

Luento 7 (verkkoluento 8)
Ohjelman toteutus järjestelmässä
(prosessi, käyttöjärjestelmä)

Prosessi
Prosessin esitysmuoto
Käyttöjärjestelmä
KJ-prosessit

15.11.2012

Copyright 2012 Teemu Kerola

1

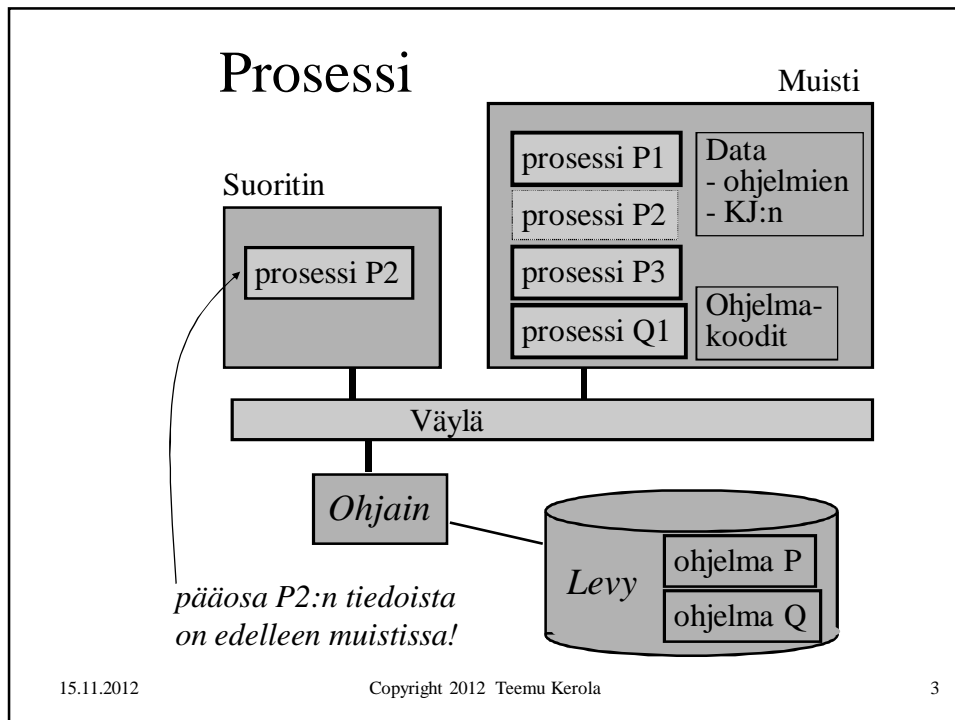
Prosessi = ohjelma järjestelmässä

- Järjestelmässä voi olla ”samalla kertaa” **monta prosessia** joko samasta tai eri ohjelmasta
 - Käyttäjän (ihmisen) näkökulma ja aikaskaala (1 min, 1 sek?)
- Suorittimella **suorituksessa on vain yksi** prosessi kerrallaan
 - Oletus: 1-ytiminen (1-core) suoritin
 - Laitteiston näkökulma ja aikaskaala (1 ms, 1 μ s, 1 ns?)
- **Muut prosessit ovat odottamassa** jotakin
 - suoritinta? I/O:ta? viestiä toiselta prosessilta?
 - vapaata muistitilaa?

