

Operációs rendszerek BSc

11. Gyak.

2022. 05. 01.

Készítette:

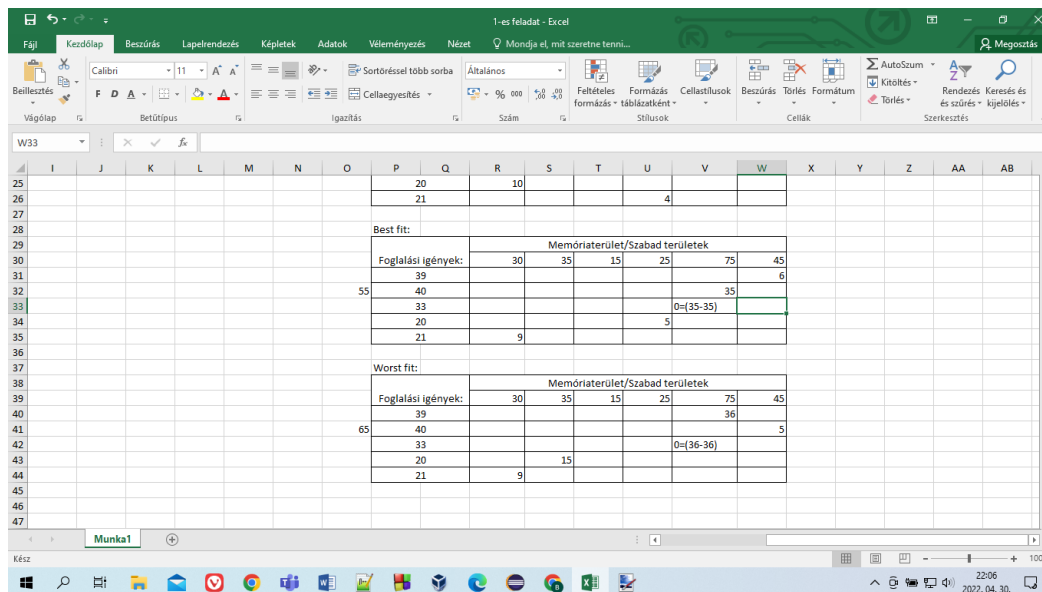
Petró Balázs Bsc
Mérnökinformatikus
FO71M2

