

LUENTO 23

Kertaus

- Koealue: (Koe to 14.12 klo 9.00 A111)
 -Kirjan luvut 9-16 (ei 13) + Appendix B.4
 -Luennot 11-23, harjoitukset 7-12
 -Opintopiiritehtävä 3

Sisältöä

- Vuorotus
- I/O
- Tiedostojärjestelmä
- Hajautettu prosessi
- Tietoturva

Vuorotus

Milloin?

- Long-term
 - oletaanko uusi prosessi suoritettavaksi?
 - mahtuuko muistiin? riittääkö swap-tila?
- Medium-term
 - milloin (heittovaihdettu) prosessi muistiin?
 - vapaata muistia?
 - moniajoaste?
- Short-term
 - mille prosessille annetaan CPU?
- I/O
 - minkä prosessin I/O pyyntö palvelee ensin?

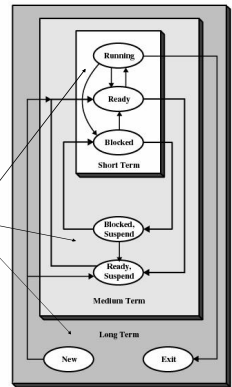


Figure 9.2 Levels of Scheduling

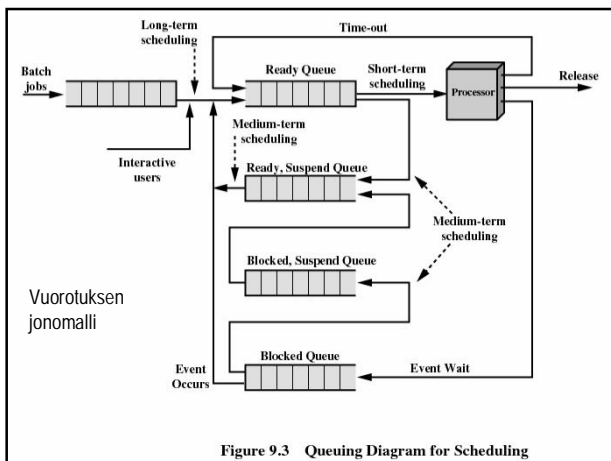
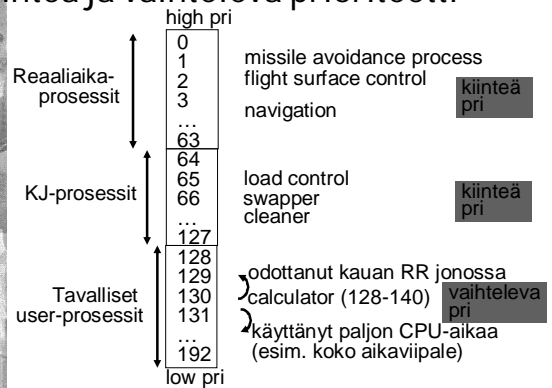


Figure 9.3 Queuing Diagram for Scheduling

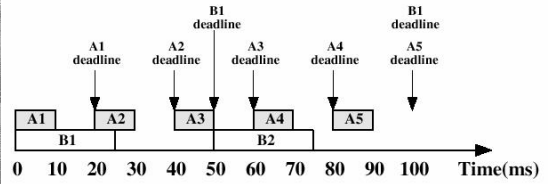
Kiinteä ja vaihteleva prioriteetti



Algoritmit

- 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

EDF – Earliest Deadline First

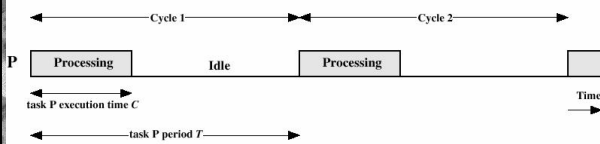


Kaksi jaksollista (periodic) työtä:

Fig 10.5 [Stal05]

- Saapumisajat A 20 ms, B 50 ms välein
- Suoritusajat A 10 ms, B 25 ms
- Vuorottaminen 10 ms:n välein
- Valm. takaraja A 20 ms, B 50 ms saapumisesta

Rate Monotonic Scheduling



- Vain jaksollisille (periodisille) töille (Fig 10.7 [Stal05])
 - Sama työ tasaisella tahdilla
 - käyttöaste yhden työn osalta on $U = C/T$
 - Jakson loppu = Hard deadline
 - Pienin jakso (T) = suurin prioriteetti
- Rate: määrä yksikköä kohden, taajuus

Ajoitettavuuden arviointi käyttöasteen avulla

Tbl 10.4 [Stal05]

- RMS: Selvästi

$$U_i = C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq 1$$

- Riittävä ehto sille, että työt voidaan ajoittaa RMS-algoritilla on

$$C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq n(2^{1/n} - 1)$$

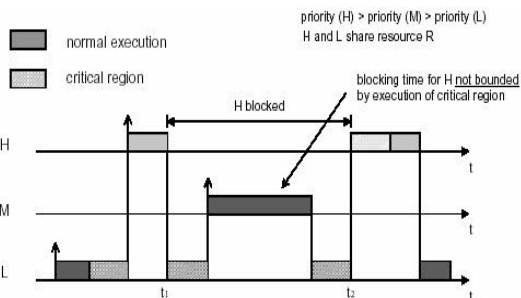
- Arvon n kasvaessa, RMS:lle yläraja lähenee

$$\ln 2 \sim 0.693 \text{ eli rajatapaus } U_i < 0.693$$

- EDF: tehokkaampi, sille riittää (riittävä ja välttävä ehto)

