# Praktikum18

# Manajemen Hardware

## POKOK BAHASAN:

- ✓ Manajemen Hardware
- ✓ BIOS dan beberapa Core Hardware (DMA,IRQ,I/O)
- ✓ Kartu ekspansi
- ✓ Devais USB
- ✓ modprobe

## TUJUAN BELAJAR:

- ✓ Mahasiswa dapat melakukan manajemen hardware
- ✓ Mahasiswa dapat melakukan setting BIOS dan beberapa Core Hardware (DMA,IRQ,I/O)
- ✓ Mahasiswa dapat melakukan setting kartu ekspansi
- ✓ Mahasiswa dapat melakukan setting devais USB
- ✓ Mahasiswa dapat menambahkan dan menghapus modul dari linux kernel

## TUGAS PENDAHULUAN

1. Apa yang disebut device driver dan kernel modul ? Apa beda keduanya ?

2. Saat dilakukan ls -al pada dua devais dibawah didapat hasil berikut.

```
$ ls -l /dev/hda
brw-rw----    1 root     disk       3,   0 Apr 11  2002 /dev/hda
$  ls -l /dev/ttyS1
crw-rw----    1 root     uucp       4,  65 Apr 11  2002 /dev/ttyS1
```

a. Pada hasil ls -l, huruf b dan c menunjukkan bahwa hda adalah block devais dan ttyS1 adalah karakter devais. Apa maksud dari blok dan charakter devais. Sebutkan contoh nya sebanyak-banyaknya.

b. Apa fungsi major node dan minor node? Pada hasil ls -l /dev/hda, ini ditunjukkan oleh 3 (major) 0 (minor)

3. Apa yang disebut pseudo devais dan apa bedanya dengan devais biasa ? Beri contoh pseudo devais di Linux.

4. Apa fungsi /proc, /sys dan /dev

## DASAR TEORI:

### Aplikasi User Space dan Kernel Space

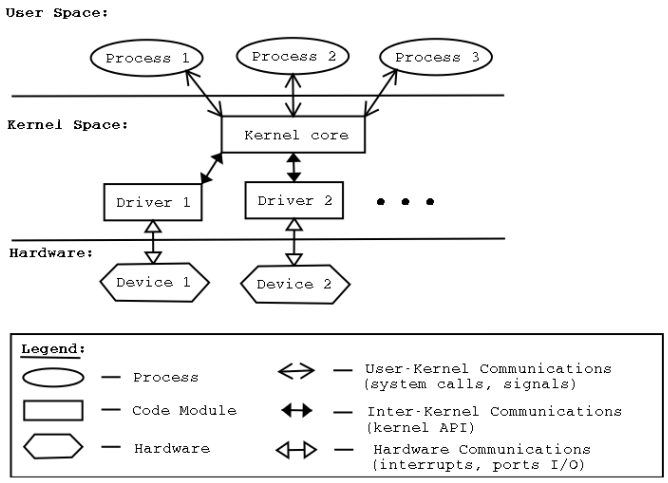
Ketika anda menulis aplikasi, perlu diperhatikan perbedaan antara aplikasi yang berada pada user space dan kernel space.

- Aplikasi Kernel space

Kernel bertugas melakukan manajemen hardware secara efisien dan memberikan layanan interface programming untuk aplikasi berbasis user space yang bersifat uniform. Kernel, lewat device drivernya membentuk semacam interface antara aplikasi untuk end-user(user space) dan hardware. Semua subrutin dan fungsi yang membentuk kernel (contohnya kernel module dan device driver) merupakan bagian dari aplikasi pada kernel space.

