

## Käyttöjärjestelmä esimerkit: WINDOWS, UNIX ja LINUX

Mikroytimet (luku 4.3)

Käyttöjärjestelmä esimerkit (2.5 – 2.8)

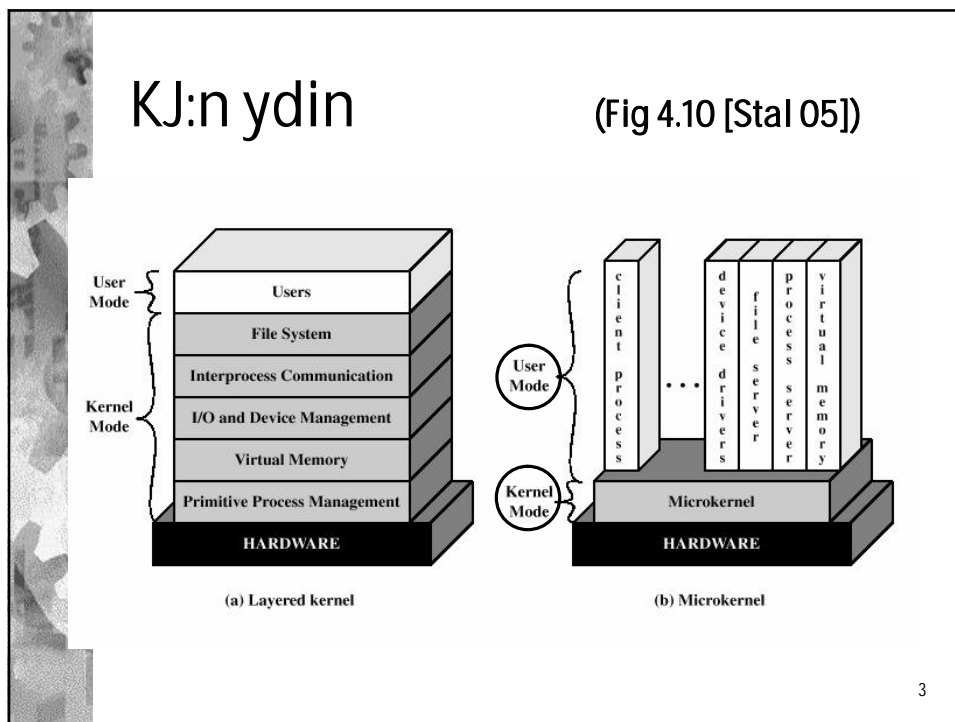
SMP (4.2)

1

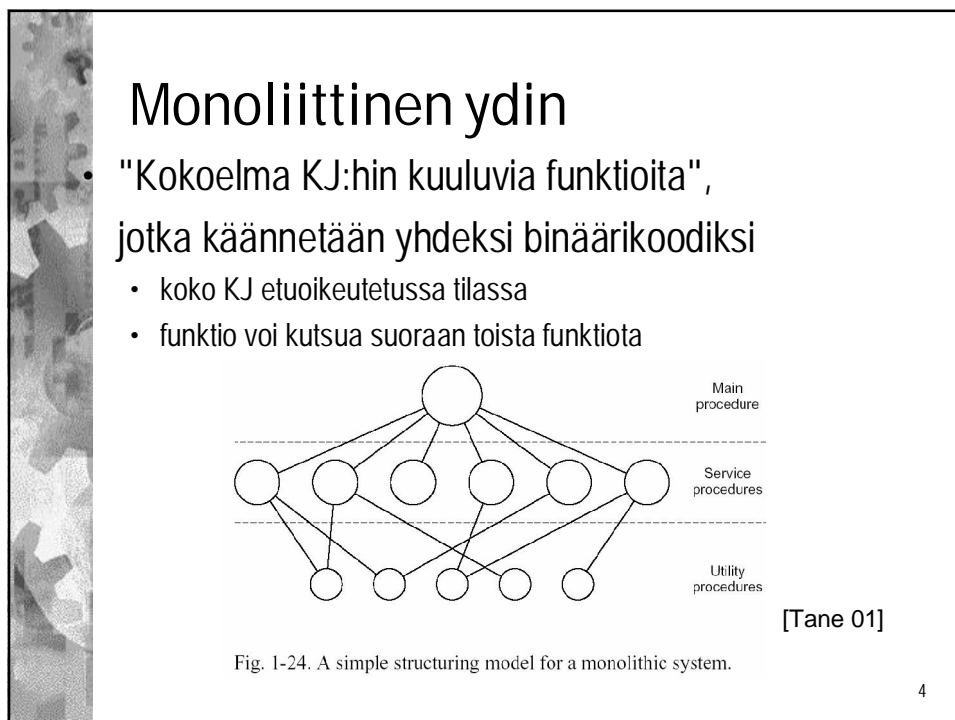
## Mikroydin (vs monoliittinen ydin)

Ch 4.3 [Stal05]

2



3



4

## Monoliittinen ydin Ongelmat?

- Periaatteessa kaikki KJ:n osat pääsevät käsiksi kaikkiin tietorakenteisiin
  - globaalien muuttujien käyttö
    - ei edes parametrien kopiointi hidastamassa
  - kopioi itse tieto suoraan paikalleen
    - ei sanomanvälitystä
    - ei *etuoikeutettu tila* ↔ *käyttäjätila* -vaihtoja KJ:n sisällä
- Sovellus käyttää palvelua palvelupyynnöllä
  - parametrit esim. pinoon, keskeytys
  - 'vastaus' suoraan annettuihin muuttujiin tai pinossa
- Nopeus, tehokkuus, yksinkertaisuus

