JEGYZŐKÖNYV

Operációs rendszerek BSc

2021. tavasz féléves feladat

Készítette: **Kércsi Bence**

Programtervező Informatikus

Neptunkód: **ILVIYV**

**A feladat leírása:**

29. Írjon C nyelvű programot, ami SIGUSR1 szignál érkezésekor növeli eggyel a változó értékét, SIGUSR2 szignál érkezésekor pedig kiírja a képernyőre a változó aktuális értékét.

A másik program pedig: futtatáskor szignál-t küld az adott processznek.

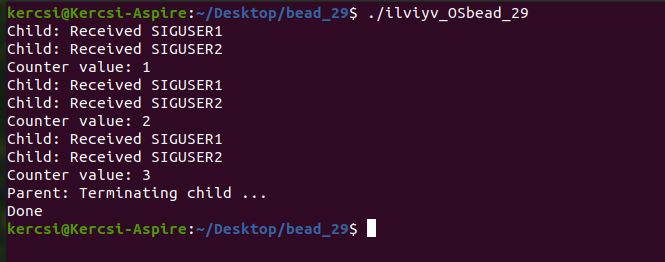
**A feladat elkészítésének lépései:**

Tanár urat megkérdeztem emailbe, és azt a választ kaptam, hogy ha a feladatot egy C programkóddal, de 2 processzként (szülő és gyermek) írom meg, az akkor helyes megoldása a feladatnak.

Ezt figyelembe véve meg született az eredmény. A kódnak 2 fő része és feladatköre van, az egyik a szülő, aminek a feladata majd a szignálok generálása, a másik a gyermek, aki addig vár, amig a szülő nem küldd neki szignált.

A szignál kezelést és 2 külön függvény helyett egybe raktam, és a függvényen belül vizsgálom switch-el, hogy melyik szignált kapta. Azt is ellenőriztem először, hogy a szignált használhatjuk-e.

**A futtatás eredménye:**



**A C kód:**

**#include <signal.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <sys/types.h>**

**#include <unistd.h>**

**#include<sys/wait.h>**

**static void sig\_handler(int);**

**int main ()**

**{**

**int i, parent\_pid, child\_pid, status;**

**if(signal(SIGUSR1, sig\_handler) == SIG\_ERR)**

**printf("Parent: Unable to create handler for SIGUSR1\n");**

**if(signal(SIGUSR2, sig\_handler) == SIG\_ERR)**

**printf("Parent: Unable to create handler for SIGUSR2\n");**

**parent\_pid = getpid();**

**if((child\_pid = fork()) == 0) {**

**for (;;) pause();**

**}**

**else {**

**int run=0;**

**do{**

**kill(child\_pid, SIGUSR1);**

**sleep(3);**

**kill(child\_pid, SIGUSR2);**

**sleep(3);**

**run++;**

**}while(run<3);**

**printf("Parent: Terminating child ...\n");**

**kill(child\_pid, SIGTERM);**

**wait(&status);**

**printf("Done\n");**

**}**

**}**

**static void sig\_handler(int signo) {**

**static int counter=0;**

**switch(signo) {**

**case SIGUSR1: /\* Incoming SIGUSR1 \*/**

**printf("Child: Received SIGUSER1\n");**

**counter++;**

**break;**

**case SIGUSR2: /\* Incoming SIGUSR2 \*/**

**printf("Child: Received SIGUSER2\n");**

**printf("Counter value: %d\n",counter);**

**break;**

**default: break;**

**}**

**return;**

**}**