

Vuorotuksen algoritmien vertailua ja 1. periodin kertaus

Koealue: (Koe to 19.10 klo 9.00 CK112)

- Kirjan luvut 1-8 (paitsi 5.1 – 6.6)
- Luennot 1-12, harjoitukset 1-6
- Opintopiiritehtävät 1 ja 2

Menetelmien vertailua

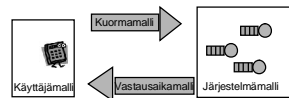
- Miten?



Malli +
Suorituskykyanalyysi

- Paljon liikkuvia osia:
 - Palveluaikojen jakauma
 - Vuorotuksen ja prosessin vaihdon yleisrasite
 - Prosessien palveluntarve (CPU vs I/O – sidonnaisuus)
 - Prosessien saapumisten jakauma
 - CPU:n ja I/O:n suorituskyky

Mallinnus



Mallinnus:

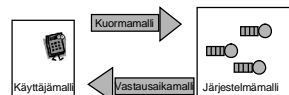
- **täsmälliset** käsitteiden määrittelyt
- oletukset järjestelmän ja ympäristön käyttäytymisestä
- tulosten tarkkuus?
- mallin validiteetin rajat?

Suorituskyvystä
ei voi puhua
määrittelemättä
työkuormaa

Suorituskyky:

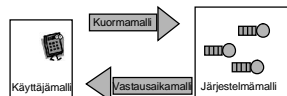
- oletettu/vaadittu kyky vastata tiettyihin palvelupyyntöihin
- järjestelmän käyttäytyminen verrattuna määrittelyyn (vallitsevaa kuormaa suorittaessaan)

Mallinnus



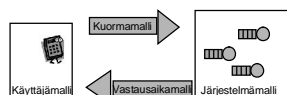
- Mallinnusprosessin tavoite
 - kuvauksien täsmällisyys
 - menetelmän arvioitavuus
 - kokeiden toistettavuus
- Mallin tavoite : oleellisen huomioonotto
 - järjestelmän toimintamekaniikan ymmärtäminen
(mitä / miten / miksi ; syy-seuraussuhteet)
 - käyttökelpoisuus
(laitteiston valinta, kapasiteetin suunnittelu, ohjauksen viritys, ...)
- Mallin ratkaisumenetelmät
 - mittaus
 - analyttiset mallit
 - simulointimallit

Mittaus



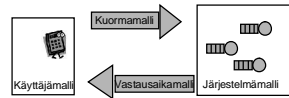
- Mallinnus
 - Mistä järjestelmän käyttäytymisestä ollaan kiinnostuneita?
 - Miten sitä voidaan mitata (suoraan tai epäsuorasti)?
 - Mitkä häiriötekijät on otettava huomioon?
- Järjestelmän instrumentointi
 - mittarien toteutus
 - mittarien häiriövaikutuksen arviointi
- Koesuunnittelu
 - kuorma
 - ajankohta, kesto
 - koetoistot
- Mittaustulosten validiteetti? Tulosten validiteetin rajat?

Simulointi



- Muodostetaan järjestelmää kuvaava toiminnallinen malli
 - komponentit
 - palveluaikajakaumat, saapumisväliaikajakaumat
 - töiden reititykset komponenttiverkon läpi
- Ohjelmoidaan simulaattori
 - haaste: verifiointi (toimiiko oikein?)
- Koesuunnittelu
 - koepisteiden määrittely: parametrien arvot kussakin pisteessä
 - kokeen määrittely: mittauksen kesto; koetoistojen määrä
- Simulointi: mittauskoe simulaattorin toiminnasta
- Haaste: validiteetti (mallintaako oikeaa järjestelmää?)

Analyttinen malli



- Muodostetaan järjestelmää kuvaava matemaattinen malli
 - jonomalli
 - jonoverkkomalli
 - määrittelyt: yleensä todennäköisyysjakaumia
- Määritetään tai määritellään mallin parametrien arvot
- Lasketaan suorituskyky suureiden arvot
- Menetelmäpaletti laaja
 - koulumatematiikka
 - jonoteoria
 - tasapainoyhtälönratkaisu (jne.)
- Mallin validiteetti? Tulosten validiteetin rajat?

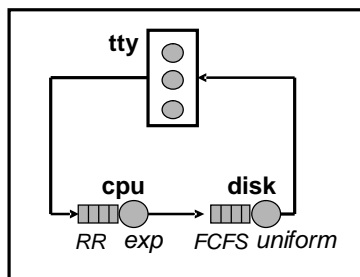
Menetelmän valinta

Kriteeri	Mittaus	Simulointi	Analyttinen
Vaihe	proton jälkeen	milloin vain	milloin vain
Ajan tarve	vaihtelee	kohtalainen	pieni
Välineistö	"mittaristo"	ohjelmointikieli (*)	analyttikko
Tarkkuus (**)	vaihtelee	kohtalainen	karkea
"Entä jos ..."	vaikeaa	onnistuu	helppoa
Kustannus	korkea	kohtalainen	pieni
"Myytävyys"	korkea	kohtalainen	hankala

(*) ohjelmistotukea on olemassa

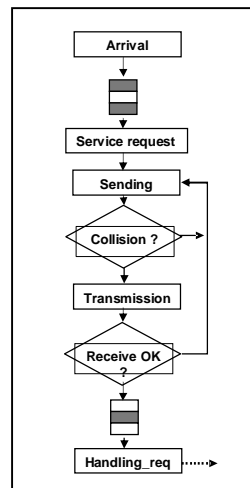
(**) aina mahdollista: tulokset ovat täysin virheellisiä

Mallin rakenne

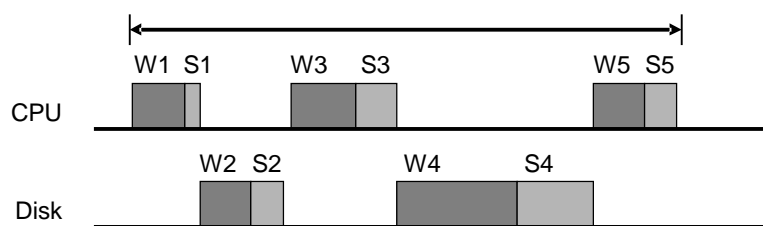


Abstrakti matemaattinen rakenne
(esim. laskentaa varten)

Yksityiskohtainen toimintakuvaus
(esim. simulointia varten)



Palvelijan vastausajan synty (1)

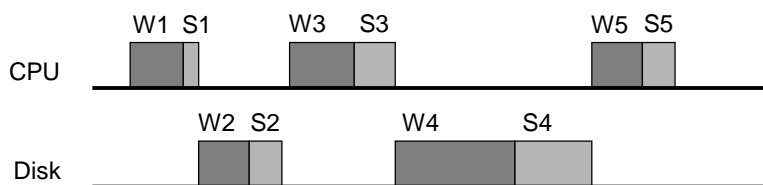


CPU :
 palvelutarve $D_{cpu} = S1+S3+S5$
 jonotusaika $Q_{cpu} = W1+W3+W5$
 läpimenoaika $R_{cpu} = D_{cpu}+Q_{cpu}$

DISK : $R_{dk} = D_{dk}+Q_{dk}$

Vastausaika $R = R_{cpu} + R_{dk}$

Palvelijan vastausajan synty (2)



$$R = D_{cpu} + Q_{cpu} + D_{dk} + Q_{dk}$$

$$= D_{cpu} + D_{dk} + Q_{cpu} + Q_{dk}$$

Kevyt kuorma $\Rightarrow R \sim R_0 = \sum D_i$

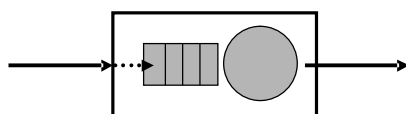
Kuorma kasvaa \Rightarrow kukin Q_i kasvaa (yli lineaarisesti)