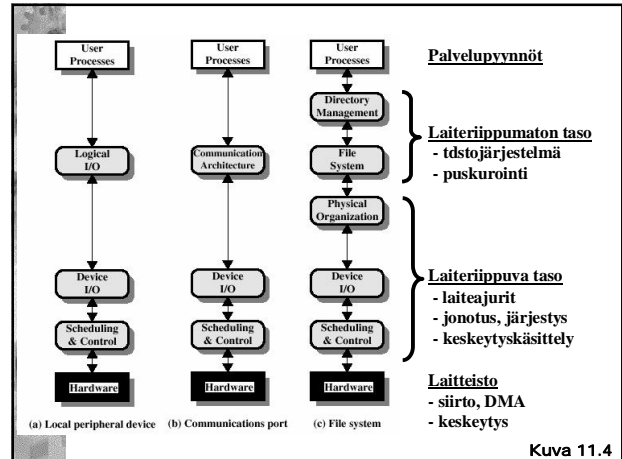
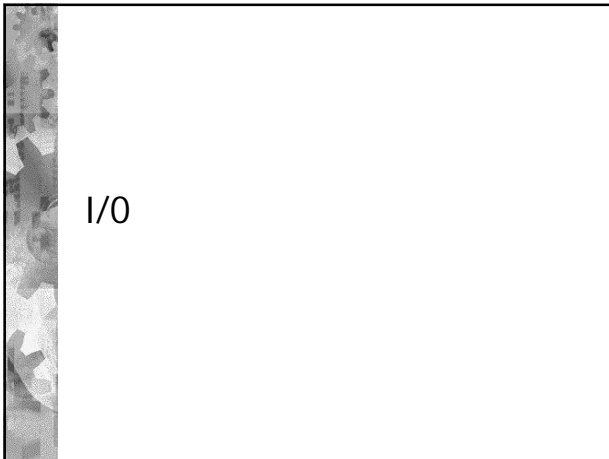
$$U_i = C_1/T_1 + C_2/T_2 + \dots + C_n/T_n \leq 1$$

Prioriteetin kääntymisen (Priority inversion)



Prioriteetin kääntymisen välttäminen

- Keskeyttämättömät kriittiset alueet
 - Luovat tarpeetonta odotusta.
 - Käyttökelpoisia vain lyhyille kriittisille alueille.
- Sisääntuloprotokolla kriittiselle alueelle
 - Prioriteetin perintä (Priority Inheritance Protocol).
 - Prioriteetin kattomenetelmä (Priority Ceiling Protocol).



Laiteajurit

- Erityyppisille laitteille omat ajurinsa
- Etsi ajuri laitenumeron perusteella laitekuvaajalistasta
 - siirtoa käynnistettäessä
 - siirron päättyessä (keskeytys!)
- Laitekuvaaja
 - laitteen tunnistus, device id
 - tilatietoa, kenelle laite varattu
 - mitä ajuria käyttää
 - mitä ajurin funktiota (handler) kutsuttava missäkin tilanteessa
 - open(), read(), write(), close() ..., keskeytys
- jono pyynnöistä parametreineen
 - mm. linkki pyynnön tehneen prosessin PCB:hen

Siirrän puskurointi

- Tarve
 - Prosessi odottaa Blocked-tilassa siirron valmistumista
 - Alue, jonne siirretään oltava silti muistissa
- Lohkoperustainen
 - Levyt, nauhat
 - kirjanpito vapaasta / varatusta tilasta lohkoittain
 - siirto laitteen ja muistin välillä lohko kerrallaan
 - hajakäsittely mahdollista (nauha?)
- Tavuperustainen
 - pääteyhteys, kirjoitin, hiiri, tiedonsiirtolinjat, ...
 - tiedon käsittely tavu kerrallaan
 - vain peräkkäiskäsittely

Levyhaku

(Fig 11.6 [Stal05])

- Laitteen vapautumisen odotus
 - ohjain käsittelee yhden pyynnön kerrallaan
- Siirtokanavan odotus
 - jos useita levyjä samassa väylässä
- Hakuvarren siirto-aika (seek time)
 - hakuvarsi oikealle uralle
- Pyörähdysviive (rotational delay/latency)
 - odota, että oikea sektori pyörähtää kohdalle
- Siirtoaika (transfer time)
 - yhden lohkon kirjoittamiseen/lukemiseen kuluva aika

Näihin voi vaikuttaa

saanti-aika (access time)

Tähän ei voi vaikuttaa

Algoritmeja

- Hakuvarren siirtoaika pisin
 - kannattaa minimoida siirrot
- Random? FIFO? PRI? LIFO?
 - huonoja, eivät huomioi hakuvarren nykyistä positiota
- Ota huomioon hakuvarren sijainti
 - SSTF
 - SCAN
 - C-SCAN
 - N-step-SCAN ja FSCAN

Levyn vuorotusalgoritmeja

Table 11.2 Comparison of Disk Scheduling Algorithms

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	Average seek length	27.5	Average seek length	27.8	Average seek length	35.8

