

# *Käyttöjärjestelmät II*

## **NYKYAIKAISET KÄYTTÖ- JÄRJESTELMÄT Ch 2.5 - 2.8 [Stal05]**

# Mitä KJ-I:ssä / KJ-II:ssa?

## KJ-I

### n Nykyaikainen KJ

- u monoliittinen KJ vs. mikroydin KJ
- u säie vs. prosessi
- u symmetrinen moniprosessointi (SMP)
- u hajautettu järjestelmä
- u olioperustainen rakenne

## Seuraavaksi KJ-II:ssa

### n Windows 2000 (Ch 2.5)

### n Perinteiset UNIX-järjestelmät (Ch 2.6)

### n Nykyaikaiset UNIX-järjestelmät (Ch 2.7-8)

# Käyttöjärjestelmät II

## *WINDOWS 2000*

### Ch 2.5 [Stal05]

(ks. myös Tan01, Ch 11)

# Windows 2000



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

# W95/98 vs. W2000

Fig 11-1 [Tan01]

XP?

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

# W2K Arkkitehtuuri

## n Muutettu (modified) mikrokernel-arkkitehtuuri

- u useat mikrokernelin ulkopuolisetkin K-palvelut suoritetaan etuoikeutetussa tilassa

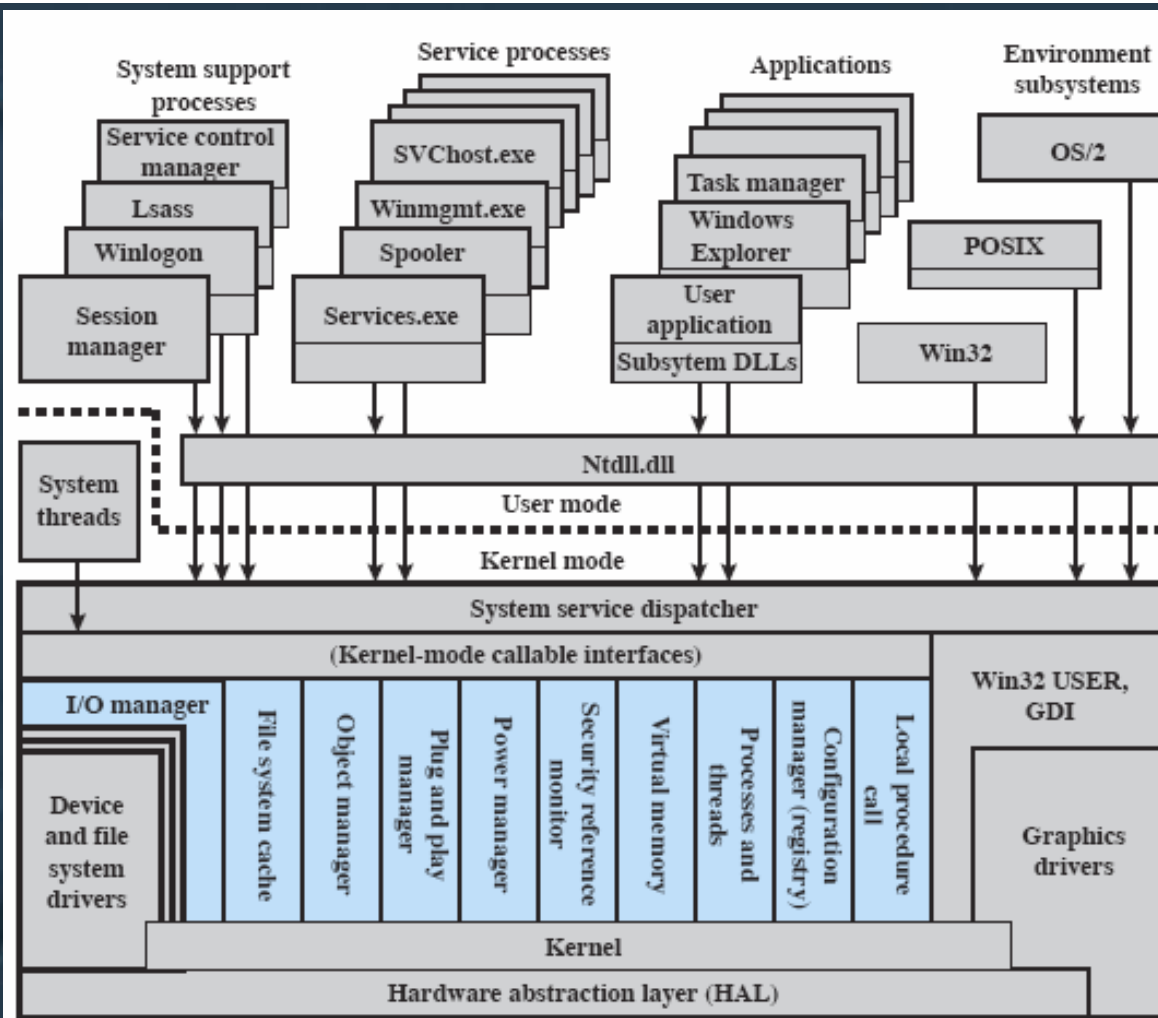
Miksi?