5

## Mikroydin (mikrokernel)

- Etuoikeutetussa tilassa toimiva pieni KJ:n ydin (core functions)
  - tärkeät laitetason liittymät
  - minimijoukko muita perustoimintoja
- Muut alijärjestelmät käyttäjätilassa
  - laiteajurit
  - tiedostojärjestelmät
  - virtuaalimuistin hallinta
  - ikkunoitu käyttöliittymä
  - turvallisuusosia, mm. käyttäjien tunnistus
- Alijärjestelmät eivät pääse suoraan käsiksi laitteistoon tai toisiinsa - käyttö sanomanvälityksen kautta

6

## Mikroytimen peruspalvelut

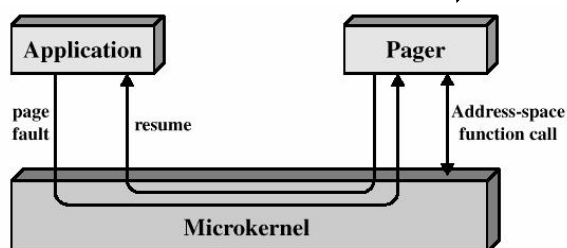
- Mitkä?
  - ei täysin yleispäteviä sääntöjä
  - ainakin laitteistoa suoraan käyttävät osat
  - tuki käyttäjätilan palveluille ja sovelluksille
    - kommunikointi
- Minimi:
  - prosessi ja prosessin vaihto (basic scheduling)
  - muistinhallinnan alimmat tasot (low-level MM)
  - prosessien välinen kommunikointi (IPC)
  - siirränän laitesidonnaiset osat
  - keskeytysten käsittely

7

## Mikroytimen peruspalvelut

(Fig. 4.11 [Stal 05])

- Muistinhallinnan alimmat tasot
  - virtuaalisivujen kuvaus fyysisiksi sivuiksi ytimessä
    - sivutaulu ja MMU:n asetukset
  - sivutus ja virtuaalimuistin hallinta mikroytimen ulkopuolella
    - sivupuutoksen käsittely
    - sivujen poisto- ja sijoitusalgoritmit



8

## Mikroytimen peruspalvelut (jatkuu)

- Prosessien välinen kommunikointi (IPC)
  - Perusmekanismina sanomanvälitys
    - pyyntö / vastaus
    - jos prosesseilla ei yhteiskäyttöistä muistia, sanomanvälitys muistista muistiin kopiointia (etuoik. tilassa)
  - Sanoma: (keneltä, kenelle, data)
    - data = esim. palvelun nimi ('portti') ja sen parametrit
    - datan paikalla voi olla osoite, josta data löytyy
  - Palveluun voi liittyä oikeuksien tarkistus KJ:ssä
    - kirjanpito 'porteista' ja pääsyoikeuksista
    - prosessi voi muuttaa pääsyoikeuksia
  - Poissulkemisen ja synkronoinnin primitiivit

9

## Mikroytimen peruspalvelut (jatkuu)

- Siirräntä
  - laiteohjaimen rekistereiden käyttäminen, DMA
  - I/O-porttien sijoittaminen prosessin osoiteavaruuteen
- Keskeytysten käsittely
  - mikroydin tunnistaa keskeytykset, mutta erillinen käyttäjätason palvelu käsittelee ne
    - keskeytysvektori: mikroydin pitää kirjata keskeytysnumeroista ja niihin liittyvistä palvelijoista
    - generoi ja lähetä sanoma käsittelijälle (esim. ajurille)
      - välitä tarvittaessa ohjaimen rekistereiden arvot
    - kuittaa keskeytys huomatuksi
    - lyhyt ja nopea?

10

## Laiteajuri

- Käyttäjätilassa toimiva prosessi / säie
  - saa pyyntöjä sovelluksilta ja laitteistolta

```
while (true) do {
    waitFor (msg, sender);
    if (sender == my_sw_client) {
        read/write I/O-ports
        ...
    }
    if (sender == my_hw_interrupt) {
        read/write I/O-ports
        reset hardware interrupt
    }
}
```

- I/O-portit ajurin osoiteavaruudessa (muuten tiedot sanomina)

11

## Mikroytimen kehuja

- Yhtenäinen palvelujen käyttötapa
  - sama mekanismi etuoikeutetussa ja käyttäjätilassa
  - sanomanvälitys: send ja receive
- Laajennettavuus
  - helppo lisätä uusia palveluja (KJ:kin kehitty...)
  - helppo lisätä uusia liittymiä vanhoihin moduuleihin
- Joustavuus
  - piirteitä poistamalla saadaan räätälöity versio
  - paikkopaketit voidaan ottaa mukaan käynnistämättä konetta uudelleen
  - esim: sulautettu järjestelmä - minimijoukko toiminnallisuutta
    - kännykkä, kämmenmikro, videonauhuri, tv, ...
- Siirrettävyys
  - laitesidonnaiset osat rajattu mikroytimeen
  - uusi laitteisto vaatii vain mikroytimen muokkaamista

12

## Mikroytimen kehuja (jatkuu)

- Luotettavuus
  - modulaarinen rakenne helpompi toteuttaa ja testata
  - jos joku palveluosa kaatuu, muut voivat silti toimia
- Sopii hajautettuun järjestelmään
  - paikallisen ja etäpalvelun käyttö samanlaista
  - 'single microkernel image': jos palvelut ja prosessit nimetty kaikissa koneissa yksikäsitteisesti
    - eri koneissa sama palveluiden nimentä ja osoitteet
- Sopii oliopohjaisen KJ:n toteutukseen
  - mikroytimen yksi olio muiden (palvelija)olioiden joukossa
  - kommunikointi vain tunnettujen rajapintojen kautta
    - private data, public interfaces

13

## Mikroytimen heikkouksia

- Tehokkuus
  - yhteiskäytössä oleva vs. paikallinen muisti
  - sanomavälitys vs. suora funktiokutsu
  - paljon *user mode* ↔ *kernel mode* -vaihtoja
    - viestien käyttö vs. suorat funktiokutsut
- Apu: enemmän toimintoja takaisin suoritettavaksi etuoikeutettuun tilaan
  - esim. Mach, Chorus, W2K
  - laiteajureita takaisin etuoikeutettuun tilaan
  - muita paljon käytettyjä palveluja
  - kompromissi - mitä menetetään?

