WEEK 6

Scheduling

Stallings, Chapter 9

1

Table 9.2 Scheduling Criteria

User Oriented, Performance Related

Turnaround time This is the interval of time between the submission of a process and its completion. Includes actual execution time plus time spent waiting for resources, including the processor. This is an appropriate measure for a batch job.

Response time For an interactive process, this is the time from the submission of a request until the response begins to be received. Often a process can begin producing some output to the user while continuing to process the request. Thus, this is a better measure than turnaround time from the user's point of view. The scheduling discipline should attempt to achieve low response time and to maximize the number of interactive users receiving acceptable response time.

Deadlines When process completion deadlines can be specified, the scheduling discipline should subordinate other goals to that of maximizing the percentage of deadlines met.

User Oriented, Other

Predictability A given job should run in about the same amount of time and at about the same cost regardless of the load on the system. A wide variation in response time or turnaround time is distracting to users. It may signal a wide swing in system workloads or the need for system tuning to cure instabilities.

2

System Oriented, Performance Related

Throughput The scheduling policy should attempt to maximize the number of processes completed per unit of time. This is a measure of how much work is being performed. This clearly depends on the average length of a process but is also influenced by the scheduling policy, which may affect utilization.

Processor utilization This is the percentage of time that the processor is busy. For an expensive shared system, this is a significant criterion. In single-user systems and in some other systems, such as real-time systems, this criterion is less important than some of the others.

System Oriented, Other

Fairness In the absence of guidance from the user or other system-supplied guidance, processes should be treated the same, and no process should suffer starvation.

Enforcing priorities When processes are assigned priorities, the scheduling policy should favor higher-priority processes.

Balancing resources The scheduling policy should keep the resources of the system busy. Processes that will underutilize stressed resources should be favored. This criterion also involves medium-term and long-term scheduling.

Tbl 9.2 (alaosa)

3

When?

- Long-term
  - Accept new process?
  - Enough memory? swap?
- Medium-term
  - Bring a suspended process back, when?
  - Enough free memory?
  - MPL, multiprocessing level?
- Short-term
  - Which process is switched to processor?
- I/O
  - Service order of I/O requests (of processes) ?

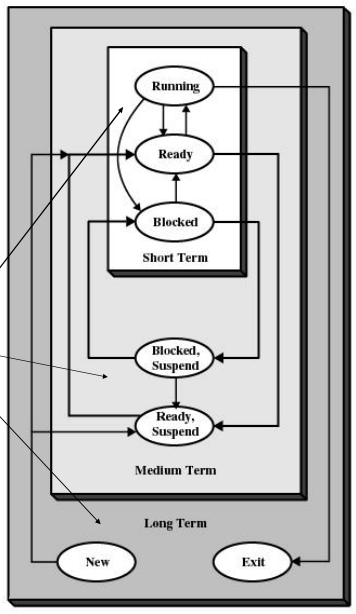
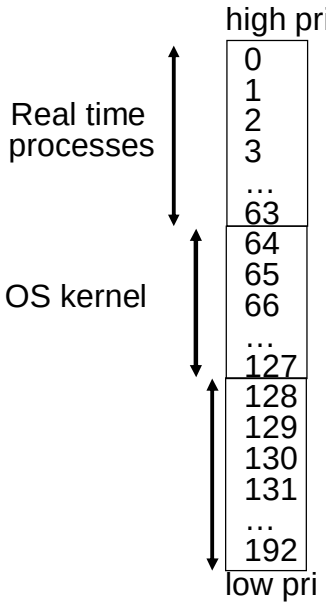


Figure 9.2 Levels of Scheduling

Priorities



5

CPU Scheduling: Algorithms examples

- First-Come-First-Served FCFS
- Round Robin RR
- Virtual Round Robin VRR
- Shortest Process Next SPN
- Shortest Remaining Time SRT
- Highest Response Ratio Next HRRN
- Multilevel Feedback feedback

- Fair Share Scheduling FSS

6

Esimerkkiprosessit

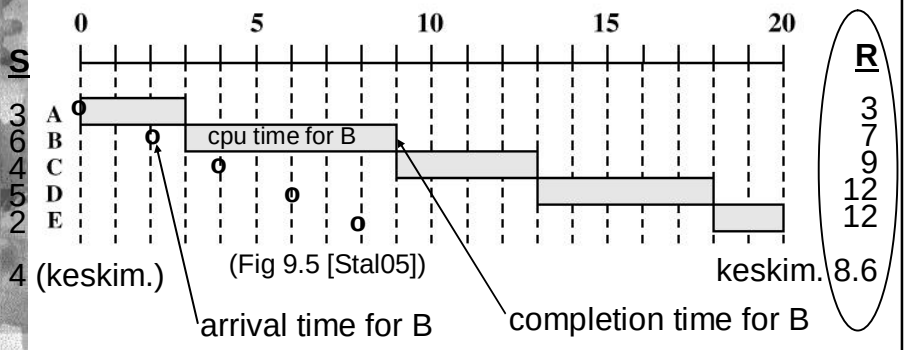
(Tbl 9.4 [Stal05])

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

- Service Time = CPU:ssa kulutettu aika
- Esimerkeissä ei mietitä I/O:n vaikutusta

7

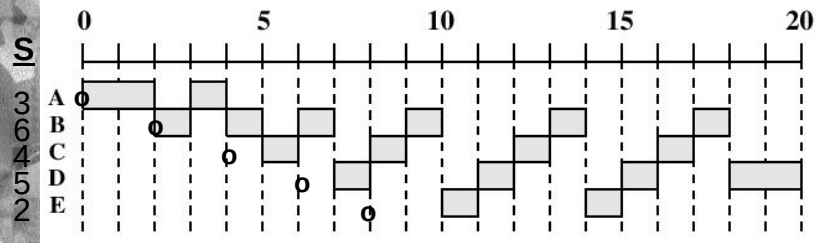
FCFS – First Come First Served



(Fig 9.5 [Stal05])

