

Bilgi Teknolojileri Yüksek Lisans Programı
İşletim Sistemleri Final Sınavı

1. Bu soru UNIX işletim sistemi ile ilgilidir.
 - (a) Sistem çağırısı nedir? Neden gereklidir? (10 puan)
 - (b) Yeni bir proses nasıl yaratılır? Yeni bir proses yaratılırken işletim sistemi hangi adımları yürütür? (10 puan)
 - (c) Bir proses iki adet çocuk proses yarattıktan sonra hem anne hem de çocuk prosesler kendilerinin ve annelerinin kimlik numaralarını ekrana bastırıp sonlanacaklardır. (15 puan)
 - i. Bu çalışmayı sağlayan programı C dilinde, LINUX işletim sistemi çağrılarını kullanarak yazın.
 - ii. Prosesler çalıştıklarında oluşacak bir örnek ekran çıktısı verin ve yorumlayın. (Anne prosesin çocukların hepsinden sonra sonlandığını varsayabilirsiniz.)
 - iii. Anne proses çocuklardan önce sonlanırsa yukarıda verdiğiniz ekran çıktısında nasıl bir değişiklik olur? Açıklayın.
 - (d) UNIX işletim sisteminde kullanılan iş sıralama algoritmasını açıklayın. (15 puan)
2. (15 puan)
 - (a) Banker algoritmasının nasıl çalıştığını kısaca açıklayın.
 - (b) Aşağıdaki tablolarda verilen durumların güvenli olup olmadıklarını belirleyin ve cevabınızı açıklayın. (Sistemde kaynaktan toplam 10 tane var.)

Proses	Atanmış	Max. İstek
A	3	9
B	2	4
C	2	7

 - i.

Proses	Atanmış	Max. İstek
A	4	9
B	2	4
C	2	7

 - ii.
3. Karşılıklı dışlama nedir? Semafor mekanizması neden karşılıklı dışlama yapmak için kullanılmaya uygundur? Açıklayın. (15 puan)
4. İşletim sistemi çekirdeğinin modüler olması ne demektir? Bu özellik çalışma anında ve çekirdek geliştirmede ne avantajlar sağlar? (10 puan)
5. (20 puan)
 - (a) Giriş/Çıkış yazılımlarının temel amaçları nelerdir?
 - (b) Giriş/Çıkış aygıtları kaç'a ayrılır? Açıklayın.

Sistem Çağrıları:

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