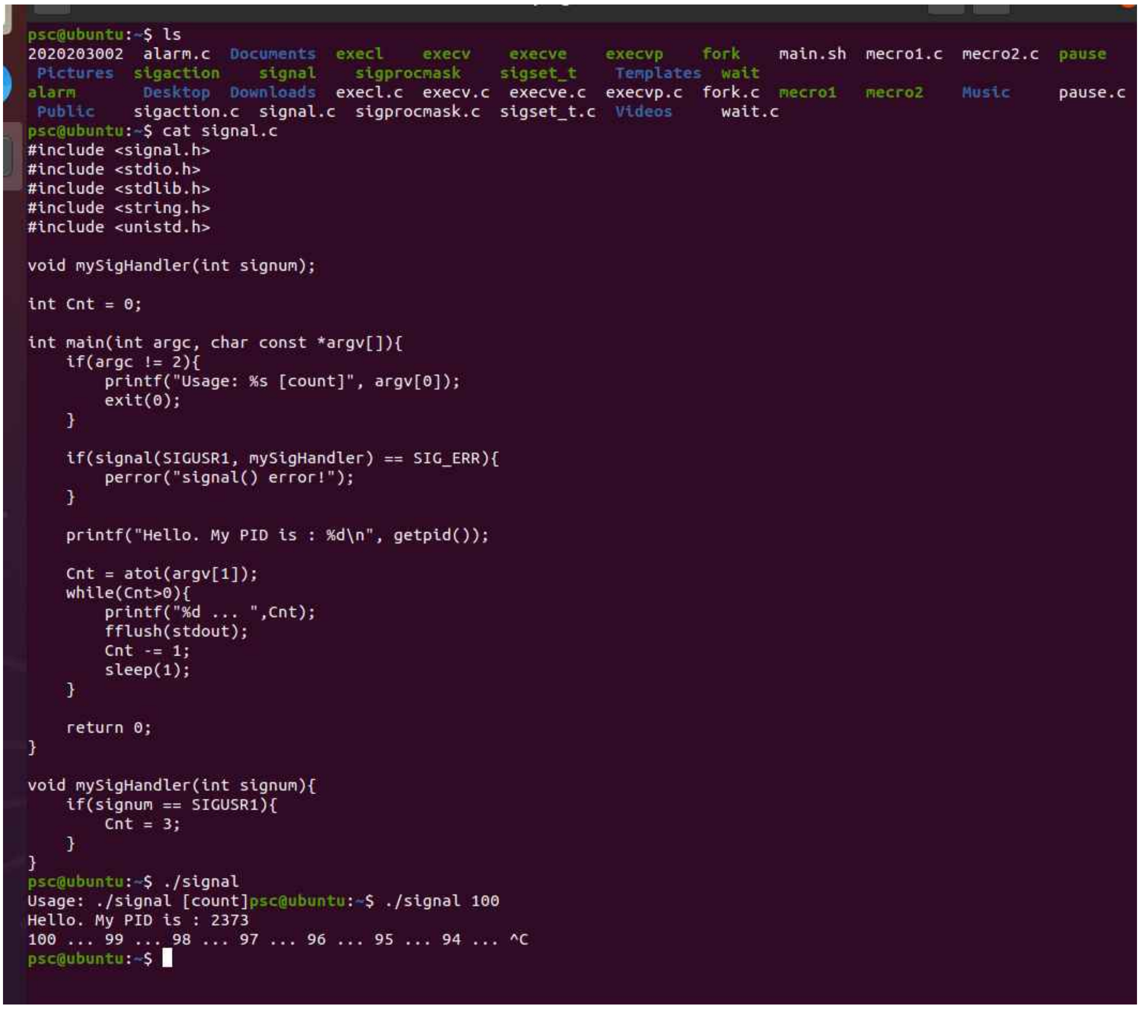
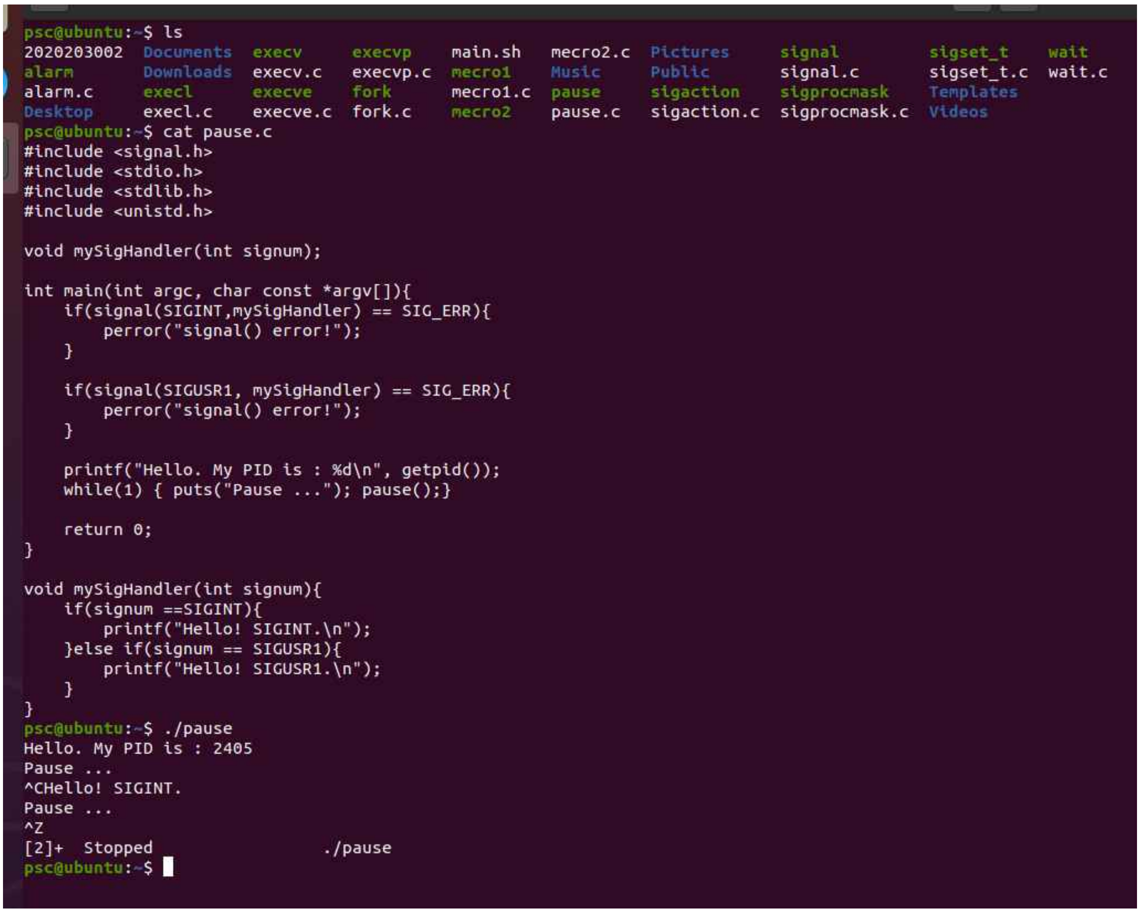
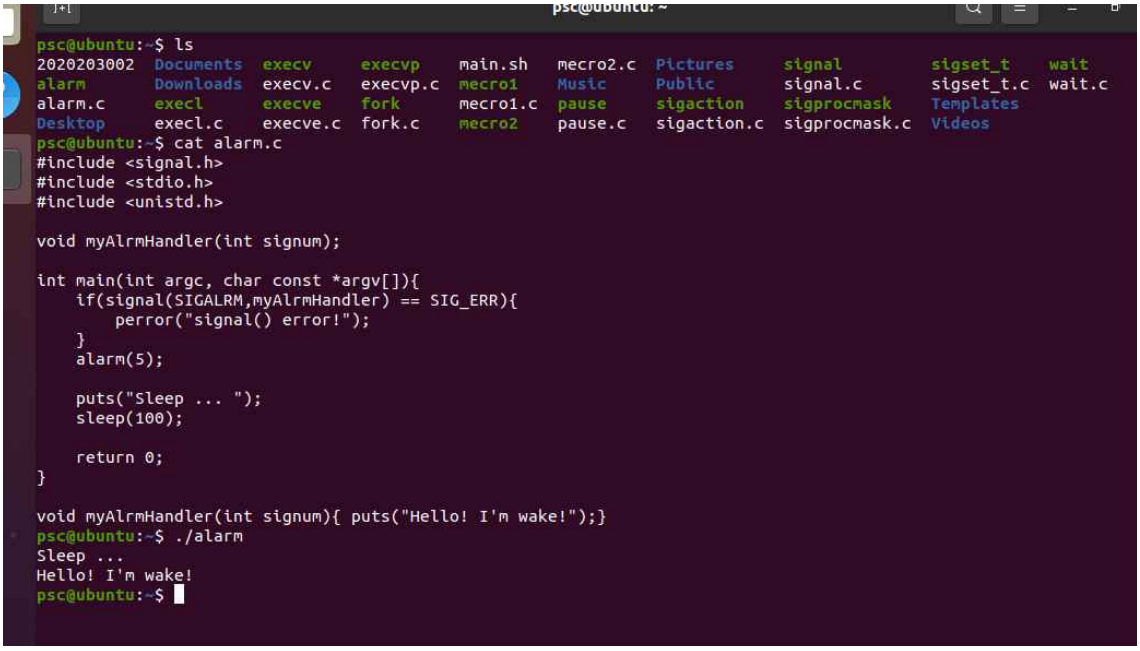
2020203002 박상천



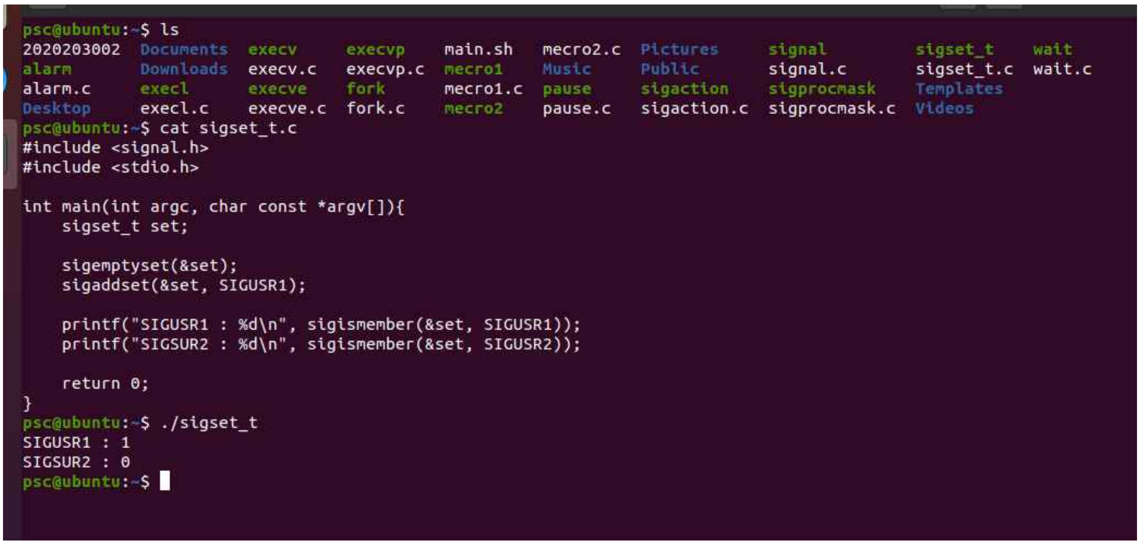
```
psc@ubuntu:~$ ls
2020203002  alarm.c  Documents  execl    execv    execve    execvp    fork    main.sh  mecro1.c  mecro2.c  pause
 Pictures  sigaction    signal    sigprocmask    sigset_t    Templates  wait
alarm        Desktop  Downloads  execl.c  execv.c  execve.c  execvp.c  fork.c  mecro1    mecro2     Music      pause.c
 Public     sigaction.c  signal.c  sigprocmask.c  sigset_t.c  Videos      wait.c
psc@ubuntu:~$ cat signal.c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