Raskas kuorma $\Rightarrow R \sim \sum W_i$ ($\gg R_0$)

Viritys/kasvatus: D_i pienenee \Rightarrow

vaikutus riippuu laitteen i kuormituksesta

Asiakkaat jonossa - jonomalli



T mittausaika
A saapujien lkm
C poistujien lkm
B "busy time"

Operationaalinen jonomalli: kaikki suureet ovat mitattavissa

$S = B/C$ keskimääräinen palveluaika

$U = B/T$ käyttöaste

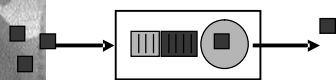
$X = C/T$ suoritusteho, läpäisy

$\lambda = A/T \sim X$
saapumistiheys

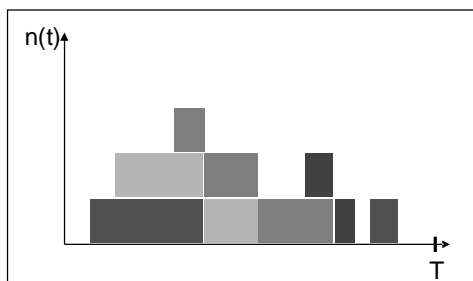
$$U = \lambda S$$

Käyttöastelaki $U = XS$

Littlen laki



Little: $N = XR$



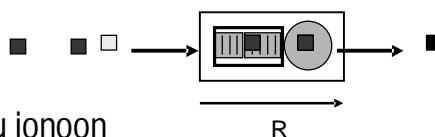
W = kokonaisoleskeluaika järjestelmässä (tumma alue)

$R = W / C$ keskimääräinen vastausaika
 $N = W / T$ keskimääräinen jonopituus (*)
 $X = C / T$ läpäisy

(*) Huom: "jononpituus" sisältää myös palveltavan

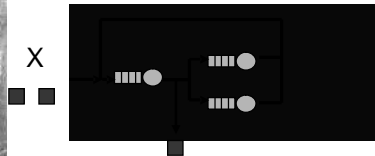
Littlen laki: "perustelu"

Saapumistiheys X



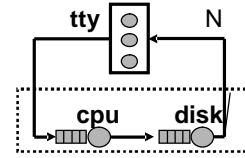
- Asiakas saapuu jonoon
 - jonon pituuden odotusarvo on N
- Asiakkaan "läpimenoaika"
 - läpimenoajan odotusarvo on R
 - tänä aikana jonoon saapuu uusia asiakkaita XR
- Asiakas poistuu jonosta
 - jälkeen jäävän jonon pituuden odotusarvo on N
- $\Rightarrow N = XR$
- Littlen laki pätee myös "varsinaiseen jonoon"

Avoin ja suljettu järjestelmä



Avoin järjestelmä

- "(ääretön) populaatio"
- asiakkaiden lkm vaihtelee
- saapumistiheys = poistumistiheys
- kysymys:
 - vastausajan käyttäytyminen
 - (käyttöaste?)



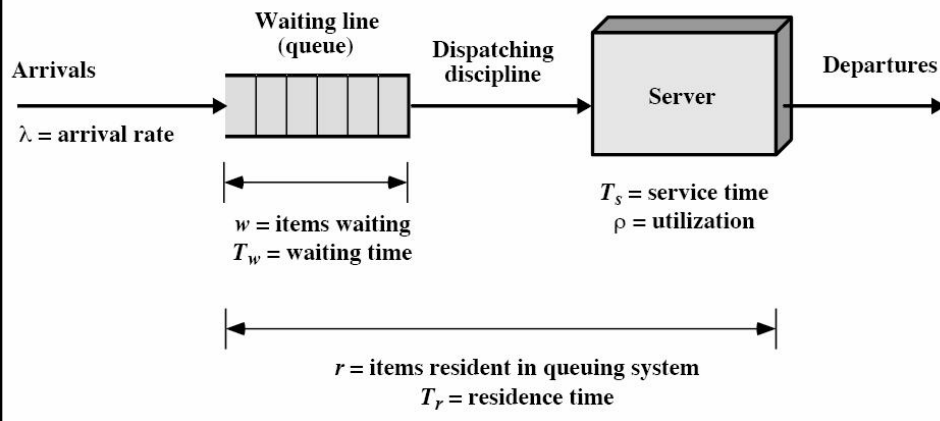
Suljettu järjestelmä

- kiinteä asiakasjoukko
- kysymys:
 - vastausajan käyttäytyminen
 - "vuon" nopeus
 - (käyttöasteet?)

Jonomalli

Fig 9.23 [Stal05]

- Lisätietoja: <http://WilliamStallings.com/StudentSupport.html>

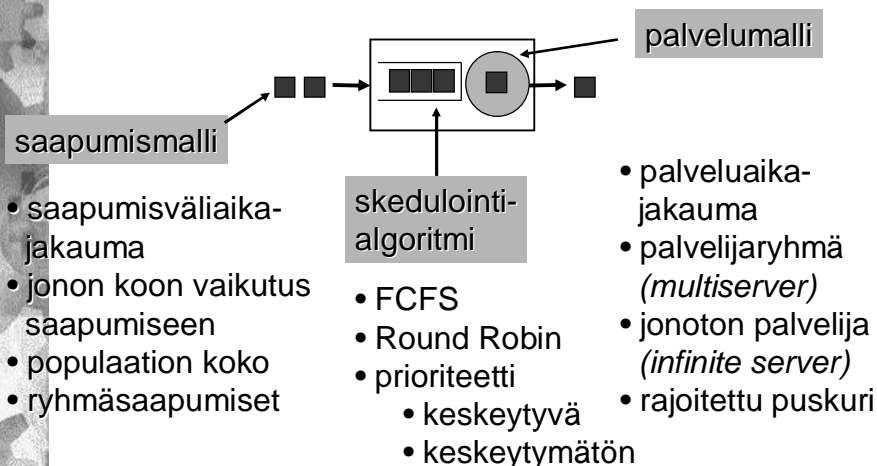


Jonomallin merkinnät

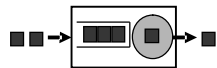
Table 9.8 Notation for Queuing Systems

λ	= arrival rate; mean number of arrivals per second
T_s	= mean service time for each arrival; amount of time being served, not counting time waiting in the queue
ρ	= utilization; fraction of time facility (server or servers) is busy
w	= mean number of items waiting to be served
T_w	= mean waiting time (including items that have to wait and items with waiting time = 0)
r	= mean number of items resident in system (waiting and being served)
T_r	= mean residence time; time an item spends in system (waiting and being served)

Jonomallin rakenne



Jonot: Kendallin notaatio



A / S / m / B / K / SD

A saapumisväliaikajakauma (M, E_k, H_k, D, G, \dots)

S palveluaikajakauma (M, E_k, H_k, D, G, \dots)

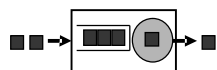
m palvelijoiden lukumäärä palvelupisteessä

B jonopuskurin koko

K populaation koko (asiakkaita maailmassa)

SD skedulointialgoritmi (service discipline)

Jonot: Kendallin notaatio (jatk)



A / S / m / B / K / SD

Jakaumamerkinnot

M eksponentiaalijakauma

E_k Erlangin jakauma

H_k hyperekspontiaalinen

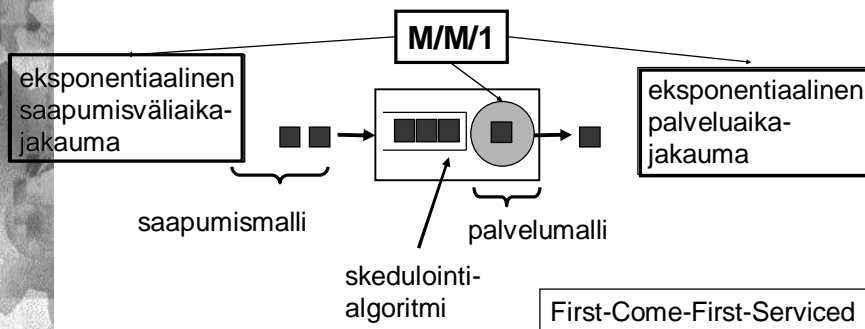
D deterministinen (yksi mahdollinen arvo)