RAID - Redundant Array of Independent Disks

- RAID 0 (ei redundanssia, ei toisintoja)
- RAID 1 (mirror, kahdennettu)
- RAID 2 (Hamming)
- RAID 3 (pariteettibitti)
- RAID 4 (pariteettilohko)
- RAID 5 (hajautettu pariteettilohko)
- RAID 6 (2 haj. pariteettilohkoa)

Lohkopuskurit

Ch 11.7 [Stal 05]

Lohkopuskurit, levypuskurit

- KJ:n data-alueella oleva puskurit muistiinluettuja levylohkoja varten
 - Paikallisuusperiaate
 - Ennaltanouto / viivästetty kirjoitus
- Poistoalgoritmit
 - LRU: Least Recently Used
 - LFU: Least Frequently Used
 - Most Recently Used – MRU FIFO (Frequency Based Replacement)

Most Recently Used – MRU Three Sections

- Most Recently Used – MRU Three Sections
 - parannus: jaa jono kolmeen osaan
 - poistot aina viimeisestä osasta
 - etuosasta pudonneelle jää aikaa vanheta
 - tulos: parempi algoritmi kuin LRU tai LFU



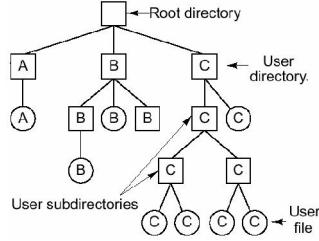
(b) Use of three sections

Fig 11.9 (b) [Stal 05]

Tiedostojärjestelmä

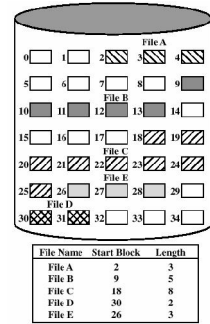
Hakemisto: Hierarkinen puurakenne

- Juurihsto, kotihakemistot, alihakemistot
- Myös prosessit voivat luoda alihakemistoja
- Juurihstolla kiinteä paikka levyllä



Koko tdsto yhdelle alueelle

- Hstoalkiossa vain ens. lohkon numero sekä tdston koko (pituus)
- Koon muuttaminen vaikeaa
 - arvioitava varausta tehtäessä
 - saatetaan joutua kopioimaan uudelle alueelle



Lohkojen ketjutus

- Varaus lohko kerrallaan vasta tarvittaessa
- Hstoalkiossa viite tdston ensimm. lohkonumeroon sekä tdston koko

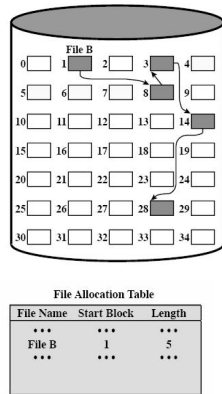
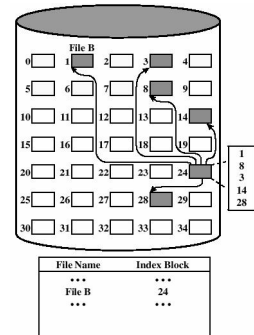


Fig 12.9 [Stal05]

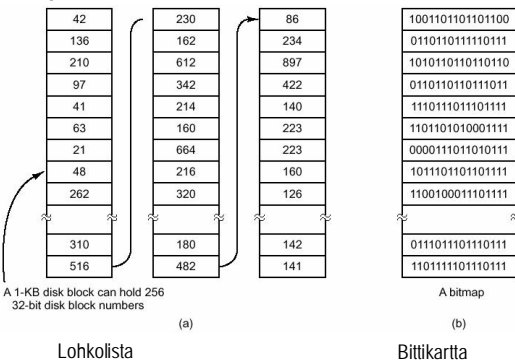
Lohkohakemisto

- Erillinen hakemisto tdstolle varatuista lohkoista
- Usein erillään omassa lohkoissa
 - hstoalkiossa vain hstolohkon numero