8

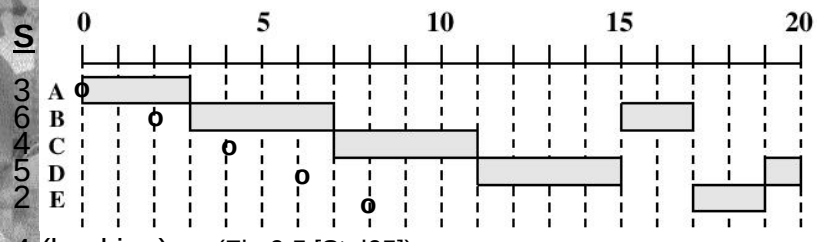
RR – Round Robin



4 (keskim.) (Fig 9.5 [Stal05]) keskim. 10.8

9

RR – effect of the time slice



4 (keskim.) (Fig 9.5 [Stal05]) keskim. 10.0

10

Virtual RR:n jonomalli

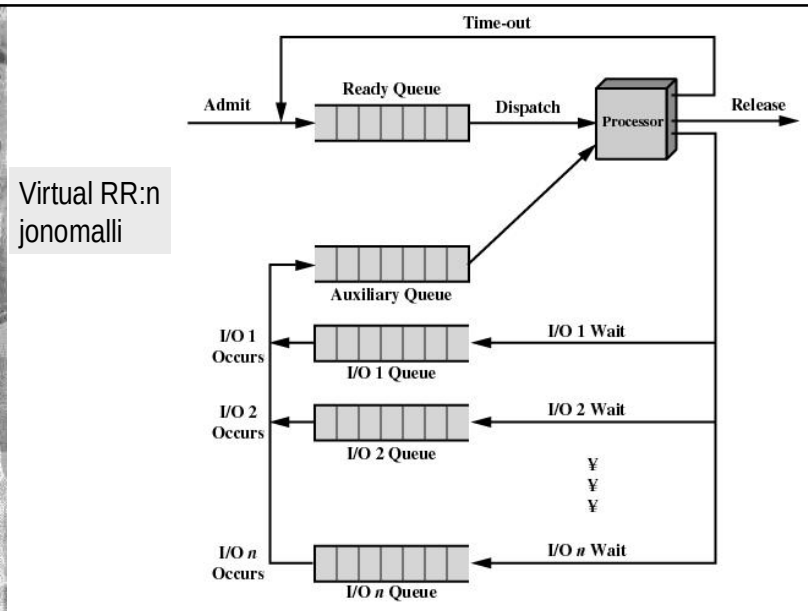
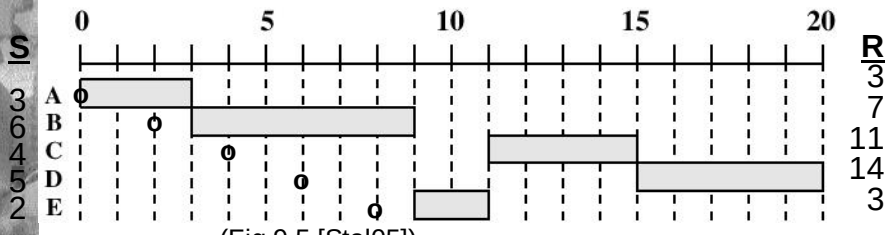


Figure 9.7 Queuing Diagram for Virtual Round-Robin Scheduler

11

SPN – Shortest Process Next



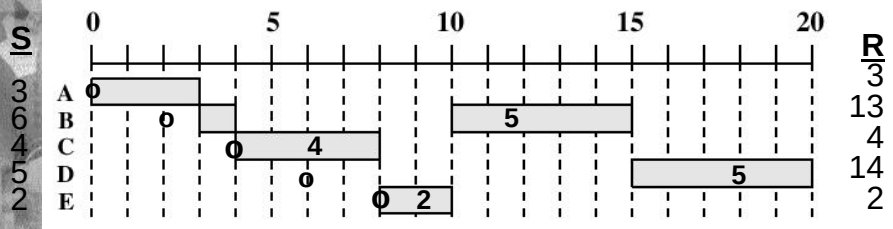
(Fig 9.5 [Stal05])

4 (keskim.)

keskim. 7.6

12

SRT – Shortest Remaining Time



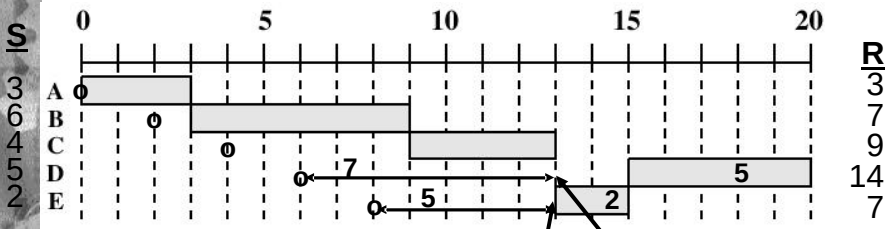
4 (keskim.)

(Fig 9.5 [Stal05])

keskim. 7.2

13

HRRN – Highest Response Ratio Next



4 (keskim.)

(Fig 9.5 [Stal05])

keskim. 8.0

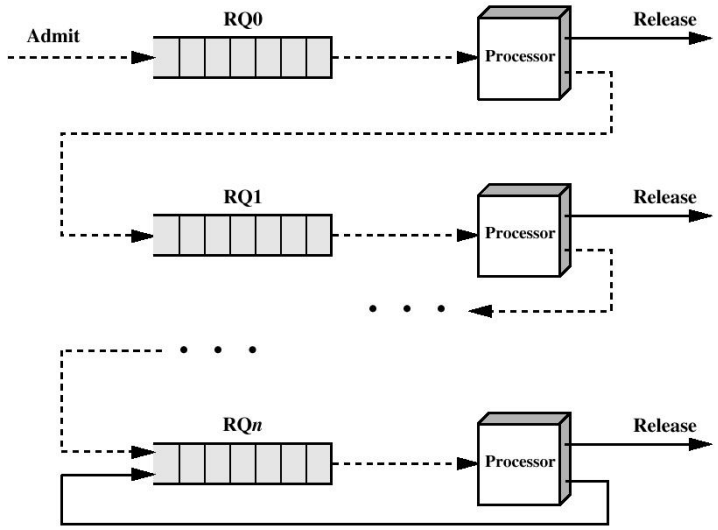
(5+2)/2 > (7+5)/5

$$\text{response ratio} = \frac{\text{time spent waiting CPU + expected service time}}{\text{expected service time}}$$

14

Multilevel Feedback

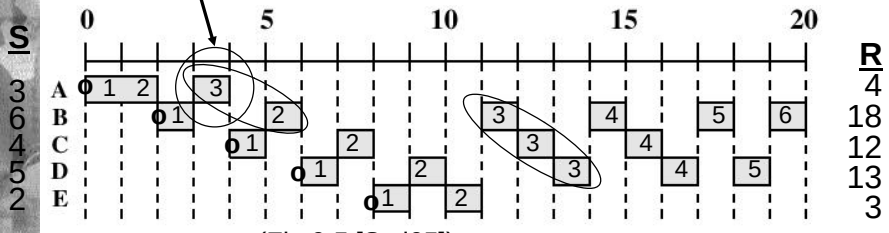
(Fig 9.10 [Stal05])



15

Feedback q=1

Virhe?



