

Kirjoita jokaiseen vastauspaperiisi kurssin nimi, pvm, oma nimi, nimikirjoitus ja opiskelijanumero. Tämä koe on tavallinen erilliskoe. Vastaa kaikkiin kysymyksiin. Kuhunkin tehtävään riittää 1-2 sivun vastaus.

1. [6 p] **Kriittisen vaiheen (alueen) ongelma**

- a. [2 p] Selitä, mikä on kriittisen vaiheen (KV) ongelma. Kuka odottaa ketä ja milloin?
- b. [2 p] Anna pseudokooditasoinen esimerkki kahden koodisegmentin KV-ongelmasta. Anna skenaario, jonka lopputulos on virheellinen, kun KV-ongelmaa ei ole ratkaistu.
- c. [2 p] Anna edellämainittuun (kohta b) esimerkkiä semaforien avulla toteutettu ratkaisu KV-ongelmaan. Selitä, miksi ratkaisu toimii myös (b-kohdassa) antamassasi skenaariossa.

2. [6 p] **Lukkiutuminen**

- a. [2 p] Anna pseudokooditason esimerkki, jossa prosessit P, Q ja R ovat lukkiutuneet. Perustelee.
- b. [2 p] Kuinka prosessien lukkiutuminen voidaan havaita. Kuinka näin havaitusta lukkiutumisesta voi toipua?
- c. [2 p] Lukkiutuminen voidaan ennakolta estää varaamalla resurssit aina tietyssä järjestyksessä. Miksi tämä menetelmä varmasti estää lukkiutumiset? Näytä, kuinka tämä menettely estää lukkiutumisen kohdan (a) esimerkissäsi. Mikä ongelma tässä lukkiutumisongelman ratkaisussa on?

3. [6 p] **Muistinhallinta ja virtuaalimuisti**

- a. [3 p] Minkä ongelman Buddy-systeemi (Buddy System) ratkaisee ja kuinka se toimii?
- b. [3 p] Minkä ongelman kello-algoritmi (Clock) ratkaisee ja kuinka se pääpiirteissään toimii?

4. [6 p] **Tosiaikajärjestelmien vuoronanto (real time scheduling)**

- a. [3 p] Miten tosiaikajärjestelmän vuoronanto eroaa tavallisen järjestelmän vuoronannosta? Miksi FIFO (FCFS) tai aikaviipalevuorottelu (Round-Robin) eivät yleensä sovi tosiaikajärjestelmien vuoronantoon?
- b. [3 p] Tosiaikajärjestelmän tulee pystyä samanaikaisesti käsittelemään (i) kaksi (ääni-) puhelua ja (ii) yhden videon lähettämisen. Äänipuhelut tarvitsevat kumpikin 5 ms välein 1 ms CPU aikaa. Videossa on 25 kehystä/s ja kukin kehys tarvitsee 15 ms CPU aikaa. Nämä kolme prosessia ovat skeduloitavissa RMS:llä (rate monotonic scheduling). Selitä, miksi prosessit ovat skeduloitavissa RMS:llä ja minkälainen vuoronanto siitä seuraa.

5. [6 p] **Levyjen hallinta**

- a. [2 p] Minkä levyjen hallintaan liittyvän ongelman SCAN (hissi) algoritmi ratkaisee? Kuinka SCAN pääpiirteissään toimii?
- b. [2 p] Miksi C-SCAN (Circular SCAN) on parempi kuin SCAN? Kuinka C-SCAN pääpiirteissään toimii?
- c. [2 p] Minkä C-SCAN:iin liittyvän ongelman Linux Deadline Scheduler -algoritmi ratkaisee? Kuinka Linux Deadline Scheduler pääpiirteissään toimii?

6. [6 p] **Tiedostojen hallinta**

- a. [4 p] Mikä on indeksoitu tiedosto (IF) ja kuinka se eroaa indeksoidusta sarjallisesta (peräkkäis) tiedostosta (ISF)? Milloin sitä (IF) kannattaa käyttää? Missä tapauksessa IF:n käyttö on hitaampaa kuin ISF:n käyttö? Kuinka monta indeksiä IF:ssä tarvitaan ja kuinka suuria ne ovat?
- b. [2 p] Miksi IF:n indeksi kannattaa usein toteuttaa B-puuna? Mitä etua B-puu toteutuksella on verrattuna tavalliseen (monitasoiseen) indeksiin?