

**IKO31204**

**Pemrograman Sistem**

**Jilid 4: Pengenalan Device Driver**

Fakultas Ilmu Komputer - Universitas Indonesia  
Sep 2012

# topik

Peran module / driver

Struktur pembagian tugas Kernel

Kategori *Device* dan *Module*

load & unload module

interaksi user & kernel level

# referensi

Linux Device Driver, 2<sup>nd</sup> edition, Alessandro  
Rubini, Jonathan Corbet, O'Reilly,  
**Chapter 1 & 2**

# module ?

pieces of code that can be loaded and unloaded into the kernel on demand. They extend the functionality of the kernel without the need to reboot the system.

# pendahuluan

- => Popularitas suatu OS
- => Setiap Vendor menginginkan Hardwarenya dapat berfungsi Di atas suatu Sistem Operasi
- => Programmer Driver semakin banyak
- => Tanpa Device Driver, keseluruhan fungsi Sistem tidak akan optimal

# Peran Driver

- => Menyembunyikan kerumitan / detail, bagaimana hardware tsb berinteraksi dengan sistem
- => Memetakan antara fungsi spesifik Hardware dengan fungsi standard
- => Aplikasi User Level cukup memanggil fungsi standard tsb

# Motivasi membuat Driver

- => Meningkatnya jumlah pengguna (potensial market)
- => Permintaan agar Hardware dapat digunakan
- => Source code terbuka luas, memungkinkan seseorang untuk mencoba “membunuh” rasa *curiosity*
- => dll

# Ide Dasar Desain *Unix* (1)

Ide Desain Pemrograman OS Unix dapat dibagi menjadi 2 bagian:

- => Apa fungsi yang harus disediakan (mekanisme)
- => Bagaimana fungsi tsb boleh / dapat digunakan (kebijakan).

Apabila konsep kedua bagian tersebut dipisahkan, implementasi dan adaptasi perubahan pada sistem relatif lebih mudah untuk dicapai



# Ide Dasar Desain *Unix* (II)

Contoh :: XWindows + Desktop Env.

=> XWindows mengenali dan mengelola hardware Grafik (cth: VGA Card) dan menyediakan fungsi generik bagi Desktop Env.

=> Desktop Env. memberikan “look and feel” bagi user. Cth: GNOME dan KDE, keduanya mengakses Hardware dgn cara yg sama (melalui XWindows)

# Prinsip Driver Devel.

write kernel code to access the hardware, but don't force particular policies on the user, since **different users have different needs.**

The driver should deal with **making the hardware available**, leaving all the issues about how to use the hardware to the **applications**

# Pembagian Peran Kernel

## => **Process management**

Create/Destroy process, Schedule resource, dll

## => **Memory management**

Virtual Addressing, handle malloc

## => **Filesystems**

Hampir semua hal di UNIX dianggap sebagai file. Kernel menciptakan filesystem terstruktur di atas hardware yang tidak terstruktur