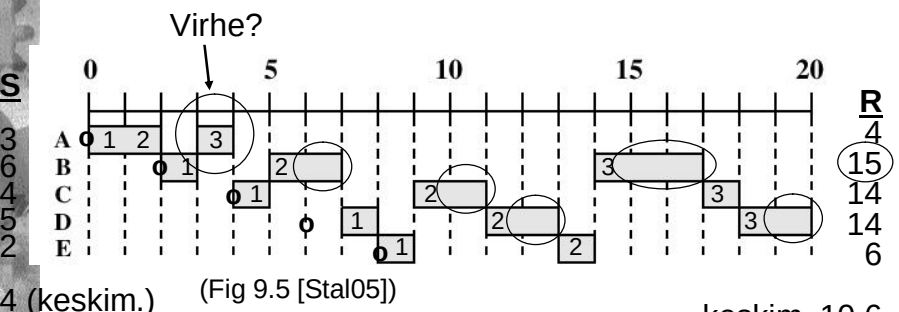
4 (keskim.)

(Fig 9.5 [Stal05])

keskim. 10.0

16

Feedback $q=2^i$



(Fig 9.5 [Stal05])

Fig 9.5 [Stal05]

17

Yhteenveto

Tbl 9.3 [Stal05]

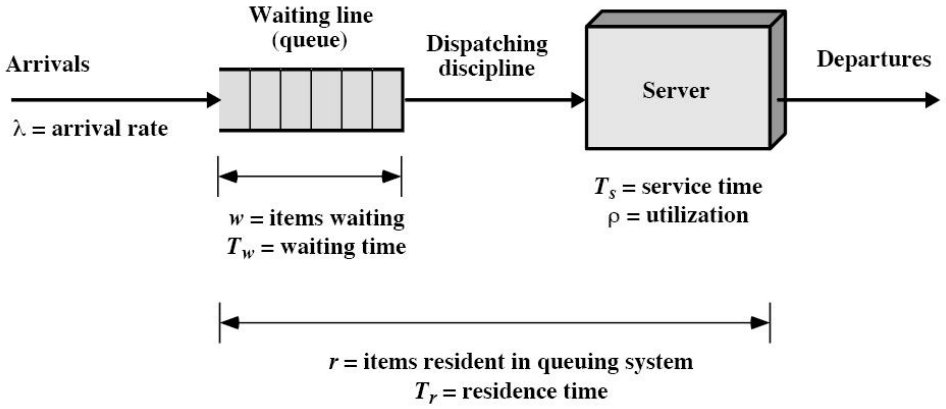
	Selection Function	Decision Mode	Throughput	Response Time	Overhead	Effect on Processes	Starvation
FCFS	max[w]	Nonpreemptive	Not emphasized	May be high, especially if there is a large variance in process execution times	Minimum	Penalizes short processes; penalizes I/O bound processes	No
Round Robin	constant	Preemptive (at time quantum)	May be low if quantum is too small	Provides good response time for short processes	Minimum	Fair treatment	No
SPN	min[s]	Nonpreemptive	High	Provides good response time for short processes	Can be high	Penalizes long processes	Possible
SRT	min[$s-e$]	Preemptive (at arrival)	High	Provides good response time	Can be high	Penalizes long processes	Possible
HRRN	$\max\left(\frac{w+s}{s}\right)$	Nonpreemptive	High	Provides good response time	Can be high	Good balance	No
Feedback	(see text)	Preemptive (at time quantum)	Not emphasized	Not emphasized	Can be high	May favor I/O bound processes	Possible

w = time spent ~~in system so far~~, waiting ~~and executing~~
e = time spent in execution so far
s = total service time required by the process, including e

Queuing systems

Fig 9.23 [Stal05]

- See: http://WilliamStallings.com/StudentSupport.html

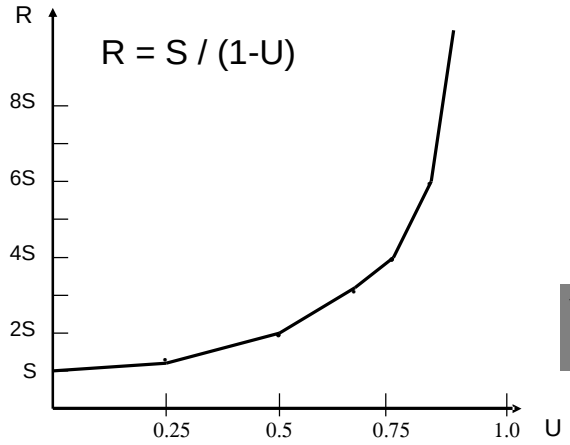


Jonomallin merkinnät

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)

Käyttöasteen vaikutus



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

Validiteettialue:
$\lambda < \mu$

21

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)

$$\lambda = \lambda_1 + \lambda_2$$
$$\rho_1 = \lambda_1 T_{s1};\ \rho_2 = \lambda_2 T_{s2}$$
$$\rho = \rho_1 + \rho_2$$
$$T_s = \frac{\lambda_1}{\lambda} T_{s1} + \frac{\lambda_2}{\lambda} T_{s2}$$
$$T_r = \frac{\lambda_1}{\lambda} T_{r1} + \frac{\lambda_2}{\lambda} T_{r2}$$

HUOM:
Eri kaavat estävälle ja keskeyttävälle

b) No interrupts; exponential service times

$$T_{r1} = T_{s1} + \frac{\rho_1 T_{s1} + \rho_2 T_{s2}}{1-\rho_1}$$
$$T_{r2} = T_{s2} + \frac{T_s - T_{s1}}{1-\rho}$$

(priority)

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

$$T_{r1} = T_{s1} + \frac{\rho_1 T_{s1}}{1-\rho_1}$$
$$T_{r2} = T_{s2} + \frac{1}{1-\rho_1}\left(\rho_1 T_{s2} + \frac{\rho T_s}{1-\rho}\right)$$

(priority with preemption)