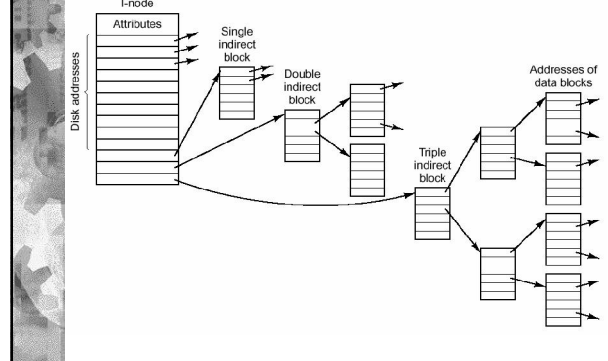
Vapaan tilan hallinta

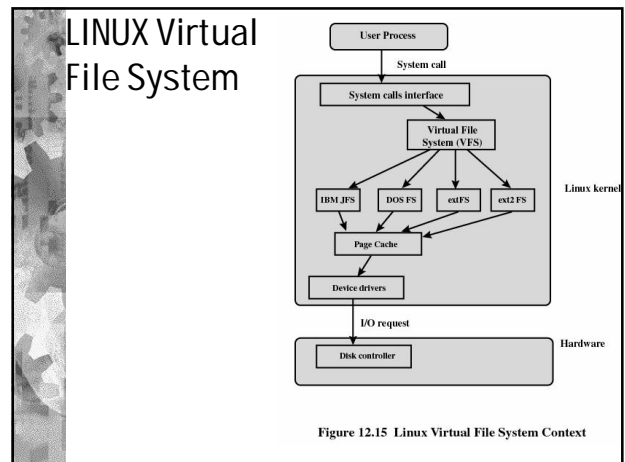
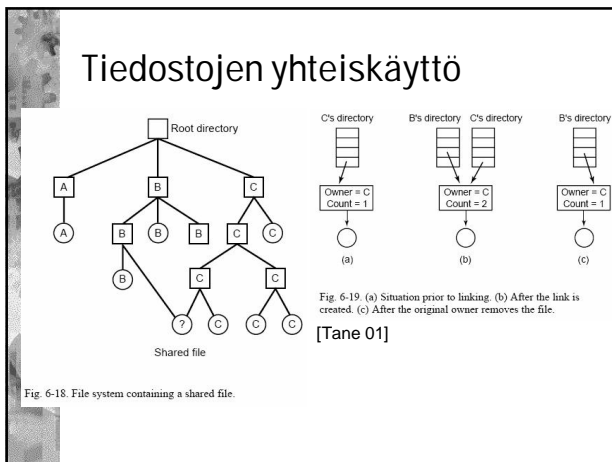
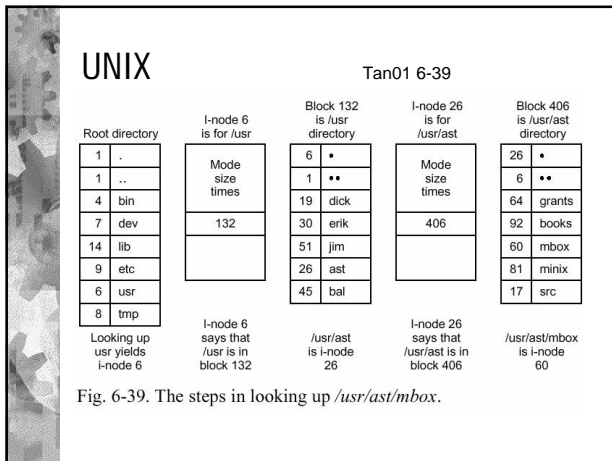
Tan01 6-21



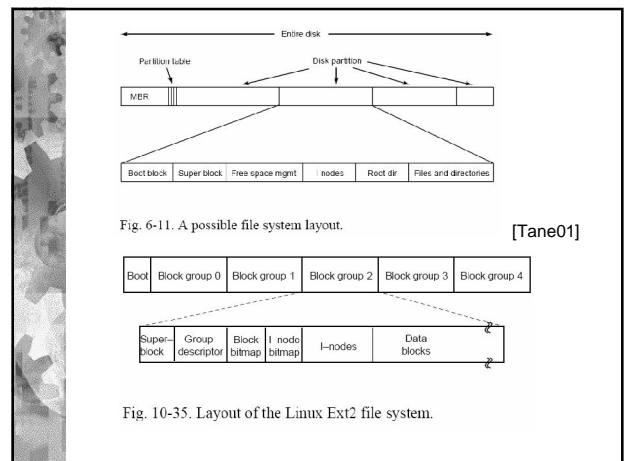
UNIX

Tan01 6-38





- ### Linux tiedostojärjestelmät
- ext2fs (second extended file system)
 - Linuxia varten kehitetty tiedostojärjestelmä
 - esikuvana BSD Fast File System (FFS)
 - lohkoryhmit
 - tehokkuus, luotettavuus
 - /proc
 - erikoistiedostot, luodaan 'lennosta'
 - esim. ytimen parametrien kysely/asettaminen
 - KJ-palvelut piilotettu tiedostojärjestelmän käyttöksi
 - käytön valvonta tiedostojärjestelmän suojauksen avulla
 - ext3fs
 - journaling file system, log-structured file system (LFS)
 - Red Hat Linux'issa



Linux ext2fs levy

- Lohkoryhmät (block groups)
 - yhtenäisesti levyllä allokoitu alue
 - datalohkot ja i-nodet fyysisesti lähellä toisiaan
 - sääsiä hakuvarren siirroissa
- Kaikki lohkot samankokoisia (1 KB)
- Kaikki i-nodet 128B (tavallinen UNIX 64B)

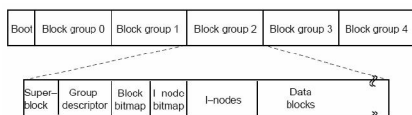


Fig. 10-35. Layout of the Linux Ext2 file system.

ext2fs superlohko (superblock)

- 1 lohko
- Kuuaa koko ext2fs-partition rakenteen
- Kopio jokaisen lohkoryhmän alussa
 - luotettavuus, virheestä toipuminen
- Ydin operoi vain lohkoryhmän 0 superblokillä ja ryhmäkuvaajilla
 - muille käyttöä, jos superblock 0 'rikki'
 - /sbin/e2fsck kopioi aika-ajoin muualle



ext2fs superlohko (superblock)