15.11.2012

Copyright 2012 Teemu Kerola

2

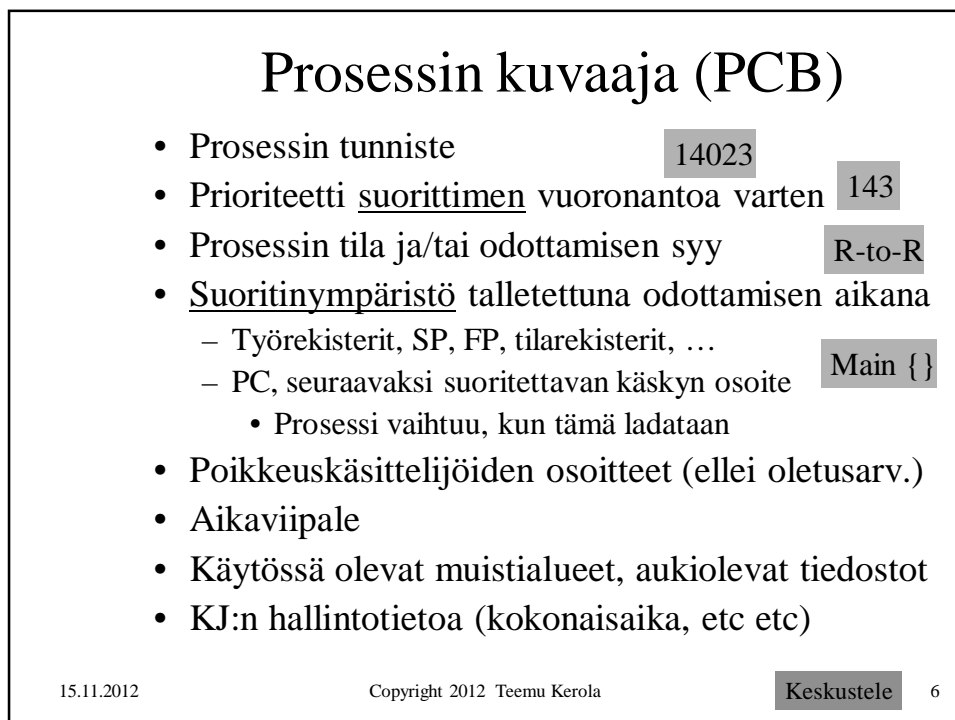
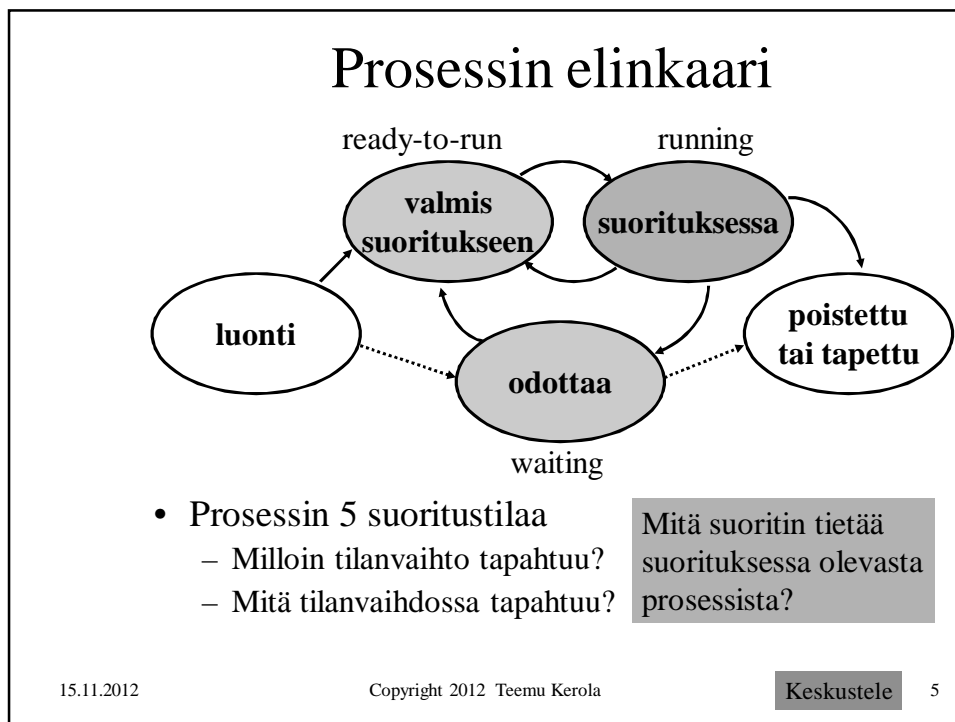