G mielivaltainen jakauma (general)

Tavallisia jonomalleja:

$M/M/1, M/M/m, M/M/m/B, M/G/1, M/D/1$

Jonomallin ratkaisu



Järjestelmän tila: asiakkaiden lukumäärä

Mallin ratkaisu: järjestelmän tilojen todennäköisyydet

Esim: $P(0)?$, $P(\text{asiakkaiden lukumäärä} > 10)?$

Eksponentiaalinen jakauma

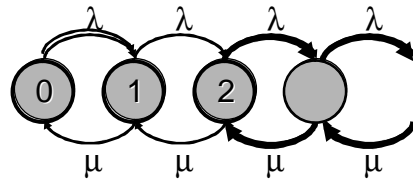
- Täysin satunnaisten tapahtuman todennäköisyys
- (Ainoa) oletus: $P\{\text{tapahtuma sattuu välillä } \Delta t\} = \lambda * \Delta t$

=>

tiheysfunktio	$f(t) = \lambda e^{-\lambda t}$	$t > 0$
kertymäfunktio	$F(t) = 1 - e^{-\lambda t}$	$t > 0$
odotusarvo $E(t)$	$1/\lambda$	
hajonta	$1/\lambda$	

- Esimerkkejä:
 - riippumattomien saapumisten väliajan jakauma
 - satunnaisten eliniän jakauma

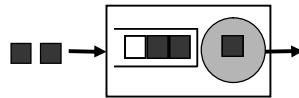
Jonon tilamalli (M/M/1)



λ saapumistiheys
 μ palvelutiheys

... HUOM: ajasta riippumattomia!
 $\rho = \lambda\mu$ käyttöaste < 1

$\lambda < \mu$



$\rho_0, \rho_1, \dots ?$
 vuotasapainoehto:
 poistumistiheys = saapumistiheys

0: $\lambda p_0 = \mu p_1$

1: $(\lambda + \mu)p_1 = \lambda p_0 + \mu p_2$

2: ...

$p_1 = \lambda/\mu p_0 = \rho p_0$

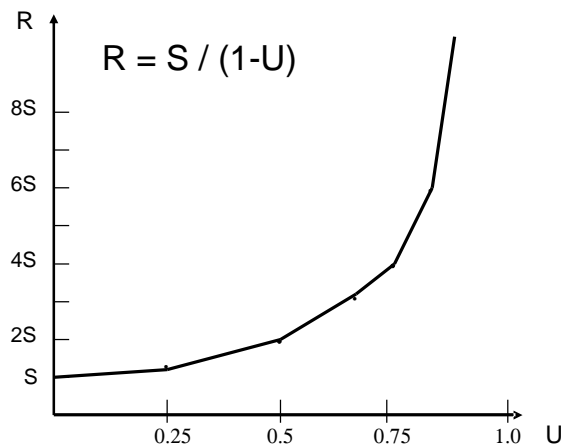
$p_2 = \rho p_1 = \rho^2 p_0$

....

$p_n = \rho^n p_0$

$p_0 = (1 - \rho) \Rightarrow p_n = \rho^n (1 - \rho)$

Käyttöasteen vaikutus



U	R
0.5	2*S
0.75	4*S
0.875	8*S

Validiteettialue:
 $\lambda < \mu$

Vuorotuksen mallinnus

- FCFS (tai FIFO) ja Round Robin mallinnettavissa, useimmat muut liian monimutkaisia mallinnettavaksi suljetulla analyttisellä mallilla
- Mallinnetaan prioriteettiskedulointia seuraavasti:
 - Kaksi prioriteetti luokkaa, keskeyttävä moniajo
 - Luokilla on eri suoritusajat
 - Graafeissa lisäoletukset:
 - Saapumistiheys λ sama molemmissa luokissa
 - Alemman luokan suoritus aika = 5* ylempään suoritus aika

Table 9.6 Formulas for Single-Server Queues with Two Priority Categories

- Assumptions:
1. Poisson arrival rate.
 2. Priority 1 items are serviced before priority 2 items.
 3. First-in-first-out dispatching for items of equal priority.
 4. No item is interrupted while being served.
 5. No items leave the queue (lost calls delayed).

(a) General Formulas (no priority)

$$\lambda = \lambda_1 + \lambda_2$$

$$\rho_1 = \lambda_1 T_{s1}; \rho_2 = \lambda_2 T_{s2}$$

$$\rho = \rho_1 + \rho_2$$

$$T_s = \frac{\lambda_1}{\lambda} T_{s1} + \frac{\lambda_2}{\lambda} T_{s2}$$

$$T_r = \frac{\lambda_1}{\lambda} T_1 + \frac{\lambda_2}{\lambda} T_2$$

HUOM:
Eri kaavat estävälle
ja keskeyttävälle

b) No interrupts; exponential service times

$$T_{r1} = T_{s1} + \frac{\rho_2 T_{s1} + \rho_2 T_{s2}}{1 - \rho_1}$$

$$T_{r2} = T_{s2} + \frac{T_{s1} - T_{s2}}{1 - \rho}$$

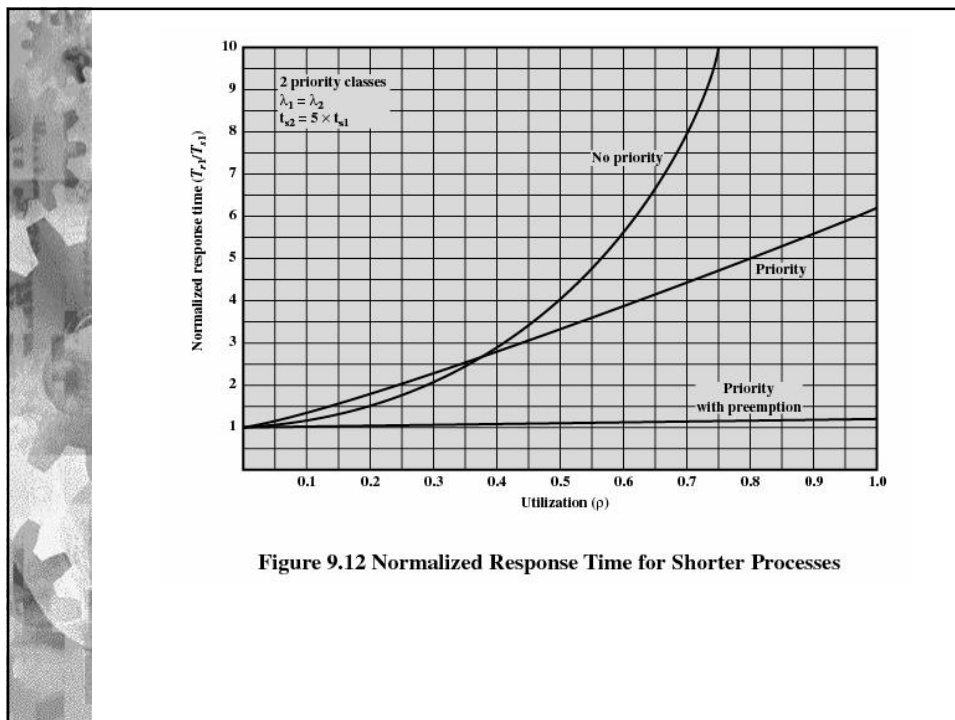
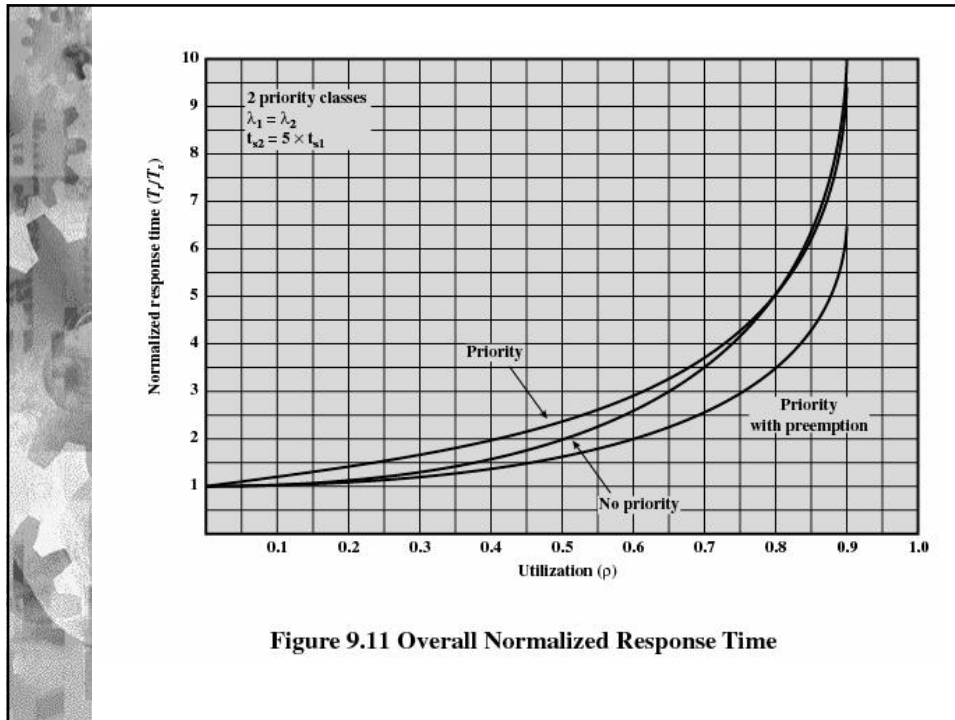
(priority)