- 1 lohko
- Kuuaa koko ext2fs-partition rakenteen
- Kopio jokaisen lohkoryhmän alussa
 - luotettavuus, virheestä toipuminen

blocksize

	0	1	2	3	4	5	6	7
0	Number of i-nodes			Number of blocks				
8	Number of reserved blocks			Number of free blocks				
16	Number of free i-nodes			First data block				
24	Block size			Fragment size				
32	Blocks per group			Fragments per group				
40	i-nodes per group			Time of mounting				
48	Time of last write		Status	Max. mnt cnt				
56	Ext2signat	Status	Error behav.		Pad word			
64	Time of last test		Max test interval					
72	Operating system			File system revision				
80	RESUID	REGID	Pad word					
Pad words								

ext2fs i-node

Access Control List



	0	1	2	3	4	5	6	7
0	Mode	Uid		File size				
8	Access time			Time of creation				
16	Time of modification			Time of deletion				
24	Gid	Link counter		No. of blocks				
32	File attributes			Reserved (OS-dependent)				
40	12 direct blocks							
88	One-stage indirect block			Two-stage indirect block				
96	Three-stage indirect block			File version				
104	File ACL			Directory ACL				
112	Fragment address			Reserved (OS-dependent)				
120	Reserved (OS-dependent)							

Kirjaava tiedostojärjestelmä

- Kirjataan kaikki muutokset (journalointi?)
- Pitää tiedostojärjestelmän eheänä
- Kirjataan
 - Vain metatieto muutoksista – journal
 - Sekä metatieto että itse data – loki
- Tarve:
 - Tiedostojärjestelmän tarkistus (check) kestää liian kauan, jos epänormaali 'kaatuminen'
 - Valtaosa levyoperaatioista on kirjoituksia, lukuoperaatiot tehdään puskuireista
 - Useimmat kirjoitukset pieniä päivityksiä
 - levyn hakuvarsi liikkuu paljon, vähän dataa siirtyä

Perusidea

- Ongelma tavallisen tiedostojärjestelmän uuden tiedoston X luomisessa:
 - kirjoita hakemiston i-node, hakemisto, tiedoston i-node ja lopulta tiedosto
 - virta poikki (tms vika) kesken kaiken? Ooops.
- Ratkaisu: tapahtumaloki, joka takaa tiedostojärjestelmän eheyden - vrt tietokantojen loki
- Esim: Microsoft NTFS, Red Hat Linux ext3fs

NTFS: Piirteitä



- Kaatumisista ja levyvirheistä toipuminen
 - LFS lokitiedoston avulla
- Käyttöoikeudet
 - pääsystilat (security descriptor)
- Sallii suuret levyt ja tiedostot
 - FAT32:ssa vain 2^{32} lohkoa, suuri allokointitaulu
- Tiedosto-oliot ovat (*arvo*, *attribuutti*) -pareja
- Mahdollisuus indeksointiin tiedoston käsittelyn nopeuttamiseksi
- Lohko, cluster
 - yksi tai useampi perakkainen sektori (esim. 512 B - 4 KB)
 - 32 GB levyllä 128 sektoria/lohko (→ lohko 64-512 KB)
 - varauksen ja kirjanpidon perusyksikkö
- Partitio, volume
 - fyysinen levyn looginen osa, jolla oma tiedostojärjestelmä

NTFS-partitio

(Fig. 12.17 [Stal 05])

partition boot sector	Master File Table	System Files	File Area
-----------------------	-------------------	--------------	-----------

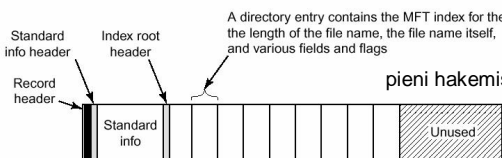
- Boottilohko
 - partitio ja tiedostojärj. rakenne, bootitietue ja -koodi
 - MFT:n sijainti
- MFT
 - tietoa tiedostoista, hakemistoista (folders) ja vapaasta tilasta
- System Files (~ 1MB)
 - kopio MFT:n alkuosasta
 - virheistätoipumisloki, bittikartta vapaat/varatut lohkot, attribuuttien kuvaustaulu
- File Area - tiedostojen lohkoille

NTFS – MFT (Master File Table)

- 1 KB:n kokoisia MFT-tietueita
 - jokainen kuvaa yhden taltiolla olevan tiedoston
 - myös hakemisto on tiedosto
 - vaihtelevanmittainen osa käytössä
 - (attribuutti, arvo) pareja (ei paikkasidonnainen!)
 - data attribuutti, 'arvo' = lohkojen sijainti
- 16 ensimmäistä tietuetta varattu ns. metadatalle
 - 16 \$-alkuisia tiedostoja
- Jos pieni tiedosto, tietue sisältää myös datan
- Jos iso tiedosto, data erillisellä tallealueella
 - MFT-tietuessa lohkonumeroita
 - kuvaus voi jatkua useampaan MFT-tietueeseen

Hakemiston MFT-tietue

(Fig. 11-38 [Tane01])



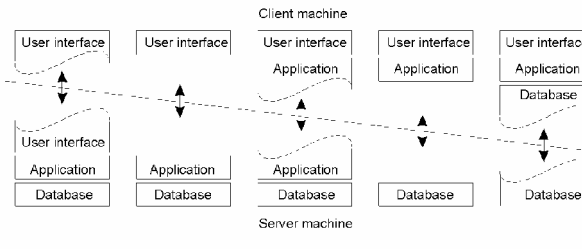
A directory entry contains the MFT index for the file, the length of the file name, the file name itself, and various fields and flags

Standard info header Index root header Directory entry Unused

- Pienissä hakemistoissa MFT-tietueet peräkkäisjärjestyksessä
- Isoissa hakemistoissa MFT-tietueessa B-puun (B-tree) indeksirakenne
 - nimen etsintä ei ole peräkkäishakua

Hajautettu prosessi

Asiakas-palvelija luokitteluja

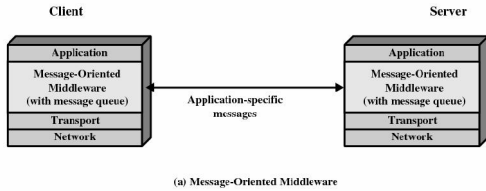


(a) (b) (c) (d) (e)

1-29 Alternative client-server organizations.

