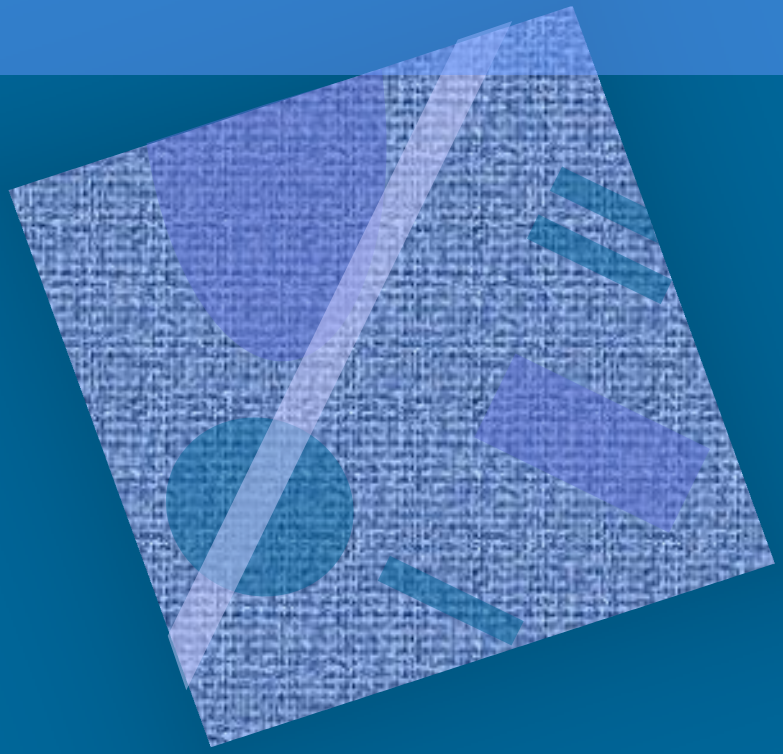
Luento 8

Ohjelman toteutus järjestelmässä (prosessi, käyttöjärjestelmä)



