

01.08.2001

Bilgi Teknolojileri Yüksek Lisans Programı
İşletim Sistemleri Final Sınavı

(Süre:120 dakika)

1. (20 puan)
 - (a) İş sıralama (scheduling) ne demektir? Neden gereklidir? Kısaca açıklayın.
 - (b) UNIX işletim sisteminde kullanılan iş sıralama algoritmasını kısaca açıklayın.
2. (20 puan)
 - (a) Bellek yönetim biriminin temel görevleri nelerdir?
 - (b) UNIX işletim sisteminde kullanılan bellek yönetimi yaklaşımını kısaca açıklayın.
3. (20 puan)
 - (a) Giriş/Çıkış yazılımlarının katmanlı bir yapıda olması neden tercih edilir? Kısaca açıklayın.
 - (b) Giriş/Çıkış yazılımlarında hangi katmanların yer alması etkin bir çalışma sağlar? Kısaca açıklayın.
4. (40 puan) Bu soruda üç prosesin paylaşılan bir bellek alanını kullanmaları ele alınacaktır. Bir A prosesi gerekli paylaşılan kaynakları (semaforlar ve paylaşılan bellek alanları) ve iki adet çocuk proses yaratacak ve gerekli ilk değer atamalarını yapacaktır. Çocuk prosesleri Ç1 ve Ç2 ile gösterirsek, A prosesi ile Ç1 ve Ç2 prosesleri iki tamsayı uzunluğunda bir paylaşılan bellek alanını ortak olarak kullanacaklardır. Bu paylaşılan bellek alanına yazılan ilk tamsayı proses numarasını, ikinci tamsayı da veriyi göstermektedir. Bu durumda A prosesi veriyi ekrana bastırmasını istediği çocuğunun kimlik değerini (tamsayı) ve veriyi (tamsayı) sırayla paylaşılan bellek alanının ilk ve ikinci kısımlarına yazacaktır. Ç1 ve Ç2 prosesleri de paylaşılan bellek alanını kontrol edecekler ve eğer bu alanın ilk kısmında kendi proses kimlik numaraları yazılıysa, ikinci kısımdaki veriyi okuyup ekrana bastıracaklardır.
 - (a) Bu çalışmayı sağlamak için kullanmayı öngördüğünüz semaforları ne amaçlarla kullanacağınızı kısaca açıklayın.
 - (b) A, Ç1 ve Ç2 proseslerine ilişkin program kodlarını sembolik bir dille yazın.
 - (c) Yukarıda tanımlanan çalışmayı sağlayan programı C programlama dili ve gerekli UNIX sistem çağrılarını kullanarak yazın.

Sistem Çağrıları:

- `int fork(void);`
- `void signal(int signum, (void *) handler);`
- `int getpid(void);`
- `int getppid(void);`
- `unsigned int sleep(unsigned int seconds);`
- `int wait(int *status);`
- **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);`