

JEGYZŐKÖNYV

Operációs rendszerek BSc
2021 tavasz féléves feladat

Készítette: **Szeszák Ádám**
Neptunkód: **AZCTJJ**

A feladat leírása:

Írjon C nyelvű programokat, ami:

- létrehoz egy üzenetsort
- ebbe az üzenetsorba betesz egy másodpercenként egy változót
- ennek a változónak a kezdőértéke legyen az aktuális processz azonosítója
- a változó mindig beírás után növekedjen 2-vel
- a másik program pedig: kiolvas

A feladat elkészítésének lépései:

A feladat elkészítéséhez két programra lesz szükség, egy amelyik az üzeneteket küldi, a másik pedig amelyik kiolvassa, a végén pedig eltávolítja az üzenetsort. A feladathoz készítettem egy header file-t is, melyben egy struktúrát definiálok az üzenet létrehozásához illetve az üzenet hosszát és számát is ebben határozom meg.

Első körben a küldő programot fogom bemutatni. A programkód:

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

void report_and_exit(const char* msg) {
    perror(msg);
    exit(-1); /* Hiba eseten kilep */
}

int main() {
    int value;
    key_t key = ftok(PathName, ProjectId); /* kulcs generalasa */
    if (key < 0) report_and_exit("nem kaphato kulcsot...");

    int qid = msgget(key, 0666 | IPC_CREAT); /* generalt kulcs hasznalata queue */
    if (qid < 0) report_and_exit("nem kaphato queue id...");
    value = qid;
    int i;
    for (i = 0; i < MsgCount; i++) {
        /* uzenetsor generalasa, queuedMessage a queue.h fileban definialt struk */
        queuedMessage msg;
        msg.value = value;
        msg.id = i;

        /* uzenet kuldes */
        msgsnd(qid, &msg, sizeof(msg), IPC_NOWAIT); /* ha az uzenetsor tele, nem */
        printf("%i uzenet elkuldve, ertek: %i\n", (int) msg.id, (int) msg.value);
        value = value + 2;
        sleep(1);
    }
    return 0;
}
```

Első lépésben egy kulcsot generálunk a queue id létrehozásához. Ez azért szükséges, mert ha egy szabadon választott számot használnánk, meg van a lehetőség arra, hogy összeütközésbe kerülünk egy más, nem ehhez kapcsolódó programmal.

Következő lépésben létrehozunk egy üzenetsort, ahol a 0666 kapcsoló a hozzáférési jogért felelős a memóriaszegmenshez, míg az IPC_CREAT a rendszernek utasítást ad, hogy készítsen egy új memóriaszegmenst az osztott memóriának.

Ez után a for cikluson belül létrehozom a küldendő üzenetet. Itt a ciklus előre megadott értékig megy, ezt végtelen ciklusra cserélve a program futása leállításig történik. A küldendő üzenet a header fájlban létrehozott struktúra, ahol a value az aktuális processz értékével kezdődik, az id pedig a küldött üzenet sorszáma.

Miután létrehoztam a küldendő üzenetet, az msgsnd használatával küldöm, ahol az IPC_NOWAIT kapcsolót használtam arra az esetre, ha tele lenne az üzenetsor ne várakozzon.

Miután elküldtem az üzenetet, kiíratom az elküldött üzenet tartalmát és értékét, majd növelem a változó értékét 2-vel és a program vár 1 másodpercet, majd küldi a következő üzenetet.

Ez után az üzenet fogadására szolgáló programra volt szükség, amely kódja:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdlib.h>
#include "queue.h"

void report_and_exit(const char* msg) {
    perror(msg);
    exit(-1); /* hiba eseten kilep */
}

int main() {
    key_t key= ftok(PathName, ProjectId); /* kulcs az uzenetsor azonositasara */
    if (key < 0) report_and_exit("nem kaphato kulcs...");

    int qid = msgget(key, 0666 | IPC_CREAT); /* eleri a qid-t, ha mar elkeszites */
    if (qid < 0) report_and_exit("nincs hozzaferes a sorhoz...");

    int i;
    for (i = 0; i < MsgCount; i++) {
        queuedMessage msg;
        if (msgrcv(qid, &msg, sizeof(msg), 0, MSG_NOERROR | IPC_NOWAIT) < 0)
            puts("msgrcv problema...");
        printf("%i uzenet megkapva, ertek: %i\n", (int) msg.id, (int) msg.value);
        sleep(1);
    }

    /** uzenetsor eltavolitasa */
    if (msgctl(qid, IPC_RMID, NULL) < 0) /* NULL = 'no flags' */
        report_and_exit("problema az uzenetsor eltavolitasaval...");

    return 0;
}
```

Ebben az esetben is szükség van kulcs generálására, hogy azonosítani tudjuk az üzenetsort.