Prosessien välinen viestintä: sanomat

- Sanomien välitys on prosessien tapa kommunikoida toisten prosessien kanssa
 - Sekä samassa että eri koneissa olevien prosessien välillä

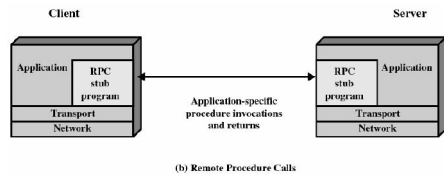


Sanomanvälityksen ominaisuuksia

- Luotettavuus
 - Luotettava takaa viestien kulun
 - Epaluotettava ei takaa, mutta yksikertainen toteuttaa
- Synkronointi
 - Asynkroninen – lähettäjä voi jatkaa heti
 - Synkroninen – lähettäjä jää odottamaan
- Sanomien pysyvyys
 - Pysyvä viestintä (Persistent communication) – viesti odottaa järjestelmässä vastaanottajaa
 - Valitton viestintä (Transient communication) – lähettäjä ja vastaanottajan olta samanaikaan paikalla
- Sidonta
 - Miten?
 - Koska?

Prosessien välinen viestintä: aliohjelmakutsut

- Aliohjelmakutsuilla (API) prosessit tyypillisesti haluavat jotain palvelua tehtäväksi
 - Samassa koneessa: funktiokutsuna
 - Eri koneissa: Remote procedure calls



RPC

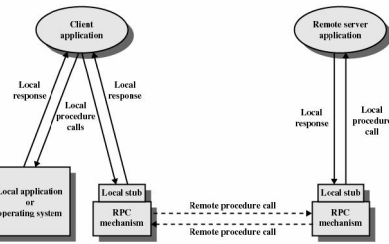
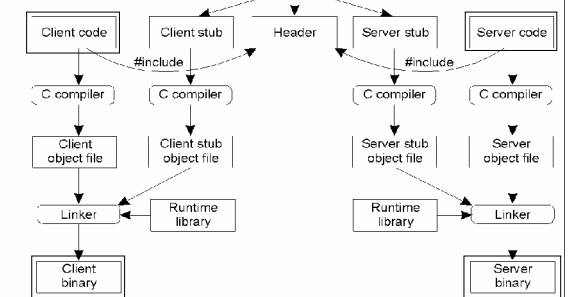


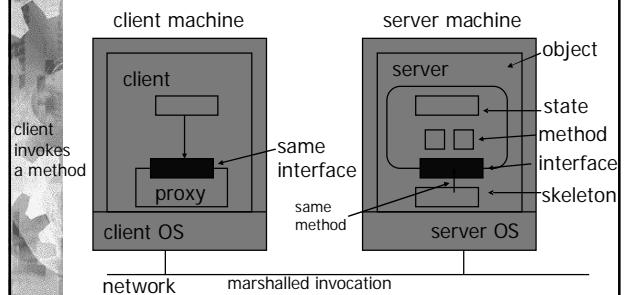
Figure 14.12 Remote Procedure Call Mechanism

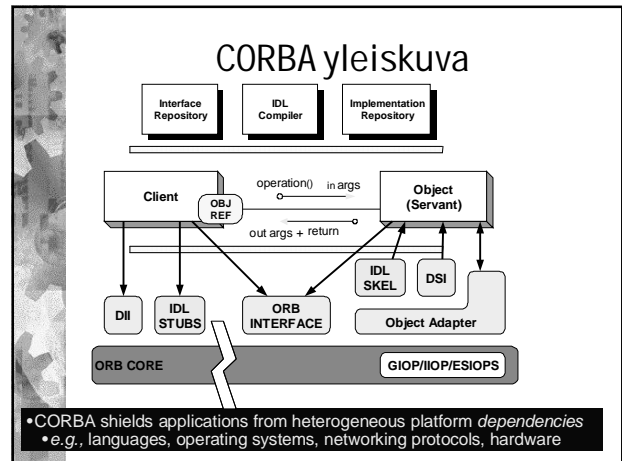
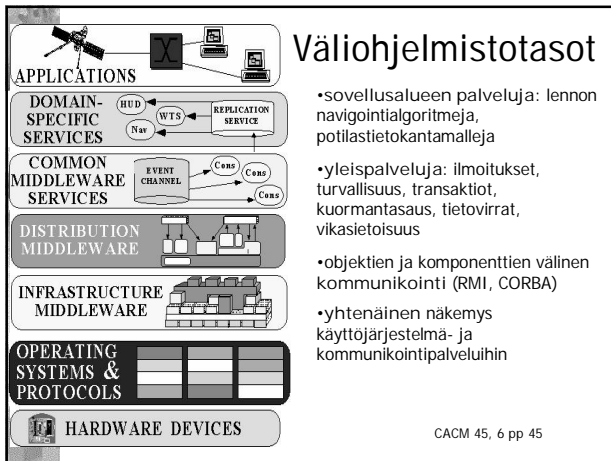
RPC: Asiakkaan ja palvelijan toteutuksen vaiheet