(c) Preemptive-resume queuing discipline; exponential service times

(priority with preemption)

$$T_{r1} = T_{s1} + \frac{\rho_1 T_{s1}}{1 - \rho_1}$$

$$T_{r2} = T_{s2} + \frac{1}{1 - \rho_1} \left(\rho_1 T_{s2} + \frac{\rho T_{s2}}{1 - \rho} \right)$$



Simulointiesimerkki

- HUOM: Simuloinnin tulokset pätevät vain kokeillulle kuormalle.
- Tässä esimerkissä:
 - Saapumistiheys λ ja
 - Palveluaika T_s satunnaistettuja:
 - Arvotaan kullekin prosessille erikseen!

50000 prosessia

$$\lambda = 0.8$$

$$T_s = 1 \quad \text{Keskiarvoja!}$$

$$\rho = \lambda T_s = 0.8$$

Tuloksia

Y-akseli on normalisoitu: Kokonaisaika on jaettu suoritusajalla

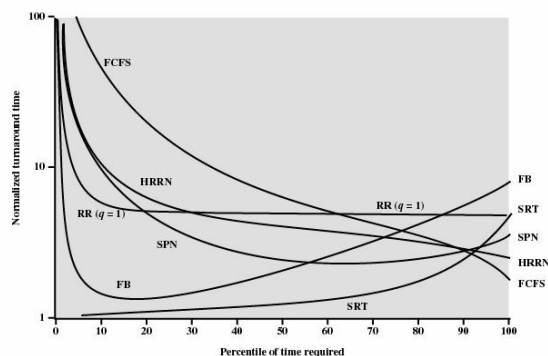
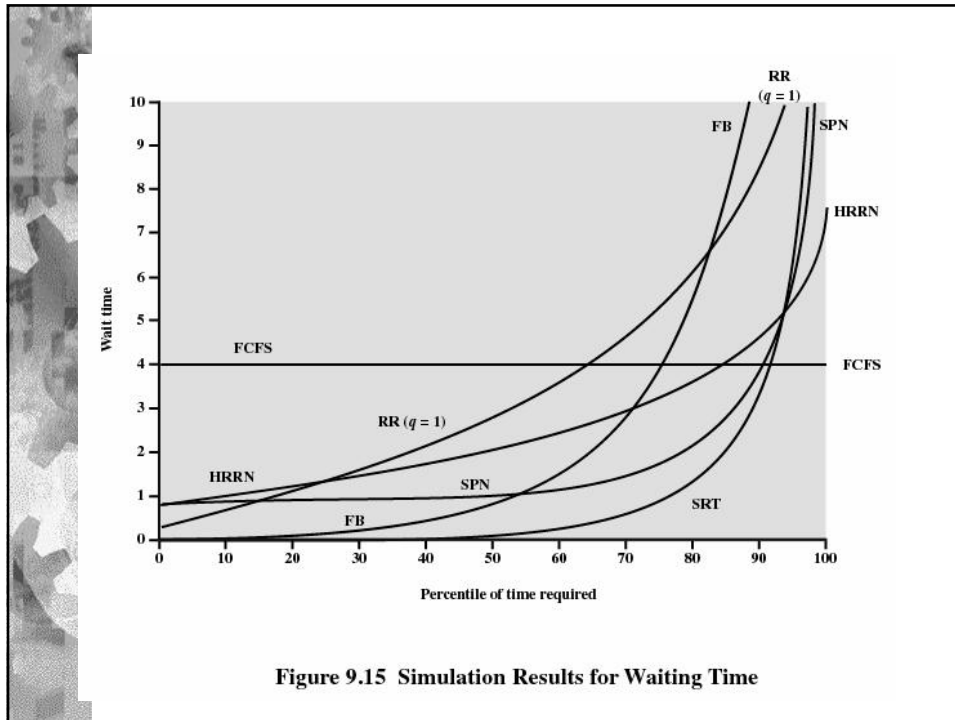


Figure 9.14 Simulation Results for Normalized Turnaround Time

- RR: pitkällä töillä vakiollinen käyttäytyminen (noin 5 kertaa suoritusajaa)
- FCFS: noin 1/3 töistä (Lyhyimmät) yli 10-kertainen läpimeno suoritusajaa nähden



Kertaus

Tärkeitä asioita 1/2

- KJ:n perusrakenne
- Prosessi ja säie
 - Kuvaaja, suoritus, tilan ja kontekstin vaihto
- Muistinhallinta
 - MMUn rakenne
 - Eri menetelmät
 - Muistin varaukset, osoitteenmuunnos
- Virtuaalimuistimekanismi
 - Sivutuksen periaate, osoitteenmuunnos
 - PTR, Sivutaulu, osoitteenpuutoskeskeytys
 - Algoritmit ja politiikat

Tärkeitä asioita 2/2

- Paikallisuus
- Keskeytysmekanismi
- Moniajo
- Etuoikeutettu tila

Muita asioita

- UNIX, Linux, Windows
- KJ:n historia
- Synkronoinnin ja poissulkemisen mekanismit
- Välimuisti

- Käy läpi kirjan "Review Questions!"
 - Jos osaat vastata kaikkiin hyvin, niin kokeessakin pitäisi pärjätä.
- Varmista, että osaat myös laskuharjoitustehtävät

Kertausta Osa2:
(vanhoja kalvoja)

Kerrosmalli

- Peruskäyttäjä näkee vain sovellukset
 - Niiden alla on varusohjelmia ja käyttöjärjestelmä
- Sovellusohjelmoija näkee ja käyttää palvelurajapintaa (esim. funktiokirjastona)
- Käyttöjärjestelmä hallinnoi laitteistoa sovellusten puolesta