14

## WINDOWS

Ch 2.5 [Stal05]  
(ks. myös Tan01, Ch 11)

15

## Windows 2000

- Historia: ks. kirja
- Moniajaja yhden käyttäjän ympäristössä
  - useita ohjelmia avoinna samanaikaisesti (multitasking)
  - joustava datan siirto ohjelmien välillä (leikepöytä)
  - GUI: graafinen käyttöliittymä
- Myös palvelinkäyttöön (server)
  - moniajaja monen käyttäjän ympäristössä, oma KJ versio
- Verkon kautta yhteys muihin koneisiin
  - sisältää keskeiset tietoliikenneprotokollat
- Parempi tuki hajautetuille sovelluksille
  - asiakas-palvelija suhde, voivat olla eri koneilla
  - hajautettu hakemistopalvelu: Active Directory

16

## W95/98 vs. W2000

Fig 11-1

[Tan01]

Item	Windows 95/98	Windows 2000
Full 32-bit system?	No	Yes
Security?	No	Yes
Protected file mappings?	No	Yes
Private addr space for each MS-DOS prog?	No	Yes
Unicode?	No	Yes
Runs on	Intel 80x86	80x86, Alpha, MIPS, ...
Multiprocessor support?	No	Yes
Re-entrant code inside OS?	No	Yes
Plug and play?	Yes	Yes
Power management?	Yes	Yes
FAT-32 file system?	Yes	Optional
NTFS file system	No	Yes
Win32 API?	Yes	Yes
Run <u>all</u> old MS-DOS programs?	Yes	No
Some critical OS data writable by user?	Yes	No

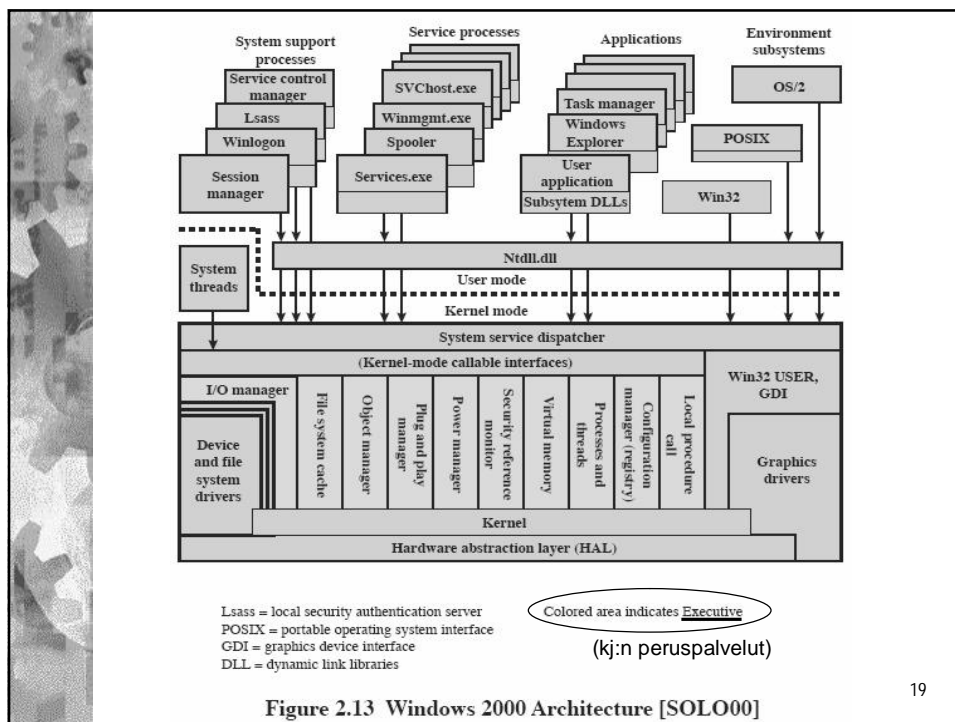
XP?

17

## W2K Arkkitehtuuri

- Muutettu (modified) mikrokernel-arkkitehtuuri
  - useat mikrokernelin ulkopuolisetkin K-palvelut suoritetaan etuoikeutetussa tilassa Miksi?
- Modulaarinen rakenne
  - yksi KJ-palvelu per moduuli
  - käyttö vain määritellyn rajapinnan kautta
  - moduuleja voi muuttaa / poistaa koskematta muihin moduuleihin
    - mm. laiteajurit
- KJ ja sovellukset eristetty toisistaan
  - käyttäjätila / etuoikeutettu tila
  - KJ:n tietorakenteita ja laitteistoa voi ronkia vain etuoikeutetussa tilassa

18



19

## W2K Etuoikeutettu tila

(ks. Fig. 2-13)

### HAL, Hardware Abstraction Layer

- piilottaa laitteiston yksityiskohdat, abstrahointi
  - väylät? keskeytykset? DMA? ajastimet? muisti? SMP?
  - laitteella muistiinkuvattu I/O vai ei? spin locks?
- ylempi taso käyttää laitteistoa yleisten toimintojen tasolla
- erilainen toteutus eri alustoille → siirrettävyys
  - Intel (PowerPC, Alpha)
- ei riittävän tehokas multimediasovelluksiin
  - DirectX rajapinta multimediasovelluksille
    - HAL – Hardware Abstraction Layer
    - HEL – Hardware Emulation Layer