## n Modulaarinen rakenne

- u yksi KJ-palvelu per moduuli
  - u käyttö vain määritellyn rajapinnan kautta
  - u moduuleja voi muuttaa / poistaa koskematta muihin moduuleihin
- F mm. laiteajurit

## n KJ ja sovellukset eristetty toisistaan

- u käyttäjätila / etuoikeutettu tila
- u KJ:n tietorakenteita ja laitteistoa voi ronkkia vain etuoikeutetussa tilassa



Lsass = local security authentication server  
 POSIX = portable operating system interface  
 GDI = graphics device interface  
 DLL = dynamic link libraries

Colored area indicates Executive

Figure 2.13 Windows 2000 Architecture [SOLO00]

# W2K Etuoikeutettu tila

## HAL, Hardware Abstraction Layer

(ks. Fig. 2-13)

- u piilottaa laitteiston yksityiskohdat, abstrahointi
  - F väylät? keskeytys? DMA? ajastimet? muisti? SMP?
  - F laitteella muistiin kuvattu I/O vai ei? spin locks?
- u ylempi taso käyttää laitteistoa yleisten toimintojen tasolla
- u erilainen toteutus eri alustoille → siirrettävyys
  - F Intel (PowerPC, Alpha)
- u ei riittävän tehokas multimediasovelluksiin
  - F DirectX rajapinta



# W2K Etuoikeutettu tila

ks. Fig. 2-13

## Ydin (“mikrokernel”, kernel)

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

# W2K Etuoikeutettu tila

ks. Fig. 2-13

## Laiteajurit

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

# W2K Etuoikeutettu tila

ks. Fig. 2-13

## Windows 2000 Executive & Executive API

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

# W2K Etuoikeutettu tila

ks. Fig. 2-13

## Windows 2000 Executive & Executive API

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

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

# W2K Etuoikeutettu tila

ks. Fig. 2-13

## Windows 2000 Executive & Executive API

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

# W2K Käyttäjätila

ks. Fig. 2-13

## n **Special system support processes**

- u Käyttäjätilassa ajettavia KJ:n palveluja
  - F tehokkuus ei keskeinen tekijä
- u mm. logon, istunnon hallinta

## n **Service processes**

- u Muita kuin em. palveluja
- u mm. tapahtumaloki, etäproseduurin kutsu

# W2K Käyttäjätila

ks. Fig. 2-13

## Environment subsystems

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

## User applications

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

# W2K WIN32 API

## n WIN32 filosofia

- u usein 3-4 tapaa tehdä joku tietty asia
- u monimutkaisia liittymiä, jopa 10 parametria
- u osa ei selvästi kuuluisikaan API:iin
  - F esim. kokonaisen tiedoston kopioiminen