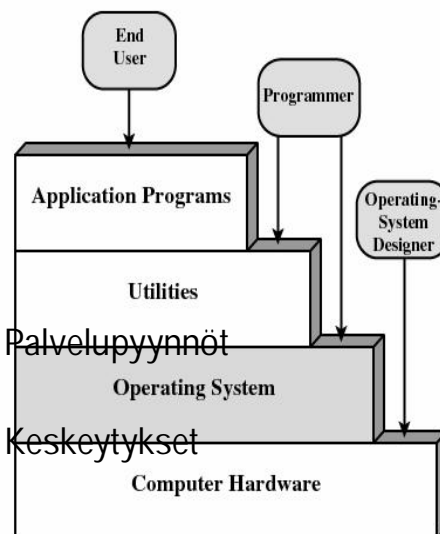
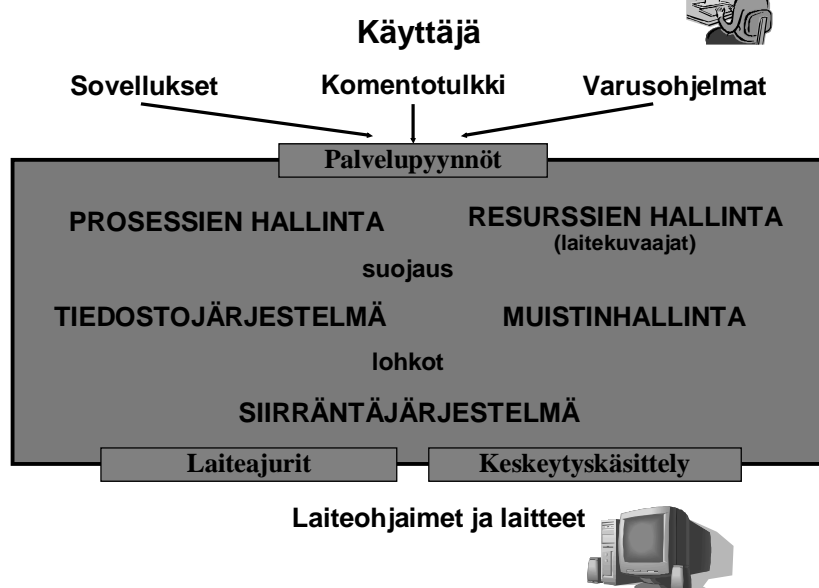


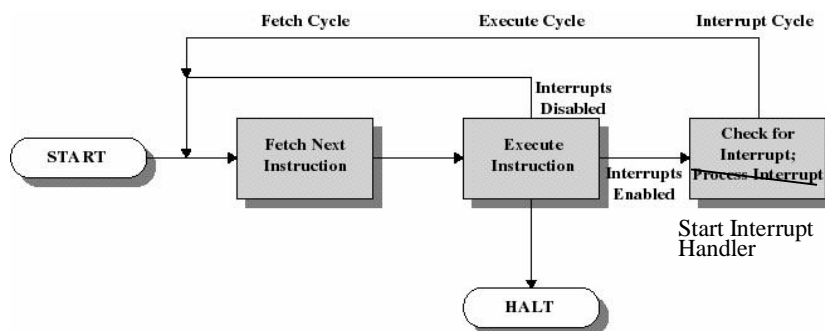
Figure 2.1 Layers and Views of a Computer System

Keskeiset KJ:n osa-alueet



Käskesykli (täydennetty)

Kuva 1.7



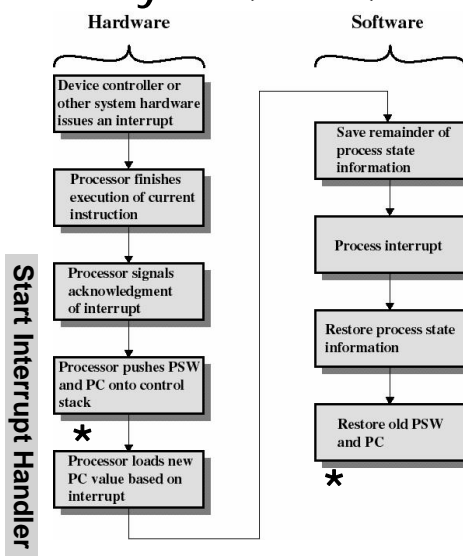
- Jos keskeytys sallittu, CPU tutkii tilarekisterin ennen seuraavan käskyn noutoa
- Jos keskeytys, suorita keskeytyskäsitteilyn käsky

Keskeytyskäsitteily

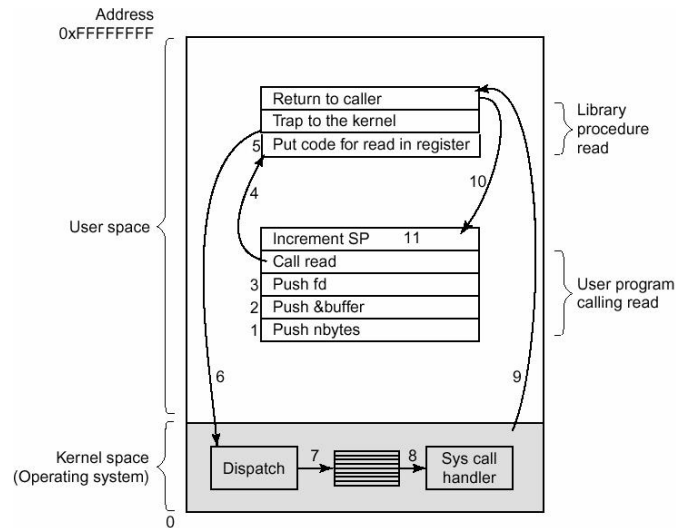
(kuva 1.10)

Kirja s. 21-25

- * Etuoikeutettu tila vs. käyttäjätila
- * Keskeytysten esto vs. salliminen

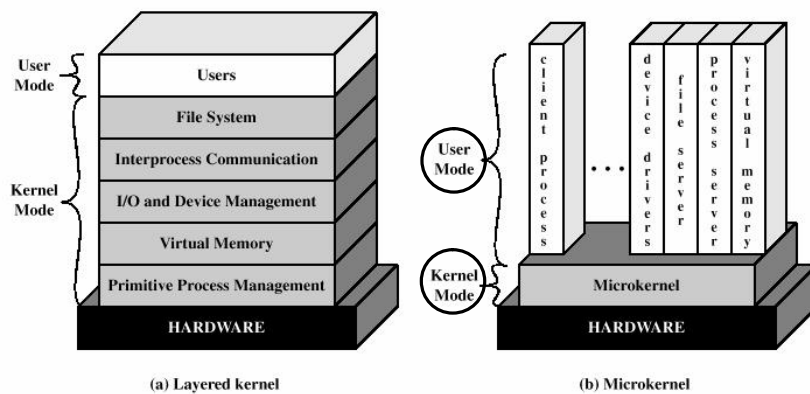


Palvelupyyntö: read(fd, buffer, nbytes) Tan01 1-17

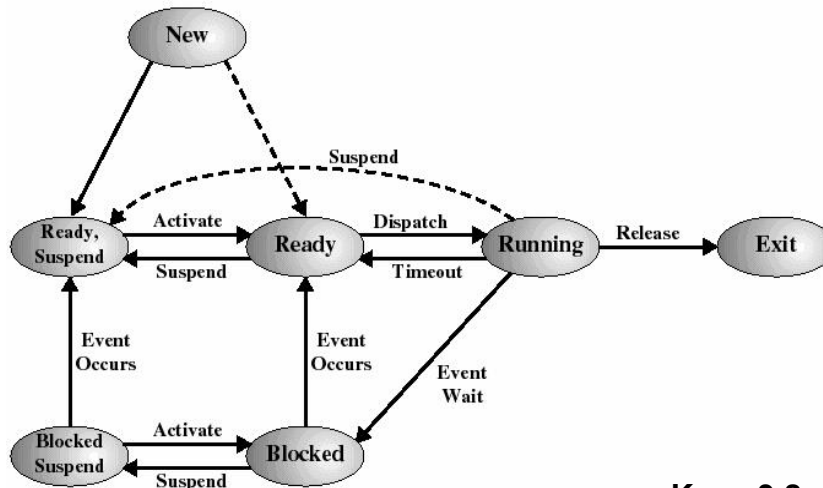


KJ:n ydin

(Fig 4.10 [Stal 05])

