Bilgisayar İşletim Sistemleri 1. Yıliçi Sınavı (Süre: 100 dakika)

İsim:
No:
İmza:
Öğretim Üyesi:

24.03.2010

Soru	1	2	3	Toplam
Puan				

Soru 1: P0 ve P1 prosesleri karşılıklı dışlama koşullarını gerçeklemek üzere aşağıda yer alan kodu verilen ilk değerler ile yürütmektedirler. Yerel değişkenler "i" ve "j" kodu yürüten prosesin indisine uygun olan değerlere sahiptirler:
P0 için i=0, j=1 ve P1 için i=1, j=0.

```
boolean bekliyor[2] = { false, false };
int sıra = 0;
```

mx_begin:

```
1.    bekliyor [i] = true;
2.    while (sıra!= i) {
3.          while (bekliyor [j]);
4.          sıra = i; }
```

mx_end:

```
1.    bekliyor [i] = false;
```

(a) Karşılıklı dışlama koşullarını gerçekleyen kullanıcı düzeyindeki bir algoritmanın karşılaması gereken koşullar nelerdir, açıklayın.

(b) Verilen çözümü inceleyerek, sözkonusu koşulları yerine getirip getirmediğini belirtin. Yanıtınızın olumsuz olması durumunda, nedenini bir örnek senaryo ile gösterin.

Soru 2: Aşağıdakileri kısacaca yanıtlayın.

(a) Kullanıcı düzeyinde çalışma ve çekirdek düzeyinde çalışma kavramlarını açıklayın.

(b) Proses bağlamı nedir? Açıklayın.

(c) Proses/iplik arasındaki üç önemli farkı yazın ve açıklayın.

(d) Bir prosesin çalışıyor durumundan çıkmasına neden olacak koşullar ve prosesin geçebileceği yeni durumlar nelerdir? Açıklayın.

(e) Tek işlemcili bir sistemde çoklu iplikli çalışmanın programa hız kazandıracağı durumları açıklayın.

(f) Zaman paylaşımlı çalışma nedir? Açıklayın.

(g) Bir s semaforu üzerinde uygulanabilen P(s) ve V(s) işlemlerini açıklayın.

(h) Semafor yapısını kullanarak, A, B ve C proseslerinin çalışmalarını, A belirli bir fonksiyon çağrısından geri dönmeden (function Fonk();) B ve C'nin çalışmaya başlamasına izin vermeyecek şekilde senkronize eden örnek kod parçasını sözde komutlar kullanarak yazın. Açıklayın.

Soru 3: Soruları aşağıdaki programa göre yanıtlayın.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int t1;

int main (void) {
  int i, f;
  int t2, *t3;

  t1=1;

  for (i=0; i<2; i++) {
    f=fork();
    if (f==0)
       break;
  }
  t2=0;
  if (f==0) {
    t3=(int *)malloc(sizeof(int));
    *t3=0;
    for (i=0; i<5; i++) {
       t1++;
       t2++;
       (*t3)++;
    }
    printf("Proses pid=%d t1=%d t2=%d t3=%d\n", getpid(),t1,t2,*t3);
    free(t3);
    exit(0);
  }
  else {
    t3=(int *)malloc(sizeof(int));
    *t3=0;
    printf("Proses pid=%d t1=%d t2=%d t3=%d\n", getpid(),t1,t2,*t3);
    wait(NULL);
    wait(NULL);
    free(t3);
    exit(0);
  }
  return (0);
}
```

(a) Program çalıştırıldıktan sonra ekrana basılacak mesajları yazın. Bu mesajların ve mesaj içindeki değişken değerlerinin (`t1`,`t2`,`*t3`) nedenini açıklayın.
(Not: Proses kimlik değerlerini uygun şekilde seçebilirsiniz).

(b) Mesajların ekrana basılma sıraları hakkında ne söyleyebilirsiniz? Açıklayın.

(c) Alttaki programdan "`wait(NULL)`" satırları kaldırılırsa ekrana basılacak mesajlarda nasıl bir değişiklik olabilir? Açıklayın.

(d) Alttaki programda prosesler yerine iplikler kullanılsaydı `t1`, `t2` ve `*t3` program sonunda hangi değerlere sahip olurdu? Açıklayın.