```