Vastaa kaikkiin kysymyksiin. Please answer all questions.

1. Selitä (6p)
   a) Osituskäyttö
   X b) Pre-emptio skeduloinnissa
   x c) Ulkoinen pirstoutuminen

   Explain (6p)
   a) Time sharing
   b) Pre-emption in scheduling
   c) External fragmentation

2. Esittele kriittisen alueen ongelman ratkaisun vaatimukset. Toteuttaako seuraava algoritmi nämä vaatimukset kahden prosessin (tässä i ja j) tapauksessa (perustele)? Jos ei niin kuinka muuttaisit toteutusta, jotta vaatimukset täyttyisivät? (6p)

   Present the requirements of a solution to the critical-section problem. Does the following algorithm satisfy all these requirements in the case of two (here i and j) processes (justify)? If not, how would you modify the solution so that it would? (6p)

```
do {
   flag[i] = true;

   while (flag[j]) {
      /* do nothing */
   }

      /* critical section */

   flag[i] = false;

     /*  remainder section */

} while(1);
```

3. Viisi prosessia P_i, $i = 1, \ldots, 5$, käyttävät kuutta tiedostoa F_j, $j = 1, \ldots, 6$. Eräällä ajanhetkellä kunkin prosessin varaamat tiedostot on esitetty taulukossa 1. Samalla hetkellä prosessit ovat pyytäneet saada avata tiedostoja taulukon 2 mukaisesti. Kuhunkin tiedostoon voidaan päästää vain yksi prosessi kerrallaan.

   Five processes P_i, $i = 1, \ldots, 5$, are using six files F_j, $j = 1, \ldots, 6$. At one moment the files opened by processes are shown in Table 1. At the same time processes have requested to open files according to Table 2. Each file can be accessed only by one process at a time.

Taulukko / Table 1.

	F_1	F_2	F_3	F_4	F_5	F_6
P_1			X			
P_2						
P_3				X		
P_4						X
P_5		X				

Taulukko / Table 2.

	F_1	F_2	F_3	F_4	F_5	F_6
P_1		X				
P_2	X					
P_3			X			X
P_4	X					
P_5				X		

sykli!

a) Selvitä graafisesti, onko järjestelmässä tapahtunut lukkiuma. Jos on, mitkä prosessit ovat osallisina lukkiumaan? (5p)

Find out graphically, if a deadlock has occurred in the system. If it has, find out which processes are deadlocked? (5p)

b) Vapaana olevaa tiedostoa 1 pyytää tällä hetkellä kaksi eri prosessia. Kummalle prosessille kyseinen tiedosto kannattaisi antaa? Perustele vastauksesi. (1p)

At the current time, a free resource, the file 1, is requested by two different processes. Which one of the processes it is better to give access to the file. Justify your answer. (1p)

4. a) Mitä tarkoitetaan vaatimussivutuksella? Milloin sivukeskeytys voi tapahtua? (2p)

Explain the term demand paging. Under what circumstances do page faults occur? (2p)

Kun muistia ei ole työmuistissa

b) Oletetaan, että järjestelmän muistinhallinnassa käytetään vaatimussivutusta. Sivunkorvausalgoritmina on LRU. Eräässä tapauksessa viitataan prosessin osoiteavaruuteen seuraavassa järjestyksessä:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0

Prosessin käytössä on kolme kehystä. Esitä kehysten sisältö kunkin viittauksen jälkeen. Kuinka monta sivukeskeytystä viittauksista aiheutuu? (3p)

A computer memory management is implemented by demand paging using LRU page-replacement algorithm. A process will access its memory space according to the following reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0

The process can use three frames. Show the content of the frames after each reference. How many page faults will occur? (3p)

c) Onko LRU tässä tapauksessa optimaalinen sivunkorvausalgoritmi? Perustele vastauksesi. (1p)

Is LRU in this situation an optimal demand paging algorithm? Justify your answer. (1p)