Miskolc, 2022

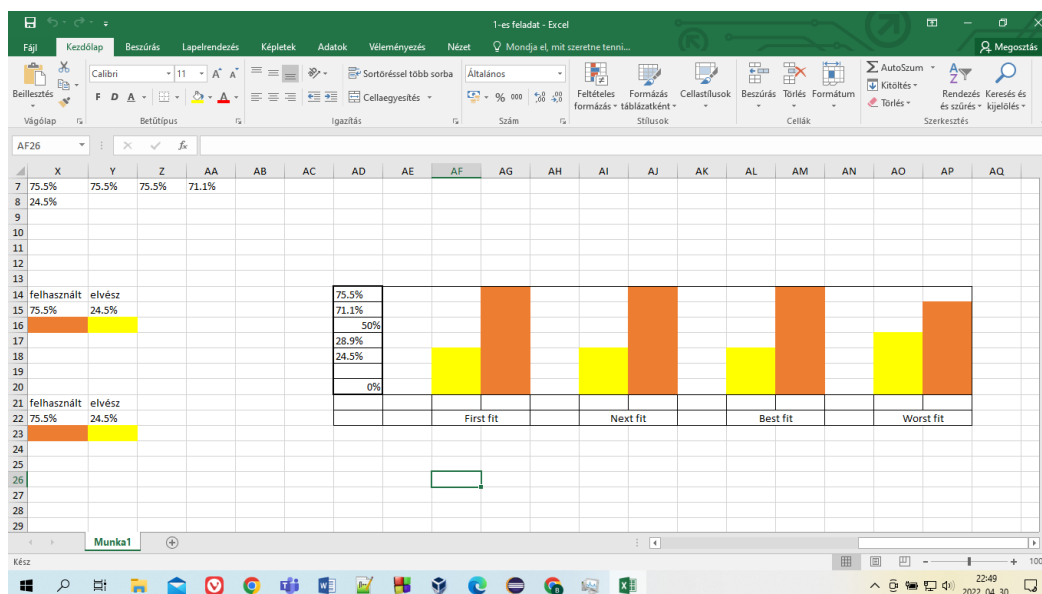
Feladatok

1. . Adott egy rendszer (foglalási stratégiák), melyben a következő : Szabad területek : 30k, 35k, 15k, 25k, 75k, 45k és a Foglalási igények: 39k, 40k, 33k, 20k, 21k állnak rendelkezésre. A rendszerben a memória 4 kbyte-os blokkokban kerül nyilvántartásra, ennél kisebb méretű töredék igény esetén a teljes blokk lefoglalásra kerül.
 - Határozza meg változó méretű partíció esetén a következő algoritmusok felhasználásával: first fit, next fit, best fit, worst fit a foglalási igényeknek megfelelő helyfoglalást – táblázatos formában (az ea. bemutatott mintafeladat alapján)!
 - Hasonlítsa össze, hogy a teljes szabad memóriaterület hány százaléka vész el átlagosan az egyes algoritmusok esetén!
 - A kapott eredményeket ábrázolja oszlop diagrammal!
 - Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

	P	Q	R	S	T	U	V	W
10	First fit:							
11	Memóriaterület/Szabad területek							
12	Foglalási igények:	30	35	15	25	75	45	
13	39					36=(75-39)	5=(45-40)	
14	40							
15	33			0=(35-35)				
16	20		10=(30-20)					
17	21				4=(25-21)			
18	Next fit:							
19	Memóriaterület/Szabad területek							
20	Foglalási igények:	30	35	15	25	75	45	
21	39						6	
22	40					35		
23	33			0				
24	20		10					
25	21				4			
26	Best fit:							
27	Memóriaterület/Szabad területek							
28	Foglalási igények:	30	35	15	25	75	45	
29	39							
30	40					35		



Oszlop Diagrammal ábrázolni a százalékokat(narancs a felhasznált, míg citrom a maradék ami elvész az egyes algoritmusok esetén , azaz átlagosan 3x 24,5% és csak a worst fitnél 28,9% vész el/azaz átlagosan számolva $(73,5 + 28,9)/4 = 25,6\%$):



Indoklás: Ebben az esetben mindegyik képes alokálni a memóriát. A worst fit a tapasztalatok szerint a legrosszabb

(átlagosan a teljes allokálható memória fele elveszik), míg a másik három átlagosan hasonló teljesítményű (a teljes allokálható memória 1/3-a veszik el). Ez itt picivel kevesebb lett (a három azonos veszteségű), de a worst fit használatával még így is többet veszítünk.

Módszerek az eredmények javítására: 1. Lapszervezés használata. Hátránya, hogy kell hozzá MMU támogatás.

2. Ha lennének nem kielégíthető igények, akkor ez esetben szemétgyűjtés (garbage collection) alkalmazása, vagyis a memória allokáció futás idejű átrendezése nem kielégíthető igények esetén. Hátránya, hogy erőforrás igényes.

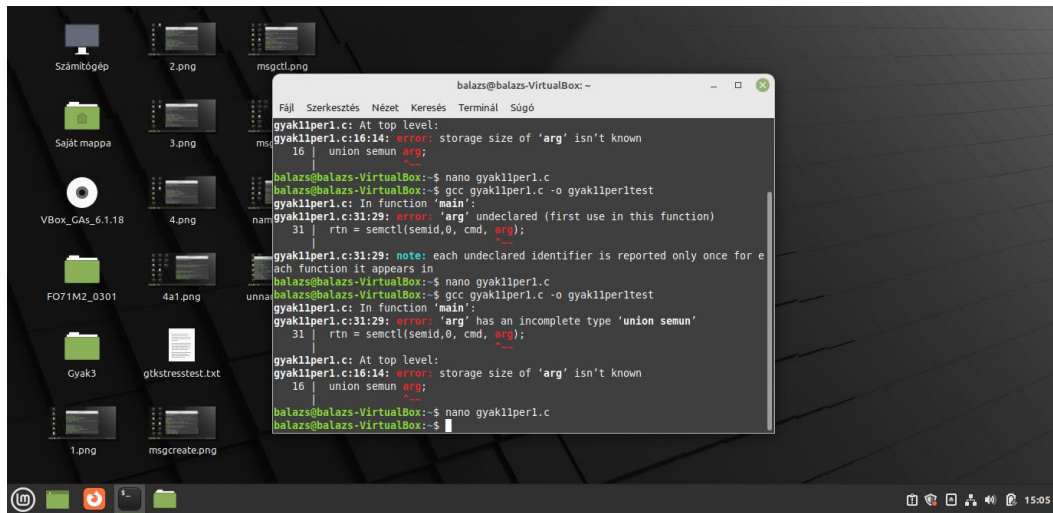
2. Gyakorló feladat: A feladat megoldásához először tanulmányozza Vadász Dénes: Operációs rendszer jegyzet, a témához kapcsolódó fejezetét (6.4)., azaz Írjon C nyelvű programokat, ahol