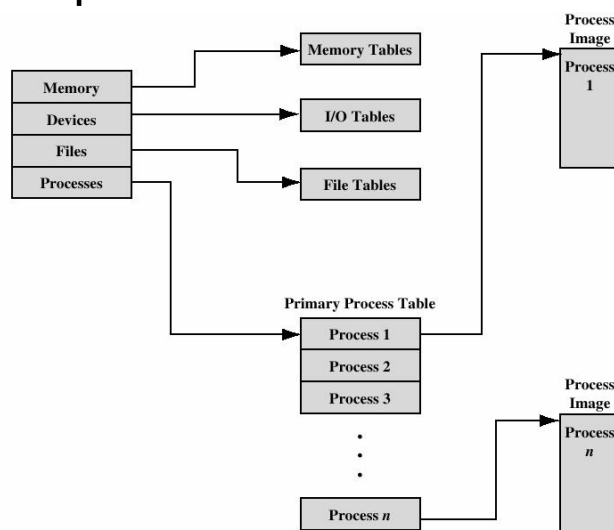


Prosessin tilakaavio (7 tilaa)

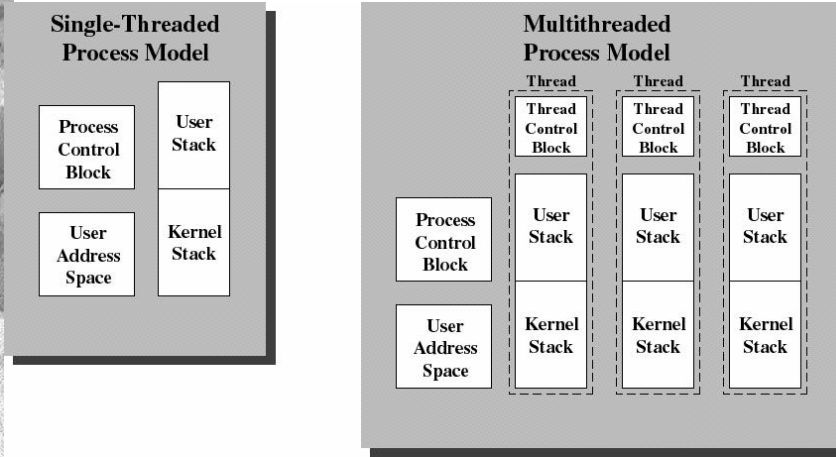


Kuva 3.8

KJ:n perustietorakenteita Kuva 3.10



Yksi säie vs. Monta säiettä



Säikeen kuvaaja TCB
tallealue rekistereille, prioriteetti, tila, ...

Solaris

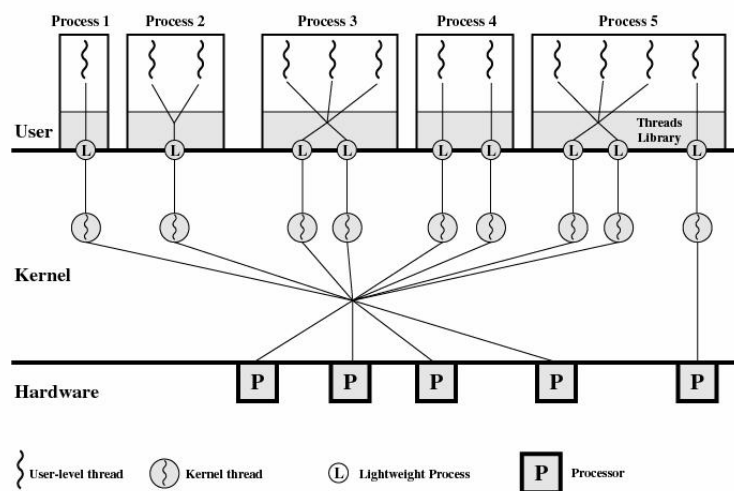
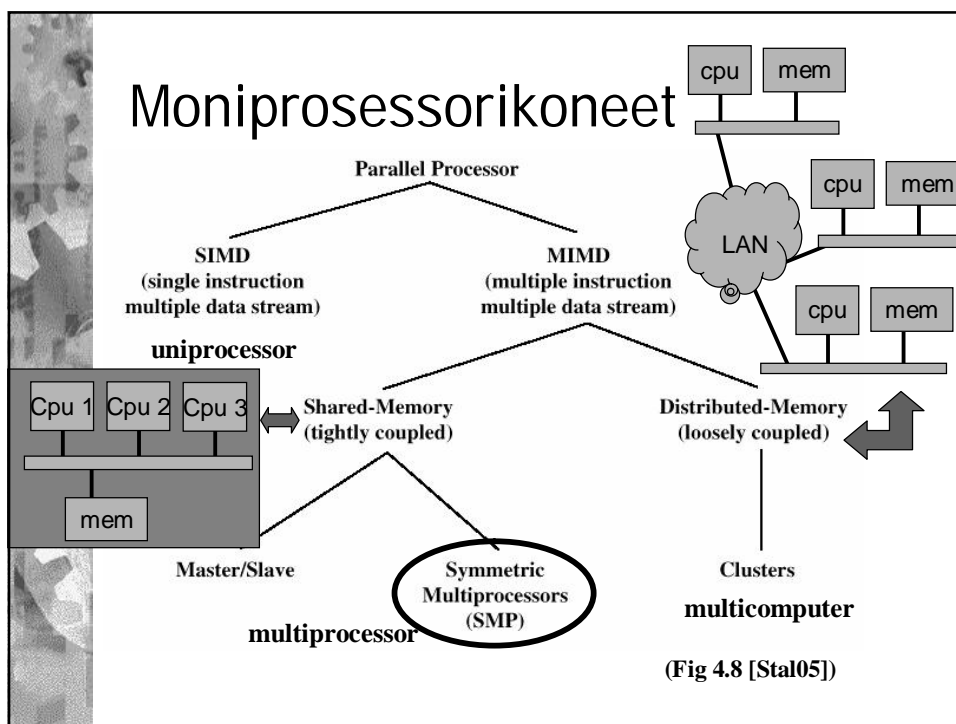


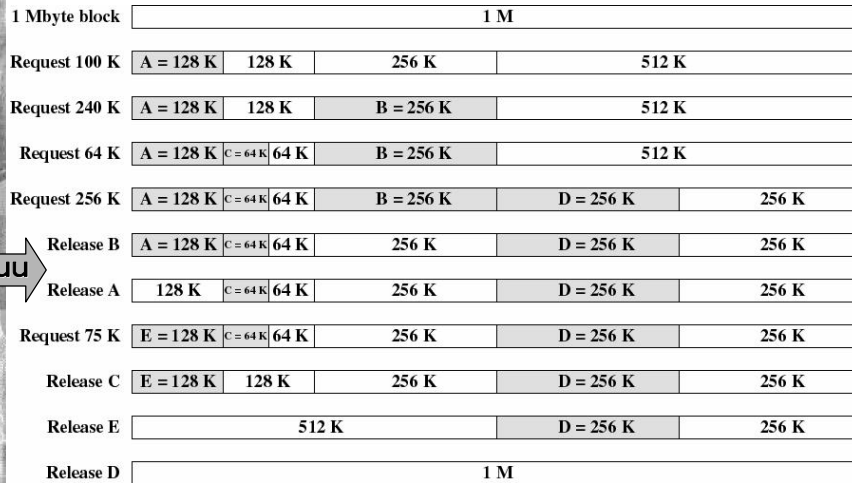
Figure 4.15 Solaris Multithreaded Architecture Example