Prosessi

Prosessin esitysmuoto

Käyttöjärjestelmä

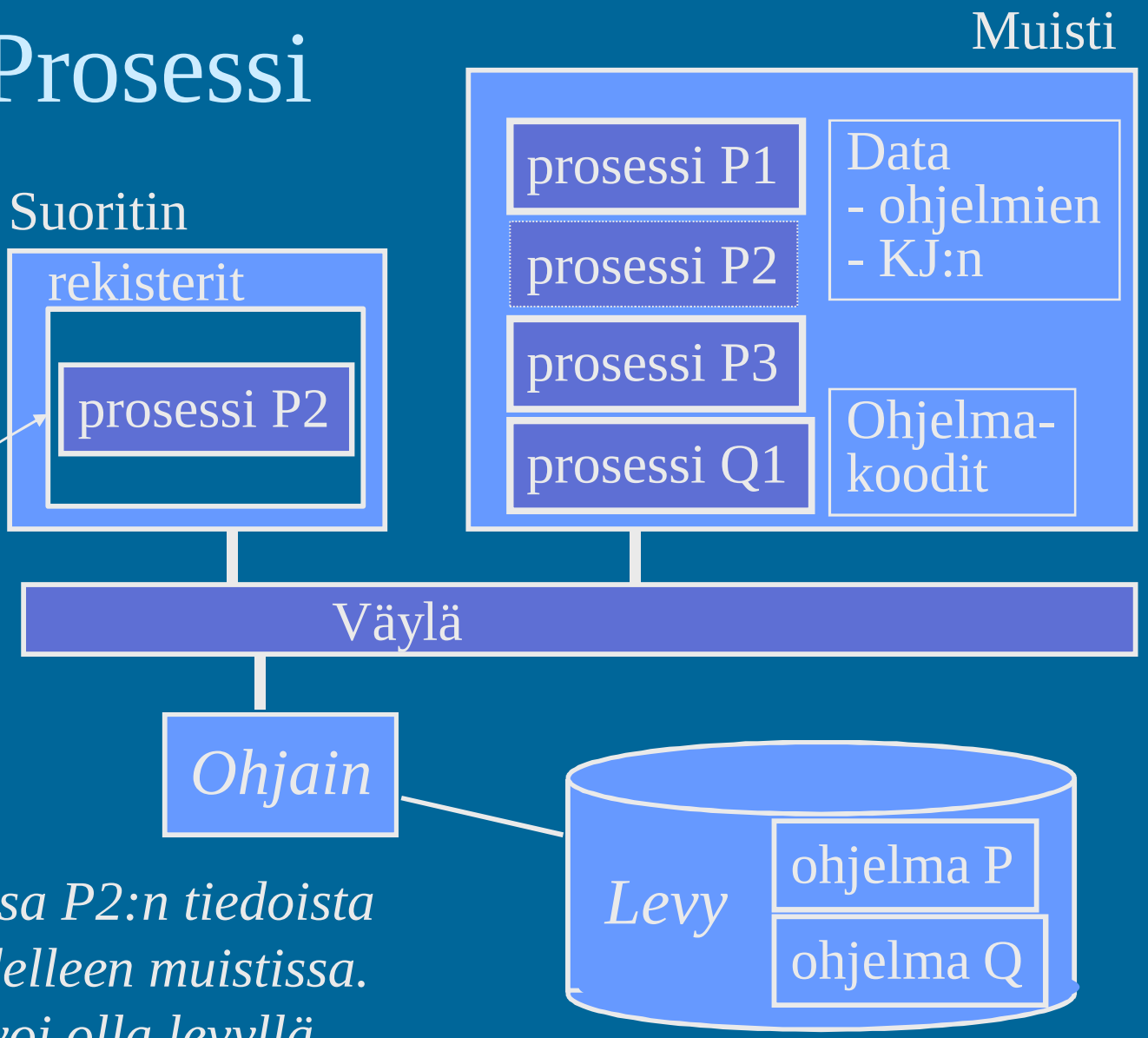
KJ-prosessit

I/O:n toteutus laiteajurin avulla

Prosessi = ohjelma järjestelmässä

- Järjestelmässä voi olla ”samalla kertaa” **monta prosessia** joko samasta tai eri ohjelmasta
  - Käyttäjän (ihmisen) näkökulma ja aikaskaala (1 min, 1 sek, 10 ms)
- Suorittimella **suorituksessa on vain yksi** prosessi kerrallaan
  - Oletus: 1 ytiminen (1 core) suoritin
  - Laitteiston näkökulma ja aikaskaala
    - 1 ns, 1 µs, 1 ms?
- **Muut prosessit ovat odottamassa** jotakin
  - suoritinta? I/O:ta? viestiä toiselta prosessilta?
  - vapaata muistitilaa?

Prosessi



Pääosa P2:n tiedoista on edelleen muistissa. Osa voi olla levyllä.

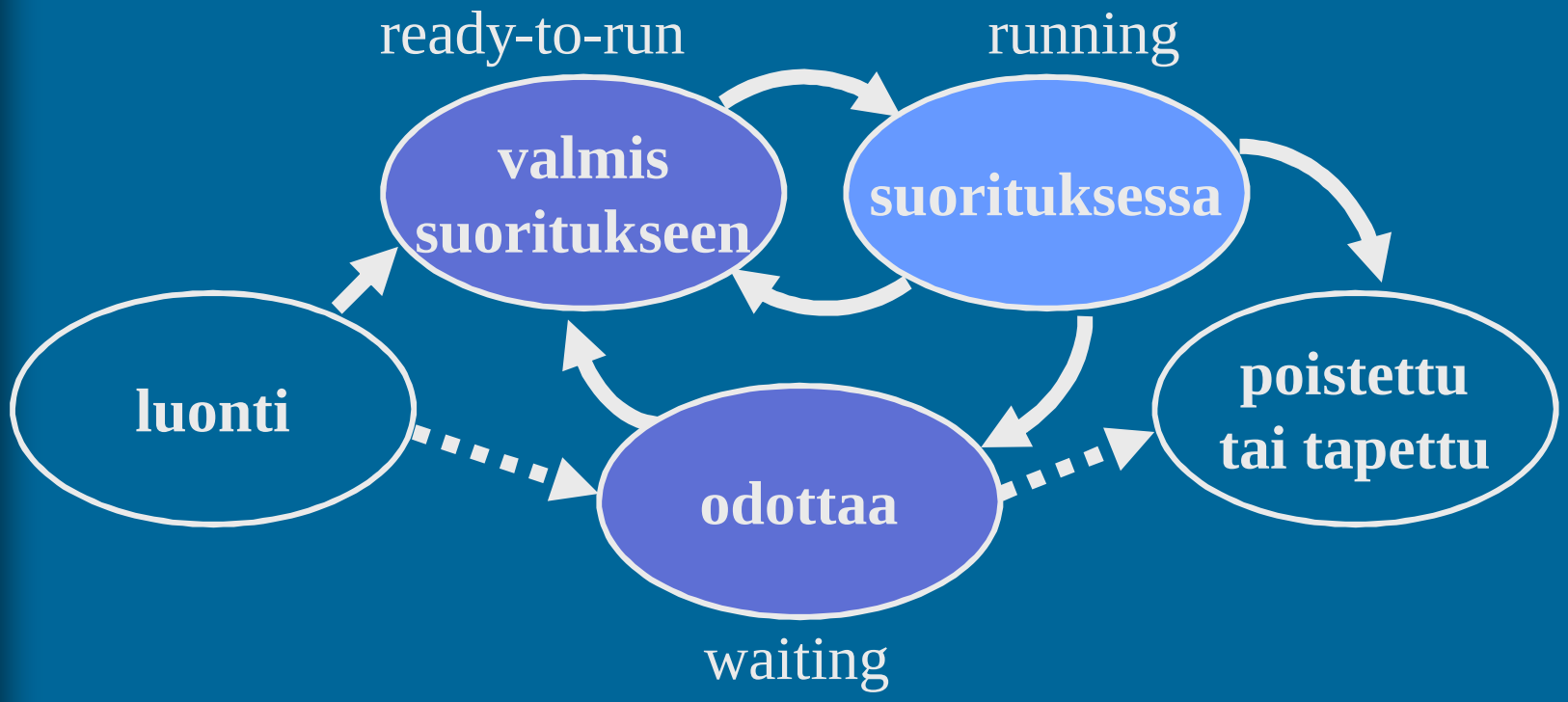
Prosessin vaihto

process switch

- Suorittimella suoritusvuorossa olevan prosessin vaihtaminen
- Tapahtuu aika usein
  - Keskimäärin esim. 2000-3000 konekäskyn välein?
  - Esim. 50-500 kertaa sekunnissa? 10ms välein?
  - Miksi?
    - Nykyinen prosessi ei voi jatkaa suoritusta
    - Nykyinen prosessi ei halua jatkaa suoritusta
    - KJ päättää, että aika vaihtaa suoritusvuoroa
      - Päästyään suorittamaan keskeytyksen kautta
- Iso operaatio - paljon kopiointia
  - montako konekäskyä tähän kuluu?

50-500?
0?

Prosessin elinkaari



- Prosessin 5 suoritustilaa
  - Milloin mikäkin tilanvaihto tapahtuu?
    - Mikä tapahtuma saa aikaan tilanvaihdon?
    - Mitä tilanvaihdossa tapahtuu?
    - Kuka jatkaa suoritusta tilanvaihdon jälkeen?

Keskustele

Prosessin kuvaaja (PCB)

Process Control Block

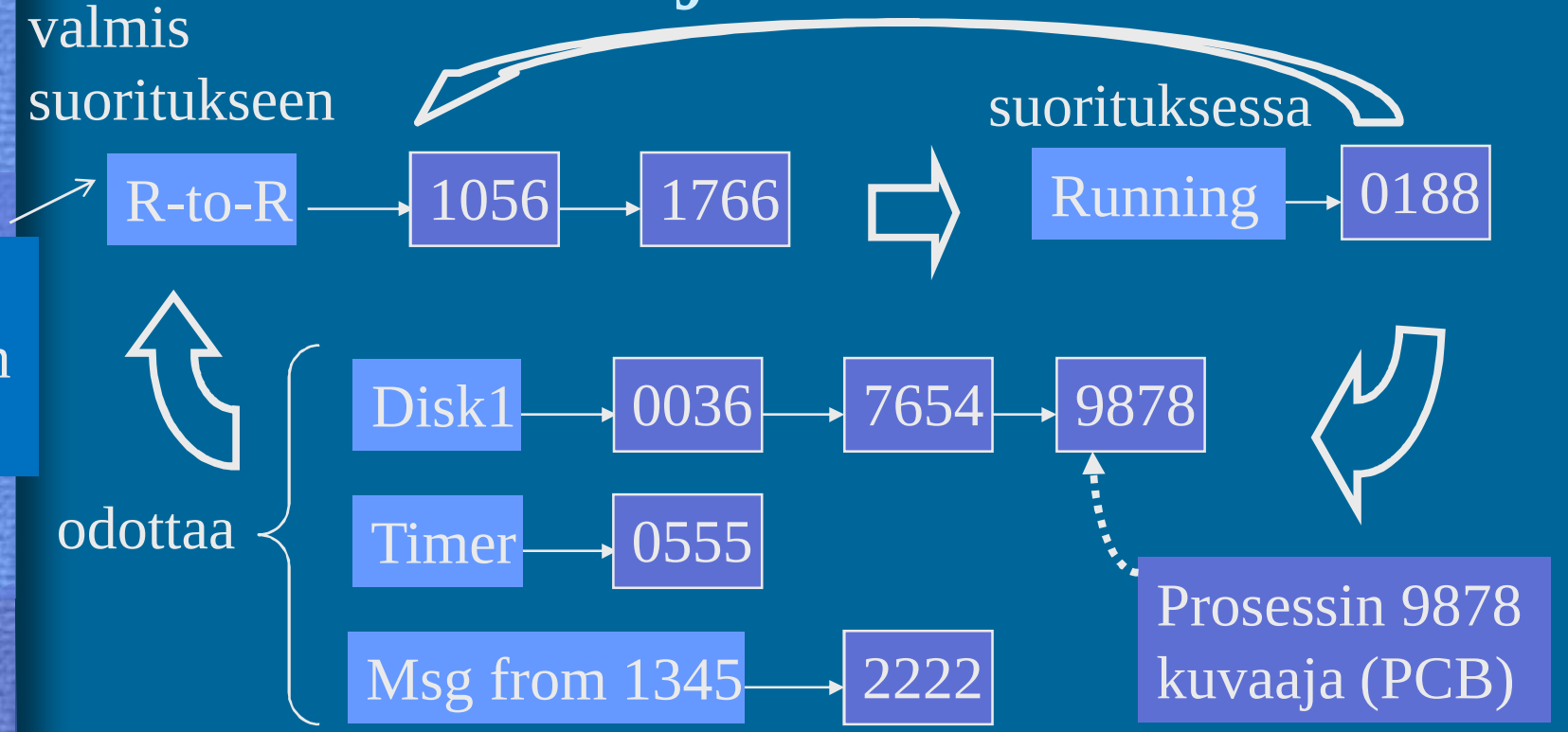
- Prosessin tunniste
  14023
- Prioriteetti suorittimen vuoronantoa varten
  143
- Prosessin tila ja/tai odottamisen syy
  R-to-R
- Suoritinympäristö talletettuna odottamisen aikana
  - Työrekisterit, SP, FP, tilarekisterit, …
  - PC, seuraavaksi suoritettavan käskyn osoite
    - Prosessi vaihtuu, kun tämä ladataan
      aluksi main {}
- Poikkeuskäsittelijöiden osoitteet (ellei oletusarv.)
- Aikaviipale (milloin KJ antaa vuoron toiselle)
- Käytössä olevat muistialueet, auki olevat tiedostot
- KJ:n hallintotietoa (kokonaisaika, etc etc)

processor context

time slice

Keskustele

Prosessit jonoissa



Vuoronanto:
valitse seuraava prosessi Ready-to-Run -jonosta ja
siirrä se suoritukseen CPU:lle
(kopioi tämän prosessin **suoritinympäristö** suorittimelle)

Prosessin vaihto

- Vaihdon tekee KJ rutiini sillä hetkellä suorittavan prosessin ympäristössä (esim. keskeytyskäsittelijässä)
- Talleta vanhan prosessin suoritinympäristö (eli suorittimen tila) suorittimelta omalle talletusalueelle muistiin
  - Rekisterit, myös PC (tästä jatketaan joskus …)
  - Ei tarvitse laittaa talteen, jos vanha prosessi tapetaan
- Kopioi uuden prosessin suoritinympäristö omalta talletusalueeltaan suorittimelle
  - Lataa rekisterit (viimeisenä PC)
- Uuden prosessin suoritus jatkuu täsmälleen siitä mihin viime kerralla jäätiin
  - Sama konekäsky, käytännössä sama suoritusympäristö
  - Yleensä keskellä prosessin vaihtoa suorittavaa KJ rutiinia
    - Keskeytyskäsittelijässä?
  - Seuraavaksi palataan tästä rutiinista ja jatketaan laskentaa

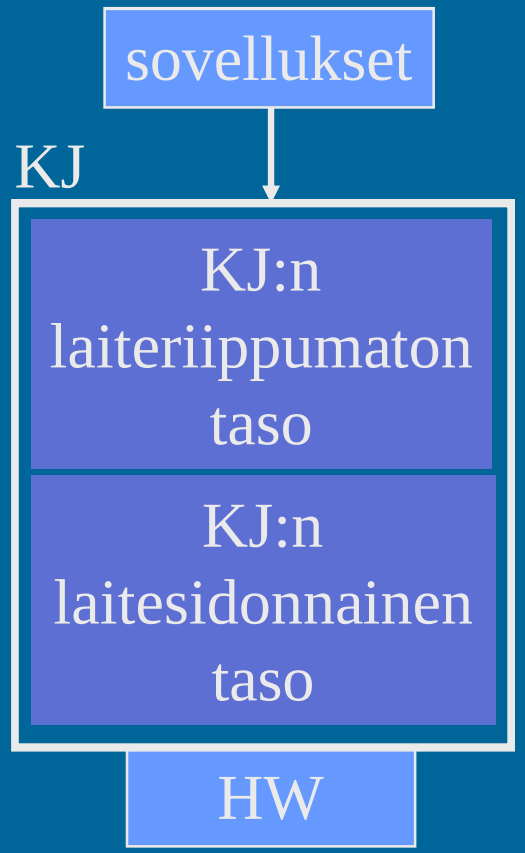
Prosessin prioriteetti

- Prosessin tärkeysjärjestys suorittimella
  - esim. pieni numero Þ 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 Þ huonompi prioriteetti
  - kauan R-to-R jonossa Þ parempi prioriteetti
    - prosessi siirretään korkeamman prioriteetin R-to-R jonoon?

Keskustele

Käyttöjärjestelmä (KJ)

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



Resurssien hallinta ja valvonta

- Suorittimen vuorottaminen, prosessien hallinnointi
  - Jaa suoritinaikaa reilusti, kukaan ei odota ikuisesti
  - Kriittiset prosessit saavat ajoissa suoritinaikaa
- Muistitilan hallinta
  - Paljonko muistitilaa kullekin prosessille
  - Mitkä muistialueet on allokoitu kullekin prosessille
  - Helppo yhteiskäyttö ja samalla tietojen suojaus
- 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

Käyttöjärjestelmän rakenne

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

Käyttöjärjestelmän (KJ) toteutus

- Joukko prosesseja ja/tai aliohjelmia
  - prosessit elävät omaa elämäänsä
    - käyttäjätason prosessi
    - etuoikeutettu prosessi, root-prosessi
    - esim. laiteajuri
  - aliohjelmat suoritetaan sen hetkisen prosessin ympäristössä (etuoikeutetussa tilassa?)
    - esim. keskeytyskäsittelijä, laiteajuri
  - saavat kontrollin aina tarvittaessa
    - aliohjelmakutsut, SVC, viestit
    - ajastimet ja muut keskeytykset
    - KJ ei tee mitään, ellei sen koodia ole suorituksessa!

daemon
windows service
windows palvelu

Keskustele

KJ palvelun kontrollin palautus

- Aliohjelmakutsut
  - CALL → RETURN
- SVC
  - SVC → IRET
- Viestit
  - viesti → vastausviesti
    (lähettäjä odottaa vastausta RECEIVE:ssä)
- Ajastimet ja muut keskeytykset
  - keskeytys → IRET
    (vuoro keskeytyneelle tai vuoronantajan valitsemalle seuraavalle prosessille)

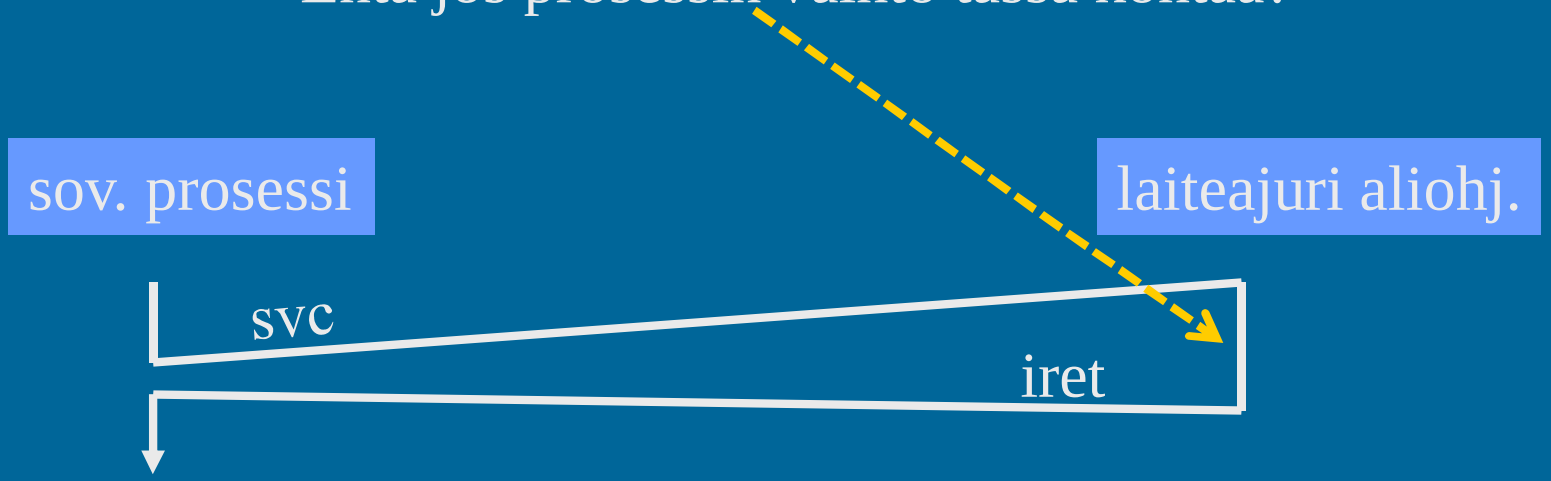
Explisiittinen KJ-palvelun kutsu = Palvelupyyntö

Implisiittinen KJ-palvelun kutsu

KJ palvelu suoritetaan keskeytetty prosessin sisällä. Prosessi ei odota Ready-jonossa.

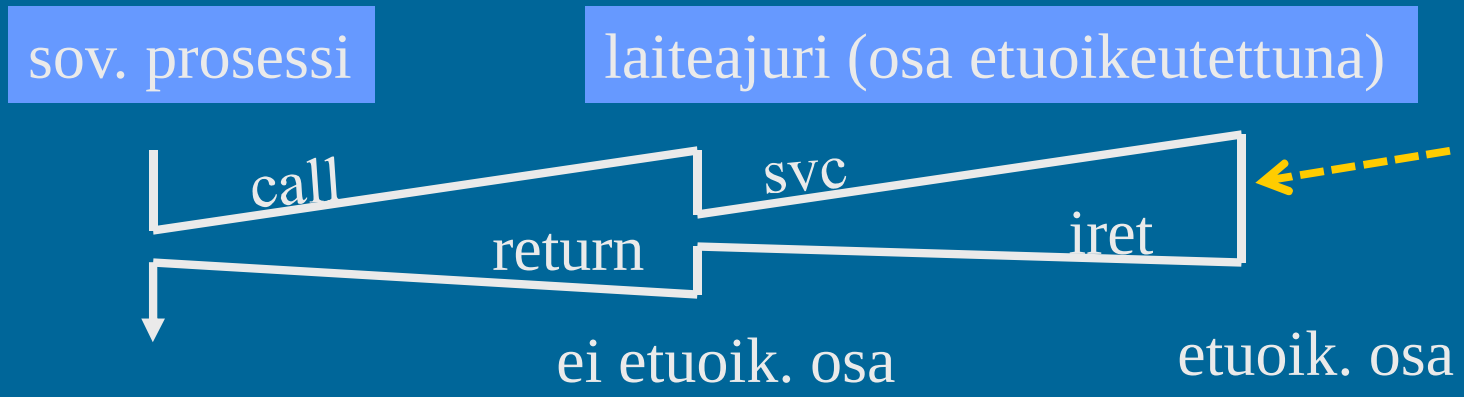
KJ esimerkki: laiteajuri

- Aliohjelmana (proseduurina, metodina)
  - laiteajuri suoritetaan KJ-rutiinina tavallisen SVC-kutsun kautta etuoikeutetussa tilassa
    - vain yksi kutsu kerrallaan suorituksessa? miksi? miten voidaan valvoa?
      - Entä jos prosessin vaihto tässä kohtaa?



KJ esimerkki: laiteajuri (jatk.)