- kreál/azonosít szemafor készletet, benne N szemafor-t. A kezdő értéket 0-ra állítja – semset.c,
- kérdezze le és írja ki a pillanatnyi szemafor értéket – semval.c
- szüntesse meg a példácskák szemafor készletét – semkill.c
- sembuf.sem_op=1 értékkel inkrementálja a szemafort – semup.c

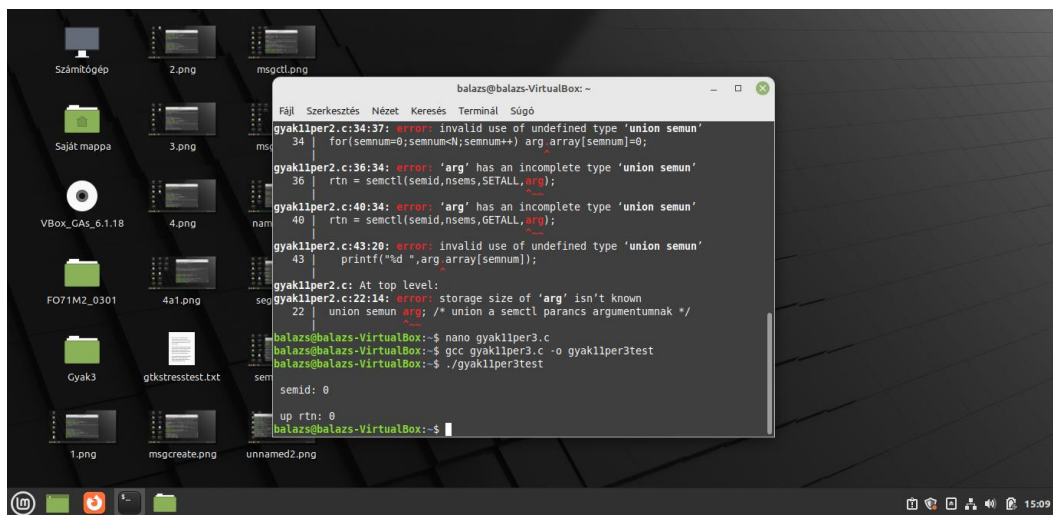
A futtatás eredményét is tartalmazza a jegyzőkönyv.

A futtatás eredmények:

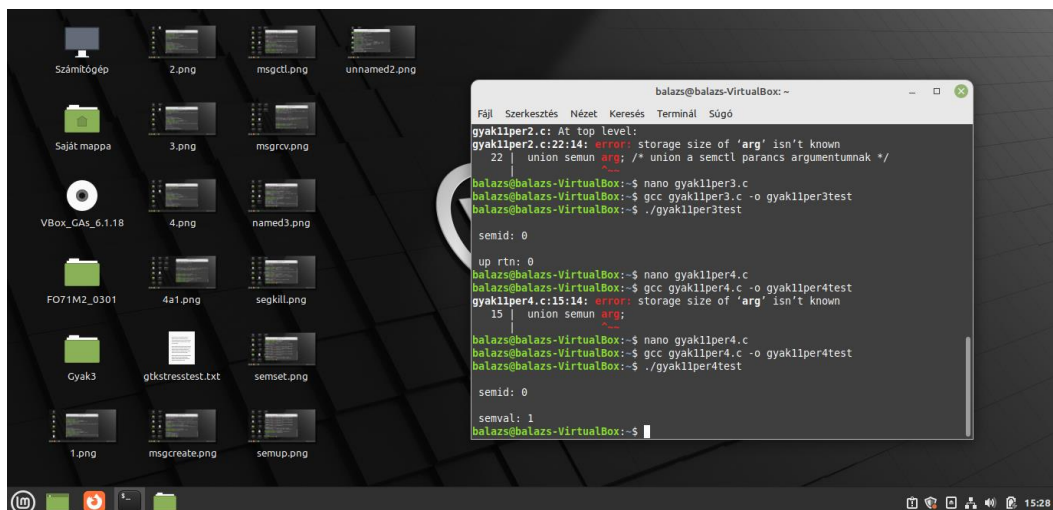
semkill.c



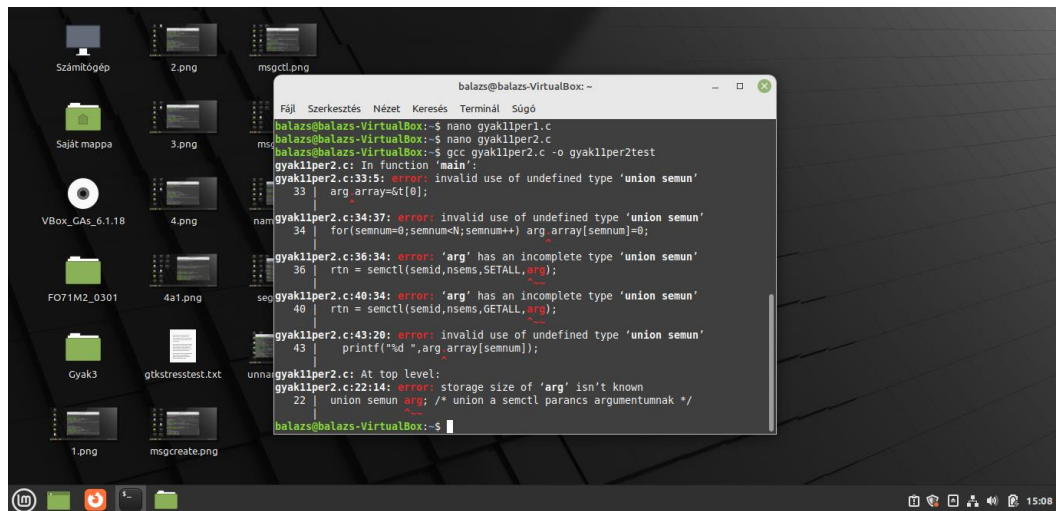
semup.c



semval.c

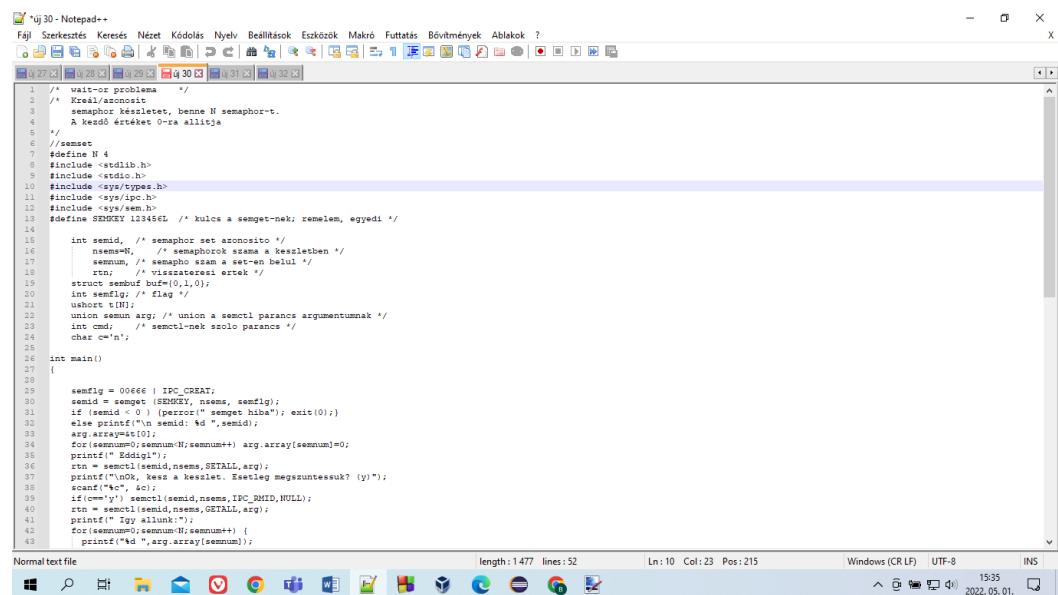


semset.c



```
balazs@balazs-VirtualBox: ~  
Fájl Szerkesztés Nézet Keresés Terminál Sütő  
balazs@balazs-VirtualBox:~$ nano gyak1lper1.c  
balazs@balazs-VirtualBox:~$ nano gyak1lper2.c  
balazs@balazs-VirtualBox:~$ gcc gyak1lper2.c -o gyak1lper2test  
gyak1lper2.c: In function 'main':  
gyak1lper2.c:33:5: error: invalid use of undefined type 'union semun'  
33 |   arg_array=t[0];  
   |   ^  
gyak1lper2.c:34:37: error: invalid use of undefined type 'union semun'  
34 |   for(semnum=0;semnum<N;semnum++) arg_array[semnum]=0;  
   |                                     ^  
gyak1lper2.c:36:34: error: 'arg' has an incomplete type 'union semun'  
36 |   rtn = semctl(semid,nsems,SETALL,arg);  
   |                                ^  
gyak1lper2.c:40:34: error: 'arg' has an incomplete type 'union semun'  
40 |   rtn = semctl(semid,nsems,GETALL,arg);  
   |                                ^  
gyak1lper2.c:43:20: error: invalid use of undefined type 'union semun'  
43 |   printf("d ",arg_array[semnum]);  
   |                   ^  
gyak1lper2.c: At top level:  
gyak1lper2.c:22:14: error: storage size of 'arg' isn't known  
22 |   union semun arg; /* union a semctl parancs argumentumnak */  
   |                   ^  
balazs@balazs-VirtualBox:~$
```