20

## W2K Etuoikeutettu tila

ks. Fig. 2-13

### Ydin ("mikrokernel", kernel)

- ei "aito": myös manager-palvelut etuoikeutetussa tilassa
  - "aidossa" ytimessä vain ydin etuoikeutetussa tilassa
- poikkeusten ja keskeytysten käsittely (alkutoimet)
- säikeiden vuorottaminen (W2K kernel ei käytä itse säikeitä!)
- prosessienvaihto (rekisterien kopiointi, MMU:n asetukset)
- SMP ja synkronointi
- virtakatkoista toipuminen
- aina muistissa, ei käytä säikeitä

21

## W2K Etuoikeutettu tila

ks. Fig. 2-13

### Laiteajurit

- muuttaa laiteriippumattomat I/O-pyyntöt laitetason operaatioiksi
- ohjaimien rekistereiden käyttäminen
- ko. laitteeseen liittyvän keskeytyksen käsittely

22

## W2K Etuoikeutettu tila

ks. Fig. 2-13

### Windows 2000 Executive & Executive API

- Object manager (Ch 2 [Stal05])
  - W2K:n olioiden luonti, nimeäminen, hallinta, poisto
  - esim. prosessi-olio, säie-olio, synkronointi-olio
  - olion luoja saa kahvan luomaansa oloon
  - oloon liittyy myös pääsyoikeuksia (access control)
- Security reference monitor (Ch 16 [Stal05])
  - pääsynvalvonta ja auditointi (US DoD "Orange Book")
  - samat mekanismit kaikille olioille
    - prosessit, tiedostot, muistiosoitteet ja I/O-laitteet
- Process/thread manager (Ch 4 [Stal05])
  - kirjanpito prosesseista ja säikeistä

23

## W2K Etuoikeutettu tila


ks. Fig. 2-13

### Windows 2000 Executive & Executive API (jatkuu)

- I/O manager (luku 11 [Stal05])
  - siirränän laiteriippumaton rajapinta sovelluksille
    - laitteiden nimeäminen, tiedostojärjestelmä
    - pyyntöjen ohjaaminen oikeille laiteajureille
- File system cache (tiedostovälimuisti)
  - tiedostojärjestelmän ja siirränän puskurimuisti
  - jokainen I/O pyyntö ei aiheuta fyysistä siirtoa

*[ Huom: useimmiten "manager" ei ole prosessi, vaan kokoelma yhteiskäyttöisiä aliohjelmia ]*

24




## W2K Etuoikeutettu tila ks. Fig. 2-13

### Windows 2000 Executive & Executive API (jatkuu)

- Local procedure call facility (LPC)
  - asiakas/palvelija mallin viestipohjainen  
"paikallisproseduurikutsu" ts. asiakas voi kutsua palvelijassa olevaa rutiinia (vrt. RPC)
- Virtual memory manager (Ch 8 [Stal05])
  - prosessin virtuaaliosoitteiden kuvaus fyysisiksi osoitteiksi (~ sivutaulujen ylläpito)
  - muistin varaus / vapautus
- Window/Graphics modules (GUI)
  - ikkunapohjainen käyttöliittymä
  - etuoikeutetussa tilassa: suora pääsy laitteistoon
  - GDI (Graphics Device Interface)


25



## W2K Käyttäjätila ks. Fig. 2-13

- Special system support processes
  - Käyttäjätilassa ajettavia KJ:n palveluja
    - tehokkuus ei keskeinen tekijä
  - mm. logon, istunnon hallinta
- Service processes
  - Muita kuin em. palveluja
  - mm. tapahtumaloki, etäproseduurin kutsu

26



## W2K Käyttäjätila


Environment subsystems ks. Fig. 2-13

- Linkki KJ-palvelujen ja sovelluksen välillä (KJ-rajapinta, API)
- DLL (Dynamic Link Library) muuttaa kirjastorutiinikutsut W2K palvelupyynnöiksi
- Tuetut alijärjestelmät: Win32, Posix, OS/2, Win 3.1, MS/DOS
- Alijärjestelmät omia suojattuja prosesseja
  - rajattu osoiteavaruus
  - vanhojen järjestelmien ohjelmia voi suorittaa kenties kääntämättä, vaikka käyttävät "vanhaa" API-rajapintaa

### User applications

- Win32, Win 3.1, MS-DOS (kaikki ohjelmat eivät ehkä toimi)
- Posix, OS/2 periaatteessa kyllä, mutta vajavainen...
  - saatavilla täydellisempänä muiden toimittamana
- Kullakin tyypillä omat DLL:nsä

27



## W2K WIN32 API

- WIN32 filosofia
  - usein 3-4 tapaa tehdä joku tietty asia
  - monimutkaisia liittymiä, jopa 10 parametria
  - osa ei selvästi kuuluisikaan API:iin
    - esim. kokonaisen tiedoston kopioiminen
- Palvelupyyntö palauttaa kahvan (handle) luomaansa olioon
  - käyttö seuraavissa kutsuissa

28

## WIN32 API palvelupyynnöt

Fig 11-31 [Tane 01]

Win32 API function	UNIX	Description
CreateFile	open	Create a file or open an existing file; return a handle
DeleteFile	unlink	Destroy an existing file
CloseHandle	close	Close a file
ReadFile	read	Read data from a file
WriteFile	write	Write data to a file
SetFilePointer	lseek	Set the file pointer to a specific place in the file
GetFileAttributes	stat	Return the file properties

API group	Description
Window management	Create, destroy, and manage windows,
Menus	Create, destroy, and append to menus and menu bars
Dialog boxes	Pop up a dialog box and collect information
Painting and drawing	Display points, lines, and geometric figures
Text	Display text in some font, size, and color
Bitmaps and icons	Placement of bitmaps and icons on the screen
Colors and palettes	Manage the set of colors available
The clipboard	Pass information from one application to another
Input	Get information from the mouse and keyboard