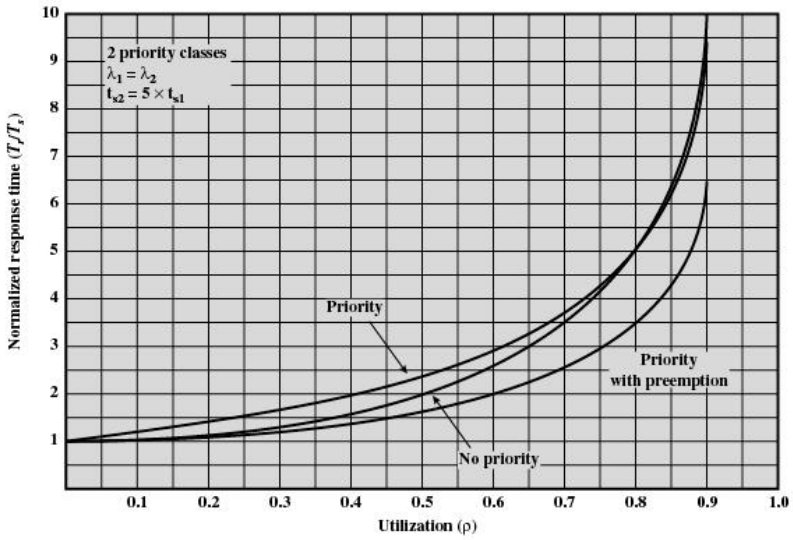


Figure 9.11 Overall Normalized Response Time

23

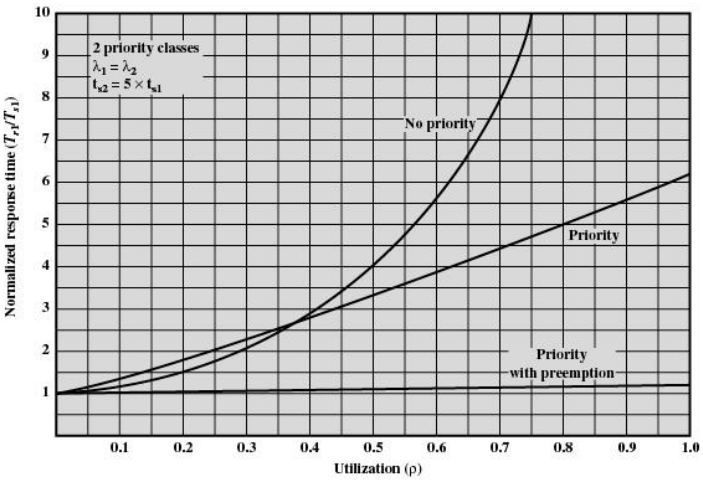


Figure 9.12 Normalized Response Time for Shorter Processes

24

Simulointiesimerkki

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

$$50000\,prosessia$$
$$\lambda = 0.8$$
$$T_s = 1$$

Odotusarvoja!

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

25

Tuloksia

Y-akseli on normalisoitu: Kokonaisaika on jaettu suoritusajalla

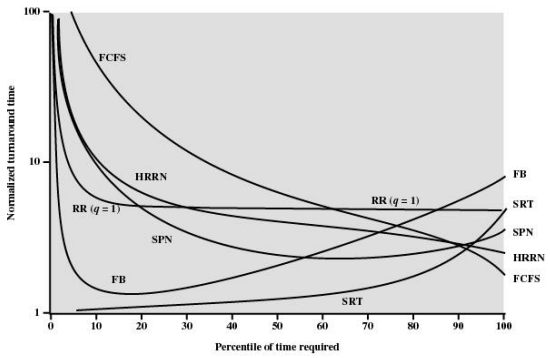


Figure 9.14 Simulation Results for Normalized Turnaround Time

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

26

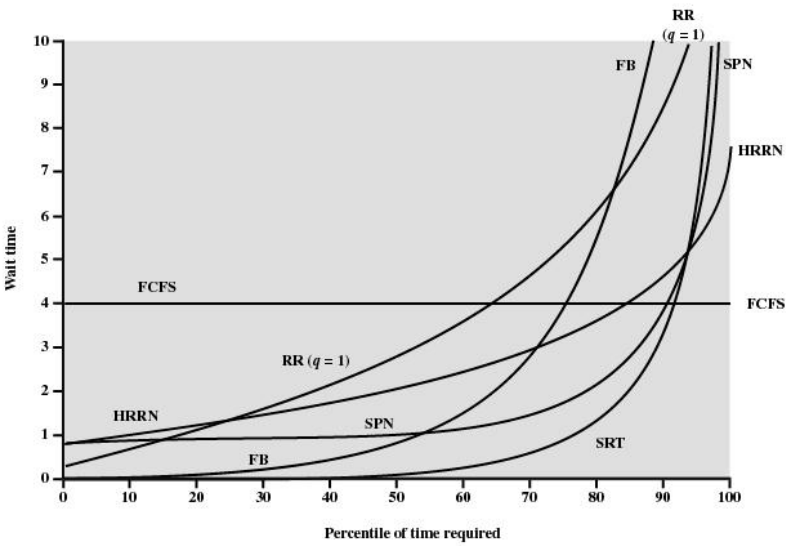


Figure 9.15 Simulation Results for Waiting Time

27

Kertaus

28

Important themes 1/2

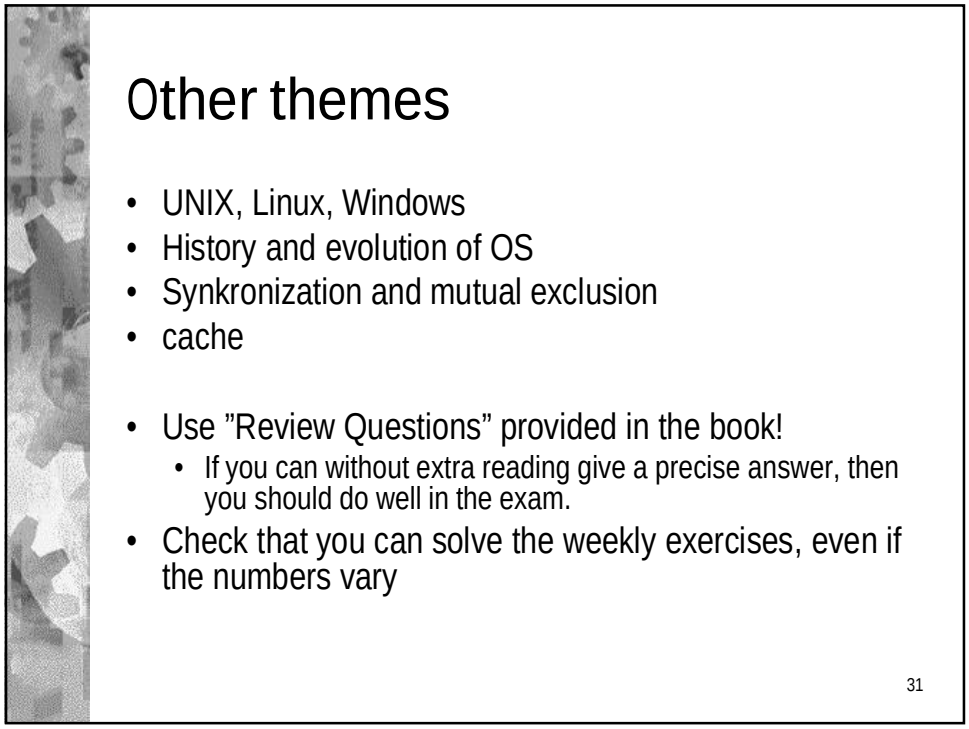
- Structure of OS
- Process and thread
  - PCB, TCB, execution, mode and context switch
- Memory management
  - MMU's structure
  - Different methods
    - Memory allocation, address translation
- Virtual memory
  - Paging, address translation
  - Page table, page fault, PTR, TLB
  - Policies, methods and algorithms