Yksinkertainen muistinhallinta

Menetelmä	kuvaus	vahvuudet	heikkoudet
Kiinteä partitio	Muisti jaettu etukäteen osiin. Prosessi vain yhdessä osassa.	helppo toteutus	sisäinen pirstoutuminen maksimi prosessimäärä rajoitettu
Dynaaminen partitio	Muistia varataan tarpeen mukaan. Prosessi vain yhdessä osassa.	ei sis. pirst. par. muistin käyttöaste	ulkoinen pirstoutuminen, tiivistämistarve
Buddy System	Muistinvar. dyn., mutta kiinteänkokoisina osina. Prosessi vain yhdessä osassa.	ei juurikaan ulkoista pirstoutumista	vähäinen sisäinen pirstoutuminen
Yks. segmentointi	Prosessi jaettu segmentteihin. Segm. sijoitettavissa vapaasti.	ei sis. pirst. par. muistin käyttöaste	ulkoinen pirstoutuminen
Yks. sivutus	Prosessi ja muisti jaettu sivuihin. Sij. vapaasti	ei ulk. pirst.	hyvin vähän sis. pirst. (vain viimeinen sivu)

Buddy System: esimerkki



Kuva 7.6

2-tasoinen sivutaulu

- Ylin hakemisto mahtuu yhteen sivuun, aina muistissa

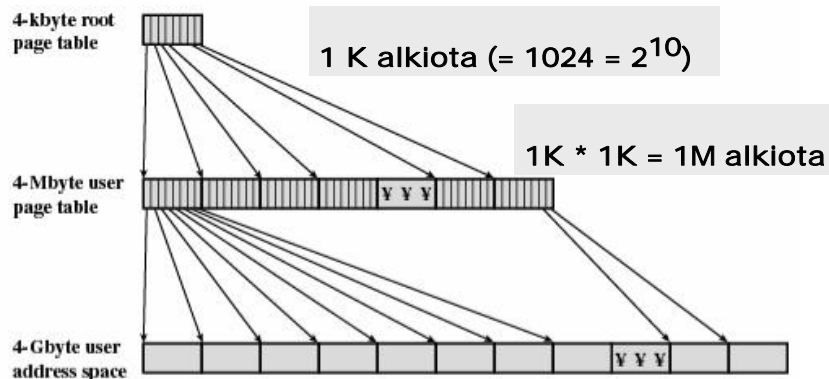


Figure 8.4 A Two-Level Hierarchical Page Table

5. ed

Käänteinen sivutaulu v.2

- Sivutilan (kehyksen) numero
 - Suoraan taulun indeksi
 - Ei tarvitse tallettaa tauluun

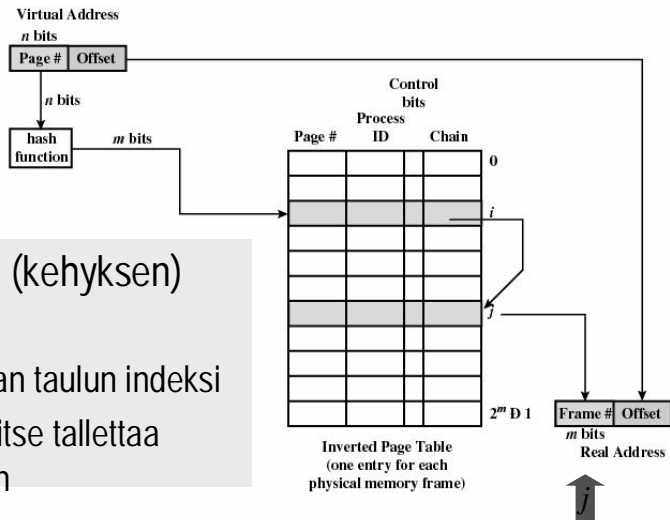
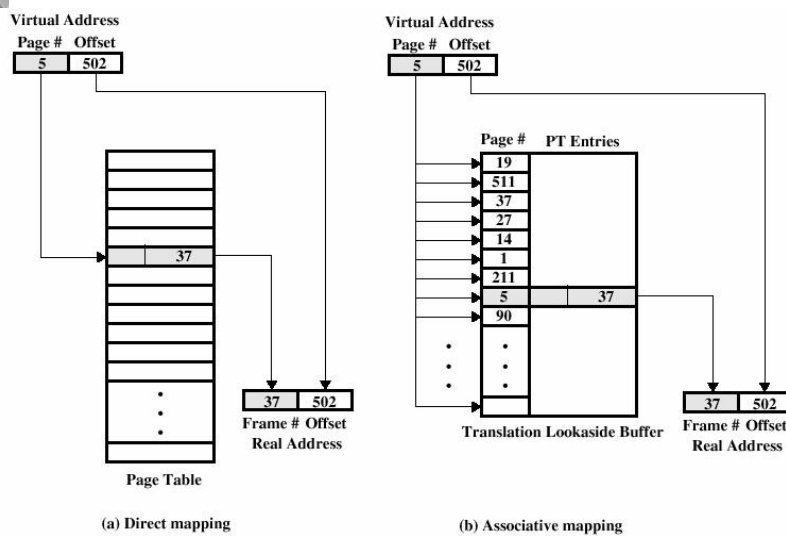
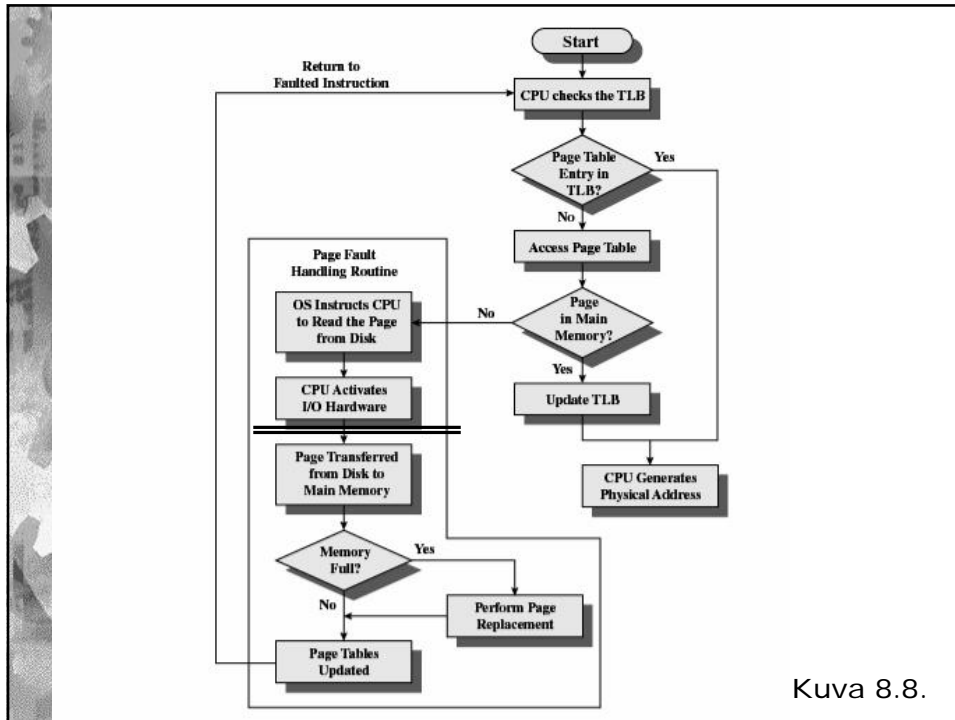