void mySigHandler(int signum);

int Cnt = 0;

int main(int argc, char const *argv[]){
    if(argc != 2){
        printf("Usage: %s [count]", argv[0]);
        exit(0);
    }

    if(signal(SIGUSR1, mySigHandler) == SIG_ERR){
        perror("signal() error!");
    }

    printf("Hello. My PID is : %d\n", getpid());

    Cnt = atoi(argv[1]);
    while(Cnt>0){
        printf("%d ... ",Cnt);
        fflush(stdout);
        Cnt -= 1;
        sleep(1);
    }

    return 0;
}

void mySigHandler(int signum){
    if(signum == SIGUSR1){
        Cnt = 3;
    }
}
psc@ubuntu:~$ ./signal
Usage: ./signal [count]psc@ubuntu:~$ ./signal 100
Hello. My PID is : 2373
100 ... 99 ... 98 ... 97 ... 96 ... 95 ... 94 ... ^C
psc@ubuntu:~$
```



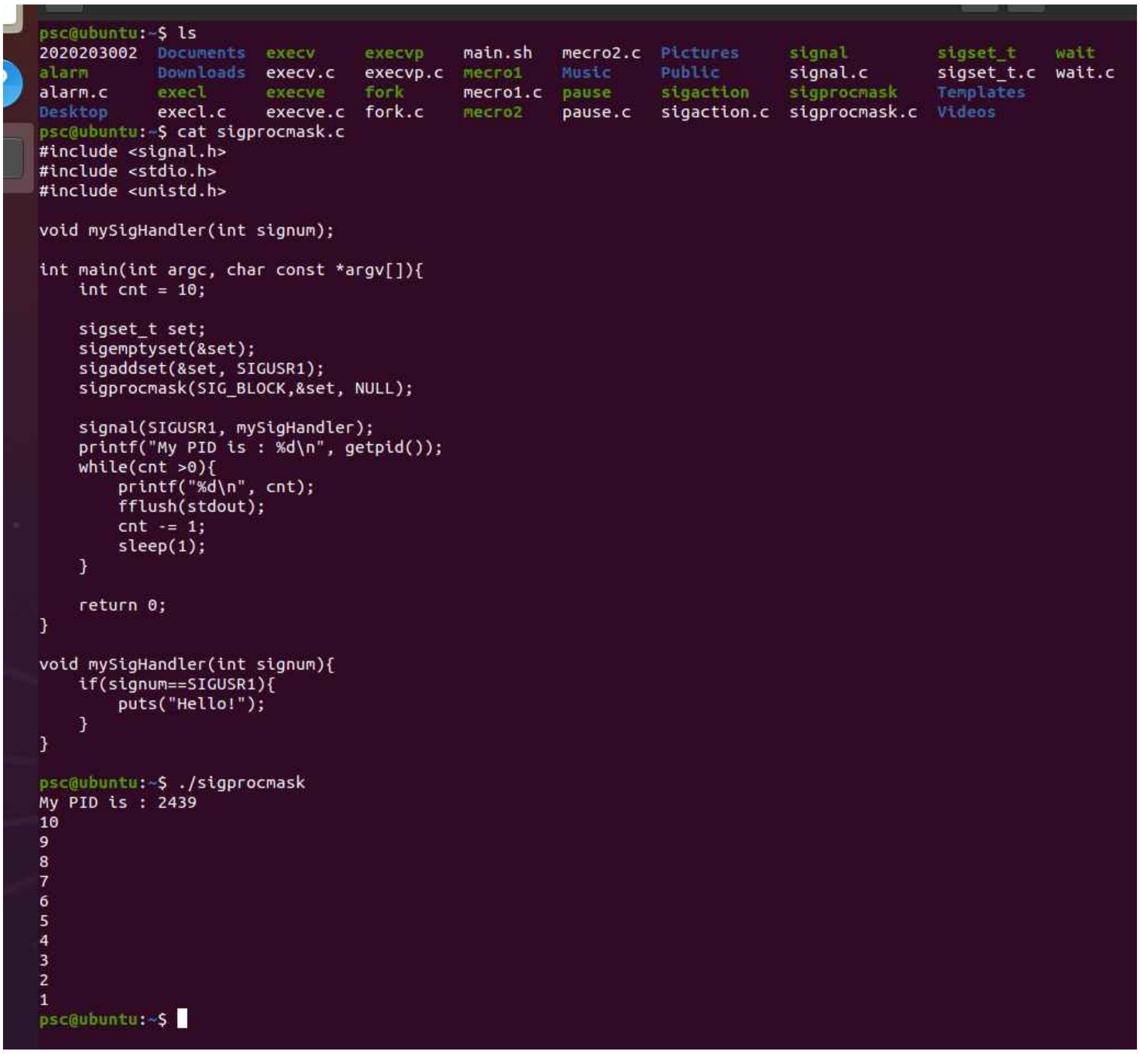
```
psc@ubuntu:~$ ls
2020203002  Documents  execv     execvp     main.sh   mecro2.c  Pictures      signal         sigset_t    wait
alarm       Downloads  execv.c   execvp.c   mecro1    Music     Public        signal.c       sigset_t.c  wait.c
alarm.c     execl      execve    fork       mecro1.c  pause     sigaction     sigprocmask    Templates
Desktop     execl.c    execve.c  fork.c     mecro2    pause.c   sigaction.c   sigprocmask.c  Videos
psc@ubuntu:~$ cat pause.c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void mySigHandler(int signum);

int main(int argc, char const *argv[]){
    if(signal(SIGINT,mySigHandler) == SIG_ERR){
        perror("signal() error!");
    }

    if(signal(SIGUSR1, mySigHandler) == SIG_ERR){
        perror("signal() error!");
    }

    printf("Hello. My PID is : %d\n", getpid());
    while(1) { puts("Pause ..."); pause();}

    return 0;
}

void mySigHandler(int signum){
    if(signum ==SIGINT){
        printf("Hello! SIGINT.\n");
    }else if(signum == SIGUSR1){
        printf("Hello! SIGUSR1.\n");
    }
}
psc@ubuntu:~$ ./pause
Hello. My PID is : 2405
Pause ...