Fig 11-29 [Tane 01] 29

## W2K Asiakas-palvelija -malli

- Hajautetun järjestelmän perusmalli
  - perustuu sanomaväilykseen
  - pyyntö-vastaus malli (request-reply)
- Sopii myös ei-hajautettuun järjestelmään
  - käytössä kaikissa W2K moduuleissa
  - yksi tai useampia prosesseja/säikeitä per KJ-palvelu
- Yksinkertaistaa W2K Executive'n ohjelmointia
  - rajapinnat helpommin sovitettavissa
  - LPC = yhtenäinen prosessien kommunikointi
- Luotettavuus
  - kukin palvelija on oma prosessinsa, samoin asiakas
  - kullakin käytössä oma muistialue
  - vain palvelija koskee laitteistoon

Local  
Procedure  
Call

30

## W2K Säikeet ja SMP

- Mikä tahansa CPU voi suorittaa KJ:n koodia
- Prosessi voi jakaantua säikeisiin (ei W2K ydin)
- Prosessin säikeitä voi suorittaa samanaikaisesti eri prosessoreilla
- Palvelijaprosessit jakautuvat säikeiksi voidakseen palvella useita pyyntöjä yhtä aikaa
- Mekanismit muistialueiden yhteiskäyttöön sekä prosessien väliseen kommunikointiin
  - poissulkeminen ja synkronointi

31

## W2K Oliot

- W2K-toteutus pääosin oliopohjaista
- Oliota esim. tiedostot, prosessit, säikeet, semaforit, ajastin, ikkunat, ...
- Kaikki ei kuitenkaan oliopohjaista
  - oliota käytetään, jos resurssia tarvetta käyttää käyttäjätalasta tai resurssi jaetussa käytössä **Fig. 2.13**
- Object Manager
  - vastaa olioiden luonnista, niiden palvelujen käytöstä ja poistamisesta (open – use – close) **Tbl. 2.5**
- Esim. ytimen oliota
  - microkernel control objects **Tbl. 6.7**
  - dispatcher and synchronization objects

32

## W2K Executive olioita

Type	Description
Process	User process
Thread	Thread within a process
Semaphore	Counting semaphore used for interprocess synchronization
Mutex	Binary semaphore used to enter a critical region
Event	Synchronization object with persistent state (signaled/not)
Port	Mechanism for interprocess message passing
Timer	Object allowing a thread to sleep for a fixed time interval
Queue	Object used for completion notification on asynchronous I/O
Open file	Object associated with an open file
Access token	Security descriptor for some object
Profile	Data structure used for profiling CPU usage
Section	Structure used for mapping files onto virtual address space
Key	Registry key
Object directory	Directory for grouping objects within the object manager
Symbolic link	Pointer to another object by name
Device	I/O device object
Device driver	Each loaded device driver has its own object

Fig 11-10 [Tan01]

33

## W2K Oliot (ks. Appendix B [Sta105])

- Kapselointi: muuttujat ja tietorakenteet vain olion sisäisiä, käyttö julkisilla palveluilla
- Luokat ja niiden ilmentymät: luokka vain malli oliosta (rajapinnat), KJ luo olion tarvittaessa
  - esim. luokka prosessi kuvaa prosessin tietorakenteet ja käsittelyssä käytettävät palvelut. Jokaista luotua prosessia kohden oma ilmentymä (olio), jossa ko. prosessia koskevat tiedot
- Perintä: olio perii ominaisuuksia yläluokalta
  - esim. hakemisto-olion lipuke "compressed" kopioidaan hakemistoon talletetuille tiedosto-olioille
  - W2K: puutteita käsittelyssä
- Polymorfismi: saman operaation voi kohdistaa olioon riippumatta sen luokasta
  - W2K: puutteita käsittelyssä

34

## PERINTEINEN UNIX-JÄRJESTELMÄ

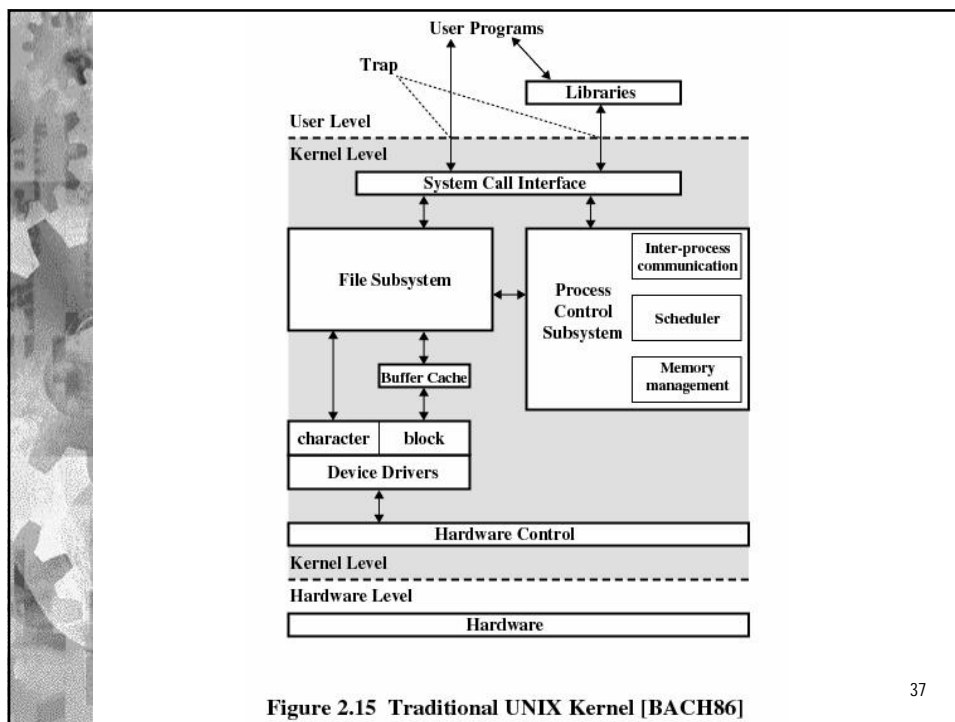
Ch 2.6 [Stal05]