kódjaik:



```
/* wait-or problema */  
/* Kérel/azonosít  
semaphor készletet, benne N semaphor-t.  
A kezdő értéket 0-ra állítja  
*/  
//semset  
#define N 4  
#include <stdlib.h>  
#include <stdio.h>  
#include <sys/types.h>  
#include <sys/ipc.h>  
#include <sys/sem.h>  
#define SEMKEY 1234567 /* Kulcs a semset-nek: remelem, egyedi */  
  
int semid, /* semaphor set azonosító */  
nsems, /* semaphorok száma a készletben */  
semnum, /* semapho szám a set-en belül */  
rtn; /* visszatérési érték */  
struct sembuf buf[5],0);  
int semflg; /* flag */  
uhort t[N];  
union semun arg; /* union a semctl parancs argumentumnak */  
int cmd; /* semctl-nek szolo parancs */  
char c='n';  
  
int main()  
{  
    semflg = 00666 | IPC_CREAT;  
    semid = semget (SEMKEY, nsems, semflg);  
    if (semid < 0 ) { perror(" semget hiba"); exit(0);}  
    else printf("\n semid: %d ",semid);  
    arg_array=t[0];  
    for(semnum=0;semnum<N;semnum++) arg_array[semnum]=0;  
    printf(" Eddig!");  
    rtn = semctl(semid,nsems,SETALL,arg);  
    printf("\nOK, kész a készlet. Esetleg megszüntessük? (y)");  
    scanf("%c", &c);  
    if (c=='y') semctl(semid,nsems,IPC_RMID,HULL);  
    rtn = semctl(semid,nsems,GETALL,arg);  
    printf(" Így állunk:");  
    for(semnum=0;semnum<N;semnum++) {  
        printf("%d ",arg_array[semnum]);  
    }  
}
```

```
*új 30 - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

16 nsema=0; /* semaphorek száma a készletben */
17 semnum; /* semapho szám a set-en belül */
18 rtn; /* visszatérési érték */
19 struct sembuf {0,1,0};
20 int semflg; /* flag */
21 ushort t[8];
22 union semun arg; /* union a semctl parancs argumentumnak */
23 int cmd; /* semctl-nek szolo parancs */
24 char c="n";
25
26 int main()
27 {
28
29     semflg = 00666 | IPC_CREAT;
30     semid = semget (SEMKEY, nsema, semflg);
31     if (semid < 0 ) {perror(" semget hiba"); exit(0);}
32     else printf("\n semid: %d ",semid);
33     arg.array=t[0];
34     for(semum=0;semum=N;semum++) arg.array[semum]=0;
35     printf(" Eddig:");
36     rtn = semctl(semid,nsema,SETALL,arg);
37     printf("\n00, tess a készlet. Esetleg megszüntessuk? (y)");
38     scanf("%c", &c);
39     if(c=='y') semctl(semid,nsema,IPC_RMID,NULL);
40     rtn = semctl(semid,nsema,GETALL,arg);
41     printf(" Így allunk:");
42     for(semum=0;semum=N;semum++) {
43         printf("%d ",arg.array[semum]);
44     }
45     /* semop(semid, sbuf, 1);
46     printf(" Húsznapin"): rtn = semctl(semid,nsema,GETALL,arg);
47     for(semum=0;semum=N;semum++) {
48         printf(" Így allunk: %d ",arg.array[semum]);
49     }
50     */
51 }
52

Normal text file length: 1477 lines: 52 Ln: 10 Col: 23 Pos: 215 Windows (CR LF) UTF-8 INS 15:36 2022.05.01.
```

```
*új 29 - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

1 /* semkill.c */
2 /* Megsünteti a példacskák semaphore-át.
3 Deklarációk magyarázatát lásd a semget.c-ben.
4 */
5 #include <stdlib.h>
6 #include <stdio.h>
7 #include <sys/types.h>
8 #include <sys/ipc.h>
9 #include <sys/sem.h>
10 #define SEMKEY 123456L /* kulcs a semget-nek; remelem, egyedi */
11
12
13 int semid,nsema,rtn;
14 int semflg;
15 struct sembuf sembuf, 'sop;
16 union semun arg;
17 int cmd;
18
19 int main()
20 {
21
22
23     nsema = 1;
24     semflg = 00666 | IPC_CREAT;
25     semid = semget (SEMKEY, nsema, semflg);
26     if (semid < 0 ) {perror(" semget hiba"); exit(0);}
27     else printf("\n semid: %d ",semid);
28     printf("\n");
29
30     cmd = IPC_RMID; /* Ez a megszüntetes parancsa */
31     rtn = semctl(semid,0,cmd,arg);
32     printf("\n kill rtn: %d ",rtn);
33     printf("\n");
34
35
36
37 }

Normal text file length: 756 lines: 37 Ln: 20 Col: 2 Pos: 412 Windows (CR LF) UTF-8 INS 15:37 2022.05.01.
```

```

1 /* semup.c */
2 /* Ez is futtatható program, a sembuf.sem_op=1 értékkel
3    inkrementálja a semaphore-t.
4    A deklarációt a semset.c-ben magyarázzuk, ill. a
5    beleértett file-okból nézhetők! */
6
7 #include <stdlib.h>
8 #include <stdio.h>
9 #include <sys/types.h>
10 #include <sys/ipc.h>
11 #include <sys/sem.h>
12 #define SEMKEY 123456L /* kulcs a semget-nek; remelem, egyedi */
13
14 int semid,nsems,rtn;
15 unsigned nsops; /* semup-ban ezzel adjuk meg, hogy semaphore
16    struktúrával foglalkozzon */
17 int semflg;
18 struct sembuf sembuf, *sop;
19
20
21 int main()
22 {
23
24     nsems = 1; /* Egy semaphore legyen */
25     semflg = 0666 | IPC_CREAT;
26     semid = semget (SEMKEY, nsems, semflg);
27     if (semid < 0) {perror(" semget hiba"); exit(0);}
28     else printf("\n semid: %d ",semid);
29     printf ("\n");
30
31     nsops = 1; /* Egy operacio van */
32     sembuf.sem_num = 0; /* A 0-ik semaphore-ral foglalkozunk */
33     sembuf.sem_op = 1; /* Inkrementációt kerünk! */
34     sembuf.sem_flg = 0666; /* Flag beállítás */
35     sop = &sembuf; /* Így kéri a semop az argumentumot */
36     rtn = semop(semid, sop, nsops);
37     /* 0-val visszatérő semop sikeres. */
38     printf("\n up rtn: %d ",rtn);
39     printf("\n");
40
41 }
42
43

