İşletim Sistemleri Uygulama 6

Unix'de semafor işlemleri

Bilgisayar Mühendisliği

İstanbul Teknik Üniversitesi
34469 Maslak, İstanbul

March 23, 2011

Bugün

İşletim Sistemleri Uygulama 6

Paylaşılan Bellek Alanları

Örnekler

PBA Veri Yapısı

```
1   struct shmid_ds {
        struct ipc_perm shm_perm;    /* Ownership and permissions */
        size_t          shm_segsz;   /* Size of segment (bytes) */
        time_t          shm_atime;   /* Last attach time */
        time_t          shm_dtime;   /* Last detach time */
6       time_t          shm_ctime;   /* Last change time */
        pid_t           shm_cpid;    /* PID of creator */
        pid_t           shm_lpid;    /* PID of last shmat(2)/shmdt(2) */
        shmatt_t        shm_nattch;  /* No. of current attaches */
        ...
11  };
```

PBA Sistem Çağrıları

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>

int shmget(key_t key, size_t size, int shmflg);

int shmctl(int shmid, int cmd, struct shmid_ds *buf);

void *shmat(int shmid, const void *shmaddr, int shmflg);
int shmdt(const void *shmaddr);
```

Örnek 1

```
#define _GNU_SOURCE
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/wait.h>
#include <sys/sem.h>
#include <sys/shm.h>
#include <unistd.h>
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define KEYSEM ftok(strcat(get_current_dir_name(),argv[0]),1)
#define KEYSEM2 ftok(strcat(get_current_dir_name(),argv[0]),2)
#define KEYSHM ftok(strcat(get_current_dir_name(),argv[0]),3)
```

Örnek 1

```
    void mysignal(void){}

    void sem_signal(int semid, int val){
4           struct sembuf semafor;
            semafor.sem_num=0;
            semafor.sem_op=val;
            semafor.sem_flg=1;
            semop(semid, &semafor,1);
9   }

    void sem_wait(int semid, int val) {
            struct sembuf semafor;
            semafor.sem_num=0;
14          semafor.sem_op=(-1*val);
            semafor.sem_flg=1;
            semop(semid, &semafor,1);
    }

19  void mysigset(int num){
            struct sigaction mysigaction;
            mysigaction.sa_handler=(void *)mysignal;
            mysigaction.sa_flags=0;
            sigaction(num,&mysigaction,NULL);
24  }
```

Örnek 1

```
1   int main (int argc, char *argv[])
    {
            int i, localcp, f, cocuk[2];

            int sonsem=0,
6               kilit=0,
                siram=0,
                shmid=0,
                *globalcp=NULL;

11          mysigset(12);

            for (i=0; i<2; i++){
                    f=fork();

16                  if (f==-1){
                            printf("FORK hata....\n");
                            exit(1);
                    }

21                  if (f==0)
                            break;

                    cocuk[i]=f;
            }
```

Örnek 1

```
    if (f!=0){
            sonsem=semget(KEYSEM2, 1, 0700|IPC_CREAT);
            semctl(sonsem, 0, SETVAL,0);

 5          kilit=semget(KEYSEM,1,0700|IPC_CREAT);
            semctl(kilit,0,SETVAL,1);

            shmid=shmget(KEYSHM, sizeof(int),0700|IPC_CREAT);

10          globalcp=(int *)shmat(shmid,0,0);
            *globalcp=0;
            shmdt(globalcp);
            sleep(2);