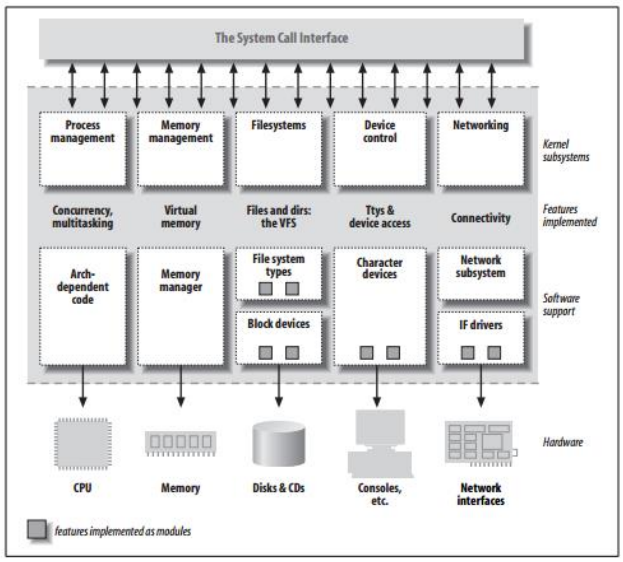
- Aplikasi User Space

Program-program yang dijalankan oleh enduser seperti shell UNIX atau aplikasi berbasis GUI termasuk dalam user space.Aplikasi user space harus berinteraksi dengan hardware yang ada, namun interaksi ini tidak berjalan secara langsung, melainkan lewat fungsi-fungsi yang disediakan kernel.




### Kernel Linux

Pada bab sebelumnya, telah dibahas mengenai konsep dibalik monolithic dan modular kernel. Hampir semua komponen devais feriferal (hardware) disimpan dalan modular kernel, komponen di kernel linux yang telah dikompilasi sebagai modul yang dapat secara dinamik diload dan dihapus ketika dibutuhkan. Pada gambar dibawah ditunjukkan diagram fungsional dari kernel linux




**Apa beda modul kernel dan device driver ?**

- Modul kernel adalah kumpulan kode yang dapat diloading atau unloading saat kernel sedang berjalan tanpa harus merestart system atau yang disebut *run-time*. Modul kernel ini dapat berupa filesystem, device driver, protokol network, fungsionalitas firewall, dst.
- Device driver adalah kumpulan kode yang digunakan oleh kernel untuk bercakap-cakap dengan hardware. Hampir seluruh hardware memiliki device driver sendiri. Seperti sopir bis (bus driver) bertugas mengemudikan bis (hardware), disini device driver berkuasa sepenuhnya atas hardware.

Kernel sendiri sebagian besar terdiri atas device driver. Beberapa hardware membutuhkan beberapa modul kernel sekaligus, dimana sebagian modul menangani hardware, sementara modul yang lain menangani protokol infrastruktur hardware tersebut . Contohnya pada WiFi.

Device driver dapat dibuilt secara statis ke kernel di disk. Contohnya pada direktory /boot, yang diloading oleh RAM saat booting dengan menggunakan boot loader (tepatnya pada stage awal proses booting). Device driver dapat pula dibuilt sebagai kernel module yang dapat diload secara dinamis pada stage proses booting berikutnya (bahkan terkadang tidak perlu diload ke kernel)

## Modular Kernel

Pada Linux, modul-modul yang dibutuhkan disimpan di /lib/modules/<versi-kernel>/. Komponen yang dibuat moduler ini sebaiknya adalah komponen yang tidak dibutuhkan sewaktu booting, misalnya komponen devais feriferal dan file system tambahan.

## Modul

Modul adalah file berukuran kecil yang dapat diload dan di-unload ke kernel sesuai permintaan admin. Dengan adanya modul, fungsionalitas kernel akan semakin bertambah. Selain itu, admin tidak perlu melakukan reboot tiap kali melakukan loading dan unloading. Salah satu tipe modul adalah device driver, yang memungkinkan kernel untuk mengakses hardware. File modul ini ditandai dengan ekstensi .ko

**Tabel 1.1 Perintah-perintah pada loadable kernel modul**

| Perintah | Tugas |
|---|---|
| depmod | Mengurusi dependensi untuk modul kernel yang loadable |
| insmod | Menginstall modul kernel yang loadable |
| lsmod | Melisting modul yang diloading |
| modinfo | Mendisplay informasi mengenai modul kernel tertentu |
| modprobe | Melakukan loading terhadap module |
| rmmod | Melakukan unloading terhadap modul tertentu |

