Michael Osei

5/8/2023

COSC 430

Week 12 Assignment

Exercises:

1.

Code:

#include <stdio.h>

#include <unistd.h>

#include <signal.h>

void my\_signal\_interrupt(int sig)

{

printf("I got signal %d\n", sig);

(void) signal(SIGINT, SIG\_DFL);

}

int main()

{

(void) signal(SIGINT,my\_signal\_interrupt);

while(1) {

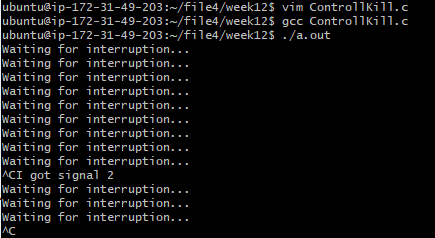
printf("Waiting for interruption...\n");

sleep(1);

}

}

Result:



#2

Code:

#include <fcntl.h>

#include <stdio.h>

#include <unistd.h>

int main(void)

{

pid\_t pid;

printf("Fork here.\n");

pid = fork();

if (pid == -1)

{

*// If fork returns -1, there was an error!*

return (1);

}

printf("\nFork successful!\n");

if (pid == 0)

{

*// Fork's return value is 0:*

*// we are now in the child process*

printf("Child: I'm the child, my internal pid is %d.\n", pid);

}

else if (pid > 0)

{

*// Fork's return value is not 0*

*// which means we are in the parent process*

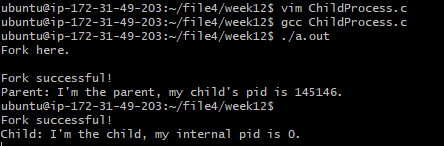
printf("Parent: I'm the parent, my child's pid is %d.\n", pid);

}

return(0);

}

Result:



#3

Code:

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

*// Child process routine:*

void child\_routine(pid\_t pid, int \*nb)

{

printf("Child: Hi! I'm the child. PID received from fork = %d\n", pid);

printf("Child: The number is %d\n", \*nb);

}

*// Parent process routine:*

void parent\_routine(pid\_t pid, int \*nb)

{

printf("Parent: I'm the parent. PID received from fork = %d\n", pid);

printf("Parent: The number is %d\n", \*nb);

\*nb \*= 2;

printf("Parent: The modified number is %d\n", \*nb);

}

int main(void)

{

pid\_t pid; *// Stores the return value of fork*

int nb; *// Stores an integer*

nb = 42;

printf("Before fork, the number is %d\n", nb);

pid = fork(); *// Creating the child process*

if (pid == -1)

return (EXIT\_FAILURE);

else if (pid == 0) *// The pid is 0, this is the child process*

child\_routine(pid, &nb);

else if (pid > 0) *// The pid is > 0, this is the parent*

parent\_routine(pid, &nb);

return (EXIT\_SUCCESS);

}

Result:

