


Jakso 8

Ohjelman suoritus järjestelmässä



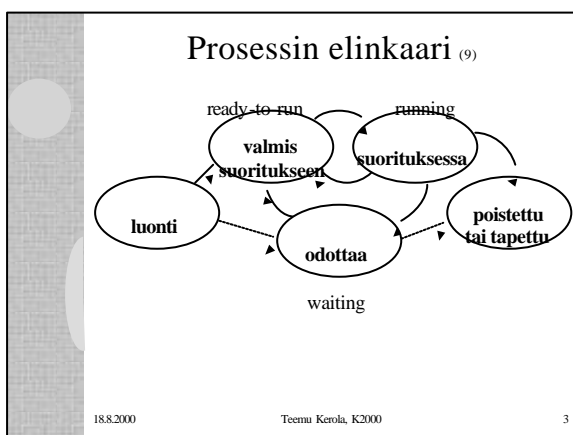
Prosessi
PCB
I/O:n toteutus

18.8.2000
Teemu Kerola, K2000
1

Prosessi ⁽⁴⁾

- Suorituksessa olevan ohjelman esitysmuoto järjestelmässä
 - ”samalla kertaa” järjestelmässä voi suorituksessa monta prosessia joko samasta tai eri ohjelmasta
 - käyttäjän (ihmisen) näkökulma ja aikaskaala (1 min, 1 sek?)
 - suorittimella on yksi prosessi kerrallaan suorituksessa
 - laitteiston näkökulma ja aikaskaala (1 ms, 1 μs, 1 ns?)

18.8.2000
Teemu Kerola, K2000
2



Prosessin vaihto ⁽⁴⁾

- Suorittimella suoritusvuorossa olevan prosessin vaihtaminen
- Tapahtuu aika usein
 - keskimäärin noin 2000-3000 konekäskyn välein?
- Iso operaatio - paljon kopiointia
 - satoja konekäskyjä

18.8.2000
Teemu Kerola, K2000
4

Prosessin esitysmuoto järjestelmässä ⁽⁴⁾

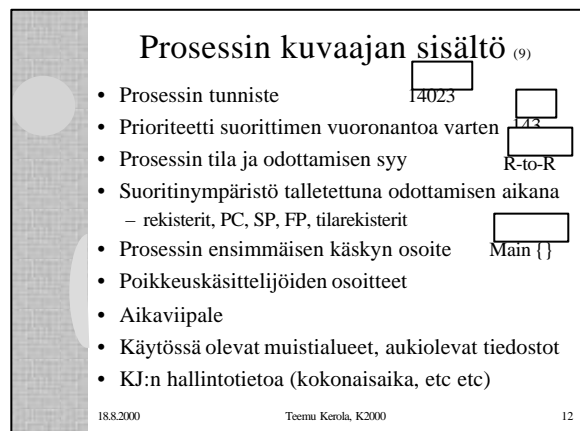
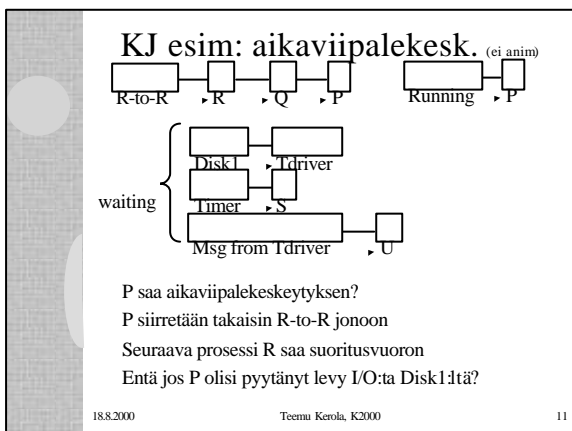
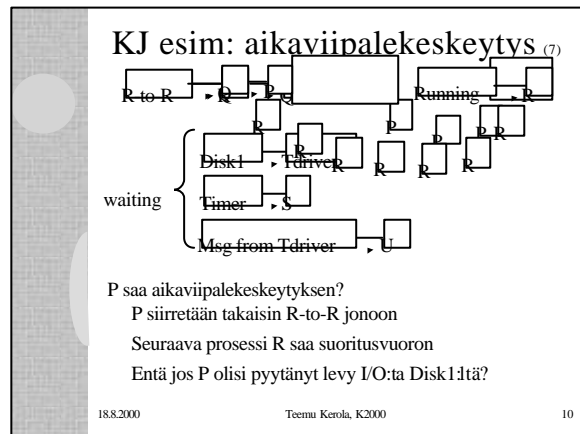
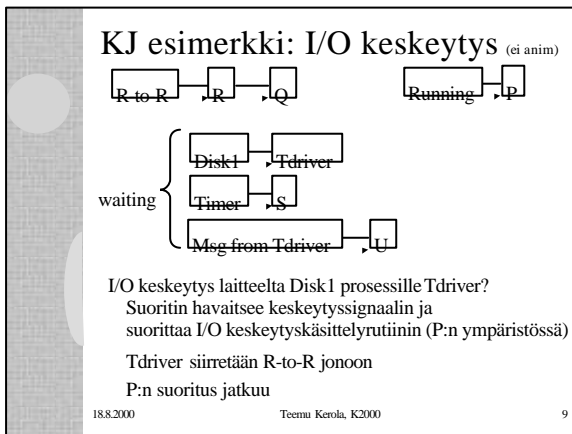
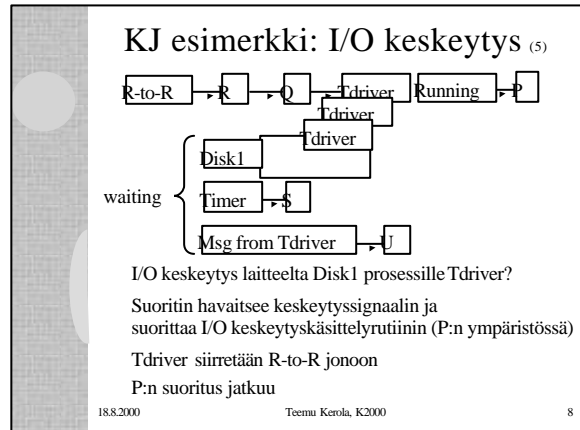
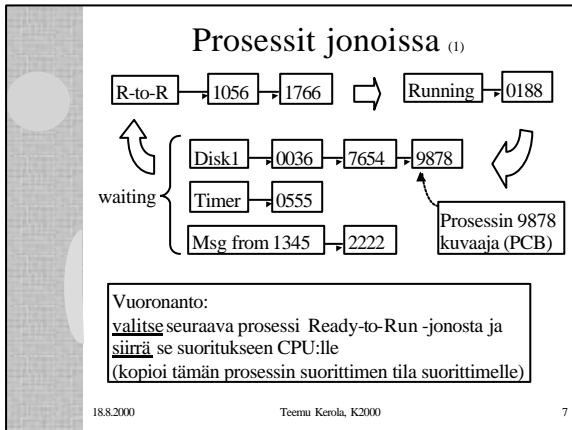
- Prosessin kuvaaja tai kontrollilohko (PCB - process control block)
 - isohko tietue, joka sisältää kaiken yhdestä prosessista
 - muistialueet, tiedostot, tiedostojen käsittelykohdat
 - ei suorituksessa oleville myös: suorittimen tila (laiterekisterit, MMU:n rekisterit, kontrollirekisterit)
 - joka prosessista oma PCB
 - käyttöjärjestelmäruutiinit manipuloivat PCB:tä

18.8.2000
Teemu Kerola, K2000
5

Prosessin tilanvaihdon toteutus ⁽⁹⁾

- Prosessin tilanvaihto tapahtuu siirtämällä prosessi (sen PCB) jonosta toiseen
 - ready-to-run jono (tai jonot)
 - running jono (jota ei oikeastaan ole olemassa)
 - waiting jono
 - joka tyyppille oma jononsa
 - esim: laitteen Disk1 I/O:n valmistumista odottavat
 - esim: näppäimistön painallusta odottavat
 - esim: kellolaittekeskeytystä odottavat
 - esim: prosessilta 1345 signaalia odottavat

18.8.2000
Teemu Kerola, K2000
6



Prosessin vaihto ⁽⁴⁾

- Vaihdon tekee KJ rutiini sillä hetkellä suorittavan prosessin ympäristössä
- Talleta vanhan prosessin suoritin ympäristö suorittimelta omalle talletusalueelle muistiin
 - talleta kaikki suorittimella olevat tiedot muistiin
- Kopio uuden prosessin suoritin ympäristö omalta talletusalueeltaan suorittimelle
 - lataa kaikki suorittimen rekisterit (myös PC!)
- Uuden prosessin suoritus jatkuu täsmälleen siitä mihin viime kerralla jäätiin
 - sama konekäsky
 - prosessin vaihtoa suorittavaan KJ rutiiniin?

18.8.2000 Teemu Kerola, K2000 13

Prosessin prioriteetti ⁽³⁾

- Prosessin tärkeysjärjestys
 - esim. pieni numero ⇒ iso prioriteetti
- Joka prioriteetti (luokalle) oma R-to-R jononsa
 - KJ prosesseilla parempi prioriteetti kuin käyttäjätason prosesseilla
 - tosiaikasovelluksen prosesseilla parempi prioriteetti kuin KJ prosesseilla
- Prioriteetti voi vaihdella prosessin elinaikana
 - paljon suoritinaikaa ⇒ huonompi prioriteetti
 - kauan R-to-R jonossa ⇒ parempi prioriteetti (prosessi siirretään korkeamman prioriteetin R-to-R jonoon)

18.8.2000 Teemu Kerola, K2000 14

Käyttöjärjestelmän toteutus ⁽⁵⁾

- Joukko prosesseja ja prosedureja
 - prosessit elävät omaa elämäänsä (etuoikeutetussa tilassa eli root'ina?) koko ajan
 - swapper (Unix) - muistinhallintaprosessi
 - init prosessi (Unix) - kaikkien käyttäjätason prosessien "äiti"
 - laiteajurit
 - proseduurit suoritetaan sen hetkisen prosessin ympäristössä (etuoikeutetussa tilassa?)
 - keskeytyskäsitelijät
 - Saavat kontrollin aina tarvittaessa
 - keskeytykset, SVC, ajastimet

18.8.2000 Teemu Kerola, K2000 15

18.8.2000 Teemu Kerola, K2000 16

Laitteiden liittäminen järjestelmään ⁽⁴⁾

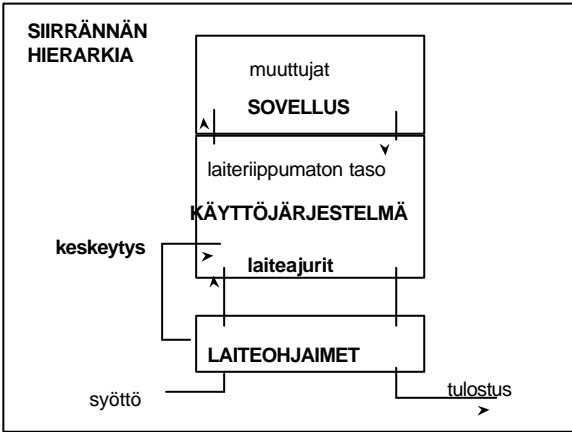
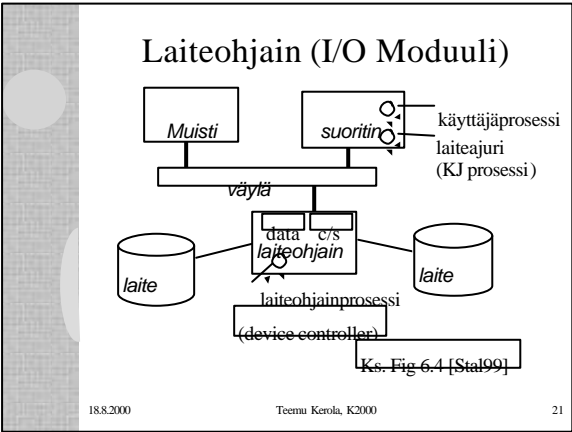
18.8.2000 Teemu Kerola, K2000 17