^CHello! SIGINT.
Pause ...
^Z
[2]+  Stopped                 ./pause
psc@ubuntu:~$ 
```



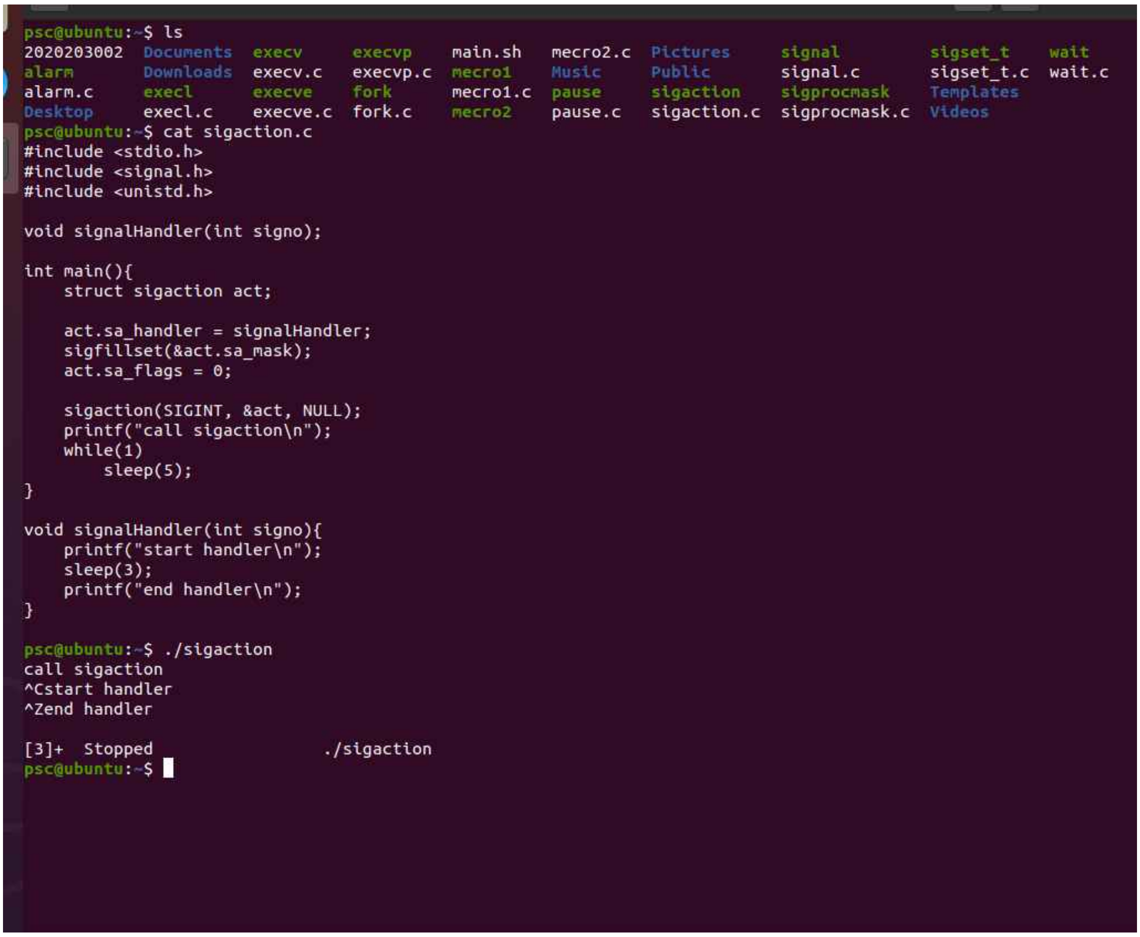
```
psc@ubuntu:~$ ls
2020203002  Documents  execv     execvp     main.sh   mecro2.c  Pictures     signal         sigset_t    wait
alarm       Downloads  execv.c   execvp.c   mecro1    Music     Public       signal.c       sigset_t.c  wait.c
alarm.c     execl      execve    fork       mecro1.c  pause     sigaction    sigprocmask    Templates
Desktop     execl.c    execve.c  fork.c     mecro2    pause.c   sigaction.c  sigprocmask.c  Videos
psc@ubuntu:~$ cat alarm.c
#include <signal.h>
#include <stdio.h>
#include <unistd.h>

void myAlrmHandler(int signum);

int main(int argc, char const *argv[]){
    if(signal(SIGALRM,myAlrmHandler) == SIG_ERR){
        perror("signal() error!");
    }
    alarm(5);

    puts("Sleep ... ");
    sleep(100);

    return 0;
}

void myAlrmHandler(int signum){ puts("Hello! I'm wake!");}
psc@ubuntu:~$ ./alarm
Sleep ...