35

## Perinteinen (vanha) UNIX

- Historia - ks. kirja
  - Thompson & Ritchie, 1969?
    - Bell Labs (v. 1970), AT&T, Berkeley University
    - SysV ja BSD
- Suosion perustat
  - Thompson, Ritchie & Kernighan, 1971
    - 16-bit PDP-11
  - siirrettävyys
    - Ritchie & Kernighan, 1973
      - C-kielellä, laitetoiminnot assemblerilla
  - monen käyttäjän moniajojärjestelmä
  - suunniteltu interaktiiviseen käyttöön
  - hierarkkinen tiedostojärjestelmä
  - vain yksi tiedostoformaatti: tavujono
    - Fig. 2-15
    - komentotulkki
    - putket 1972
- UNIX filosofia: *pieni on kaunista*
  - pienten ohjelmien yhdistely
    - yhden tuloste on toisen syöte
  - palvelupyynnöitä vain minimimäärä

36



37


## Perinteinen UNIX

- Monoliittinen kernel
  - kaikki KJ:n osat käännettiin yhdeksi binäärikoodiksi
  - yksi yhteinen osoiteavaruus
    - helppo pääsy KJ:n tietorakenteisiin ja funktioihin
- Yhden prosessorin järjestelmä
- Ei virtuaalimuistia, vain heittovaihto
  - ihan alussa, pian kyllä myös virtuaalimuisti
- Sovellus käyttää KJ-palveluja palvelupyynnöillä
  - keskeytys → siirtyminen KJ:n koodiin tai kirjastorutiinien välityksellä
    - hieman mukavampi liittymä palvelupyyntöihin
- *Bach: The Design of the Unix Operating System, Prentice Hall, 1986.*

Fig. 2-15

Unix 3BSD, 1979  
32-bit VAX

38




## Perinteinen UNIX

Fig. 2-15

- Process control subsystem
  - prosessien luonti, vuorottaminen
  - muistinhallinta
  - prosessien välinen vuorovaikutus
- File subsystem
  - tiedostojärjestelmä
  - siirräntäjärjestelmä
    - lohkolaitteet (levyt), merkkilaitteet
  - lohkolaitteilla puskurivälimuisti
- Device drivers
  - laitetta vastaa tiedosto hakemistossa /dev
    - major, minor device number → ajuri löytyy laitekuvaajalistasta

39



## NYKYAIKAINEN UNIX-JÄRJESTELMÄ

Ch 2.7-8 [Stal05]

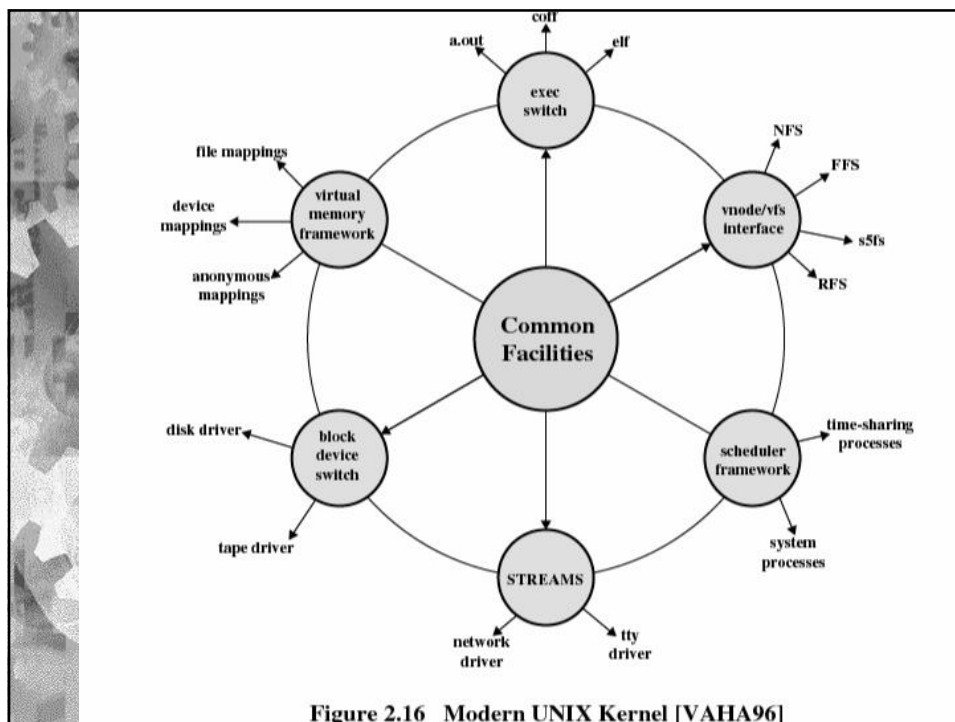
40

## Nykyaikainen UNIX

- Historian hyvät piirteet yhdessä paketissa
  - POSIX palvelupyynnörajapinta
- Virtuaalimuistin käyttö
  - perinteisessä oli vain heittovaihto (swapping)
- SMP-tuki
  - KJ suorituksessa useammalla prosessorilla
    - è poissulkeminen, synkronointi
- Tuki erilaisille tiedostojärjestelmille (virtual file syst)
  - hajautettu tiedostopalvelu NFS, ynnä muut
- Vaihtoehtoisia vuorottamisalgoritmeja
- Erilaisia binääriformaatteja (executable)
  - a.out, ELF (Executable and Linking Format),
  - COFF (Common Object File Format), Portable Executable COFF<sup>41</sup>