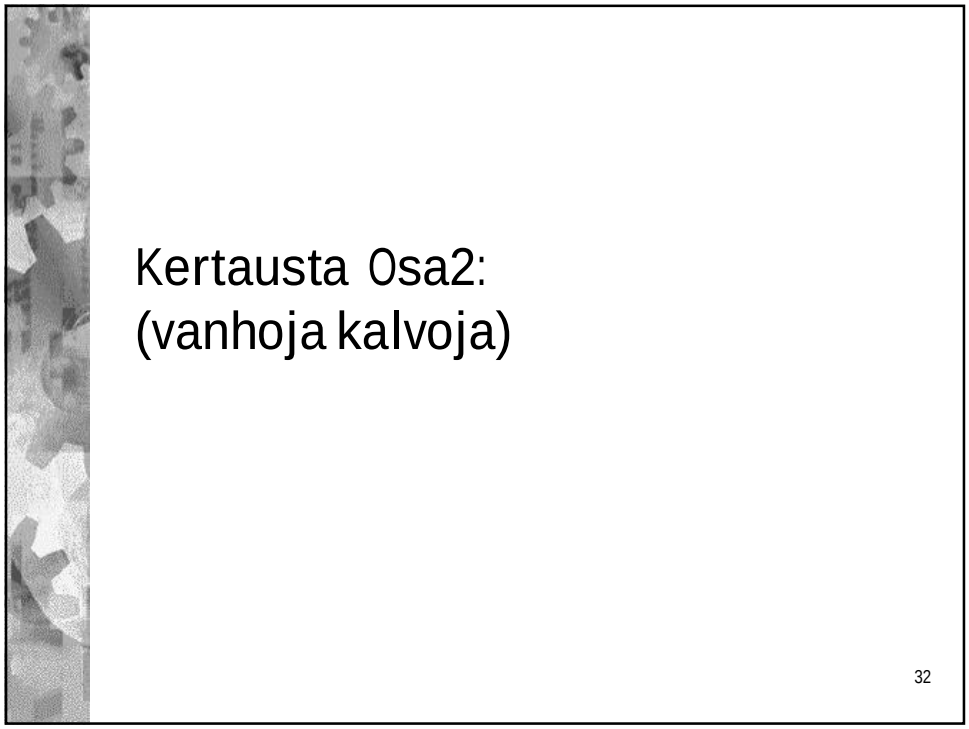
29

Important themes 2/2

- Locality
- Interrupts and interrupt handling, execution cycle
- Multiprogramming
- User mode, kernel mode

30


Other themes
• UNIX, Linux, Windows
• History and evolution of OS
• Synkronization and mutual exclusion
• cache
• Use ”Review Questions” provided in the book!
• If you can without extra reading give a precise answer, then you should do well in the exam.
• Check that you can solve the weekly exercises, even if the numbers vary
31


Kertausta Osa2:
(vanhoja kalvoja)
32

Layer model

- System calls
- Interrupts

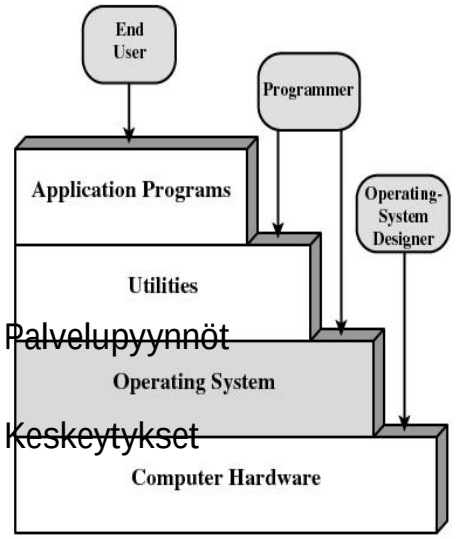


Figure 2.1 Layers and Views of a Computer System

OS structure

user

applications shell Varusohjelmat

System call

PROCESS MANAGEMENT RESOURCE MANAGEMENT

protection

FILSE SYSTEM MEMORY MANAGEMENT

blocks

I/O MANAGEMENT

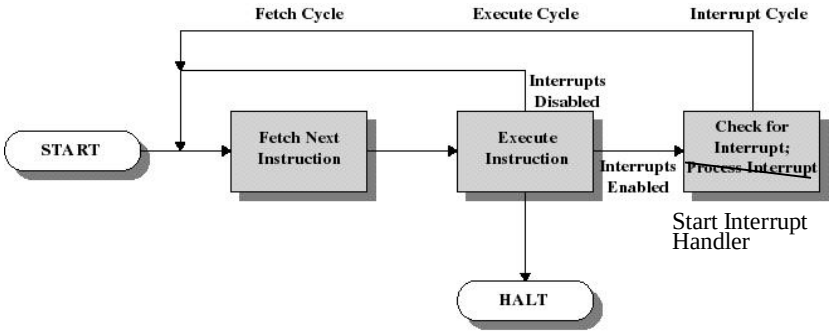
Laiteajurit Keskeytyskäsittely

Laiteohjaimet ja laitteet

34

Execution cycle

Kuva 1.7



35

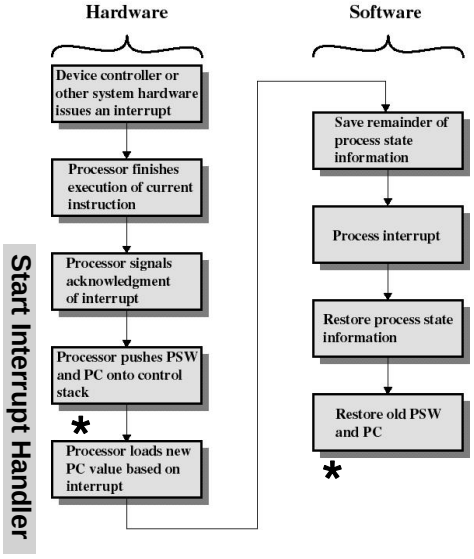
Interrupt handling

(kuva 1.10)

