Operációs rendszerek BSc

10. Gyak.

2022. 04. 13.

Készítette:

Závodszki Máté

Mérnökinformatikus

B2C7B0

Miskolc, 2022

Feladatok

1, Az előadáson bemutatott mintaprogram alapján készítse el a következő feladatot

	MAX. IGÉNY				FOGLAL			KIELÉGÍTETLEN IGÉNYEK			
	R1	R2	R3		R1	R2	R3	R1	R2	R3	
P0	7	5	3		0	3	0	7	2	3	
P1	3	2	2		2	0	0	1	2	2	
P2	9	0	2		3	0	2	6	0	0	
P3	2	2	2		2	1	1	0	1	1	
P4	4	3	3		0	0	2	4	3	1	
				foglaltak	7	4	5	készlet-giény			
				összesen	10	5	7	R1	R2	R3	
				Szabad er	3	1	2	-5	1	-2	--
								1	3	0	Runnable
								-4	2	-1	--
								2	0	-1	--
								-2	3	0	--

	MAX. IGÉNY				FOGLAL			KIELÉGÍTETLEN IGÉNYEK			
	R1	R2	R3		R1	R2	R3	R1	R2	R3	
P0	7	5	3		0	1	0	7	4	3	
P1	3	2	2		3	0	2	0	2	0	
P2	9	0	2		3	0	2	6	0	0	
P3	2	2	2		2	1	1	0	1	1	
P4	4	3	3		0	0	2	4	3	1	
				foglaltak	8	2	7	készlet-giény			
				összesen	10	5	7	R1	R2	R3	
				Szabad er	2	3	0	-5	1	0	--
								2	3	0	Runnable
								-4	2	-1	--
								2	0	-1	--
								-2	3	0	--

	MAX. IGÉNY				FOGLAL			KIELÉGÍTETLEN IGÉNYEK			
	R1	R2	R3		R1	R2	R3	R1	R2	R3	
P0	7	5	3		0	1	0	7	4	3	
P1	3	2	2		2	0	0	1	2	2	
P2	9	0	2		3	0	2	6	0	0	
P3	2	2	2		2	1	1	0	1	1	
P4	4	3	3		3	3	2	1	0	1	
				foglaltak	10	5	5	készlet-giény			
				összesen	10	5	7	R1	R2	R3	
				Szabad er	0	0	2	-5	1	-2	--
								1	3	0	Runnable
								-4	2	-1	--
								2	3	-1	--
								1	3	0	Runnable

2, Írjon három C nyelvű programot, ahol készít egy üzenetsort és ebbe két üzenetet tesz bele – msgcreate.c, majd olvassa ki az üzenetet - msgrcv.c, majd szüntesse meg az üzenetsort (takarít) - msgctl.c.

main.c

```c
    int msgid;
    key_t key;
    int msgflg;
    int rtn, msgsz;

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflg);
    if ( msgid == -1) {
                perror("\n Az msgget rendszerhivás sikertelen!");
                exit(-1);
        }
    printf("\n Az msgid %d, %x : ", msgid, msgid);




    msgp = &sndbuf;
    msgp->mtype = 1;
    strcpy(msgp->mtext,"Az egyik üzenet");
    msgsz = strlen(msgp->mtext) + 1;


    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);
    printf("\n Az 1. msgsnd visszaadott %d-t", rtn);
    printf("\n A kiküldött üzenet: %s", msgp->mtext);


    strcpy(msgp->mtext,"Másik üzenet");
    msgsz = strlen(msgp->mtext) + 1;
    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);
    printf("\n A 2.  msgsnd visszaadott %d-t", rtn);
    printf("\n A kiküldött üzenet: %s \n", msgp->mtext);

    exit (0);
}
```

