

Nimi	Nimikirjoitus	Opiskelijanumero	Pisteet

## Käyttöjärjestelmät, minikoe 1, 5.2.2020 (12 p)

Kirjoita vastauksesi tälle koepaperille kunkin tehtävän kohdalle. Huomaa, että koepaperi on 2-puolinen.

- a) [1 p] Selitä, kuinka monisäikeisen prosessin P käyttäjätason säie (ULT, user level thread) S voi olla tilassa Suorituksessa (Running), vaikka S ei oikeasti ole suorituksessa sillä hetkellä.
- b) [3 p] Järjestelmässä on neljä ydintä ja se tukee monisäikeisiä prosesseja. Prosessilla P on 17 säiettä. Yksi säie (S) suorittaa kaiken I/O:n ja muut ovat laskentasäikeitä. Haluamme, että laskentasäikeet voivat suorittaa laskentaa samaan aikaan, kun säie S tallettaa tuloksia levyille tai odottaa käyttäjän syötettä.

Olisiko P:n säikeet järkevää toteuttaa käyttäjätason säikeinä (ULT, user level threads) vai ytimen tason säikeinä (KLT, kernel level threads)? Perustele, miksi valitsemasi vastausvaihtoehto on oikein ja se toinen väärin.

- c) [4 p] Anna pseudokoodiesimerkki kriittisen vaiheen ongelmasta, jossa kriittisen vaiheen muodostaa kaksi erillistä koodisegmenttiä.

Anna skenaario, jonka lopputulos on oikein. Anna skenaario, jonka lopputulos on virheellinen.

Selitä, kuinka tämä kriittisen vaiheen ongelma ratkaistaan monitorin avulla.

- d) [4 p] Meillä on 80 merkin puskuri B. Prosessi P aika ajoin kirjoittaa B:hen. Jos B on täynnä, P odottaa, kunnes sinne tulee tilaa. Prosessi Q aika ajoin lukee B:stä. Jos B on tyhjä, Q odottaa, kunnes sinne tulee merkkejä. Toteuta pseudokooditasolla P:n ja Q:n välisen synkronointiongelman ratkaisu semaforien avulla. Muista alustaa käyttämäsi semaforit.