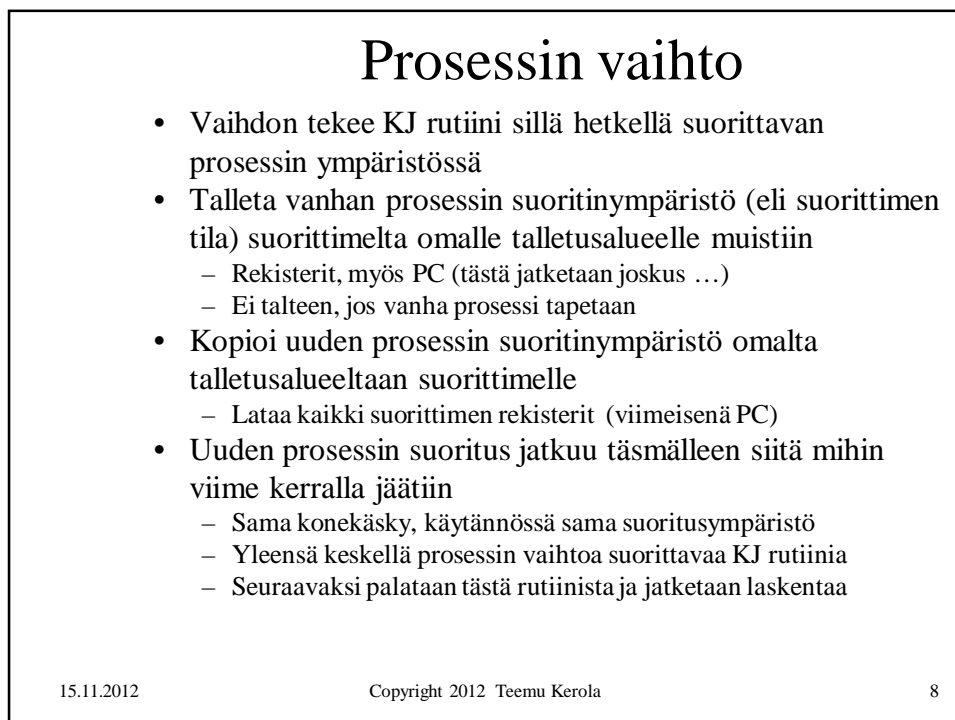
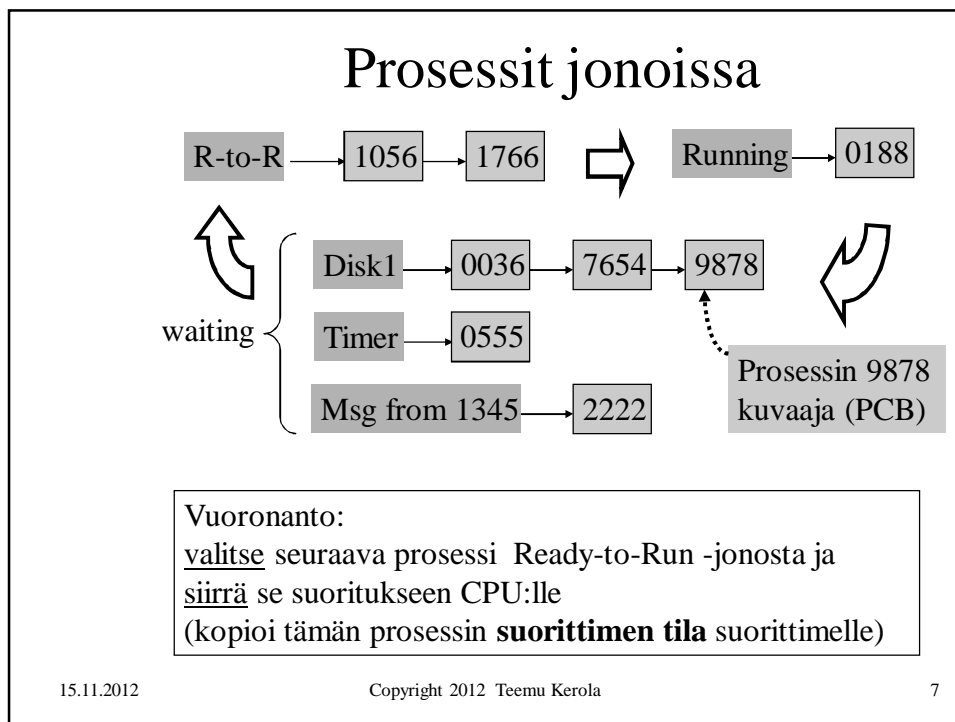
Prosessin vaihto

