

25.04.2000

Bilgi Teknolojileri Tezsiz Yüksek Lisans Programı
İşletim Sistemleri Final Sınavı

(Süre: 120 dakika)

1. (15 puan) Bildiğiniz işletim sistemi türlerini yazın ve bunları kısaca açıklayın.
2. (20 puan)
 - (a) Proses nedir? Kısaca tanımlayın.
 - (b) UNIX işletim sisteminde prosesler arası haberleşme için hangi mekanizmalar yer almaktadır?
 - (c) Semaforlar hangi amaçlarla kullanılır?
3. (15 puan) Bellek yönetim biriminin temel görevleri nelerdir? Bellek yönetim sistemleri kaç'a ayrılır?
4. (15 puan)
 - (a) Kilitlenme (deadlock) nedir? Hangi durumlarda oluşabilir?
 - (b) Kilitlenme olması durumunda genelde hangi yollar izlenir?
5. (15 puan)
 - (a) Giriş/Çıkış yazılımlarının katmanlı bir yapıda olması neden tercih edilir?
 - (b) Giriş/Çıkış yazılımlarında hangi katmanların yer alması etkin bir çalışma sağlar?
6. (20 puan) Bir anne proses bir tamsayı uzunluğunda bir paylaşılan bellek alanı ve o alanı kullanırken karşılıklı dışlama işlemini gerçeklemek amacıyla gerekecek olan semaforu yaratıp paylaşılan bellek alanına "0" ve semafora da "1" başlangıç değerlerini verecektir. Anne proses daha sonra iki adet çocuk proses yaratacaktır. Paylaşılan bellek alanını iki çocuk proses kullanmaktadır. Her iki proses de bu bellek gözündeki sayıyı okuyup ekrana bastırmakta ve sonra değerini bir artırmaktadır. Sayı 100 değerine ulaştığında her iki proses de sonlanacaktır. Anne prosesin sonlanması için dışarıdan 3 numaralı sinyalin yollanması gerekmektedir. Bu sinyali alan anne proses yarattığı paylaşılan bellek alanı ve semaforu sisteme geri verdikten sonra sonlanacaktır. Bu istenen çalışmayı sağlayan programı C programlama dilinde ve UNIX sistem çağrılarını kullanarak yazınız. Kullanılması gereken başlık dosyalarının isimleri ve gerekli sistem çağrılarının sentaksları ekte verilmiştir.

Kullanılacak başlık dosyaları:

sys/types.h, sys/ipc.h, sys/sem.h, sys/shm.h, signal.h, unistd.h, stdio.h

Sistem Çağrıları:

- `int fork(void);`
- `void signal(int signum, (void *) handler);`
- Semafor için gerekli sistem çağrıları:
 - `int semget (int key, int nsems, int semflg);`
 - `int semop (int semid, struct sembuf *ops, unsigned nops);`
 - `int semctl (int semid, int semnum, int cmd, union semun arg);`
- Semaforlar için gerekli veri yapılarının tanımı:
 - `union semun {
 int val; /* value for SETVAL */
 struct semid_ds *buf; /* buffer for IPC_STAT, IPC_SET */
 unsigned short int *array; /* array for GETALL, SETALL */
 struct seminfo *__buf; /* buffer for IPC_INFO */
};`
 - `struct sembuf
{
 short sem_num; /* semaphore number: 0 = first */
 short sem_op; /* semaphore operation */
 short sem_flg;
};`
- Paylaşılan bellekler için gerekli sistem çağrıları:
 - `int shmget(key_t key, int size, int shmflg);`
 - `void *shmat (int shmid, const void *shmaddr, int shmflg);`
 - `int shmdt (void *shmaddr);`
 - `int shmctl(int shmid, int cmd, struct shmid_ds *buf);`