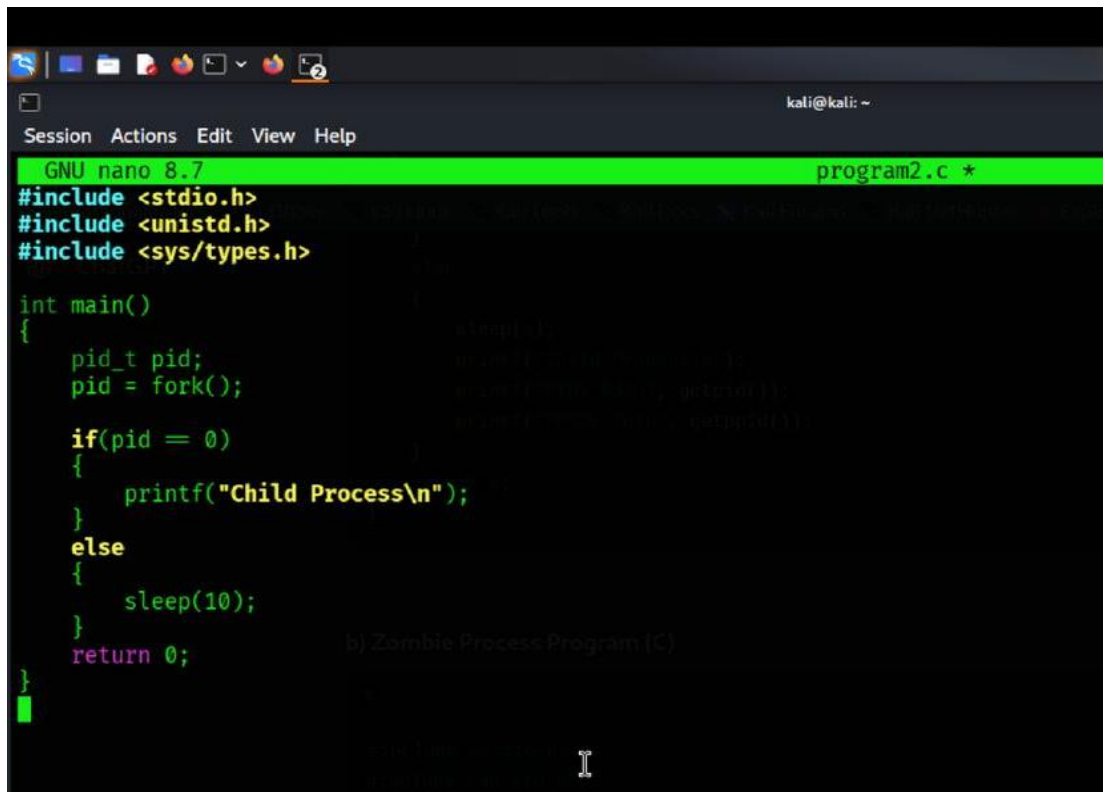


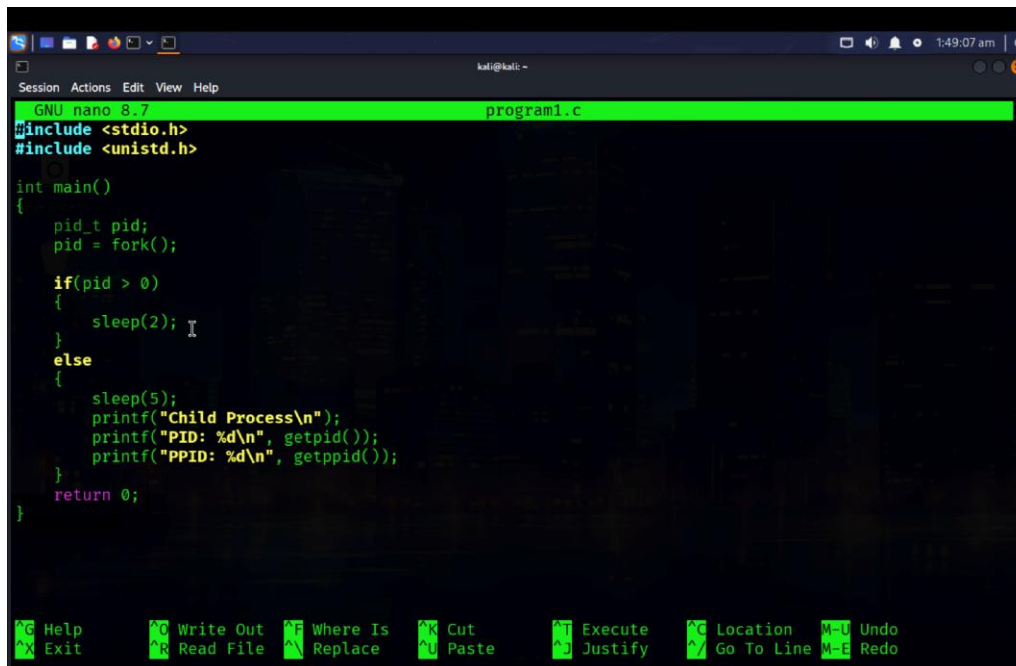
- Zombie Process Code



```
GNU nano 8.7 program2.c *
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main()
{
    sleep(1);
    pid_t pid;
    pid = fork();
    if(pid == 0)
    {
        printf("Child Process\n");
    }
    else
    {
        sleep(10);
    }
    return 0;
}
```