Book pages 21-25

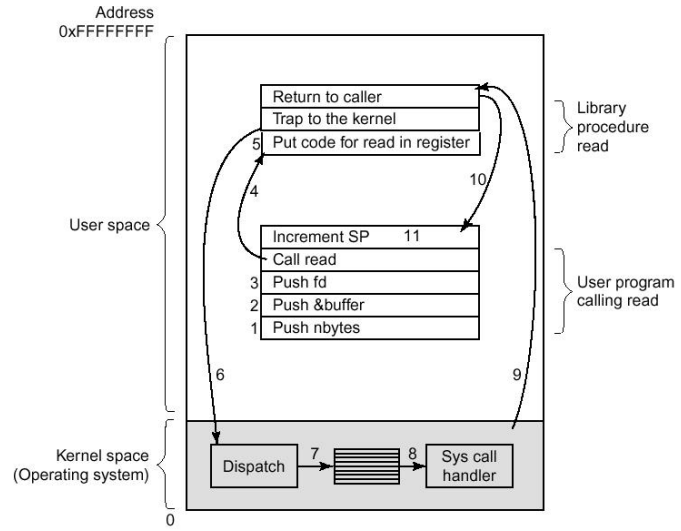
*** Kernel mode vs user mode**

*** Disabling and enabling interrupts**



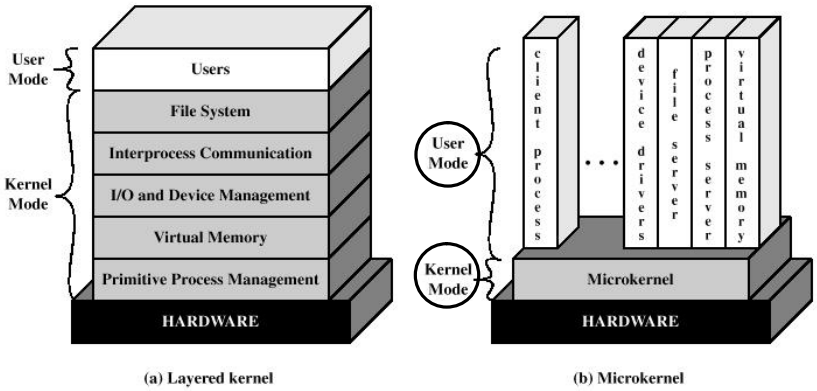
36

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



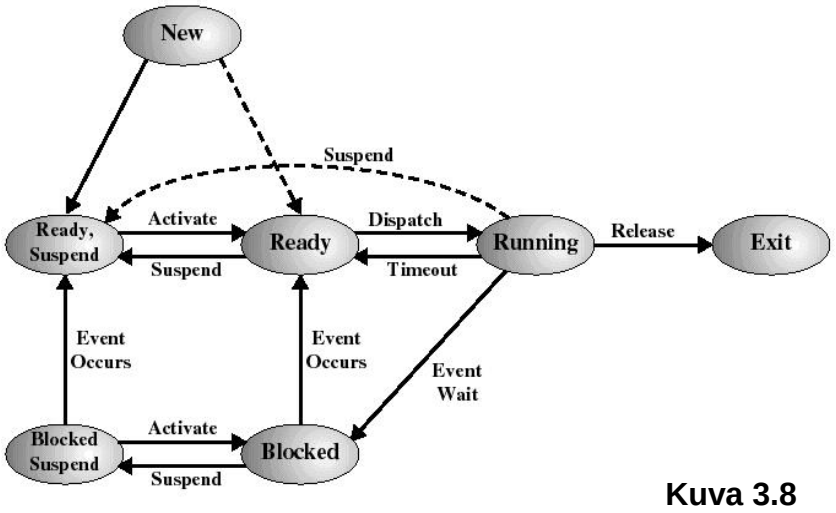
37

Kernel (Fig 4.10 [Stal 05])