15          printf("anne kaynaklari yaratti ve cocuklari baslatacak.\n");
            for (i=0; i<2; i++)
                    kill(cocuk[i],12);

            sem_wait(sonsem,2);
20          printf("Cocuklarin isi bitti\n");

            semctl(sonsem,0,IPC_RMID,0);
            semctl(kilit,0,IPC_RMID,0);
            shmctl(shmid,IPC_RMID,0);

            exit(0);          }
```

Örnek 1

```
    else{
            siram=i;
            pause();
 4          kilit=semget(KEYSEM,1,0);
            sonsem=semget(KEYSEM2,1,0);
            shmid=shmget(KEYSHM, sizeof(int),0);
            globalcp=(int *) shmat(shmid,0,0);
            printf("cocuk %d basliyor ....\n",siram);
 9          for (i=0; i<5; i++){
                    sem_wait(kilit,1);
                    printf(" cocuk %d: degeri %d buldum i:%d\n",
                                    siram, *globalcp, i);
                    localcp=*globalcp;
14                  sleep(1);
                    localcp+= i;
                    *globalcp=localcp;
                    printf(" cocuk %d: yeni degeri %d yaptim i:%d\n",
                                    siram, *globalcp, i);
19                  sem_signal(kilit,1);
                    sleep(1);
            }
            shmdt(globalcp);
            sem_signal(sonsem,1);
            exit(0);
    }
    return 0;           }
```

Örnek 1 Çıktısı

```
anne kaynaklari yaratti ve cocuklari baslatacak.
cocuk 0 basliyor ....
cocuk 1 basliyor ....
 cocuk 0:  degeri 0 buldum i:0
 cocuk 0:  yeni degeri 0 yaptim i:0
 cocuk 1:  degeri 0 buldum i:0
 cocuk 1:  yeni degeri 0 yaptim i:0
 cocuk 0:  degeri 0 buldum i:1
 cocuk 0:  yeni degeri 1 yaptim i:1
 cocuk 1:  degeri 1 buldum i:1
 cocuk 1:  yeni degeri 2 yaptim i:1
 cocuk 0:  degeri 2 buldum i:2
 cocuk 0:  yeni degeri 4 yaptim i:2
 cocuk 1:  degeri 4 buldum i:2
 cocuk 1:  yeni degeri 6 yaptim i:2
 cocuk 0:  degeri 6 buldum i:3
 cocuk 0:  yeni degeri 9 yaptim i:3
 cocuk 1:  degeri 9 buldum i:3
 cocuk 1:  yeni degeri 12 yaptim i:3
 cocuk 0:  degeri 12 buldum i:4
 cocuk 0:  yeni degeri 16 yaptim i:4
 cocuk 1:  degeri 16 buldum i:4
 cocuk 1:  yeni degeri 20 yaptim i:4
Cocuklarin isi bitti
```

Örnek 2

```
#include <stdio.h>
#include <sys/shm.h>
#include <sys/stat.h>

int main (){
        int segment_id;
        char* shared_memory;
        struct shmid_ds shmbuffer;
        int segment_size;
        const int shared_segment_size = 0x6400;

        /*ortak bellek bolgesini al*/
        segment_id = shmget(IPC_PRIVATE, shared_segment_size,
                            IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR);

        /* PBA baglanti kur*/
        shared_memory = (char*) shmat(segment_id, 0, 0);
        printf("PBA baglanti adresi %p\n", shared_memory);

        /* Segman buyuklugunu ogren*/
        shmctl(segment_id, IPC_STAT, &shmbuffer);
        segment_size = shmbuffer.shm_segsz;
        printf("Segman buyuklugu: %d\n", segment_size);

        /* PBA ya bir katar yaz. */
        sprintf(shared_memory, "Hello, World.");
```

Örnek 2

```
        /*Baglantiyi kopar */
        shmdt(shared_memory);

4       /* Farkli bir adreste PBA baglantisi kur */
        shared_memory = (char*) shmat(segment_id, (void*) 0x5000000, 0);
        printf("PBA yeniden baglanti adresi %p\n", shared_memory);

        /* PBA dan katari oku */
9       printf("%s\n", shared_memory);

        /* Baglantiyi kopar*/
        shmdt(shared_memory);

14      /*PBA yi iade et*/
        shmctl(segment_id, IPC_RMID, 0);

        return 0;
    }
```

Örnek 2 Çıktısı

```
PBA bağlantı adresi 0xb77e3000
Segman büyüklüğü:  25600
PBA yeniden bağlantı adresi 0x5000000
Hello, World.
```