

S-38.3600 UNIX Application Programming  
Exam 10.12.2014

1. (10p) List and describe methods that two processes can use to communicate. Luettele ja esittele eri menetelmiä, joilla kaksi prosessia voi kommunikoida.
2. (10p) Describe how pipes operate, from point of view of both ends. Consider life-time issues. What are the limitations of pipes as an IPC method? Kuvaile kuinka putket toimivat, kummankin pään näkökulmasta. Ota huomioon putken elinkaari. Mitä rajoituksia putkilla on IPC-menetelmänä?
3. (10p) What are the differences between threads and processes? Give a list of things a programmer must be aware of when using threads instead processes. Mitkä ovat erot säikeiden ja prosessien välillä? Anna lista asioista joista ohjelmoijan tulee olla tietoinen käyttäessään säikeitä prosessien sijaan.
4. (10p) Suppose you need to implement a service (daemon) that implements a message-passing service for different programs within one computer. The programs that wish to communicate are executed on different user accounts with low privileges. The service has the tasks of receiving, auditing (validating and logging) and retransmitting the messages. How would you implement such a service? Describe structure of the service software and the interfaces you would use for communication, handling messages and user identification. The service should be both fast and fair; how should it handle cases where one program produces too many or too large messages? Logs should be written to disk, can this cause problems? Details of daemonization are not important.

Halutaan toteuttaa palvelu (daemon) joka toteuttaa viestienvälitysmekanismiin erillisten ohjelmien välille yhden tietokoneen sisäisessä käytössä. Viestivät ohjelmat suoritetaan erillisillä käyttäjätunnuksilla joilla on matalat käyttöoikeudet. Palvelun tulee vastaanottaa, auditoida (tarkastaa ja kirjata logiin) ja välittää viestit edelleen. Kuvaile kuinka toteuttaisit tällaisen palvelun? Kuvaile palvelinohjelmiston rakenne ja mitä rajapintoja käyttäisit viestintään, viestien käsittelyyn ja käyttäjien tunnistamiseen. Palvelun tulisi olla sekä nopea että reilu; miten käsittelet tilanteen jossa yksi ohjelma tuottaa liian monta tai liian isoja viestejä? Logi tulisi kirjoittaa levyille, voiko tämä aiheuttaa ongelmia? Prosessin daemonisointia ei tarvi selittää.