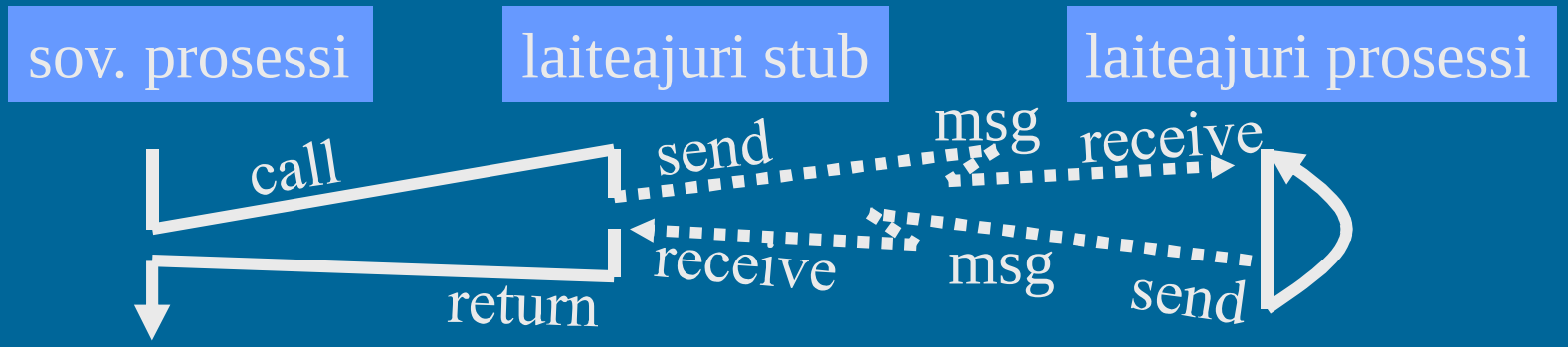
- Aliohjelmana (eli proseduurina)
  - laiteajuri suoritetaan KJ-rutiinina tavallisen aliohjelmakutsun ja/tai SVC-kutsun kautta
    - osa tai kaikki koodista voi olla etuoikeutettua
    - vain yksi kutsu kerrallaan suorituksessa? miksi? miten voidaan valvoa?



Keskustele

KJ esimerkki: laiteajuri

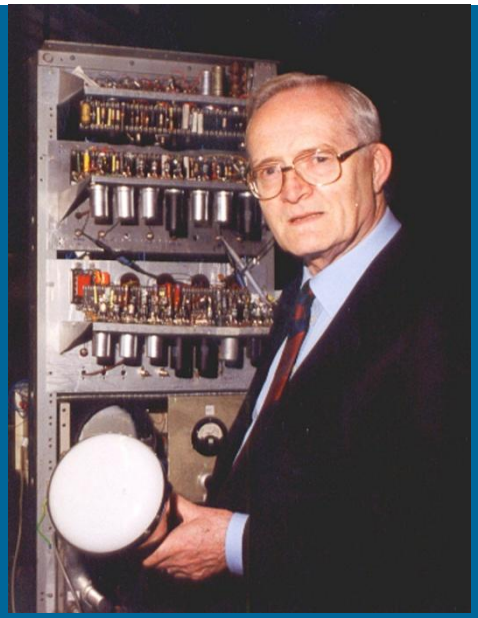
- Prosessina
  - proseduurina kutsuttu laiteajurin tynkä (stub) lähettää I/O-pyynnön viestinä laiteajuriprosessille ja odottaa vastausta
    - tynkä voi olla käyttäjätilainen
    - ajuriprosessi voi olla (joskus) etuoikeutettu
    - vaatii prosessien välistä viestintää



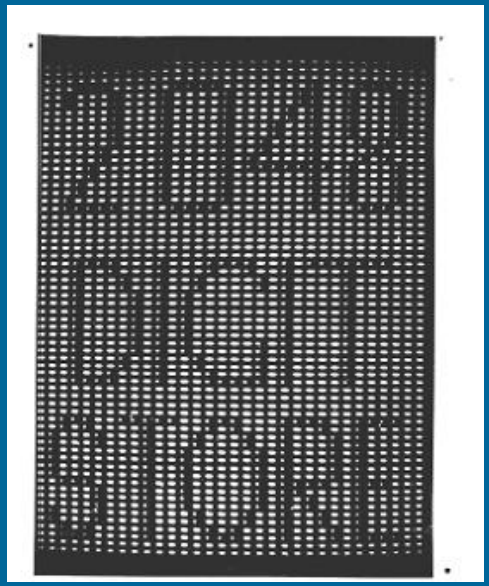
Miksi nyt ei ongelma, jos monta pyyntöä samanaikaisesti?

-- 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