## Nykyaikainen UNIX

- Muutettavuus ja laajennettavuus huomioitu
- Pieni ydin, modulaarisia palveluja, joita muut KJ:n prosessit käyttävät Fig. 2-16 [Stal05]
- Daemon-prosessit
  - KJ-palveluprosessit, jotka on luotu konetta käynnistettäessä



## Nykyaikainen UNIX SVR4 (System V, Release 4)



- AT&T, Sun Microsystems
- SVR3:n, 4.3BSD, Xenix, SunOS "parhaat palat"
  - sysV: IPC (messages, shared memory, semaphores)
  - BSD: pistokkeet (sockets)
- reaaliaikaprocessit, vuorottamislakat, tietorakenteiden dynaaminen allokointi, virtuaalimuisti, virtuaalitiedostojärjestelmä, pre-emptive kernel
- toimii PC:ssä ja superkoneessakin

(pre-empt ~ "anastaa etuoikeuden nojalla", estää, etuilla)

(pre-emption – etuosto-oikeus)

44

## Nykyaikainen UNIX



### Solaris 2.x

- Sun Microsystemsin versio SVR4:stä
  - lisäksi: fully preemptable kernel, SMP-tuki, säikeet, oliopohjainen tiedostojärjestelmä
  - tärkein kaupallinen Unix-versio
- 
- *Goodheart, Cox: The Magic Garden Explained; The Internals of UNIX System V release 4. Prentice Hall, 1994.*

45

## Nykyaikainen UNIX



### 4.4 BSD

- Berkeley Software Distribution
  - BSD:llä keskeinen rooli UNIX-kehittämisessä
  - Käytetty paljon akateemisissa piireissä
    - UNIX-järjestelmien esikuva
  - Pohjana useille kaupallisille toteutuksille
    - Mac OS X
- 
- *McKusick, Bostic, Karels, Quarterman: The Design and Implementation of the 4.4 BSD Operating System. Addison Wesley. 1996.*

46

## 4.4 BSD ydin

System calls				Interrupts and traps			
Terminal handing		Sockets	File naming	Map-ping	Page faults	Signal handling	Process creation and termination
Raw tty	Cooked tty	Network protocols	File systems	Virtual memory			
	Line disciplines	Routing	Buffer cache	Page cache	Process scheduling		
Character devices		Network device drivers	Disk device drivers		Process dispatching		
Hardware							

Fig. 10-3 [Tan01]

Vrt. W2000 kuvassa Fig. 2.13 [Stal05]

47

## LINUX




NYKYAIKAINEN UNIX-JÄRJESTELMÄ

Ch 2.8 [Stal05]  
(ks. myös Ch 10 [Tan01])

48


# Linux



- Internet-ajan ryhmätyö
  - rakentajina ekspertit ympäri maailmaa
  - 1991 ->
- HY/TKTL: "*Linux on täältä kotoisin*"
  - Linus Torvalds opiskeli ja työskenteli TKTL:llä, aloitti tekemään Linuxia *tämän kurssin* suorittamisen jälkeen
- Free Software Foundation & GNU Public License
  - vapaasti levitettävä ytimen koodi (C & assembler)
  - vapaasti levitettäviä työvälineohjelmia
    - ohjelmankehitysympäristö: GNU C, ...
    - X Window ikkunointiympäristö: Gnome, KDE, ...
    - palvelimille tarkoitettuja ohjelmia: Apache, Samba, ...
- Useita kaupallisia jakelupaketteja
  - RedHat, SuSe, Debian, Mandrake, TurboLinux, jne...

49

# Linux



- POSIX + SysV:n ja BSD:n hyvät piirteet
  - LSB: Linux Standard Base
    - mitä pitää löytyä kaikista distribuutioista
- Toimii erilaisissa ympäristöissä
  - i386, IA64, Alpha, MIPS ...
- Upotetuissa järjestelmissä (embedded systems)
  - PDA-laitteet, digital-TV, jääkaappi, "rannetietokone"...
- Konfiguroitavuus
  - ytimeen käännetään vain ne osat, joita tarvitaan
  - koodia voi muuttaa omia tarpeitaan varten
- Optimointi
  - eri alustoilla erilaisia tarpeita
  - koodia saa muuttaa tarpeiden mukaan
    - Copyleft, GNU General Public License (GNU GPL)

50

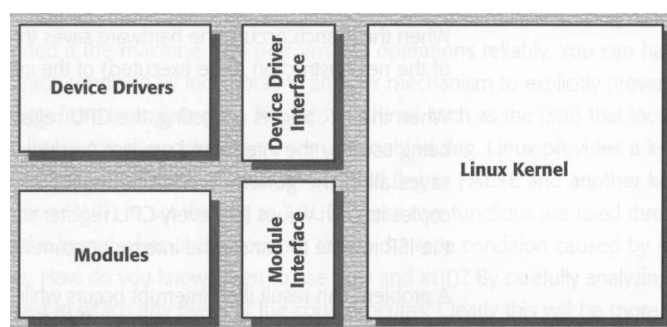
## Linux

- Perustuu alkuaan monoliittiseen ytimeen
  - ydin on yksi iso binäärikoodi
  - nopea ja vaivaton tapa välittää tietoa ytimen osien välillä
- Erikseen ladattavat moduulit
  - kaikkea ei tarvitse kääntää ytimeen
  - muistiin lisää koodia tarvittaessa (autom. / käsin)
    - dynaaminen linkitys
  - esim. laiteajurit, tiedostojärjestelmät (ext2, fat, ..)...
  - poisto, kun ei tarvita
- Moduulien riippuvuussuhteet
  - hierarkia moduulien käytössä (stackable modules)
  - keskeiset toiminnot omaksi 'kirjasto'-moduuliksi
  - ylempänä oleva käyttää alemman funktioita