process switch

- Suorittimella suoritusvuorossa olevan prosessin vaihtaminen
- Tapahtuu aika usein
 - keskimäärin noin 2000-3000 konekäskyn välein?
 - esim. 50-500 kertaa sekunnissa?
 - Miksi?
 - nykyinen prosessi ei voi jatkaa suoritusta
 - nykyinen prosessi ei voi jatkaa suoritusta
 - KJ päättää, että aika vaihtaa suoritusvuoroa
- Iso operaatio - paljon kopiointia 50-500?
 - montako konekäskyä tähän kuluu? 0?

15.11.2012 Copyright 2012 Teemu Kerola 4





Prosessin prioriteetti

- **Prosessin tärkeysjärjestys suorittimella**
 - esim. pieni numero \Rightarrow iso (parempi) prioriteetti (tai toisinpäin)
- **Joka prioriteetti(luokalle) oma R-to-R jononsa**
 - KJ prosesseilla parempi prioriteetti kuin käyttäjätason prosesseilla
 - tosiaikasovelluksen prosesseilla parempi prioriteetti kuin KJ prosesseilla
 - muista antaa KJ:lle aikaa aina joskus!
- **Prioriteetti voi vaihdella prosessin elinaikana**
 - paljon suoritinaikaa \Rightarrow huonompi prioriteetti
 - kauan R-to-R jonossa \Rightarrow parempi prioriteetti
 - prosessi siirretään korkeamman prioriteetin R-to-R jonoon

15.11.2012

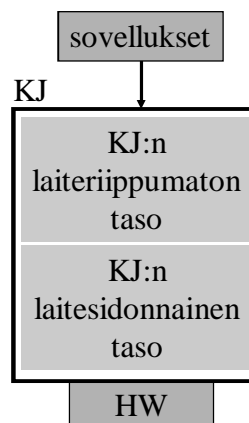
Copyright 2012 Teemu Kerola

Keskustele

9

Käyttöjärjestelmä

- **Laiteriippumaton (HW-riippumaton) rajapinta laitteistoon**
 - Helpottaa käyttöä
 - Jakaa palvelua kaikille reilusti
 - Resurssien hallinta ja valvonta
 - Sovellukset on helpompi toteuttaa ja siirtää muualta



15.11.2012

Copyright 2012 Teemu Kerola

10

Resurssien hallinta ja valvonta

- Suorittimen vuorottaminen
 - jaa suoritinaikaa reilusti kaikille
 - kukaan ei odota suoritinta ikuisesti
 - kriittiset prosessit saavat ajoissa suoritinaikaa
- Tiedostojen (koodi, data) tehokas käyttö
 - laitteesta ja sijainnista riippumaton käyttö
 - helppo yhteiskäyttö ja samalla tietojen suojaus
- Tietoliikenneverkkojen käyttö
 - laiteriippumaton käyttö
 - helppo yhteiskäyttö ja samalla tietojen suojaus
- Hallintokirjanpito

15.11.2012

Copyright 2012 Teemu Kerola

11

Käyttöjärjestelmän rakenne

- Prosessien hallinta
- Muistin hallinta
- Tiedostojen ja laitteiden hallinta
- Verkon hallinta

15.11.2012

Copyright 2012 Teemu Kerola

12

Käyttäjärjestelmän toteutus

- Joukko prosesseja ja/tai aliohjelmia
 - prosessit elävät omaa elämäänsä (etuoikeutettuna root'ina?)
 - aliohjelmat suoritetaan sen hetkisen prosessin ympäristössä (etuoikeutetussa tilassa?)
 - keskeytyskäsitteijät
 - saavat kontrollin aina tarvittaessa
 - aliohjelmakutsut, SVC, viestit
 - ajastimet ja muut keskeytykset

15.11.2012

Copyright 2012 Teemu Kerola

Keskustele 13

KJ palvelun kontrollin palautus

Explisiittinen
KJ-palvelun kutsu
= Palvelupyyntö

- Aliohjelmakutsut
 - CALL → RETURN
- SVC
 - SVC → IRET
- Viestit
 - viesti → vastausviesti
(lähettäjä odottaa vastausta RECEIVE:ssä)