38

Process states

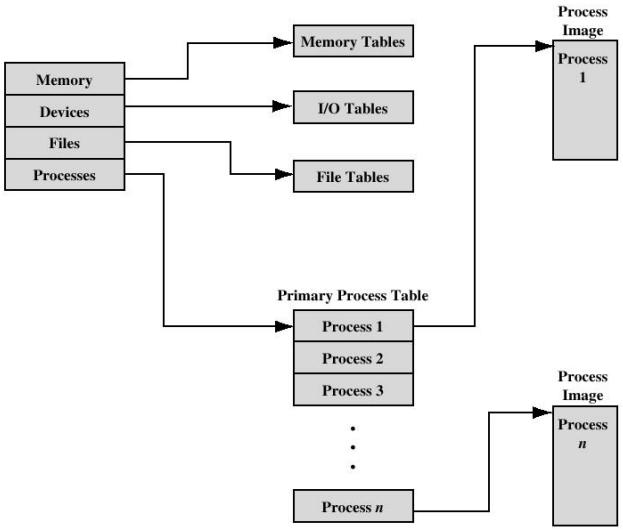


Kuva 3.8

39

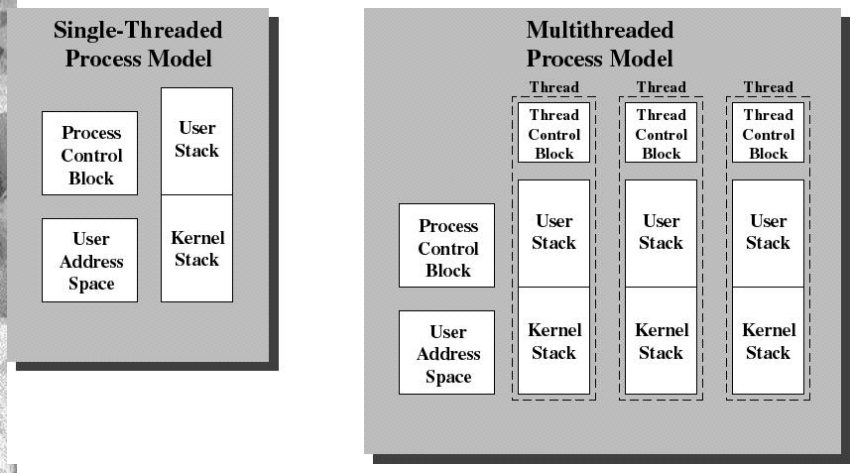
OS data structures

Kuva 3.10



40

Threads



TCB

41

Solaris

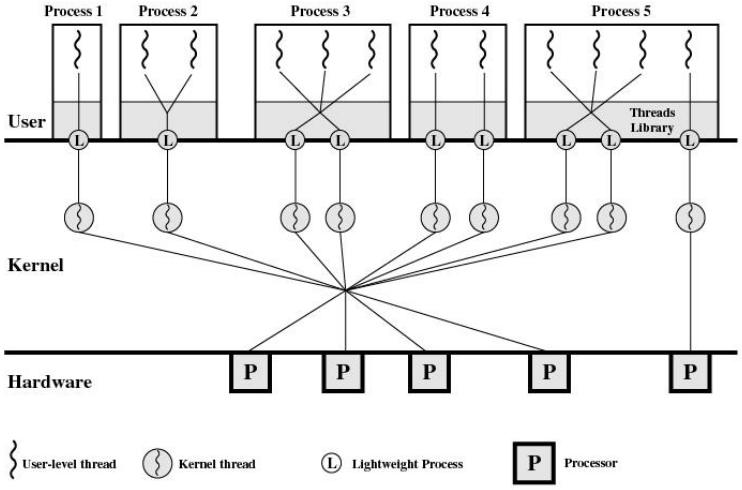


Figure 4.15 Solaris Multithreaded Architecture Example

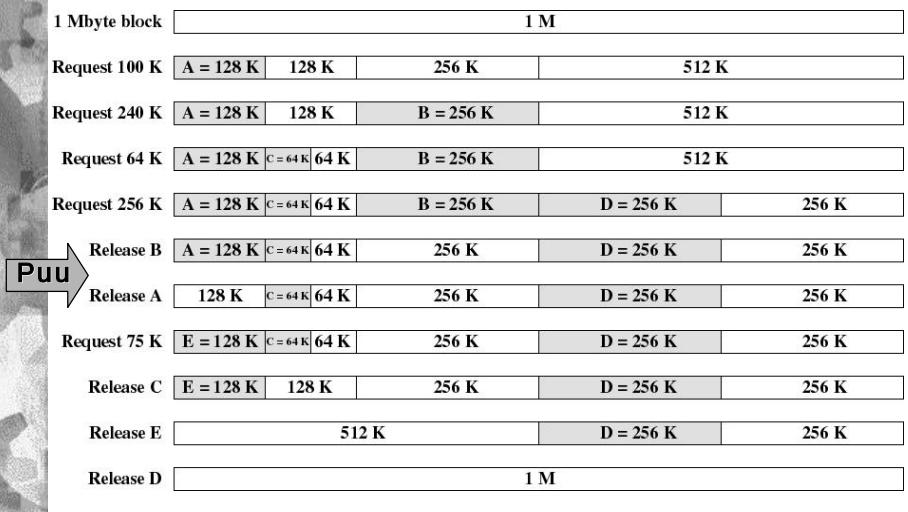
42

(Simple) memory management

Technique	Description	Strengths	Weaknesses
Fixed partitioning	**Memory divided to static partitions at system generation time**	**Simple to implement, little overhead**	**Internal fragmentation**
Dynamic partitioning	**Allocated when needed and only the amount requested**	**No internal fragmentation**	**External fragmentation and need for compaction**
Buddy System	**Dynamic allocation using fixed sizes. Process in one allocated area**	**Practically no external fragmentation**	**Small amount of internal fragmentation**
Simple segmentation	**Process divided to segments. Each segment allocated dynamically**	**No internal fragmentation**	**External fragmentation**
Simple paging	**Memory divided to frames, process to pages**	**No external fragmentation**	**Internl fragmentation only on the last page**

43

Buddy System

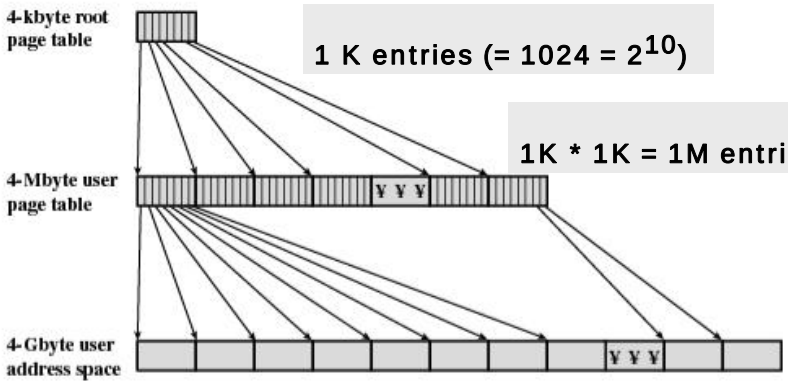


Kuva 7.6

44

Two-level hierarchical page table

- Top most level in one page and always in the memory

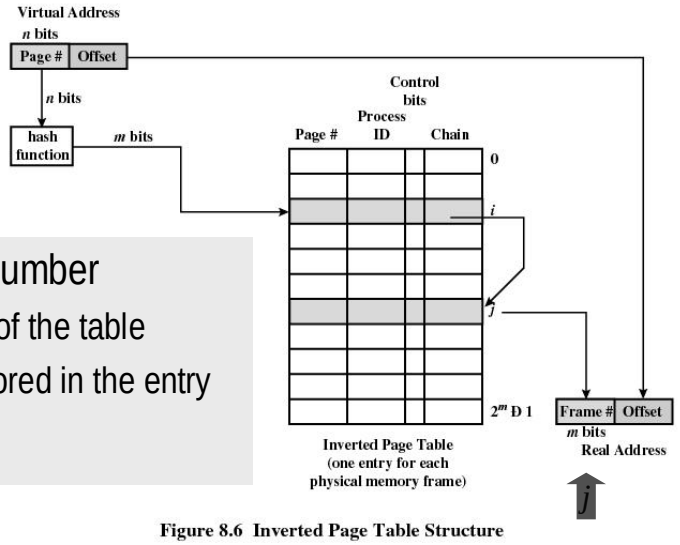


5

Figure 8.4 A Two-Level Hierarchical Page Table

Inverted page table v.2

5. ed



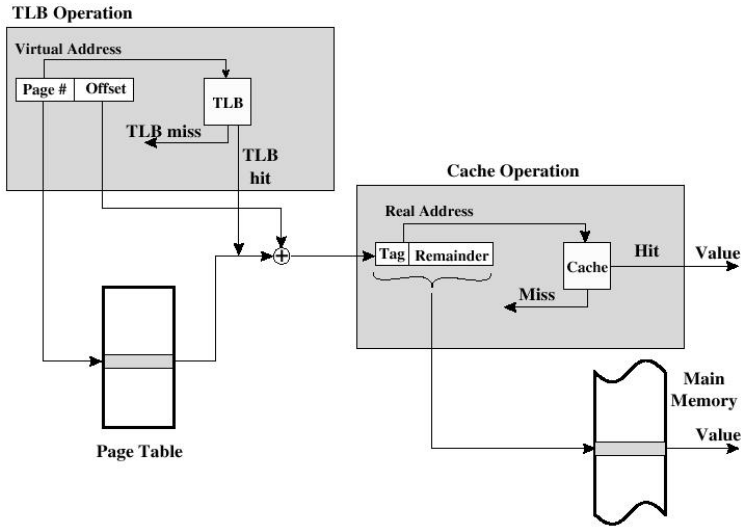
- Frame number
  - Index of the table
  - Not stored in the entry