```

Normal text file length: 1168 lines: 43 Ln: 20 Col: 1 Pos: 547 Windows (CR LF) UTF-8 INS 15:34 2022.05.01.

```

1 /* semval.c */
2 /* Lekérdi és kiírja a pillanatnyi semaphore értéket.
3    Változók értelmezését lásd a semset.c-ben */
4 #include <stdlib.h>
5 #include <stdio.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/sem.h>
9 #define SEMKEY 123456L /* kulcs a semget-nek; remelem, egyedi */
10
11 int semid,nsems,rtn;
12 int semflg;
13 struct sembuf sembuf, *sop;
14 union semun arg;
15 int cmd;
16
17 int main()
18 {
19
20     nsems = 1;
21     semflg = 0666 | IPC_CREAT;
22     semid = semget (SEMKEY, nsems, semflg);
23     if (semid < 0) {perror(" semget hiba"); exit(0);}
24     else printf("\n semid: %d ",semid);
25     printf ("\n");
26
27     cmd = GETVAL; /* E parancsra a semctl visszatér a current
28     semaphore értékhez. Itt az rtn-be. */
29     rtn = semctl(semid,0, cmd, NULL);
30
31     printf("\n semval: %d ",rtn);
32     printf("\n");
33
34 }
35
36
37

```

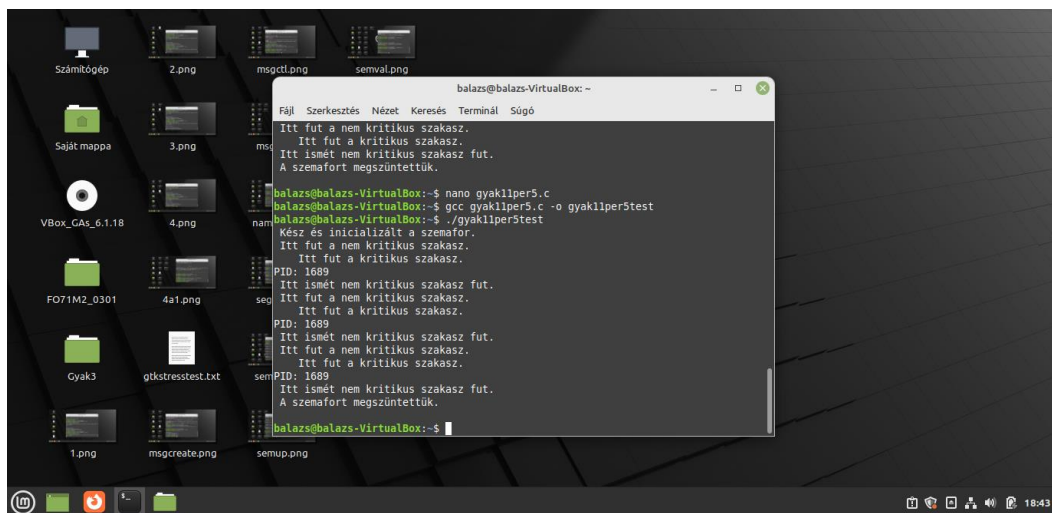
Normal text file length: 819 lines: 37 Ln: 14 Col: 32 Pos: 376 Windows (CR LF) UTF-8 INS 15:33 2022.05.01.

2a. Írjon egy C nyelvű programot, melyben

- egyik processz létrehozza a szemafor (egyetlen elemi szemafor; inicializálja 1-re, vagy x-re, ha még nem létezik),
- másik processz használja a szemafor, belépési szakasz (down), a kritikus szakaszban alszik 2-3 sec-et, m pid-et kiír, kilépési szakasz (up), ezt ismételve 2x-3x (és a hallgató egyszerre indítson el 2-3 ilyen processzt),
- harmadik processzben, ha létezik a szemafor, akkor megszünteti”.