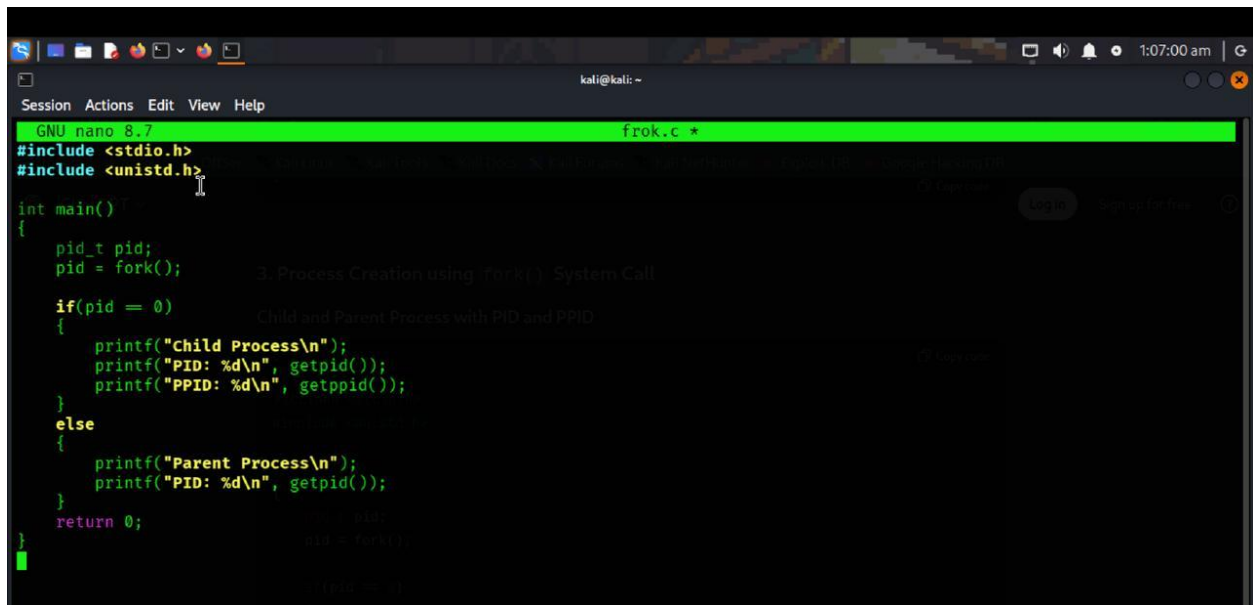
- Orphan Process Code



```
GNU nano 8.7 program1.c
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid;
    pid = fork();
    if(pid > 0)
    {
        sleep(2);
    }
    else
    {
        sleep(5);
        printf("Child Process\n");
        printf("PID: %d\n", getpid());
        printf("PPID: %d\n", getppid());
    }
    return 0;
}
```

- Fork



```
GNU nano 8.7 frok.c *
#include <stdio.h>
#include <unistd.h>

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

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