Amos(Mohan) Li

Lab10 Signal

CPE2600-121

Prof. Turney

**Part 1: Signal Research**  
Research POSIX signals, be able to answer the following questions:  
• What is a signal disposition?

It is the action taken by a process when it receives a signal.

• What is a signal handler? What is it used for?

A user-defined function that is executed in response to receiving a specific signal, which override the default disposition.

• Name and describe each of the five (5) default dispositions?

**Terminate**: The process terminates immediately when the signal is received. Example: SIGTERM.

**Ignore**: The signal is ignored, and the process continues executing. Example: SIGCHLD.

**Stop**: The process is paused until resumed by a SIGCONT signal. Example: SIGSTOP.

**Continue**: A paused process resumes execution. Example: SIGCONT.

**Core Dump and Terminate**: The process terminates and generates a core dump file for debugging. Example: SIGSEGV.

• Name and describe one way to programmatically send a signal to a process. Be  
able to give an example (the code) to send a signal.

kill(pid, SIGUSR1); //send SIGUSR1 to the child process

• Name and describe one way to send a signal to a process from the command line.  
Be able to give an example (the command, key combination, etc.) to send a signal.

kill -SIGTERM <PID>  
Each signal has a corresponding type. Research POSIX signal types. For EACH of the  
following signal types:  
• **SIGINT** interrupt signal, typically sent when pressing Ctrl+C in the terminal Default Disposition: Terminates the process. The Disposition Can Be Overridden: this signal are often used to trigger specific behaviors in an application, such as reloading configuration files.

• **SIGTERM** termination signal, used to request a process to terminate gracefully terminal Default Disposition: Terminates the process. The Disposition Can Be Overridden.  
• **SIGUSR1** User-defined signal, reserved for application-specific purposes Default Disposition: Terminates the process. The Disposition Can Be Overridden.  
• **SIGKILL** kill signal, used to forcefully terminate a process, it terminates the process immediately, and the disposition cannot be overridden.  
• **SIGSTOP** this is the stop signal, used to pause a process. The disposition cannot be overridden.  
Be able to:  
• Name and describe the signal  
• Define the default disposition taken by the operating system if a process does not  
define a signal handler  
• Can the disposition be overridden by a signal handler? Why do you think this is the  
case?

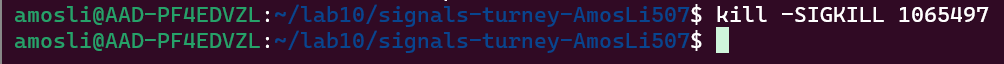
SIGKILl and SIGSTOP cannot be overridden because they are critical to the operating system’s ability to maintain control over all processes.

**Part 2: Working with a Signal Handler**

1. ***Ctrl + C*** to send the Sigint signal, type in ctrl + C send the signal when the program run to interrupt it
2. Use command ***kill -SIGINT <PID>***

Use ***ps aux | grep signal\_handler*** to get pid

1. If exit(1) is deleted in the code, we have to use command ***kill -SIGKILL <PID>*** to make the process exit, run it in another terminal



A screenshot of a computer

Description automatically generated

**Part 3: Signal Sent from the Operating System**

**Using SIGALRM**

A screen shot of a computer

Description automatically generated

**Handling SIGSEGV:**

a signal handler for SIGSEGV is added, and it print a message that a segmentation fault was received. This program retry the null pointer dereference, and leads to an infinite loop of signal handling.



**Part 3: Getting Details from a Received Signal**

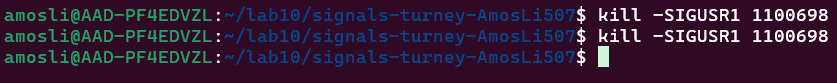
The SA\_SIGINFO flag in the sigaction structure is used to enable the signal handler to receive detailed information about the signal through the siginfo\_t structure.

Signal\_sigaction.c send the SIGUSER1 in another signal: by command

**kill -SIGUSR1 <PID>,** after receiving the signal, the pid of the signal sending process is printed.

A computer screen with numbers and text

Description automatically generated

****

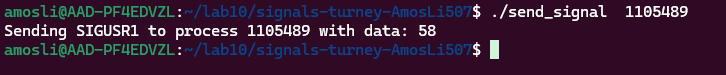
**Part 4 Sending Data with a Signal**

Send\_signal.c: This program sends the signal SIGUSER1 to the assigned PID. I random generated number is stored in the signal

Recv\_signal.c:  This program uses sigaction to register a handler for the SIGUSR1 signal, and it print out the random number data along with the signal SIGUSER.

**A screenshot of a computer

Description automatically generated**

****