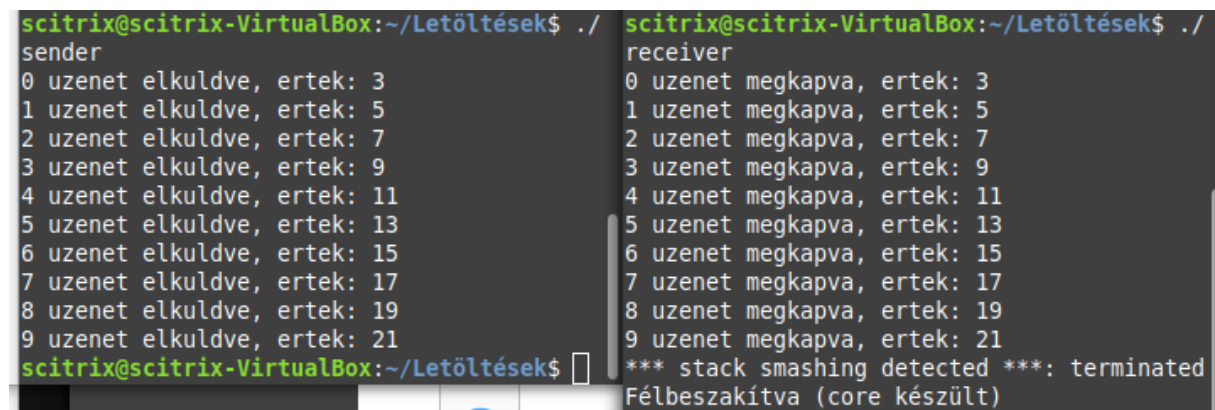
A fogadó program nem készít üzenetsort, így megtévesztő lehet az IPC_CREAT kapcsoló, de ez azt jelenti ebben az esetben, hogy „ha szükséges elkészíti, egyébként hozzáfér”.

Míg a küldő program az msgsnd hívást használja a küldéshez, addig a fogadó program a fogadásra az msgrcv hívást. Itt a 0 kapcsoló az üzenetek fogadását érkezési sorrendben hajtja végre, az MSG_NOERROR kapcsoló arra szolgál, hogyha a kapott adat mérete nagyobb mint a megadott érték(msgsz), ne térjen vissza hibával, az IPC_NOWAIT pedig azért, hogy ha nem érhető el üzenet, ne várjon rá, hanem térjen vissza azonnal.

Ez után kiíratom az üzenetben kapott értékeket, majd itt is várok 1 másodpercet, ugyanis a következő üzenet akkor fog érkezni.

A fogadó program végén, miután megkapta az összes üzenetet szükséges eltávolítani az üzenetsort az msgctl hívással.

A futtatás eredménye:



```
scitrix@scitrix-VirtualBox:~/Letöltések$ ./sender
0 uzenet elkuldve, ertek: 3
1 uzenet elkuldve, ertek: 5
2 uzenet elkuldve, ertek: 7
3 uzenet elkuldve, ertek: 9
4 uzenet elkuldve, ertek: 11
5 uzenet elkuldve, ertek: 13
6 uzenet elkuldve, ertek: 15
7 uzenet elkuldve, ertek: 17
8 uzenet elkuldve, ertek: 19
9 uzenet elkuldve, ertek: 21
scitrix@scitrix-VirtualBox:~/Letöltések$

scitrix@scitrix-VirtualBox:~/Letöltések$ ./receiver
0 uzenet megkapva, ertek: 3
1 uzenet megkapva, ertek: 5
2 uzenet megkapva, ertek: 7
3 uzenet megkapva, ertek: 9
4 uzenet megkapva, ertek: 11
5 uzenet megkapva, ertek: 13
6 uzenet megkapva, ertek: 15
7 uzenet megkapva, ertek: 17
8 uzenet megkapva, ertek: 19
9 uzenet megkapva, ertek: 21
*** stack smashing detected ***: terminated
Félbeszakítva (core készült)
```

Itt látható, hogy első körben a futtató programot indítottam el, ahol ki is írja az elküldött értéket, majd a másik terminal ablakban a fogadó program futása látható, ahol a kapott értéket és az üzenet sorszámát is kiírja. Miután megkapta az összes üzenetet, törli az üzenetsort és kilép.