## n Palvelupyyntö palauttaa kahvan (handle) luomaansa olioon

- u käyttö seuraavissa kutsuissa



# WIN32 API palvelupyyntöjä

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

Lock

Unlo

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]

# W2K Asiakas-palvelija –malli

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

Local  
Procedure  
Call

# W2K Säikeet ja SMP

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

# W2K Oliot

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

# W2K Executive objects

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]

# W2K Oliot

(ks. Appendix B [Stal05])

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

# Käyttöjärjestelmät II

## *PERINTEINEN UNIX-JÄRJESTELMÄ*

Ch 2.6 [Stal05]

# Perinteinen (vanha) UNIX

## n Historia - ks. kirja

- u Bell Labs (v. 1970), AT&T, Berkeley University
- u SysV ja BSD

## n Suosion perustat

- u siirrettävyys
  - F C-kielellä, laitetoiminnot assemblerilla
- u monen käyttäjän moniajojärjestelmä
- u suunniteltu interaktiiviseen käyttöön
- u hierarkkinen tiedostojärjestelmä
- u vain yksi tiedostoformaatti: tavujono

## n UNIX filosofia: *pieni on kaunista*

- u pienten ohjelmien yhdistely
  - F yhden tuloste on toisen syöte
- u palvelupyynnöt vain minimimäärä

Fig. 2-15



# Perinteinen UNIX

## n Monoliittinen kernel

- u kaikki KJ:n osat käännettiin yhdeksi binäärikoodiksi
- u yksi yhteinen osoiteavaruus
  - F helppo pääsy KJ:n tietorakenteisiin ja funktioihin

## n Yhden prosessorin järjestelmä

## n Ei virtuaalimuistia, vain heittovaihto

- u ihan alussa, pian kyllä myös virt. muisti

## n Sovellus käyttää KJ-palveluja palvelupyynnöillä

- u keskeytys → siirtyminen KJ:n koodiin

## tai kirjastorutiinien välityksellä

- u hieman mukavampi liittymä palvelupyyntöihin

## n *Bach: The Design of the Unix Operating System, Prentice Hall, 1986.*

# Perinteinen UNIX

Fig. 2-15

## n Process control subsystem

- u prosessien luonti, vuorottaminen
- u muistinhallinta
- u prosessien välinen vuorovaikutus

## n File subsystem

- u tiedostojärjestelmä
- u siirräntäjärjestelmä
  - F lohkolaitteet (levyt), merkkilaitteet
- u lohkolaitteilla puskurivälimuisti

## n Device drivers

- u laitetta vastaa tiedosto hakemistossa /dev
  - F major, minor device number → ajuri löytyy laitekuvaajalistasta

# Käyttöjärjestelmät II

## *NYKYAIKAINEN UNIX-JÄRJESTELMÄ*

Ch 2.7-8 [Stal05]

# Nykyaikainen UNIX

## n Historian hyvät piirteet yhdessä paketissa

- u POSIX palvelupyntörajapinta
  - F standardoidut palvelupyynnöt (~ SysV  $\cap$  BSD)

## n Virtuaalimuistin käyttö

- u perinteisessä oli vain heittovaihto (swapping)

## n SMP-tuki

- u KJ suorituksessa useammalla prosessorilla
  - è poissulkeminen, synkronointi

## n Tuki erilaisille tiedostojärjestelmille (virtual file syst)

- u hajautettu tiedostopalvelu NFS, ynnä muut

## n Vaihtoehtoisia vuorottamisalgoritmeja

## n Erilaisia binääriformaatteja (executable)

- u a.out, ELF (Executable and Linking Format),
- u COFF (Common Object File Format), Portable Executable COFF

# Nykyaikainen UNIX

- n Muutettavuus ja laajennettavuus huomioitu
- n Pieni ydin, modulaarisia palveluja, joita muut KJ:n prosessit käyttävät
- n Daemon-prosessit
  - u KJ-palveluprosessit, jotka on luotu konetta käynnistettäessä