- Siirännän hierarkia
 - **sovellusohjelmataso**
 - loogisia kokonaisuuksia, tietueita ja tiedostoja
 - ohjelman sisäisiä nimiä
 - Readln (File1, X)
 - Open (Tdsto, RW)
 - käyttöjärjestelmätaso
 - laitteisto

18.8.2000 Teemu Kerola, K2000 18

- Siirränän hierarkia
 - sovellusohjelmataso
 - **käyttöjärjestelmätaso**
 - rutiinit, jotka toteuttavat ja valvovat siirrantää
 - laiteriippumaton siirrantää
 - sovellukselle yhtenäinen tapa käyttää kaikkia siirrantäpalveluita
 - laiteriippuva siirrantää
 - laitteiden todelliseen käyttöön liittyvä ohjausohjelmisto
 - koodattu **laiteajureihin**
 - laitteisto

- Siirränän hierarkia
 - sovellusohjelmataso
 - käyttöjärjestelmätaso
 - **laiteisto**
 - siirrantää voidaan toteuttaa kokonaan prosessorin valvonnassa
 - ei hyödynnetä rinnakkaisuutta
 - **laiteohjain** (siirrantään erikoistunut prosessori) huolehtii itsenäisesti siirrantästä
 - prosessorin ja ohjainten välinen kommunikointi



- laitekuvaaja
 - yksi kutakin laitetyyppiä varten
 - talletettavat tiedot riippuvat laitteesta
 - laitteen yksilöivä tunnus (väyläosoite)
 - ohjeet laitteen käytöstä
 - urien, sektorien ja levypintojen määrä, lohkon koko
 - viitteet näppäimistön merkinmuunnostauluihin
 - laitteen tilatieto
 - varattu/vapaa/rikki
 - viitteet jonottaviin palvelupyntöihin
 - viite laitetta käyttävän prosessin kuvaajaan

- laiteriippumattoman siirrantän tehtäviä
 - loogisesta tiedostonimestä => käytettävän laitteen tyyppi
 - pitää kirjata levytilan vapaista ja varatuista alueista
 - siirrantän puskurointi (levylohko)
 - luku/kirjoituskohdan ylläpito
 - tarvittaessa käynnistää fyysisen siirrantän
 - antaa laiteajurille tehtävän

- **laiteajurin tehtäviä**
 - tehtävät riippuvat laitteesta
 - muodostaa parametrien ja laitekuvaajan perusteella laitetta ohjaavat käskyt
 - esim. levylohkonumeroiden muuttaminen levypinnan, uran ja sektorin numeroiksi
 - levypyyntöjen optimointi
 - ohjaimella tehtävän fyysisen siirränän käynnistys
 - siirränän kirjanpito
 - siirron oikeellisuuden tarkistukset ja virheiden korjausyritykset

Laitteiden käytön toteutus ⁽⁵⁾

ks. laiteohjainkuva

- Käyttäjäohjelma kutsuu käyttöjärjestelmän laiteajuria tekemään I/O:n. Laiteajuri suoritetaan samalla suorittimella kuin käyttöjärjestelmän.
- Laiteajuri ohjaa laitteen toimintaa laitteen laiteohjaimella olevien kontrollirekisterien (muistialue) avulla
- Laiteajuri voi lukea laitteen tilatietoa laiteohjaimella olevien statusrekisterien (muistialue) avulla
- Laiteajuri voi lukea (kirjoittaa) laitteen lukemaa (laitteelle kirjoitettavaa) tietoa laiteohjaimella olevien datarekistereiden (muistialue) avulla
- Kontrolli-, status- ja datarekisteri kolmikko muodostaa "I/O portin" suorittimen näkökulmasta