Mentés: gyak11_2.c A futtatás eredményét is tartalmazza a jegyzőkönyv.

Futtatás eredménye:

A screenshot of a Linux desktop environment. The desktop background is dark with a grid of icons. A terminal window is open in the center, displaying the output of a C program. The output shows the program running in a non-critical section, entering a critical section, sleeping for 2-3 seconds, and then exiting. The program is run multiple times, and the output shows the PID of each run. The terminal window title is 'balazs@balazs-VirtualBox: ~'. The desktop icons include 'Számítógép', 'Saját mappa', 'VBox_GAs_6.1.18', 'FO71M2_0301', 'Gyak3', '1.png', '2.png', '3.png', '4.png', '4a1.png', 'msgctf.png', 'msgcreate.png', 'semval.png', 'semup.png', and 'gtkstress.txt'. The system tray at the bottom shows the time as 18:43.

Kódja:

```
C:\Users\Balazs\OneDrive\Aztallal gépi\FO71M2_0426\gyak11_2.c - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

#include "mysem.h"
#include <stdlib.h>
#include <unistd.h>

int main() {
    int id; /* A szemafor azonosítója */

    if ((id = semget(KEY, 1, 0)) < 0) { /* Még nem létezik. */
        if ((id = semget(KEY, 1, PERM | IPC_CREAT)) < 0) {
            perror("A szemafor nem nyitható meg.");
            exit(-1);
        }
    }

    else {
        perror("Már létezik a szemafor.");
        exit(0);
    }

    if (semctl(id, 0, SETVAL, 1) < 0)
        perror("Nem lehetett inicializálni.");
    else puts("Kész és inicializált a szemafor.");

    //exit(0);
}
```

C source file length: 1432 lines: 111 Ln: 111 Col: 2 Pos: 1433 Windows (CR LF) UTF-8 INS 18:45 2022.05.01.

```
C:\Users\Balazs\OneDrive\Aztallal gépi\FO71M2_0426\gyak11_2.c - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

//exit(0);

struct sembuf up[] = {0, 1, SEM_UNDO};
struct sembuf down[] = {0, -1, SEM_UNDO};

if ((id = semget(KEY, 1, 0)) < 0) {
    perror("Még nem létezik a szemafor.");
    exit(-1);
}

for (int i=0; i<3; i++){
    puts("Itt fut a nem kritikus szakasz.");

    semop(id, down, 1); /* A belépési szakasz */
    puts("Itt fut a kritikus szakasz.");
    sleep(3);
    printf("PID: %d\n", getpid());
    semop(id, up, 1); /* Ez a kilépési szakasz */

    puts("Itt ismét nem kritikus szakasz fut.");
}

//exit(0);
}
```

C source file length: 1432 lines: 111 Ln: 111 Col: 2 Pos: 1433 Windows (CR LF) UTF-8 INS 18:46 2022.05.01.

```
C:\Users\Balazs\OneDrive\Asztali gép\F071M2_0426\gyak11_2.c - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

gyak11_2.c

70
71     sleep(1);
72
73     printf("PID: %d\n", getpid());
74
75     semop(&id, up, 1); /* Ez a kilépési szakasz */
76
77
78     puts("Itt ismét nem kritikus szakasz fut.");
79
80
81 }
82
83 //exit(0);
84
85
86
87 if ((id = semget(KEY, 1, 0)) < 0) {
88
89     perror("Már nem létezik a semafor. ");
90
91     exit(-1);
92
93 }
94
95
96
97 if (semctl(id, 0, IPC_RMID, 0) < 0) {
98
99     perror("Nem sikerült törölni.");
100
101     exit(-1);
102
103 }
104
105
106
107 puts("A semafor megsemmisültük. \n");
108
109 exit(0);
110
111

length: 1432 lines: 111 Ln: 111 Col: 2 Pos: 1433 Windows (CR LF) UTF-8 INS
18:47
2022.05.01.
```