Fig. 2-16 [Stal05]

# Nykyaikainen UNIX



## SVR4

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

(preempt ~ "anastaa etuoikeuden nojalla")

# Nykyaikainen UNIX

## Solaris 2.x

- u Sun Microsystemsin versio SVR4:stä
- u lisäksi: fully preemptable kernel, SMP-tuki, säikeet, oliopohjainen tiedostojärjestelmä
- u tärkein kaupallinen Unix-versio



n *Goodheart, Cox: The Magic Garden Explained; The Internals of UNIX System V release 4. Prentice Hall, 1994.*

# Nykyaikainen UNIX



## 4.4 BSD

- u Berkeley Software Distribution
- u BSD:llä keskeinen rooli UNIX-kehittämisessä
- u Käytetty paljon akateemisissa piireissä
  - F UNIX-järjestelmien esikuva
- u Pohjana useille kaupallisille toteutuksille
  - F Mac OS X



- n *McKusick, Bostic, Karels, Quarterman:*  
*The Design and Implementation of the 4.4 BSD Operating System.*  
*Addison Wesley. 1996.*



# 4.4BSD 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]

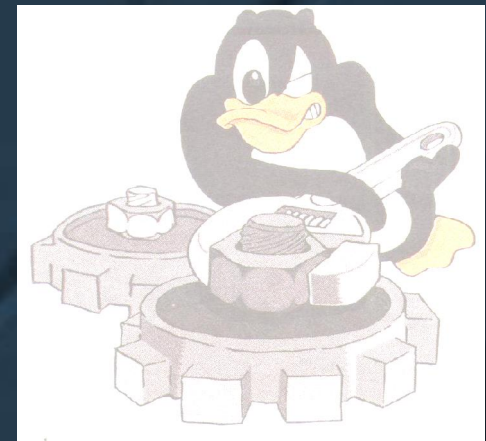
# Käyttöjärjestelmät II

NYKYAIKAINEN UNIX-JÄRJESTELMÄ

*LINUX*

Ch 2.8 [Stal05]

(ks. myös Ch 10 [Tan01])



# Linux



## n Internet-ajan ryhmätyö

- u rakentajina ekspertit ympäri maailmaa
- u 1991 ->

## n HY/TKTL: "*Linux on täältä kotoisin*"

- u Linus Torvalds opiskeli ja työskenteli TKTL:llä, aloitti tekemään Linuxia *tämän kurssin* suorittamisen jälkeen

## n Free Software Foundation & GNU Public License

- u vapaasti levitettävä ytimen koodi (C & assembler)
- u vapaasti levitettäviä työvälineohjelmia
  - F ohjelmankehitysympäristö: GNU C, ...
  - F X Window ikkunointiympäristö: Gnome, KDE, ...
  - F palvelimille tarkoitettuja ohjelmia: Apache, Samba, ...

## n Useita kaupallisia jakelupaketteja

- u RedHat, SuSe, Debian, Mandrake, TurboLinux, jne...

# Linux

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



# Linux

## n Perustuu alkuaan monoliittiseen ytimeen

- u ydin on yksi iso binäärikoodi
- u nopea ja vaivaton tapa välittää tietoa ytimen osien välillä