(Uudigen) DCE:n malli



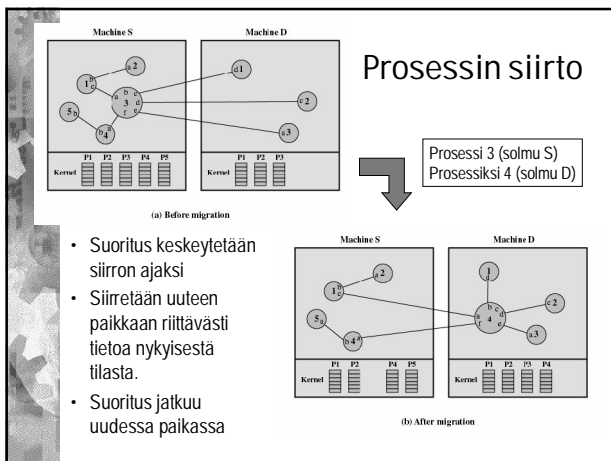
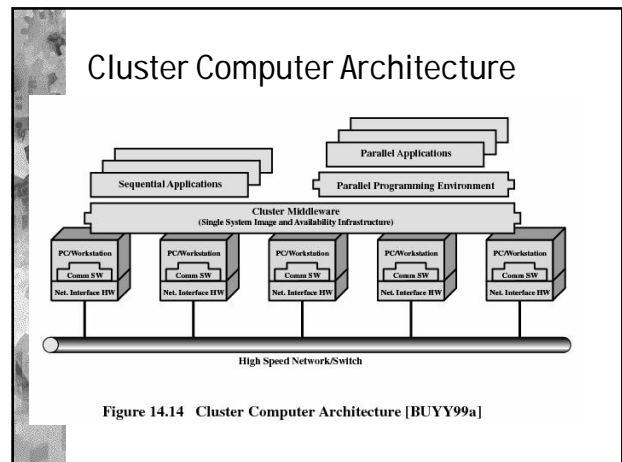
Hajautettuja olioita (Distributed objects)



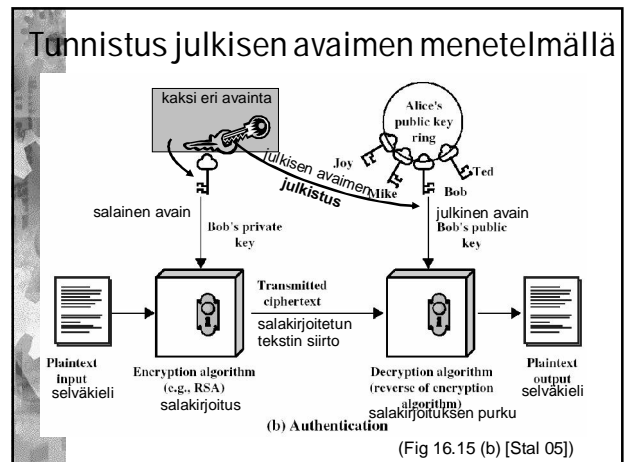
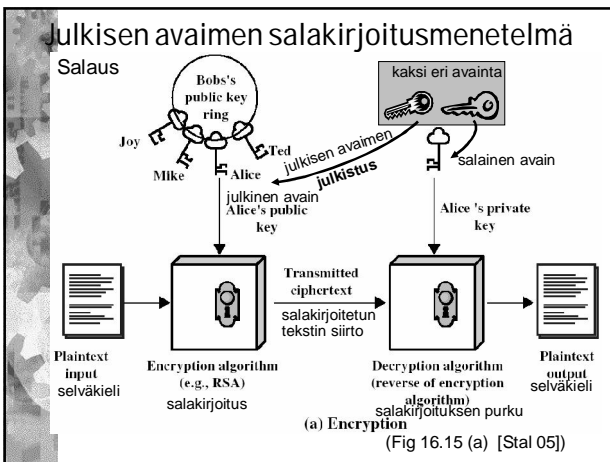
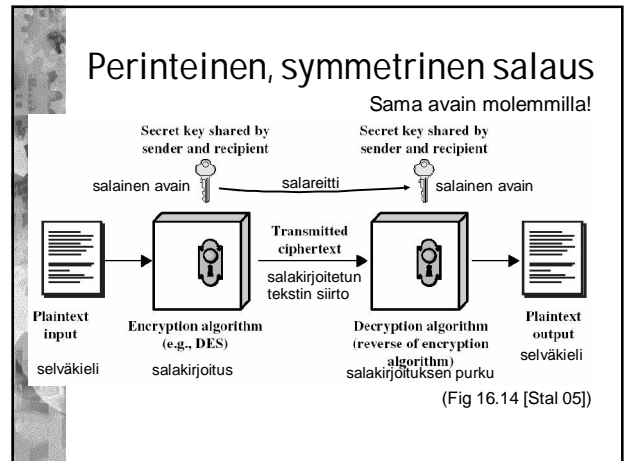
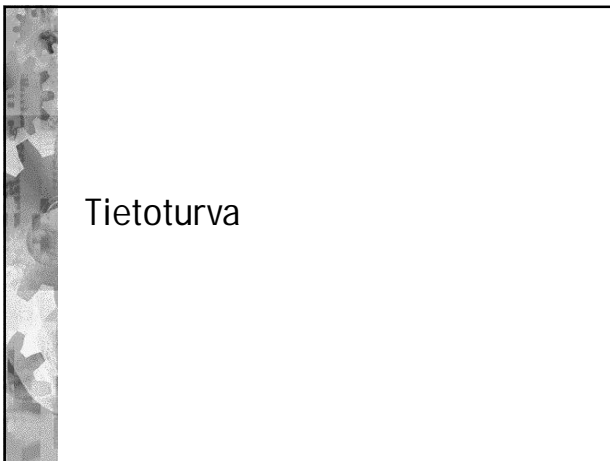