18.8.2000 Teemu Kerola, K2000 26

Laiteohjaimen rekistereihin viittaaminen ⁽⁵⁾

ks. laiteohjainkuva

- Ongelma: miten suorittimella suorittavan laiteajuri viittaa eri kortilla oleviin rekistereihin?
- Ratkaisu 1: omat I/O-konekäskyt tätä tarkoitusta varten
 - käskyssä annetaan laiteohjaimen identifikaatio ja rekisterin nro (I/O osoitevaraus)
 - vaikea laajentaa käyttöä uusiin laitteisiin, joilla rekisterit voivat olla hyvinkin erilaisia

x86: IN, OUT KOKSI: IN R1, =KBD, OUT R2, =CRT

18.8.2000 Teemu Kerola, K2000 27

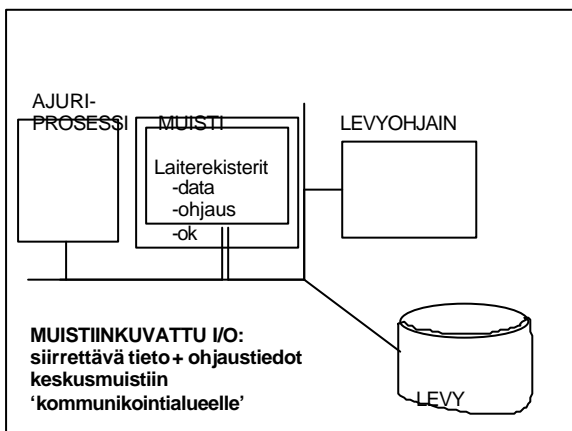
Ratkaisu 2: muistiinkuvattu I/O ⁽⁵⁾

ks. laiteohjainkuva

- Laiteajuri lukee/kirjoittaa laiteohjaimella olevia rekistereitä (data, status/kontrolli) tavallisilla muistin luku/kirjoitus käskyillä
 - ei tarvita erillisiä I/O-konekäskyjä!
 - laiteohjaimella olevat "rekisterit" ovat samanlaista viitattavaa muistia kuin "normaali muisti"
 - muistisoitteen ensimmäiset bitit valitsevat, mille laitteelle (vai tavallisen muistiin) viittaus kohdistuu
 - voidaan käyttää rinnan I/O käskyjen kanssa (laiterekistereihin voi siis viitata sekä I/O-käskyillä että muistiinkuvattun I/O:n avulla)

esim. Intelin arkkitehtuurit

18.8.2000 Teemu Kerola, K2000 28



I/O tyypit ⁽²⁾

ks. laiteohjainkuva

- Suora I/O: laiteajuri odottaa tiukassa silmukassa, kunnes laiteohjaimen statusrekisteri ilmoittaa I/O-pyyntönsä valmistuneen (direct I/O)
 - laiteajuri siirtää tietoa muistin ja datarekisterin välillä
- Epäsuora I/O: I/O:n odotusaika suoritetaan jollain muulla ohjelmalla (indirect I/O interrupt driven I/O)
 - Kun I/O-pyyntö valmistuu, laiteohjain antaa keskeytyksen (laittekeskeytyks, I/O interrupt) suorittimelle, joka (jonkin ajan kuluttua) jatkaa kesken jäänyttä I/O-pyyntönsä esittänyttä ohjelmaa.
 - laiteajuri siirtää tietoa muistin ja datarekisterin välillä