# Pembagian Peran Kernel (II)

## => **Device Control**

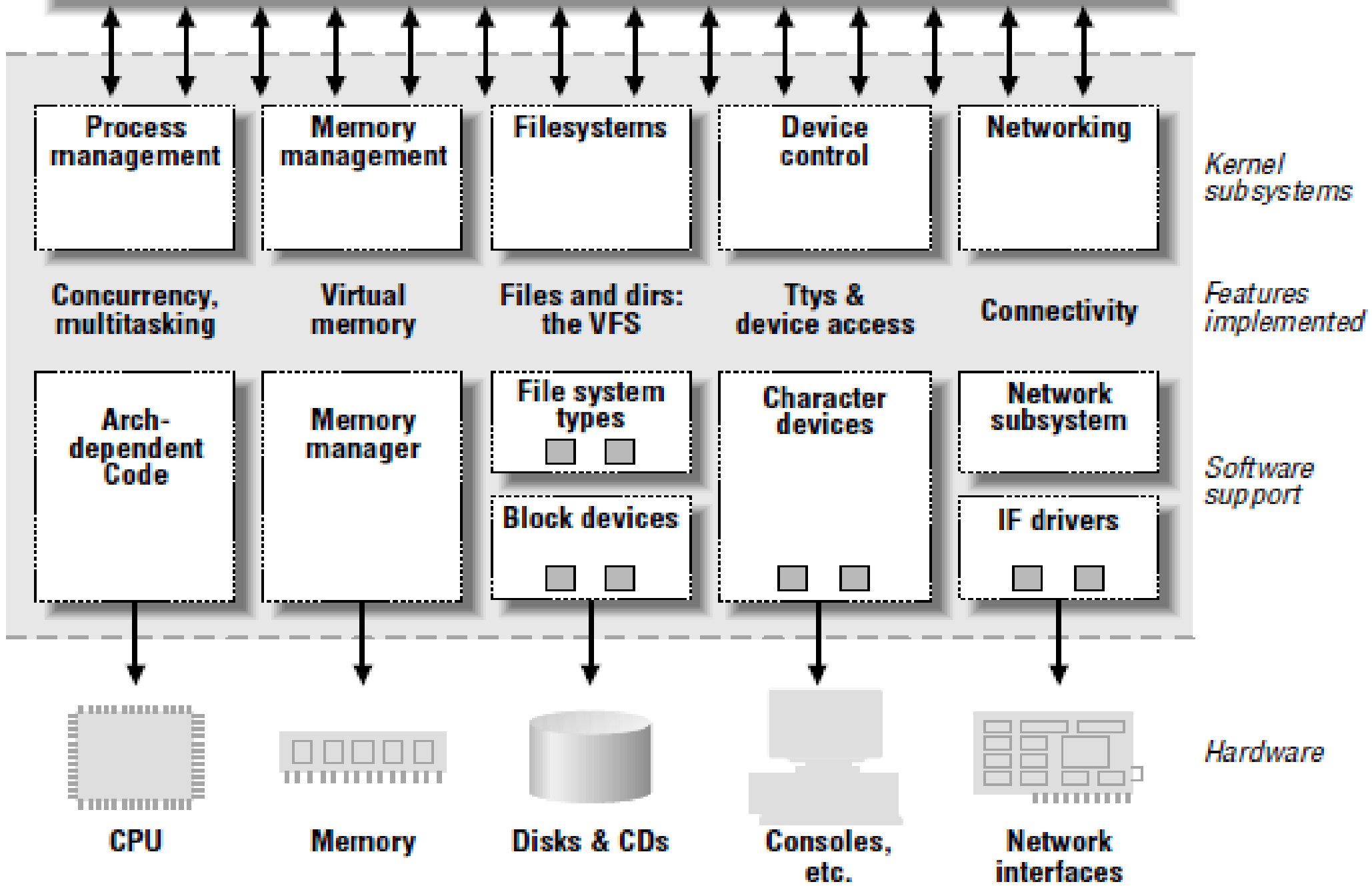
Mengendalikan hardware (kecuali prosesor & memori)

## => **Networking**

Perlu ditangani terpisah karena aktifitas networking berkaitan dengan banyak proses & driver (tidak spesifik ditangani oleh sebuah proses saja)

NB: Coba ingat-ingat 4 Layer TCP/IP

# The System Call Interface



 *features implemented as modules*

# Security

## => **Process management**

Create/Destroy process, Schedule resource, dll

## => **Memory management**

Virtual Addressing, handle malloc

## => **Filesystems**

Hampir semua hal di UNIX dianggap sebagai file. Kernel menciptakan filesystem terstruktur di atas hardware yang tidak terstruktur

# lokasi module

BASE PATH:

/lib/modules/[kernel version]/kernel

KHUSUS HARDWARE DRIVER:

[BASE PATH]/drivers/\*

# perkakas module

```
# man insmod  
# man lsmod  
# man rmmod  
# man modinfo  
dll
```



# load/unload module

Load:

```
# insmod [abs path] [params]
```

atau

```
# modprobe [alias path] [params]
```

# load/unload module

```
# insmod /lib/modules/2.4.27-2-  
386/kernel/drivers/block/loop.o  
max_loop=32
```

atau

```
# modprobe loop max_loop=32
```

# load/unload module

cth:

# lsmod

Module	Size	Used by	Not tainted
ide-cd	27072	0	
cdrom	26212	0	[ide-cd]
rtc	5768	0	(autoclean)
af_packet	11048	1	(autoclean)
.....			

# rmmmod ide-cd

# rmmmod cdrom

# interaksi user & kernel layer

## System Call

cth: open, read, write, close, wait, fork,  
dll

## PROC filesystem

cth:  
/proc/cpuinfo  
/proc/uptime  
dll

latihan  
perkakas  
module &  
proc fs  
(20 menit)

1. Cari & coba modul yang berkaitan dengan salah satu PROC FS (/proc).  
Temukan lokasi profile modul tersebut pada /proc/??

cth: bonding, usbserial,

2. Coba ganti parameter modul e1000 dengan debug=16

latihan script  
programming  
dengan kernel  
message  
(20 menit)

buat script untuk mendeteksi kabel jaringan tersambung / terputus.

- Apabila tersambung menampilkan "TERSAMBUNG", apabila jaringan putus menampilkan "PUTUS".
- Interval pemeriksaan 1 detik sekali.
- Tulisan "TERSAMBUNG" dan "PUTUS" hanya tampil kalau ada perubahan saja

HINT:

1. gunakan while true BASH, if-else-fi
2. dmesg, cut, sleep
3. cari kata-kata "link up" / "link-down"



tanya jawab