## n Erikseen ladattavat moduulit

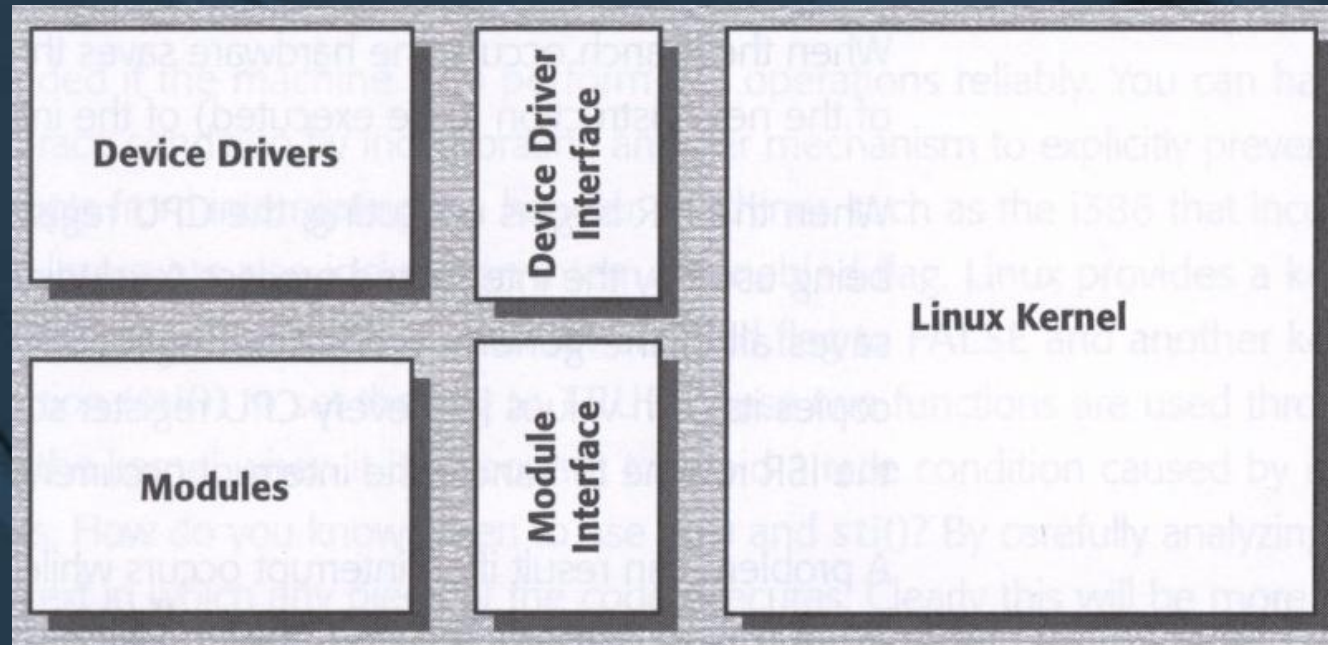
- u kaikkea ei tarvitse kääntää ytimeen
- u muistiin lisää koodia tarvittaessa (autom. / käsin)
  - F dynaaminen linkitys
- u esim. laiteajurit, tiedostojärjestelmät (ext2, fat, ..)...
- u poisto, kun ei tarvita

## n Moduulien riippuvuussuhteet

- u hierarkia moduulien käytössä (stackable modules)
- u keskeiset toiminnot omaksi 'kirjasto'-moduuliksi
- u ylempänä oleva käyttää alemman funktioita



# Linux: Ladattavat moduulit



## n Moduulit rekisteröitävä ytimelle

- u `init_module()`, `delete_module()`, ...
- u `register_blkdev()`, `unregister_blkdev()`, ...
- u `register_filesystem()`, `unregister_filesystem()`, ...