51

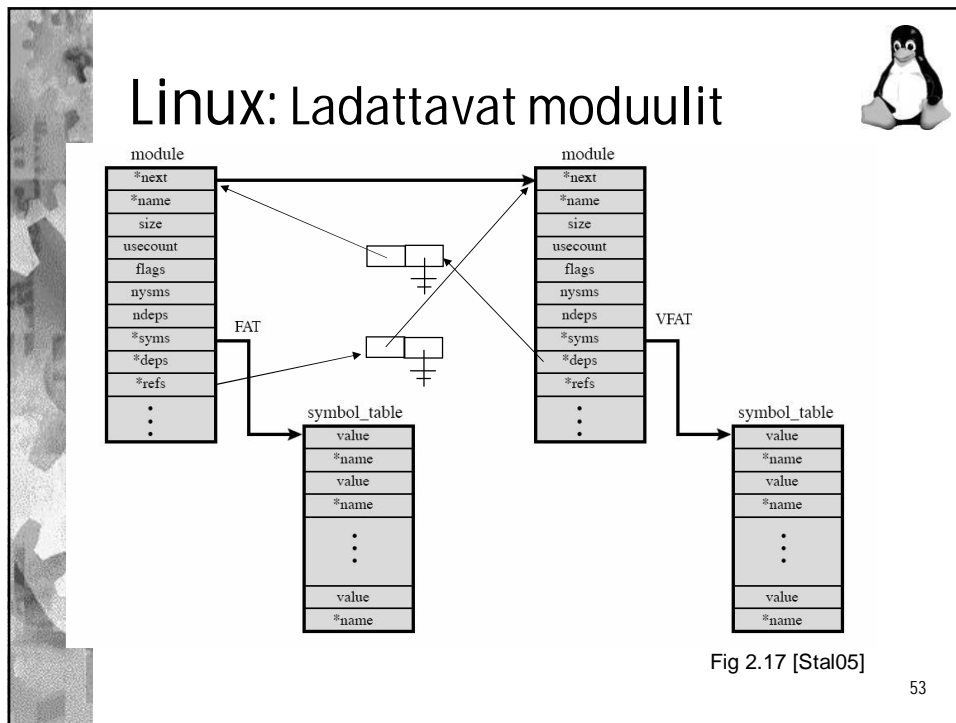
## Linux: Ladattavat moduulit



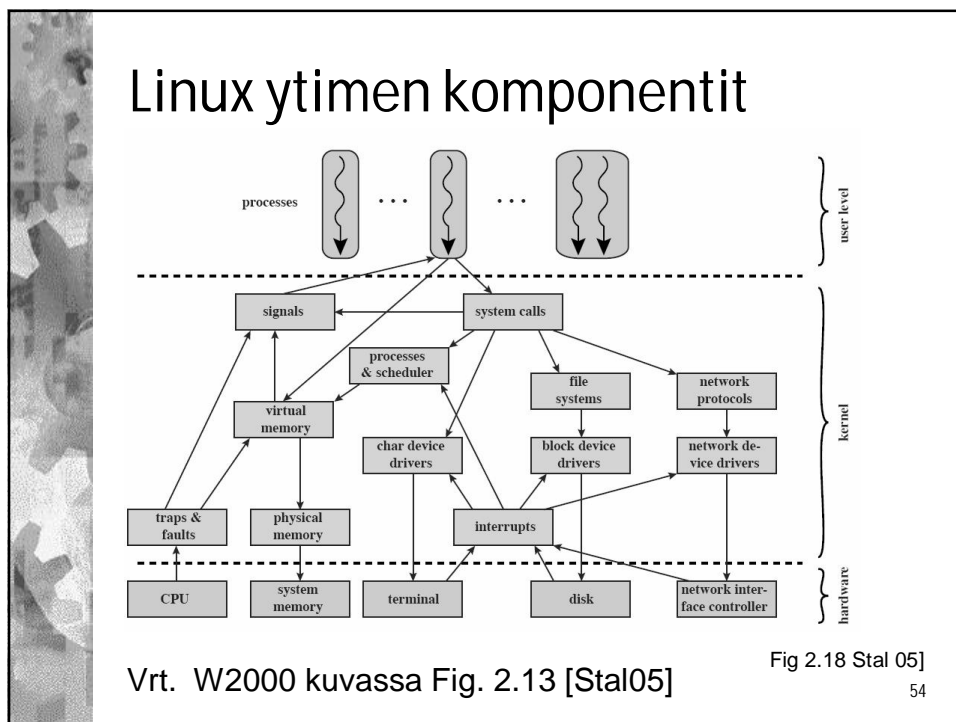
- Moduulit rekisteröitävä ytimelle
  - `init_module()`, `delete_module()`, ...
  - `register_blkdev()`, `unregister_blkdev()`, ...
  - `register_filesystem()`, `unregister_filesystem()`, ...



52



53



54

## Linux: Tiedon lähteitä



### Toteutusperiaatteet

- Bovet D.P., Cesati M.:  
*Understanding the Linux Kernel*. O'Reilly, 2<sup>nd</sup> ed., 2003.
- Beck M., Böhme H. & al. :  
*Linux Kernel Programming*. Addison-Wesley, 3<sup>rd</sup> ed., 2002
- Rubini A., Corbet J.:  
*Linux Device Drivers*. O'Reilly. 2001. 2nd ed.

### Koodiin tutustuminen

- <http://lxr.linux.no>

55