

28.04.2010

Bilgisayar İşletim Sistemleri 2. Yılı Sınavı
Süre: 120 dakika

Ad-Soyad:

No:

İmza:

Öğretim Üyesi:

Soru	1	2	3	4	Toplam
Puan					

Soru 1: Bir hizmet alımı için n adet proses ($P_i, i=1..n$) birer adet teklif verecek ve içlerinden değeri en düşük olan (en ucuz teklifi veren) seçilecektir. Teklif verme işlemi, bir prosesin “*teklif-ver*” fonksiyonunu çağırıp, önerdiği değeri ve kendi indis değerini (i) bildirmesiyle sağlanacaktır. Bir kez teklif sunan bir prosesin yeni teklifler üretmesini engellemek amacıyla, proseslerin sonuç açıklanana kadar sözü edilen fonksiyon çağrısından geri dönmelerine izin verilmeyecektir. Son (n) teklifi verdiğini belirleyen proses, toplanan değerler arasından en küçük olanı seçip bir ortak bellek alanına teklifin değerini ve teklifi veren prosesin indisini yazdıktan sonra, diğer proseslerin de tekrar çalışmalarını sağlayacaktır. Bu çalışma düzenini sağlayacak olan *teklif-ver* fonksiyonuna ait kodu sözde komutlar ile semafor yapısını ve ilkelerini kullanarak yazın ve bir prosesin bu fonksiyonu nasıl kullanacağını gösterin.

Soru 2:

a) Bir işletim sisteminin iş sıralama algoritması sistemde yer alacak olan iş yükünün nitelikleri ve gereksinimleri göz önüne alınarak belirlenmelidir. Aşağıda belirtilen iş yükleri için uygun olacağını düşündüğünüz iş sıralama yöntemlerinin adlarını veriniz ve seçim nedeninizi açıklayınız.

i. Uçak hareketlerinin izlendiği ve yönetildiği bir hava uçuş denetim merkezi

ii. Kullanıcılarının çoğunlukla elektronik mesajlarını okuma ve internette gezinme amacıyla kullandıkları bir çok kullanıcı sistem

iii. Proseslerin sistemde bekleme sürelerinin en aza indirilmeye çalışıldığı bir ‘batch’ sistemi

b) Bir çalışma ortamında yer alan prosesler aşağıda belirtilen sürelerce işlemci zamanına gereksinim duyarlar ve verilen öncelik değerlerine sahiptirler. Proseslerin P1, P2, P3, P4, P5 sırasında sisteme girdikleri bilinmektedir.

<u>Proses</u>	<u>Sisteme Giriş zamanı</u>	<u>MİB Zamanı</u>	<u>Öncelik</u>
P1	0	10	3
P2	2	1	1
P3	4	2	3
P4	6	1	4
P5	8	5	2

Aşağıdaki sıralama yöntemleri için proseslerin çalışmalarını gösteren zaman diyagramlarını çizin ve kısaca açıklayın.

i. FIFO

P1																										
P2																										
P3																										
P4																										
P5																										
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		

ii. En kısa iş-ilk önce

P1																										
P2																										
P3																										
P4																										
P5																										
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		

iii. Dönüşümlü sıralama (zaman dilimi=1)

P1																										
P2																										
P3																										
P4																										
P5																										
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		

iv. Önceliğe göre kesintili sıralama (yüksek sayısal değer yüksek öncelik belirtir)

P1																										
P2																										
P3																										
P4																										
P5																										
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		

Soru 3: Bir sistemde iki adet farklı kaynak (K1 ve K2) vardır. Sistemde toplam altı birim K1 ve dört birim de K2 bulunmaktadır. Dört adet proses (P1, P2, P3, P4) bu kaynakları paylaşmaktadır. Çalışma başlangıcında prosesler tüm çalışma boyunca kullanacakları toplam kaynak sayılarını Tablo-1'de olduğu şekilde belirtmişlerdir. Belirli bir anda proseslerin ellerinde tuttıkları kaynaklar Tablo-2'de verilmiştir. Sistem bu durumdayken P2 prosesi bir adet daha K2 kaynağından isterse Banker algoritmasına göre bu istek karşılanır mı? Açıklayın.

Tablo-1

	K1	K2
P1	4	3
P2	1	2
P3	4	4
P4	5	4

Tablo-2

	K1	K2
P1	2	1
P2	1	1
P3	1	1
P4	1	0

Soru 4: Aşağıda anlatılan çalışma C programlama dilinde, Linux sistem çağrıları da kullanılarak gerçekleştirilecektir. Gerekli programı yazıp kısaca açıklayınız.

“ Bir tamsayı uzunluğundaki bir bellek alanı iki adet proses (P1 ve P2) arasında paylaşılmaktadır. P1 prosesi gerekli *prosesler arası etkileşim kaynaklarını* oluşturduktan ve ilk değer atamalarını yaptıktan sonra bir adet çocuk proses (P2) yaratacaktır. Bu noktadan sonra P1 prosesi sonsuz bir döngü içinde kullanıcıdan tamsayı değerler okuyacak ve her okuduğu değeri, eğer boşsa, paylaşılan bellek bölgesine yazacaktır. P2 prosesi de paylaşılan bellek bölgesinde veri hazır olduğunda bu veriyi okuyarak paylaşılan bellek bölgesini boşaltacak ve veriyi ekrana bastıracaktır.”

Gerekli olabilecek Linux sistem çağrılarının prototipleri ve bazı veri yapıları:

```
pid_t fork(void);
int execlp(const char *file, const char *arg, ...);
sighandler_t signal(int signum, sighandler_t handler);
int kill(pid_t pid, int sig);
pid_t wait(int *status);
int pause(void);
int shmget(key_t key, size_t size, int shmflg);
void *shmat(int shmid, const void *shmaddr, int shmflg);
int shmdt(const void *shmaddr);
int shmctl(int shmid, int cmd, struct shmid_ds *buf);
int semget(key_t key, int nsems, int semflg);
int semop(int semid, struct sembuf *sops, unsigned nsops);
int semctl(int semid, int semnum, int cmd, ...);
struct sembuf {
    unsigned short sem_num;
    short          sem_op;
    short          sem_flg;
};
union semun {
    int val;
    struct semid_ds *buf;
    unsigned short *array;
    struct seminfo *__buf;
};
```