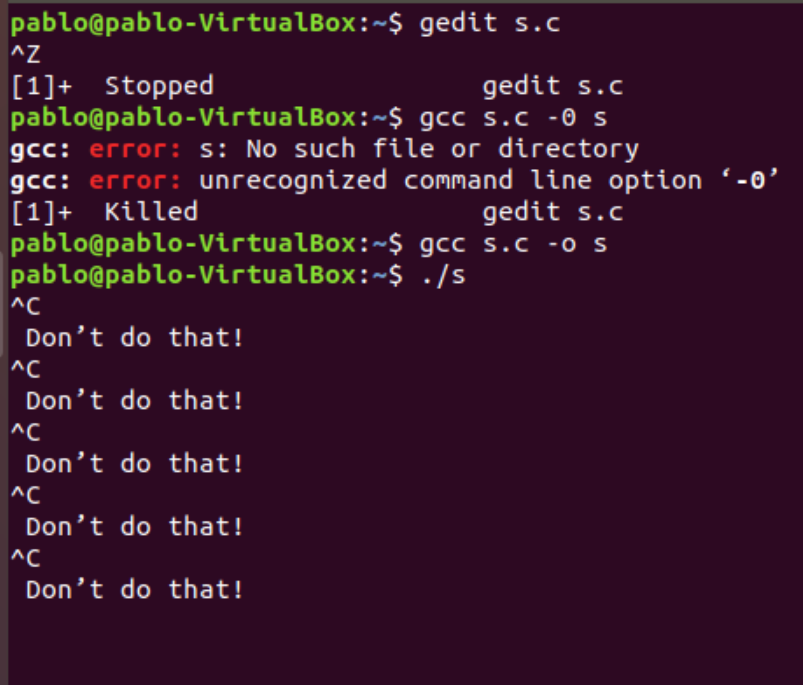
1. The code sigint\_example.c shows how to catch a signal in a C program. The main method registers the signal to be handled with the function signal(SIGNAL, handler), where SIGNAL is the signal to be handled and handler is the handler method which must be implemented, in this case only a message is printed (fflush forces a write of all buffered data). The code sigalrm\_example.c does something similar except this time the signal is sent explicitly using an alarm. Examine these codes and run them. (10 points)



2. Each signal has a unique identifier. In Linux, every signal has a name that begins with characters SIG. Find information about (at least) 3 different signals, look for their name, number and in what scenario they are used. (15 points)

| Nombre de señal | Nº | Acción | Propósito. Se envía al proceso cuando... |
| --- | --- | --- | --- |
| **SIGHUP** | 1 | Term | Cuelgue detectado en la terminal de control o muerte del proceso de control |
| **SIGQUIT** | 3 | Core | Terminación procedente del teclado |
| **SIGCHLD** | 20,17,18 | Ign | Proceso hijo terminado o parado |
| **SIGALRM** | 14 |  | ... cuando termina un temporizador. |

3. Write a program that creates a child process that will handle SIGINT, SIGUSR1 and SIGUSR2. The parent process should also handle the SIGINT function. Each handler function should print a message specifying the type of signal that was handled. The parent process must ask the user which signal to send and then send it using kill(), use sleep(1); in the parent process after sending each signal, to make your program execution easier to understand. (75 points)

Your program output should be something similar to this:

Child process created

Pick signal to send: 1(SIGINT) 2(SIGUSR1) 3(SIGUSR2) 4(Exit): 1

Parent process sending SIGINT

Child process received SIGINT 1

Parent process sending SIGINT

Child process received SIGINT 2

Parent process sending SIGUSR1

Child process received SIGUSR1 3

Parent process sending SIGUSR2

Child process received SIGUSR2 2

Parent process sending SIGUSR1

Child process received SIGUSR1 ^C

Parent process received SIGINT

Child process received SIGINT 4

Killing child process and exit