Figure 8.6 Inverted Page Table Structure

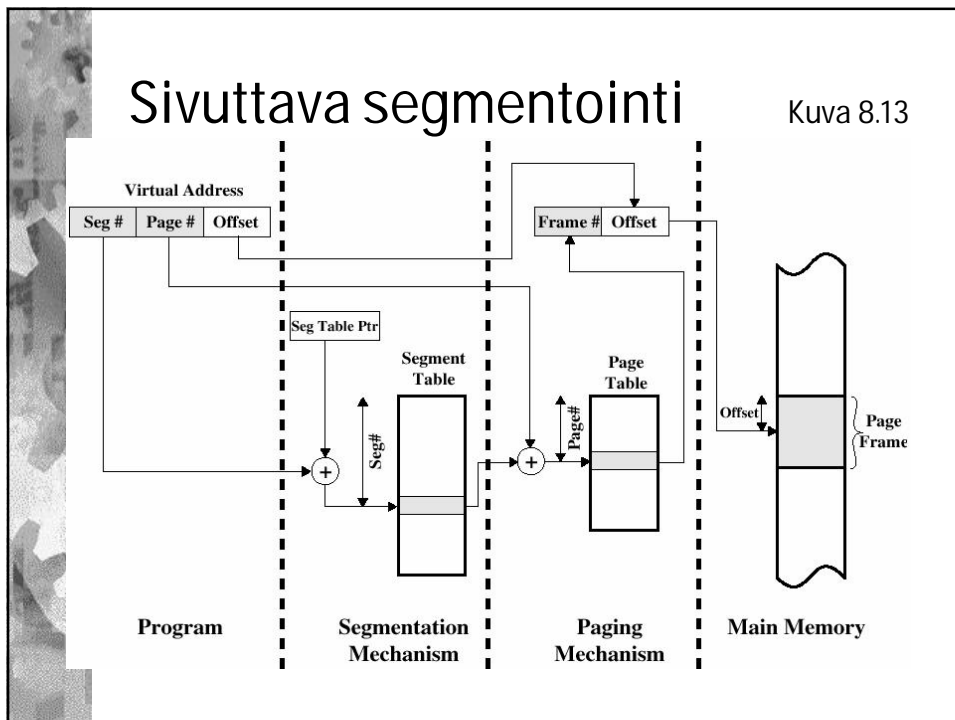
Etsintä TLB:stä

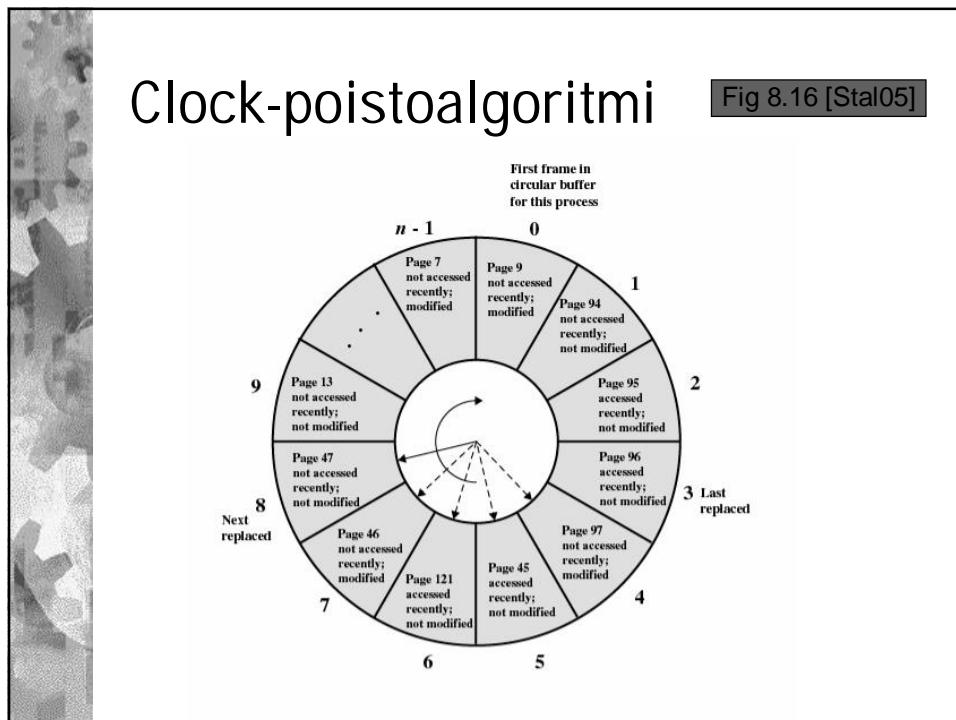
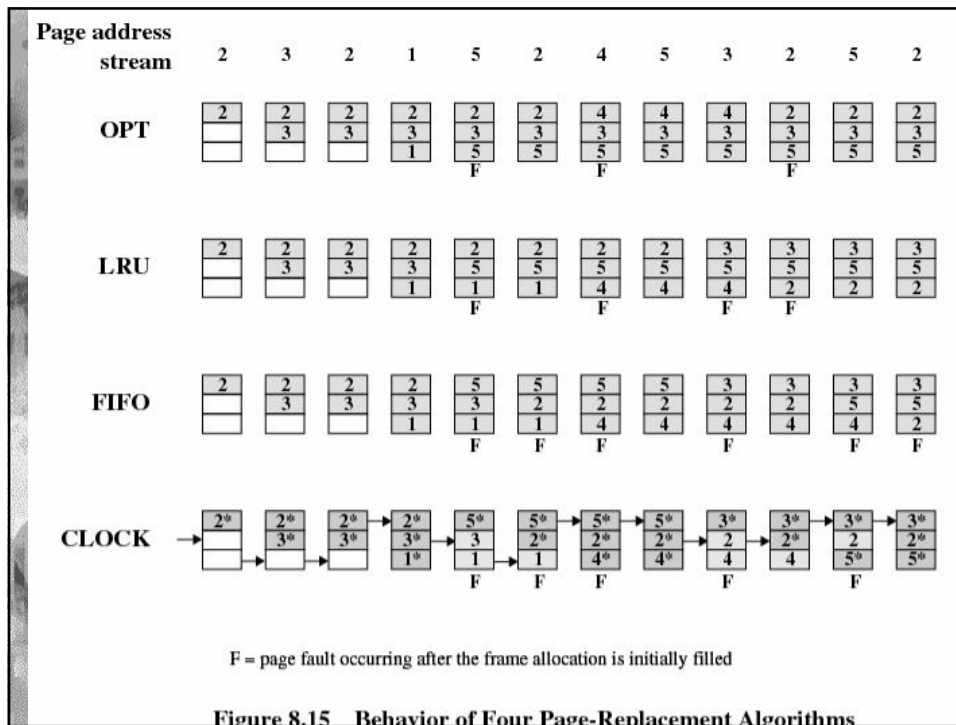
Kuva 8.9





Kuva 8.8.





Esimerkki: Käyttöjoukko

Sequence of
Page
References

Window Size, Δ

Fig 8.19 [Stal05]

	2	3	4	5
24	24	24	24	24
15	24 15	24 15	24 15	24 15
18	15 18	24 15 18	24 15 18	24 15 18
23	18 23	15 18 23	24 15 18 23	24 15 18 23
24	23 24	18 23 24	•	•
17	24 17	23 24 17	18 23 24 17	15 18 23 24 17
18	17 18	24 17 18	•	18 23 24 17
24	18 24	•	24 17 18	•
18	•	18 24	•	24 17 18
17	18 17	24 18 17	•	•
17	17	18 17	•	•
15	17 15	17 15	18 17 15	24 18 17 15
24	15 24	17 15 24	17 15 24	•
17	24 17	•	•	17 15 24
24	•	24 17	•	•
18	24 18	17 24 18	17 24 18	15 17 24 18