Hello! I'm wake!
psc@ubuntu:~$ 
```



```
psc@ubuntu:~$ ls
2020203002  Documents  execv     execvp    main.sh   mecro2.c  Pictures     signal         sigset_t    wait
alarm       Downloads  execv.c   execvp.c  mecro1    Music     Public       signal.c       sigset_t.c  wait.c
alarm.c     execl      execve    fork      mecro1.c  pause     sigaction    sigprocmask    Templates
Desktop     execl.c    execve.c  fork.c    mecro2    pause.c   sigaction.c  sigprocmask.c  Videos
psc@ubuntu:~$ cat sigset_t.c
#include <signal.h>
#include <stdio.h>

int main(int argc, char const *argv[]){
    sigset_t set;

    sigemptyset(&set);
    sigaddset(&set, SIGUSR1);

    printf("SIGUSR1 : %d\n", sigismember(&set, SIGUSR1));
    printf("SIGSUR2 : %d\n", sigismember(&set, SIGUSR2));

    return 0;
}
psc@ubuntu:~$ ./sigset_t
SIGUSR1 : 1
SIGSUR2 : 0
psc@ubuntu:~$ 
```



```
psc@ubuntu:~$ ls
2020203002  Documents  execv     execvp    main.sh   mecro2.c  Pictures     signal         sigset_t    wait
alarm       Downloads  execv.c   execvp.c  mecro1    Music     Public       signal.c       sigset_t.c  wait.c
alarm.c     execl      execve    fork      mecro1.c  pause     sigaction    sigprocmask    Templates
Desktop     execl.c    execve.c  fork.c    mecro2    pause.c   sigaction.c  sigprocmask.c  Videos
psc@ubuntu:~$ cat sigprocmask.c
#include <signal.h>
#include <stdio.h>
#include <unistd.h>

void mySigHandler(int signum);

int main(int argc, char const *argv[]){
    int cnt = 10;

    sigset_t set;
    sigemptyset(&set);
    sigaddset(&set, SIGUSR1);
    sigprocmask(SIG_BLOCK,&set, NULL);

    signal(SIGUSR1, mySigHandler);
    printf("My PID is : %d\n", getpid());
    while(cnt >0){
        printf("%d\n", cnt);
        fflush(stdout);
        cnt -= 1;
        sleep(1);
    }

    return 0;
}

void mySigHandler(int signum){
    if(signum==SIGUSR1){
        puts("Hello!");
    }
}

psc@ubuntu:~$ ./sigprocmask
My PID is : 2439
10
9
8
7
6
5
4
3
2
1
psc@ubuntu:~$ 
```



```
psc@ubuntu:~$ ls
2020203002  Documents  execv     execvp    main.sh   mecro2.c  Pictures     signal         sigset_t    wait
alarm       Downloads  execv.c   execvp.c  mecro1    Music     Public       signal.c       sigset_t.c  wait.c
alarm.c     execl      execve    fork      mecro1.c  pause     sigaction    sigprocmask    Templates
Desktop     execl.c    execve.c  fork.c    mecro2    pause.c   sigaction.c  sigprocmask.c  Videos
psc@ubuntu:~$ cat sigaction.c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

void signalHandler(int signo);

int main(){
    struct sigaction act;

    act.sa_handler = signalHandler;
    sigfillset(&act.sa_mask);
    act.sa_flags = 0;

    sigaction(SIGINT, &act, NULL);
    printf("call sigaction\n");
    while(1)
        sleep(5);
}

void signalHandler(int signo){
    printf("start handler\n");
    sleep(3);
    printf("end handler\n");
}

psc@ubuntu:~$ ./sigaction
call sigaction
^Cstart handler
^Zend handler

[3]+  Stopped                 ./sigaction
psc@ubuntu:~$
```