Implisiittinen
KJ-palvelun kutsu

- Ajastimet ja muut keskeytykset
 - keskeytys → IRET

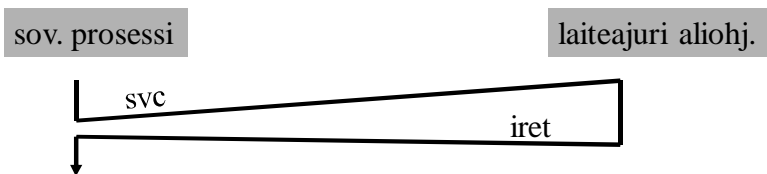
15.11.2012

Copyright 2012 Teemu Kerola

14

KJ esimerkki: laiteajuri

- Aliohjelmana (eli proseduurina)
 - laiteajuri suoritetaan KJ-rutiinina tavallisen SVC-kutsun kautta käyttäjätilassa
 - vain yksi kutsu kerrallaan suorituksessa?
miksi? miten voidaan valvoa?



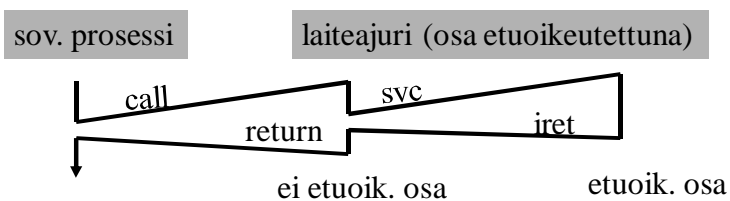
15.11.2012

Copyright 2012 Teemu Kerola

15

KJ esimerkki: laiteajuri (jatk.)

- Aliohjelmana (eli proseduurina)
 - laiteajuri suoritetaan KJ-rutiinina tavallisen aliohjelmakutsun ja/tai SVC-kutsun kautta
 - osa tai kaikki koodista voi olla etuoikeutettua
 - vain yksi kutsu kerrallaan suorituksessa?
miksi? miten voidaan valvoa?



15.11.2012

Copyright 2012 Teemu Kerola

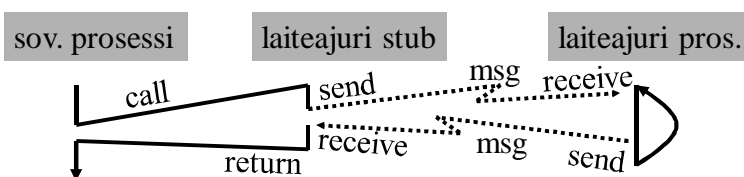
Keskustele 16

KJ esimerkki: laiteajuri

- Prosessina

- proseduurina kutsuttu laiteajurin tynkä (stub) lähettää I/O-pyynnön viestinä laiteajuriprosessille ja odottaa vastausta

- tynkä voi olla käyttäjätilainen
- ajuriprosessi voi olla (joskus) etuoikeutettu
- vaatii prosessien välistä viestintää



15.11.2012

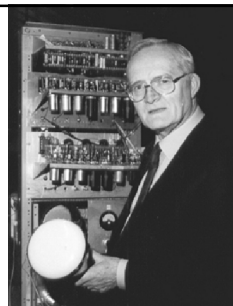
Copyright 2012 Teemu Kerola

17

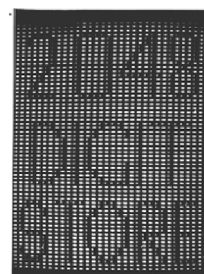
-- Luennon loppu --

- Williams Tube

- 1946, Williams & Kilburn
- katodisädeputki
- ensimmäinen suuri "RAM" muisti
- kallis: \$1000 / 1 kk / putki
- Small Scale Experimental Machine ("Baby"), 1947
- Ferranti Mark I, ensimmäinen yleiskäyttöinen kaupallinen tietokone, 1951 (10000 bitin muisti)



Tom Kilburn holding a Cathode Ray Tube



Storing 2048 bits on a CRT in 1947

15.11.2012

Copyright Teemu Kerola 2001