# Linux: Ladattavat moduulit

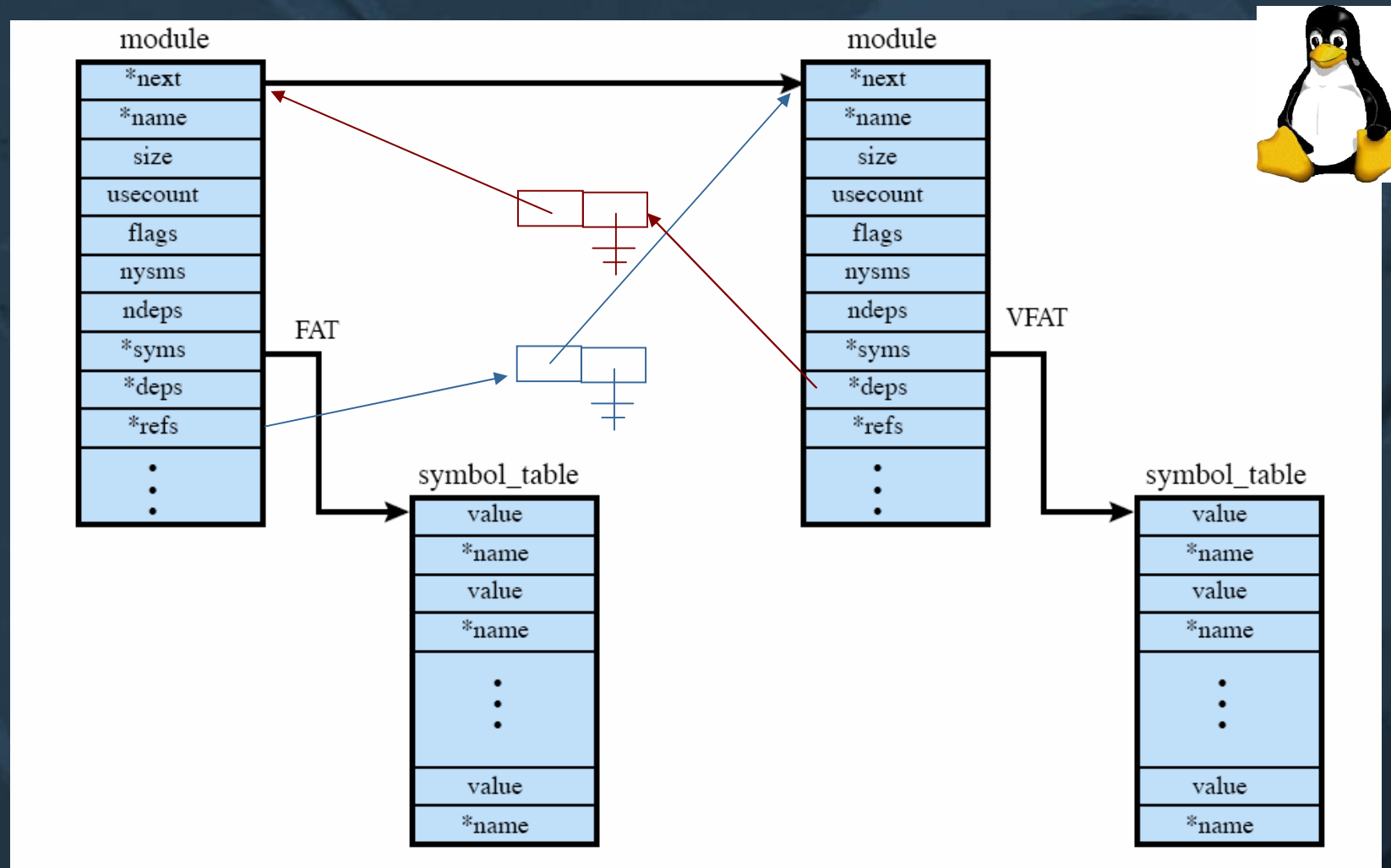


Fig 2.17 [Stal05]