main.c

```c
    char mtext[512];
} rcvbuf, *msgp;

struct msqid_ds ds, *buf;

int main()
{
    int msgid;
    key_t kulcs;
    int tipus, msgflg;
    int rtn, meret;

    kulcs = MSGKEY;
    msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

    msgid = msgget(kulcs, msgflg);
    if ( msgid == -1) {
                perror("\n Az msgget rendszerhivás sikertelen!");
                exit(-1);
        }
    printf("\n Az üzenet ID: %d",msgid);

    msgp = &rcvbuf;
    buf = &ds;
    meret = 20;
    tipus = 0;
    rtn = msgctl(msgid,IPC_STAT,buf);
    printf("\n Az uzenetek szama: %ld",buf->msg_qnum);

    while (buf->msg_qnum) {


    rtn = msgrcv(msgid,(struct msgbuf *)msgp, meret, tipus, msgflg);
    printf("\n Visszatérés: %d, A fogadott üzenet:%s\n",rtn, msgp->mtext);
    rtn = msgctl(msgid,IPC_STAT,buf);

}
    exit (0);
```

main.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdlib.h>
#define MSGKEY 654321L

int main()
{
    int msgid, msgflg,  rtn;
    key_t kulcs;
    kulcs = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget(kulcs, msgflg);

    rtn = msgctl(msgid, IPC_RMID, NULL);
    printf ("\n Visszatért: %d", rtn);

    exit (0);
}
```

3, Írjon egy C nyelvű programot, melyben az egyik processz létrehozza az üzenetsort, és szövegeket küld bele, exit üzenetre kilép, másik processzben lehet választani a feladatok közül: üzenetek darabszámának lekérdezése, 1 üzenet kiolvasása, összes üzenet kiolvasása, üzenetsor megszüntetése, kilépés.

*main.c

```c
struct msqid_ds ds, *buf;




int main()
{
    int msgid
    key_t key;
    int msgflg;
    int rtn, msgsz;

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflg);
    if ( msgid == -1) {
                perror("\n Az msgget rendszerhívás sikertelen!");
                exit(-1);
        }


    msgp = &sndbuf;
    msgp->mtype = 1;
    strcpy(msgp->mtext,"Az egyik üzenet");
    msgsz = strlen(msgp->mtext) + 1;

    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);

    strcpy(msgp->mtext,"Másik üzenet");
    msgsz = strlen(msgp->mtext) + 1;
    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);

if (msgp->mtext == "exit"){
    exit(0);
}


    key_t kulcs;
```

4, Gyakorló feladat:

main.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdlib.h>
#define SHMKEY 13

int main()
{
  int shmid;
  key_t kulcs;
  int meret = 512;
  int flag;

  kulcs = SHMKEY;
  flag = 0;
  if((shmid = shmget(kulcs, meret, flag)) < 0)
  {
     flag = 00666 | IPC_CREAT;
     if((shmid = shmget(kulcs, meret, flag)) < 0)
     {
    perror("\n Sikertelen volt az shmget!\n");
    exit(-1);
     }
  }
  printf("Az azonosító: %d\n", shmid);
  return 0;
}
```

*main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define SHMKEY 13

int main()
{
   int shmid;
   key_t kulcs;
   int meret = 512;
   int flag, rtn, sm;
   struct shmid_ds shmid_ds, *buf;

   buf = &shmid_ds;
   kulcs = SHMKEY;
   flag = 0;

   if((shmid = shmget(kulcs, meret, flag)) < 0)
   {
      perror("\n Sikertelen volt az shmget!\n");
      exit(-1);
   }

rtn = shmctl(shmid, IPC_STAT, buf);
printf("\n Méret: %ld", buf->shm_segsz);
printf("\n PID: %d\n", buf->shm_lpid);


   exit(0);
}
```

*main.c

```c
{
   int shmid;
   key_t kulcs;
   int meret = 512;
   int flag;

   struct str {
      int lng;
      char text[512-sizeof(int)];
   } *segment;

   kulcs = SHMKEY;
   flag = 0;

   if((shmid = shmget(kulcs, meret, flag)) < 0)
   {
      perror("\n Sikertelen volt az shmget!\n");
      exit(-1);
   }

   flag = 00666 | SHM_RND;
   segment = (struct str *)shmat(shmid, NULL, flag);

   if(segment == (void *) - 1)
   {
      perror("\n Sikertelen csatlakozás!");
      exit(-1);
   }

   if(strlen(segment->text) > 0)
   {
      printf("\n A régi szöveg : %s", segment->text);
   }

    printf("\n Adja meg az új szöveget : ");
    scanf("%s", segment->text);
    printf("\n\n Az új szöveg : %s \n", segment->text);
```