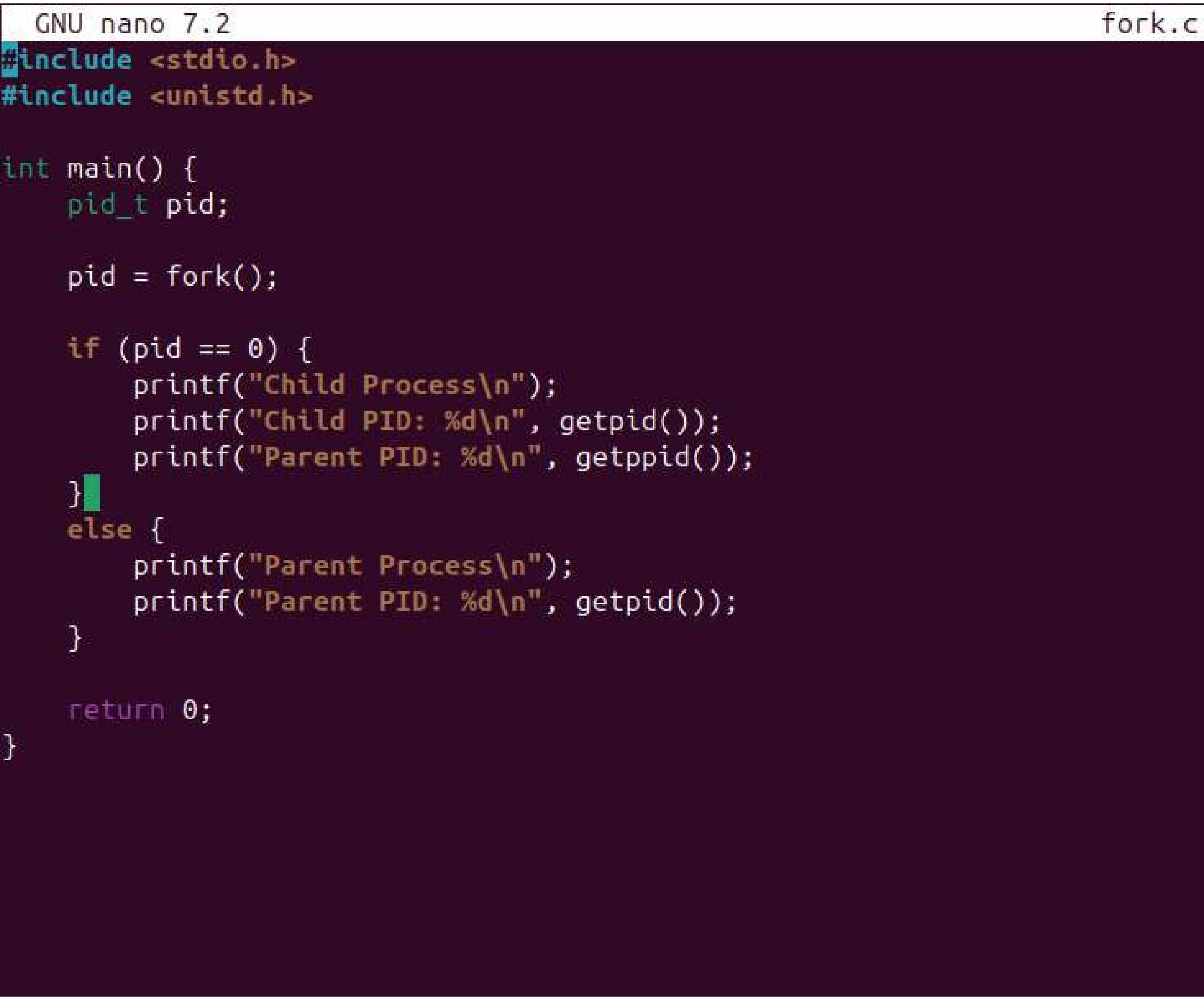
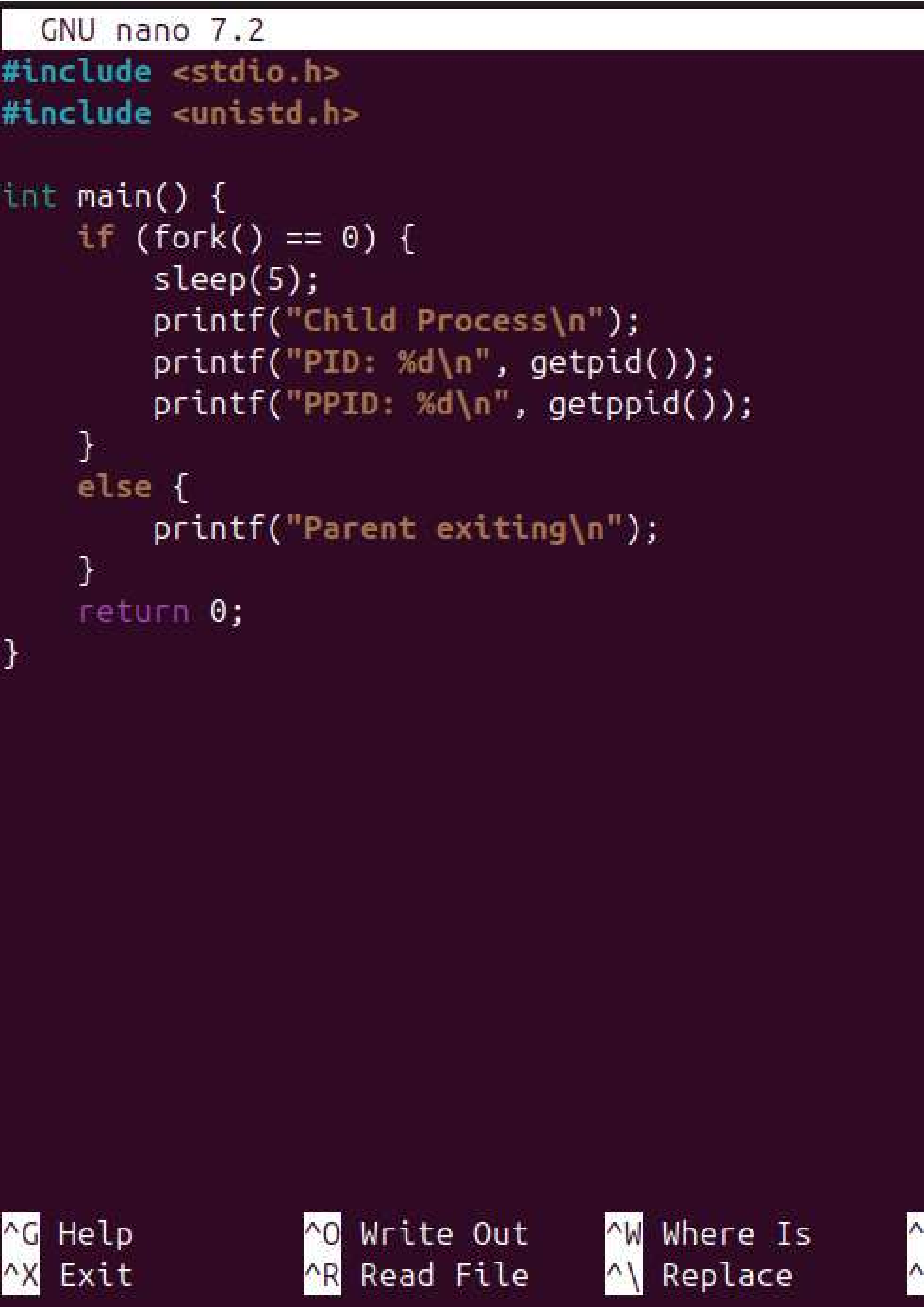
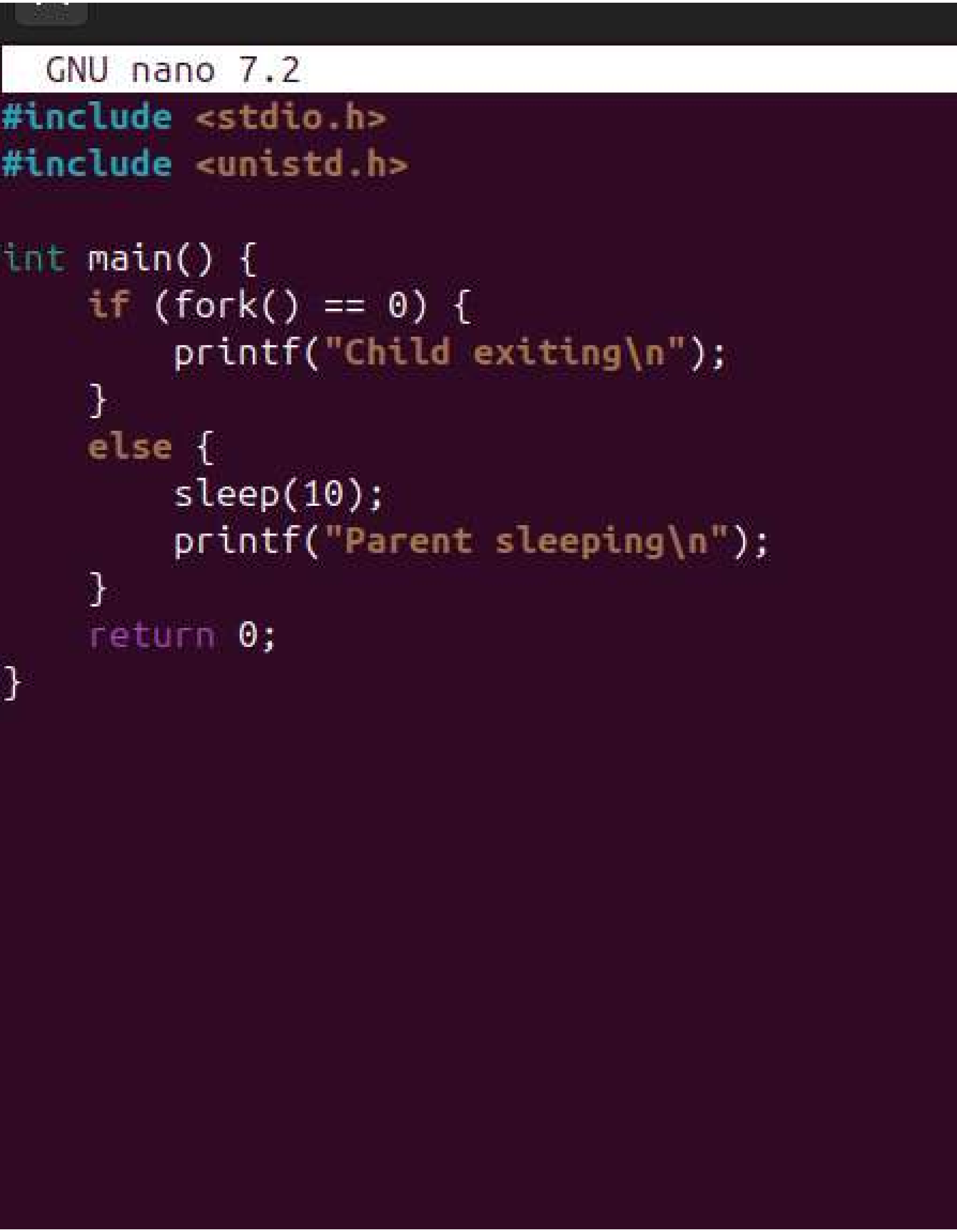


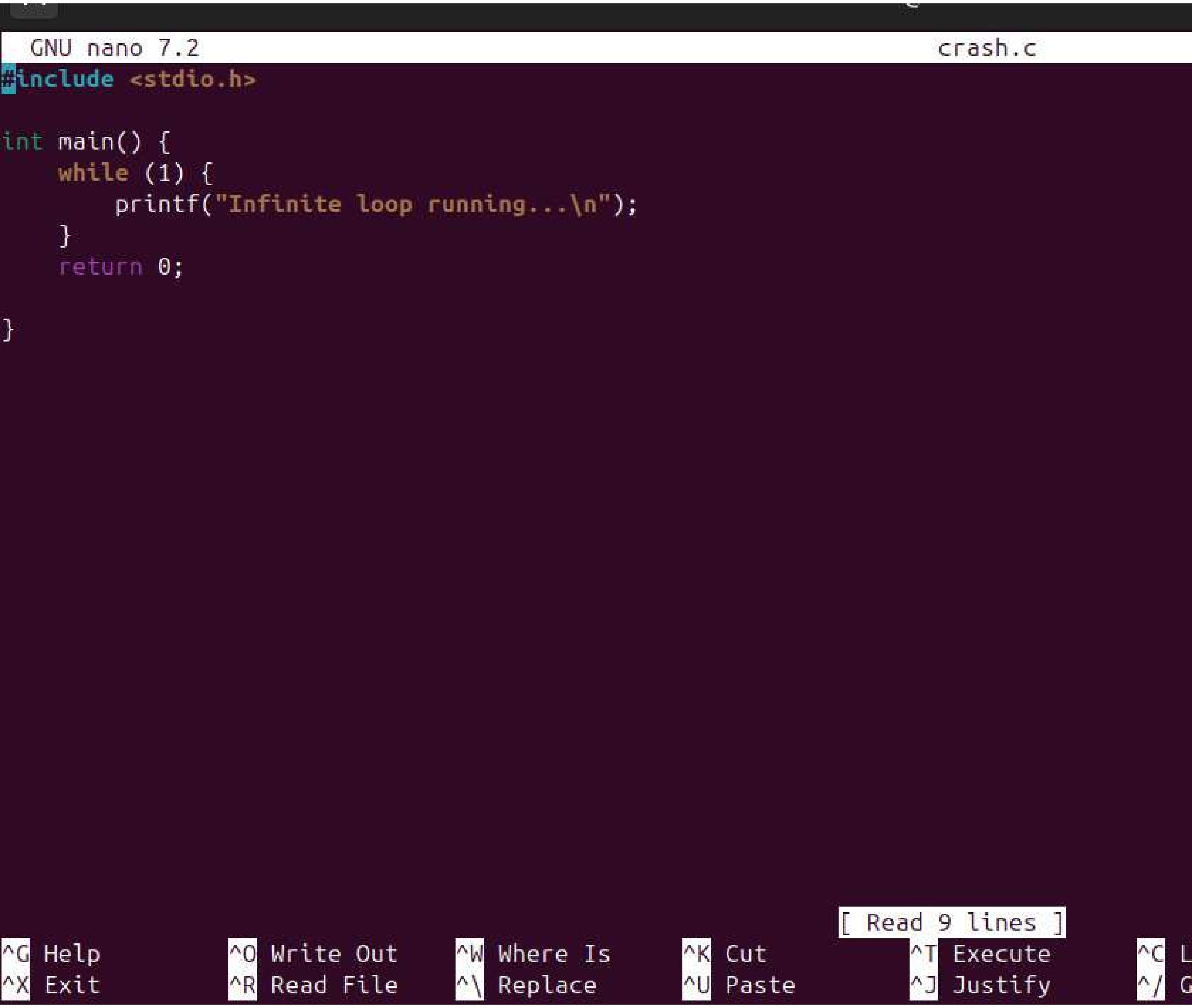
```
  GNU nano 7.2                                                fork.c
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid;

    pid = fork();

    if (pid == 0) {
        printf("Child Process\n");
        printf("Child PID: %d\n", getpid());
        printf("Parent PID: %d\n", getppid());
    }
    else {
        printf("Parent Process\n");
        printf("Parent PID: %d\n", getpid());
    }

    return 0;
}
```



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

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



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

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



```
  GNU nano 7.2                                                    crash.c
#include <stdio.h>

int main() {
    while (1) {
        printf("Infinite loop running...\n");
    }
    return 0;

}
```

```
[ Read 9 lines ]
```