

Jegyzőkönyv

Operációs rendszerek BSc
2021. tavaszi féléves feladat.

Készítette: Csősz Péter

Neptunkód: NOR38

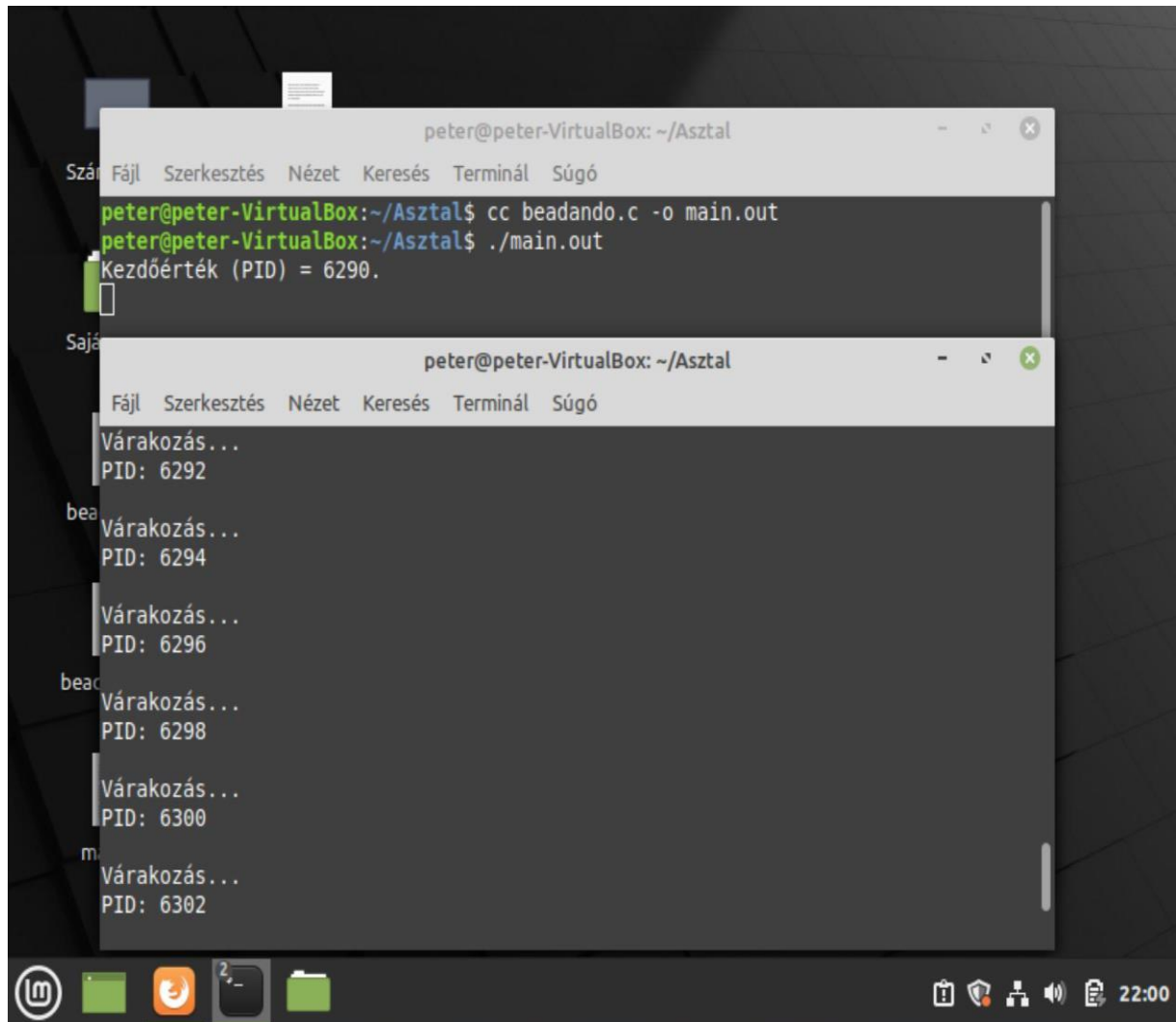
A feladat leírása:

"28. Írjon C nyelvű programokat, ami létrehoz egy üzenetsort ebbe az üzenetsorba betesz egy másodpercenként egy változót ennek a változonak a kezdőértéke legyen az aktuális processz azonosítója a változó mindig beírás után nő

A feladat elkészítésének lépései:

- Header állományok inkludálása.
- Kettő program elkészítése. Az elsőben betesz egy változót másodpercenként, amit megnövel kettővel. A kezdőérték a processz azonosítója. A második program pedig kiolvassa..
- Hiba ellenőrzést elvégzi
- Megírtam az íráshoz/olvasáshoz szükséges kódokat.
- Futtatás:

A futtatás eredménye:



```
peter@peter-VirtualBox: ~/Asztal
Szám Fájl Szerkesztés Nézet Keresés Terminál Súgó
peter@peter-VirtualBox:~/Asztal$ cc beadando.c -o main.out
peter@peter-VirtualBox:~/Asztal$ ./main.out
Kezdőérték (PID) = 6290.
Várakozás...
PID: 6292
Várakozás...
PID: 6294
Várakozás...
PID: 6296
Várakozás...
PID: 6298
Várakozás...
PID: 6300
Várakozás...
PID: 6302
```

A programkódok:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/msg.h>
#include <unistd.h>

#define MSGSZ 255
#define KULCS 10

typedef struct msgbuf
{
    long pid;
    char mtext[MSGSZ+1];
} message_buf;

int main()
{
    message_buf sbuf;
    sbuf.pid = getpid();

    printf("Kezdőérték (PID) = %ld.\nKüldés folyamatban...", sbuf.pid);

    key_t key = KULCS; // Kulcs inicializálás
    int mask = 0644 | IPC_CREAT; // Üzenetsor létrehozása
    int msgid = msgget(KULCS, mask);
    if (msgid < 0)
    {
        fprintf(stderr, "Nem sikerült a kulcs létrehozása.%d\n", KULCS);
        exit(EXIT_FAILURE);
    }

    do
    {
        sleep(1);
        sbuf.pid = sbuf.pid + 2;
        msgsnd(msgid, &sbuf, strlen(sbuf.mtext)+1, 1);
    }while(1);

    return 0;
}
```

```

#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/msg.h>

#define MSGSZ 255
#define KULCS 10

typedef struct msgbuf
{
    long mtype;

} message_buf;

static void kezelo(int);

int main()
{
    key_t key = KULCS;
    int msgflg = 0644;
    message_buf rbuf;

    int msqid = msgget(KULCS, msgflg);

    if (msqid < 0)
    {
        perror("msgget() hiba!\n");
        exit(1);
    }

    signal(SIGINT, kezelo);

    rbuf.mtype = 1;
    for ( ; ; )
    {
        printf("Várakozás...\n");
        ssize_t byteok = msgrcv(msqid, &rbuf, MSGSZ, 0, 0);
        if (byteok < 0)
        {
            perror("msgrcv() hiba!\n");
            exit(1);
        }
        printf("PID: %ld\n\n", rbuf.mtype);
    }

    return 0;
}

static void kezelo(int sig)
{
    signal(sig, SIG_IGN);
    int mask, msgid;
    key_t key = KULCS;
    mask = 0644;
    msgid = msgget(KULCS, mask);

```

```
if (msgid == -1)
{
    printf("Az üzenetsor nem létezik!\n");
    exit(EXIT_SUCCESS);
}

if (msgctl(msgid, IPC_RMID, NULL) == -1)
{
    fprintf(stderr, "Az üzenetsort nem sikerült törölni.\n");
    exit(EXIT_FAILURE);
}
else
    printf("\nAz üzenetsor sikeresen törölve.\n");
exit(0);
}
```