Operating Systems Design Lab
Computer Engineering Department
Fall 2023/2024
Lab 6: Signals

Objectives

1. To understand signals and learn how to change their default behavior.

Prelab

1. Read the manual pages of the following systems calls:

```
sighandler_t signal(int signum, sighandler_t handler);
int kill(pid_t pid, int sig);
```

Experiment

1. Implement the following program and execute it. Then, open a new terminal window and send the signal SIGINT to the running program using the command (not system call) `kill` (read more about it using “`man 1 kill `”);

```
    void main(){
        printf("My PID is %d\n", getpid());
        while(1){
            printf("*\n");
            sleep(1);
        }
    }
```

   What happened to the running program?

2. Add the following function ”`void handler(int signo)`” to the program in the previous part, and modify the main function to look as shown below.

```
void handler(int signo){
    write(1, "Hello", 6);
}

void main(){
    signal(SIGINT, handler);
    printf("My PID is %d\n", getpid());
    while(1){
        printf("*\n");
        sleep(1);
    }
}
```

   Run the program and send SIGINT to it again. Is the result different from the previous case? why?

3. Replace “`signal(SIGINT, handler);`” in the previous part with “`signal(SIGINT, SIG_IGN);`” and repeat the same procedure. What was the reaction of the process to the SIGINT signal? Why?

4. Write a program that prints the numbers from 1 to 15 to the screen with a delay of 1 second between every pair of consecutive numbers. Your program should react as follows to signals:

   (a) It ignores SIGARLM

   (b) It stops printing numbers when receiving a SIGINT signal (but remains alive), and resumes the printing (from the value it stopped at) when receiving another SIGINT signal.

   (c) It restarts the printing from 1 again when it receives a SIGHUP signal.