

Siirräntäjärjestelmät I

1. Harjoituksissa on materiaalia UNIX:n `close()`-systeemikutsusta. Missä tilanteissa systeemikutsu voi epäonnistua? Kuinka monen ohjelman luulisit tarkistavan virhetilanteita, joissa `close()` epäonnistuu? Mitä ohjelma voisi tehdä `close()`:n epäonnistumisen jälkeen?
2. Single UNIX Specification, Version 2 (Copyright 1997 The Open Group) määrittelee rutiinin `read()`, jonka paluuarvosta sanotaan: "return a non-negative integer indicating the number of bytes actually read". Miten tämä vaikuttaa ohjelmien tekemiseen? (Luet esim. 1000 tavua ja käyttöjärjestelmä sanookin, että luettiin vain 942 tavua?) Kuinka monen ohjelman arvioit varautuvan tähän virhetilanteeseen?
3. Harjoituksissa on tiedostojärjestelmää käsittelevä esimerkkiohjelma. Selvitä manuaalisivujen avulla mitä ohjelma tekee (ei pelkästään tulostus vaan mitä informaatiota eri systeemikutsut käsittelevät).
4. Ns. *tertiary*-tason massamuisteissa (mm. nauha-asetat ja CD-ROM -levyt) jotkut moni-ajojärjestelmät antavat laitteen käyttöön vain yhdelle prosessille kerrallaan. Miten tämä toteutetaan? Miksi näin toimitaan? Voisiko laitteen käyttö olla toteutettu toisin? Miten muut selkeästi yhden kohteen oheislaitteet (kuten näppäimistö ja hiiri) on jaettu useamman prosessin kesken?