Klusterit: luokittelua

Clustering Method	Description	Benefits	Limitations
Passive Standby	A secondary server takes over in case of primary server failure.	Easy to implement.	High cost because the secondary server is unavailable for other processing tasks.
Active Secondary	The secondary server is also used for processing tasks.	Reduced cost because secondary servers can be used for processing.	Increased complexity.
Separate Servers	Separate servers have their own disks. Data is continuously copied from primary to secondary server.	High availability.	High network and server overhead due to copying operations.
Servers Connected to Disks	Servers are cabled to the same disks, but each server owns its disks. If one server fails, its disks are taken over by the other server.	Reduced network and server overhead due to elimination of copying operations.	Usually requires disk mirroring or RAID technology to compensate for risk of disk failure.
Servers Share Disks	Multiple servers simultaneously share access to disks.	Low network and server overhead. Reduced risk of downtime caused by disk failure.	Requires lock manager software. Usually used with disk mirroring or RAID technology.



- ### Siirtopoliitikoja
- Eager (all):
 - Siirrä kaikki (muistialueet+muut)
 - Precopy:
 - Prosessia vielä suoritetaan, kun muistialueita jo kopioidaan
 - Eager (dirty):
 - Siirrä vain se osa muistivaruutta, joka on keskusmuistissa ja jota on muutettu
 - Copy-on-reference:
 - Siirrä sivua vain viitattaessa
 - Flushing:
 - Kopioi prosessin muuttuneet sivut levyille



Passiiviset hyökkäykset (kuuntelu, nuuskinta)

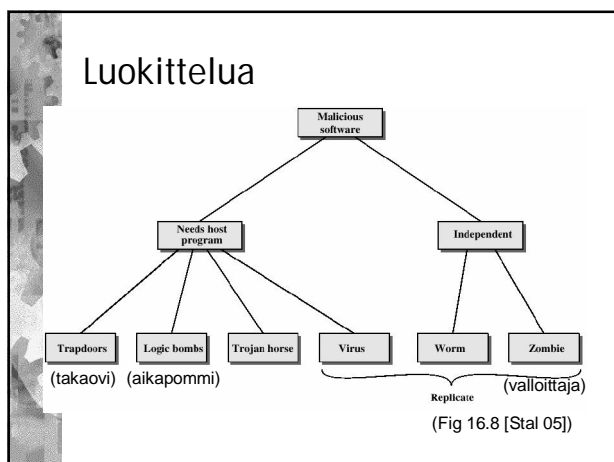
Fig 16.3 [Stal 05]

- Luottamuksellisuus rikkoontuu, eheys ei rikkoudu
- Tietoliikenneyhteydet, -verkko
 - salakuuntelu, tarkkailu, vuotaminen julkisuuteen (release of contents)
 - puhelut, sähköposti, tiedostojensiirto
 - salaus, salakirjoitus
 - silti analysoitavissa (traffic analysis)

Aktiiviset hyökkäykset

- Tiedon eheys rikkoontuu
- Tietoliikenneyhteydet, -verkko
 - lähettäjä teeskentelee olevansa joku muu (masquerade)
 - virheellinen toisto (replay)
 - viivyttäminen, muuttaminen, uudelleenjärjestely (modification of msg contents)
 - käytön esto (DoS = denial of service)
 - ylikuormitus, yhteyksien sabotointi
 - yritetään havaita ja toipua nopeasti

Luokittelua



Miten sisään yritetään?

- Arvaa / kokeile salasanoja
 - standarditunnuksia + oletussalasana / ei salasanaa
 - järjestelmällisesti lyhyitä salasanoja
 - käytä apuna järjestelmän sanastoa tai jotain muuta valmista "top100"-listaa
 - käytä käyttäjään liittyviä tietoja
 - puh., nimet, seinällä olevat sanat, ...
- Käytä Troijan hevosta
 - hyötyohjelma, joka myös kokoaa käyttäjätietoa
- Salakuuntele verkkoa
 - tunnus/salasana voi olla selväkielisenä

Virustorjunta

Aiemmin kokeessa kysyttyä

- Vuorotus
 - Yhden prosessorin: Multilevel feedback, Fair share, ...
 - Moniprosessori: kimppavuorotus
 - Reaaliaikajärj.: Rate monotonic
- I/O ja tiedostonhallinta
 - Puskurointi, lohkopuskurien allokointi
 - Levyhakuja: SCAN, FIFO, SSTF
 - Ext2fs, NTFS, tiedostojen suojaus
- Tietoturva
- Hajautuksesta:
 - Klusterit
 - *RPC, CORBA*

-- END --



Käyttöjärjestelmät