Figure 8.6 Inverted Page Table Structure

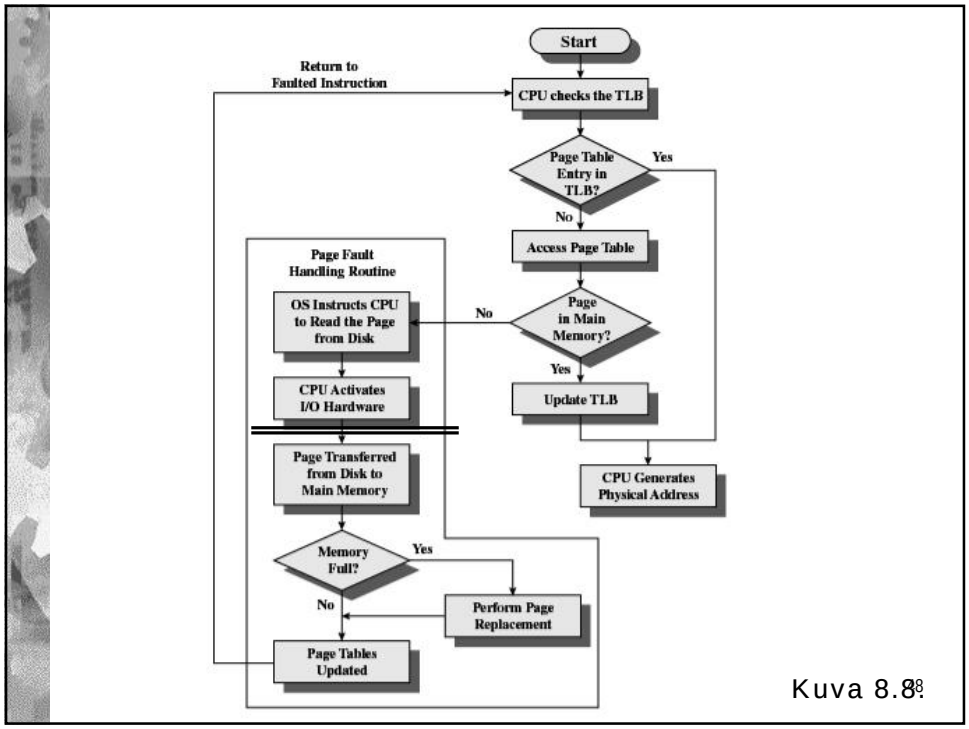
46

TLB and cache

Kuva 8.10



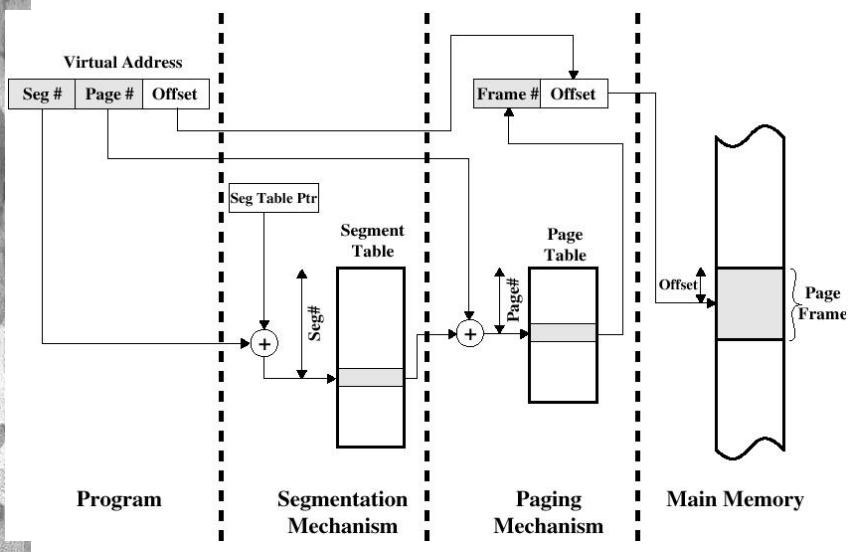
47



Kuva 8.8

Combined segmentation and paging

Kuva 8.13



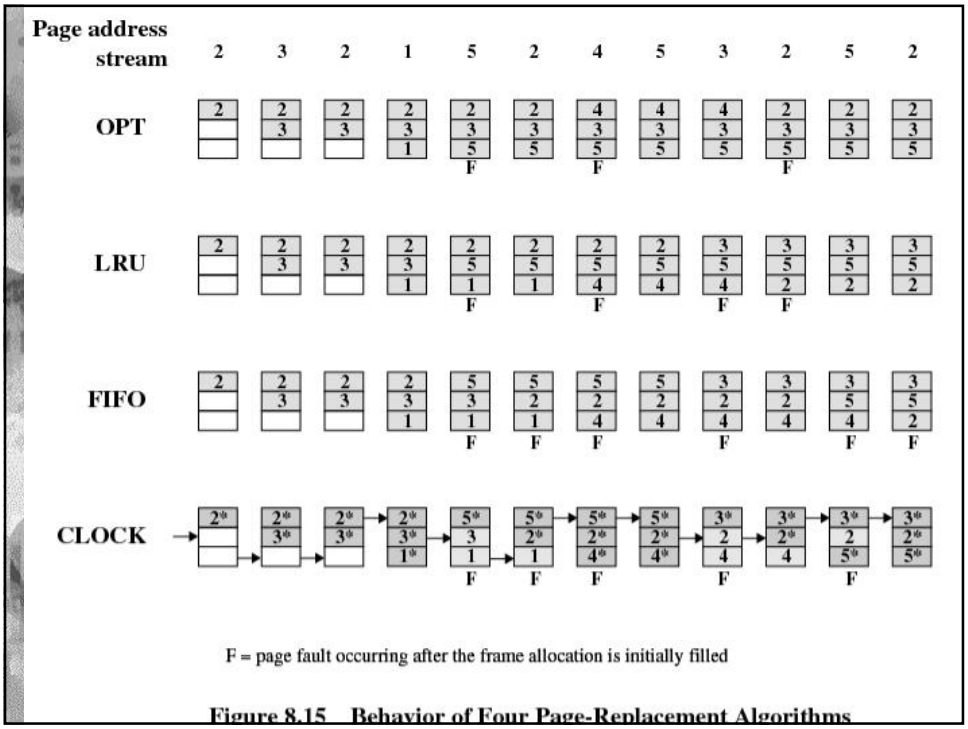


Figure 8.15 Behavior of Four Page-Replacement Algorithms

Clock page replacement

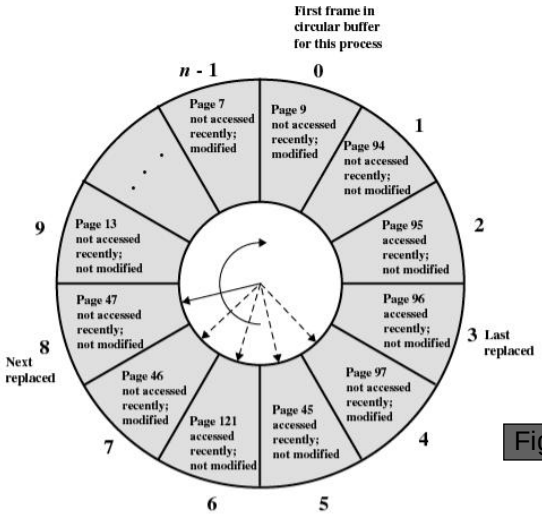


Fig 8.16 [Stal05]

51

Working set

Fig 8.19 [Stal05]

Sequence of Page References	Window Size, Δ			
	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

2. period

- Starts on Monday 29.10
- Our sessions continue on Thursday 1.11.