CMPS 3600 Medel Antonio Lab 02

```
/* CMPS 3600 05
   * Medel Antonio
   * Lab 02

* * * */

finclude <stdio.h>
   * #include <stdib.h>

#include <signal.h>

// woid handler(int signum) // signal handler

fprintf("Hello World!\n");
   //exit(1); // exit after printing

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *

// *
```

Caption 1 Here is the signal_base.c that runs infinitely

```
[amedel@odin:~/public_html/cmps3600/lab02$ ./signal_base
Hello World!
Turing was right!
^C
amedel@odin:~/public_html/cmps3600/lab02$
```

Caption 2 Here is the output of the signal case.c that runs indefinitely

```
/*
 * CMPS 3600 OS
 * Medel Antonio
 * Lab 02
 *
 * */
#include <stdio.h>
#include <stdib.h>
#include <stgnal.h>
#include <unistd.h>
int loop = 0;

void handler(int signum) // signal handler
{
    printf("Hello World!\n");
    //exit(1); // exit after printing
}

void handler2(int sigint) // second handler
{
    printf("\nloop = %d \n", loop); // display loop count
    exit(1);
}

int main(int argc, char * argv[])
{
    signal(SIGINT, handler2);
    while(1)
    {
        signal(SIGINT, handler2);
        while(1);
        // schedule a SIGALRM for 1 second
        sleep(5);
        printf("Turing was right\n");
        |/ loop++; // increment loop
}

return 0; // new exit
}
```

Caption 3 Here is the signal_base.c that runs indefinitely and checks for the SIGINT signal (check for CTRL-C) to call handler2 and print the loop counter.

```
amedel@odin:~/public_html/cmps3600/lab02$ ./signal_base2
Hello World!
Turing was right!
^C
loop = 10
```

Caption 4 Output of signal_base.c that runs indefinitely AND has a check for CTRL-C to print loop counter.

In conclusion, I created two programs that run indefinitely and print something in response to a signal. signal_base or Medel_Antonio_Lab02-1 is just an infinite loop. signal_base2 or Medel_Antonio_Lab02-2 responds to a user signal (CTRL-C) and prints the loop counter after receiving that signal. I think seeing a program respond to user input better showed me what signals are and can do. Without signals programs could not respond to events happening outside the program.