# Linux ytimen komponentit

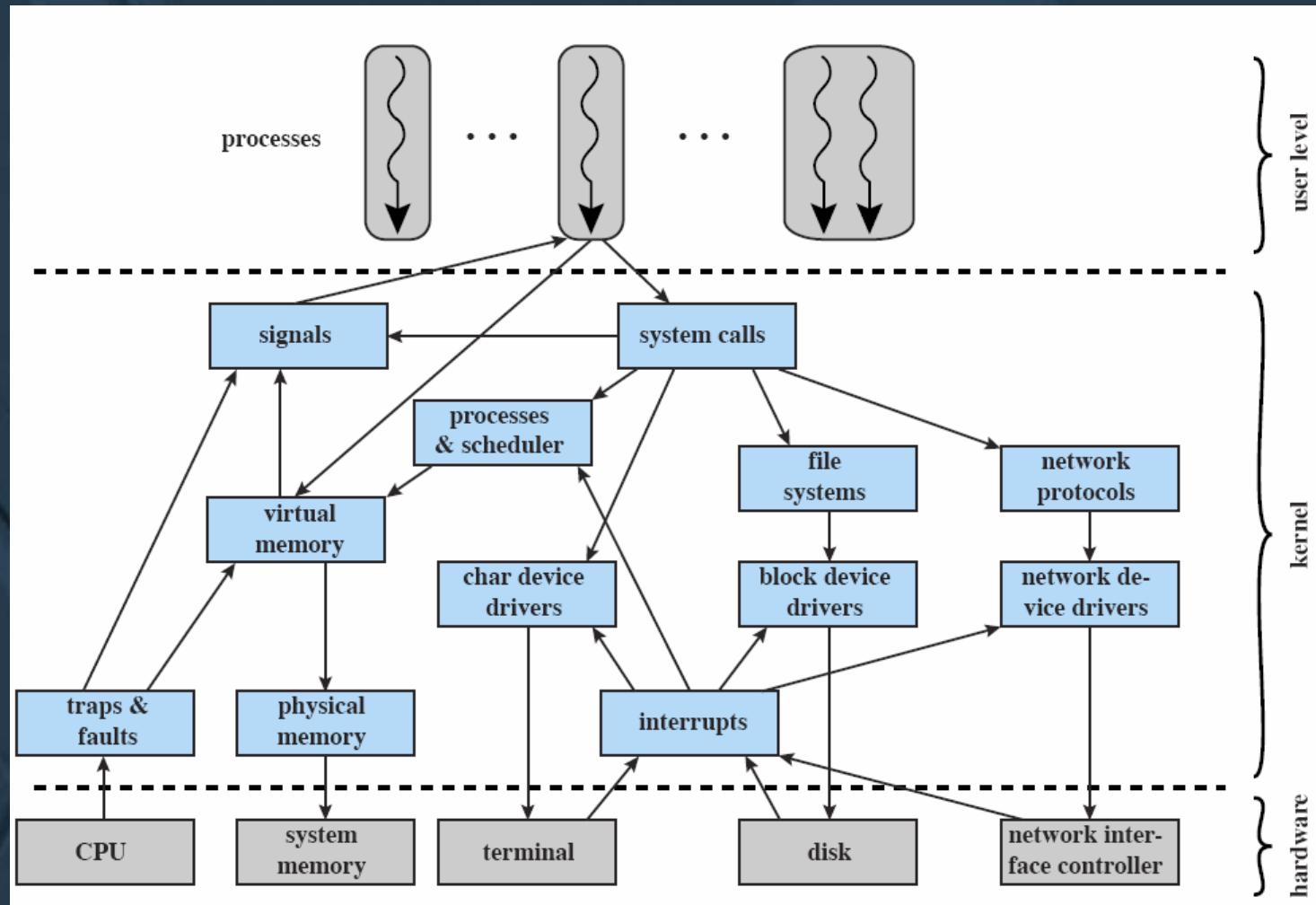


Fig 2.18 Stal 05]

Vrt. W2000 kuvassa Fig. 2.13 [Stal05]



# Linux: Tiedon lähteitä

## Toteutusperiaatteet

n Bovet D.P., Cesati M.:

*Understanding the Linux Kernel. O'Reilly, 2<sup>nd</sup> ed., 2003.*

n Beck M., Böhme H. & al. :

*Linux Kernel Programming. Addison-Wesley, 3<sup>rd</sup> ed., 2002*

n Rubini A., Corbet J.:

*Linux Device Drivers. O'Reilly. 2001. 2nd ed.*

## Koodiin tutustuminen

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