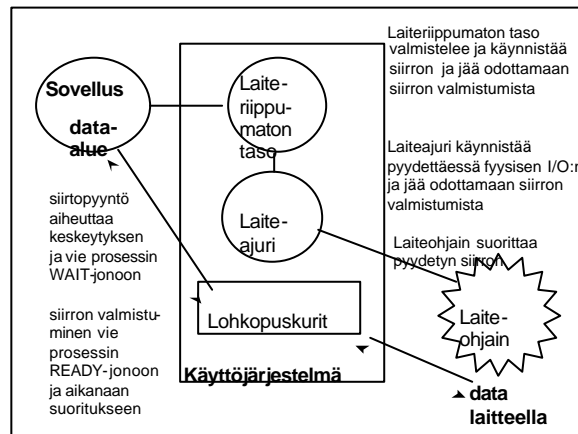
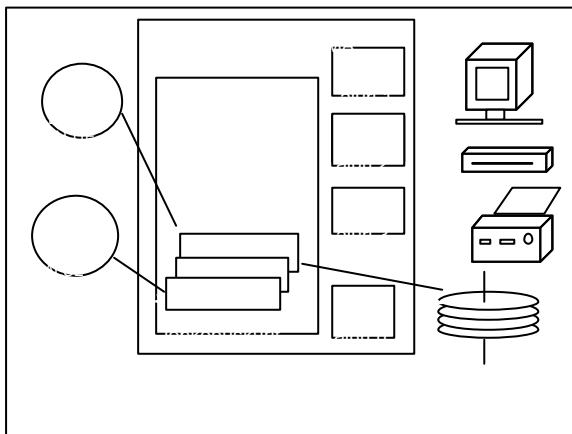
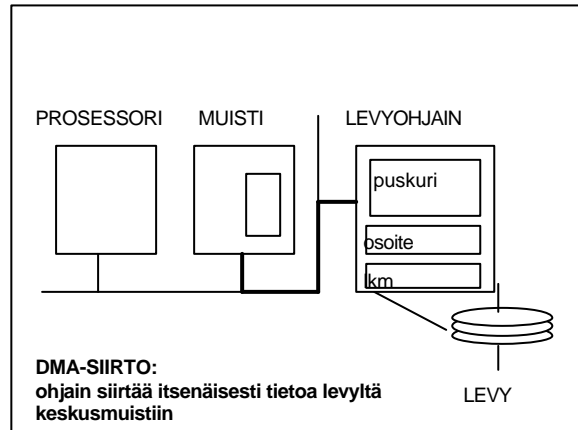
18.8.2000 Teemu Kerola, K2000 30

I/O tyypit (jatkoa) ⁽⁴⁾

ks. laiteohjainkuva

- DMA - Direct Memory Access
 - älykkäämpi laiteohjain
 - laiteohjain voi suoraan kopioida tiedot keskusmuistiin (laiteajurin ei tarvitse siirtää tietoa muistin ja datarekisterin välillä)
 - laiteohjain tekee paljon suuremman määrän työtä itsenäisesti (kuin epäsuorassa I/O:ssa) ennen suorittimelle annettavaa laitekeskeytystä

18.8.2000
Teemu Kerola, K2000
31



-- Jakson 8 loppu --

[Tane99]

```

// Open files for input and output.
inhandle = CreateFile("data", GENERIC_READ, 0, NULL, OPEN_EXISTING, 0, NULL);
outhandle = CreateFile("new", GENERIC_WRITE, 0, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);

// Copy the file.
do {
    n = ReadFile(inhandle, buffer, BUF_SIZE, &count, NULL);
    if (n > 0 && count > 0) WriteFile(outhandle, buffer, count, &count, NULL);
    while (n > 0 && count > 0);
}

// Close the files.
CloseHandle(inhandle);
CloseHandle(outhandle);
    
```

Figure 8-91. A program fragment for copying a file using the Windows NT API functions. This fragment is in C because Java lacks the low-level system calls to open and close them.

Lisää tietoa? KJ kurssit, RIO, Hajjar

18.8.2000
Teemu Kerola, K2000
35