

No	Ad Soyad	İmza
----	----------	------

S1	S2	S3	S4	Toplam
----	----	----	----	--------

Bilgisayar İşletim Sistemleri Yılsonu Sınavı

Soru 1: (20 puan)

- a) İşletim sisteminin, kısıtlı ana bellek koşullarında bile çok sayıdaki prosesi eş zamanlı olarak çalıştırabilmesine olanak sağlayan bellek hizmeti nedir, açıklayın.
- b) Bir bilgisayarın belleği 4 sayfa çerçevesi alabilmektedir. Sayfa isteklerinin aşağıdaki sırada oluştuğunu kabul ederek **FIFO** ve **LRU** yöntemleri uygulandığı takdirde her sayfa isteği karşılandıktan sonra bellekte hangi sayfaların yer alacağını ve eğer yer değiştirme söz konusu olursa bellekten hangi sayfanın dışarı taşındığını tablo üzerinde gösterin. Sayfa hatası oluşan adımları belirtin. İki yöntemi bulgularınızla karşılaştırın.

6 , 1, 2, 4, 8, 1, 8, 2, 7, 8, 1, 2

Soru 2: (20 puan)

- a) Ölümcül kilitlenme nedir, tanımlayınız ve ölümcül kilitlenme ile sonuçlanan bir hatalı semafor kullanımına örnek veriniz (semaforların ilk değerlerini belirtiniz).

- b) Bir sistemde 5 adet proses ve 3 farklı tipte kaynak yer almaktadır: A (10 adet), B (5 adet) ve C (7 adet). Yürütmenin belirli bir anında içinde bulunulan durum aşağıda gösterilmiştir:

Sahip olunan kaynaklar

	A	B	C
P0	0	1	0
P1	2	0	0
P2	3	0	2
P3	2	1	1
P4	0	0	2

Maksimum istek

	A	B	C
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3
A	B	C	
3	3	2	

Boş kaynaklar

- i) Bu durumun güvenli bir durum olup olmadığını Banker Algoritmasını uygulayarak belirleyiniz. Güvenli ise, proseslerin sorunsuz bir şekilde sonlanabilecekleri bir çalışma sırası veriniz. Güvenli değil ise, sistemin hangi noktada tıkanıldığını belirleyiniz.
- ii) Sistem yukarıda verilen durumda iken P1 prosesinin (1,0,2) isteğinde (A kaynağından 1 adet ve C kaynağından da 2 adet) bulunduğunu kabul edelim. Bu isteğin karşılanıp karşılanmayacağını belirleyin.
- iii) Sistemi güvenli olmayan duruma sokacak bir istek örneği veriniz.

Soru 3: (30 puan)

- a) Dosya sistemi yapısında diskteki boş blokların tutulması için kullanılan iki farklı yöntemi açıklayın. Bu yöntemlerin hangisinin hangi durumlarda avantajlı olduğunu açıklayın.
- b) Blok boyunun 1K olduğu ve blokların 16 bit ile adreslendiği bir sistemde, 32MB kapasiteli bir sabit disk için yukarıdaki anlattığınız iki yöntem için gereken maksimum blok sayısını adım adım hesaplayınız.
- c) UNIX V7 dosya sisteminde kullanılan i-node yapısı içinde 10 blok adresi tutulduğunu varsayın. Ayrıca bu dosya sisteminde tek aşamalı, çift aşamalı ve üç aşamalı dolaylı (single/double/triple indirect) erişim kullanıldığını da göz önünde bulundurun. Blok boyunun 2K olduğu ve blokların 32 bit ile adreslendiği bu sistemde maksimum dosya boyu ne olur? Hesabınızın adımlarını gösterin ve açıklayın.

Soru 4: (30 puan) Bu soruda iki prosesin paylaşılan bir bellek alanını ortak olarak kullanmaları ele alınacaktır.

- Bir A prosesi gerekli paylaşılan kaynakları ve bir adet çocuk proses (proses B) yaratıp, gerekli ilk değer atamalarını yapacaktır.
- Başlangıç işlemlerinden sonra, A prosesi bir paylaşılan bellek alanına rasgele ürettiği bir sayıyı yazacaktır. (**Not:** rasgele tamsayı üreten `uret(void)` isimli bir fonksiyon olduğunu varsayıp kullanabilirsiniz.)
- B prosesi, paylaşılan bellek alanında veri hazır olduğunda bu sayıyı okuyacak ve ekrana yazdıracaktır. Ardından kendisi bir başka rasgele sayı üretip aynı bellek alanına yazacaktır.
- A prosesi bellek alanında B tarafından yazılmış bir veri varsa bu veriyi okuyacak ve ekrana yazdıracaktır. Ardından kendisi de bu alana bir sayı yazacaktır.
- Bu işlemler sonsuz bir döngü içinde devam edecektir.
- Çalışmanın sonlanması için A prosesine bir sinyalin komut satırından gönderilmesi gerekmektedir. Bu sinyali alan A prosesi çocuğu B prosesine bir sinyal yolladıktan sonra semaforları ve paylaşılan bellek alanlarını da sisteme iade edip sonlanacaktır. Annesinden sinyal alan B prosesi de sonlanacaktır.

(a) Bu çalışmayı sağlayan programı sözde komutlarla yazın.

(b) Programı C programlama dili ve gerekli LINUX sistem çağrılarını kullanarak yazın.

LINUX sistem çağrıları:

```
pid_t fork(void);
typedef void (*sighandler_t)(int);
sighandler_t signal(int signum, sighandler_t handler);
int kill(pid_t pid, int sig);
int pause(void);
pid_t getpid(void);
pid_t getppid(void);
pid_t wait(int *status);
unsigned int sleep(unsigned int seconds);
int semget(key_t key, int nsems, int semflg);
int semctl(int semid, int semnum, int cmd, union semun arg);
int semop(int semid, struct sembuf *sops, size_t nsops);
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);
```