**Grupo: CCOMP6-1**

**Profesor: Julio Santisteban**

Uso de la señalización entre procesos

1. Implementar un comando que permita enviar una señalización a un proceso.

$ kill -9 8823243

Donde :

-9 indica el tipo de señalización

8823243 es el process id

Para determinar el process id se debe consultar los procesos en ejecución:

$ ps -Af

# este comando muestra los procesos

$ ps -Af | egrep myprograma

# egrep es un comando que filtra el estándar input por filas y el filtro es el argumento

# como resultado solo mostrara las líneas que contengan el argumento, en este caso myprograma

# la segunda columna identifica el porcess id y la tercera columna el parent process id

~/shell$ ps -Af

UID PID PPID C STIME TTY TIME CMD

runner 1 0 0 18:26 ? 00:00:00 /nix/store/rxw1s8ykfzd10lrpc96rb-pid1/bin/pid1

runner 16 1 0 18:28 pts/0 00:00:00 /nix/store/8kgsjv5i4j-bash-interactive-5.1-p12/bi

runner 19 1 0 18:28 pts/1 00:00:00 /nix/store/8kgsjv5bash-interactive-5.1-p12/bi

runner 124 19 0 18:28 pts/1 00:00:00 ps -Af

1. Implementar un programa que pueda capturar una señalización

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <signal.h>

void sighandler(int);

int main () {

signal(SIGINT, sighandler);

while(1) {

printf("Going to sleep for a second...\n");

sleep(1);

}

return(0);

}

void sighandler(int signum) {

printf("Caught signal %d, coming out...\n", signum);

}

1. (Tarea) Implementar un programa que permita enviar una señalización a otro proceso. La señal y proceso deberá ser pasado como argumentos del programa.

Ejemplo del paso de argumentos al programa

#include <stdio.h>

main(argc, argv)  
int argc;  
char \*argv[];  
{

if(argc!=2) {

printf("Ha olvidado su nombre.\n");  
exit(1);

}  
printf("Hola %s", argv[1]);

}

Donde:

argc es el número de argumentos, el nombre del programa se considera parte de la lista de argumentos

argv es el vector de argumentos es decir es una matriz

Los argumentos que contienen un espacio deben ser pasados entre comillas, por ejemplo:

$ ./programa.exe “Marco Antonio”

Referencia:

<https://linuxhint.com/signal_handlers_c_programming_language/>

<http://courses.cms.caltech.edu/cs124/lectures-wi2016/CS124Lec15.pdf>

<https://www.thegeekstuff.com/2012/03/catch-signals-sample-c-code/>

c++

<http://www.yolinux.com/TUTORIALS/C++Signals.html>