## Mengapa device driver diterapkan sebagai kernel modul ?

Sering kali, di Linux, driver dibuilt sebagai modul kernel, bukan di-link secara statis ke kernel, karena alasan fleksibilitas (mengurangi ukuran kernel). Selain itu juga disebabkan :

- Penggunaan fitur initrd menyebabkan OS mampu melakukan booting tanpa menggunakan device driver yang vital sekalipun
- Dengan melakukan built secara statis, menyebabkan OS bersifat statik. Artinya, jika OS tahu driver mana yang nantinya akan dibutuhkan, maka OS tidak memerlukan modul kernel. Konsekuensinya, OS akan selalu menggunakan device driver tersebut. Masalahnya, seringkali, driver yang tidak penting juga di-load oleh OS. Dengan menggunakan modular kernel, OS hanya akan melakukan loading modul-modul yang dibutuhkan saja dan melakukan unloading modul ketika modul-modul tersebut tidak dibutuhkan.
- Jika ada perubahan kecil pada kernel, anda tidak harus melakukan building dari awal, namun cukup mengkompile secara terpisah dan kemudian di-load dalam kernel.
- Dengan memisahkan komponen menjadi modul, maka debugging modul menjadi lebih mudah karena jika ada kesalahan cukup booting lewat base kernel tanpa modul untuk melihat apakah system error/crash terjadi saat booting. Jika booting berjalan baik, maka anda dapat menambahkan modul yang anda curigai sebagai penyebab crash untuk melihat apakah booting berjalan dengan baik atau tidak. Jika terjadi crash, maka modul tersebut adalah penyebabnya. Anda tinggal melakukan debug pada modul tersebut

- Tidak perlu melakukan rebooting system ketika terjadi perubahan code.
- Tidak semua kernel modul adalah driver, contohnya fitur terbaru dari linux kernel adalah kernel modul dengan proses scheduler yang bervariasi.

### File /etc/modules

File /etc/modules.conf digunakan untuk menyimpan parameter modul (IRQ dan port I/O) namun isi dari file ini kebanyakan berisikan sejumlah alias.

alias : merujuk ke devais yang dengan nama yang berbeda.

Contoh: devais ethernet pertama yang disebut sebagai eth0, bukan dengan nama driver devais tersebut (natsemi)

options : perintah untuk menjelaskna spesifikasi modul

```
#cat /etc/modules.conf
alias eth0 natsemi
alias eth1 ipw2200
alias snd-card-0 snd-ali5451
options snd-card-0 index=0
options snd-ali5451 index=0
remove snd-ali5451 { /usr/sbin/alsactl store 0 >/dev/null
2>&1
|| : ; }; /sbin/modprobe -r --ignore-remove snd-ali5451
```

**Dependensi**

Beberapa modul bersifat dependen terhadap modul-modul yang lain. Artinya ketika modul tersebut di-loading atau di-unloading, dibutuhkan modul-modul lain. File database dari dependensi modul ini disimpan di file /lib/modules/<kernel-version>/modules.dep. File ini dibuat berdasarkan perintah depmod yang dijalankan saat booting (lewat skrip rc.sysinit).

Bagaimana jika driver hardware kita belum terkompilasi ?

- Beberapa driver hardware yang dijumpai menjadi bagian dari source kernel, namun tidak terkompilasi
- Hal ini dapat dilihat pada kernel source tree
- Atau lewat dokumentasi pada /usr/src/linux/Documentation
- Untuk itu, kita cukup melakukan kompilasi terhadap driver tersebut, tanpa harus mengkompilasi keseluruhan kernel

**Konfigurasi Hardware**

- Untuk mengkonfigurasi driver dari harware yang tidak dikenal, pertama anda harus mengetahui informasi yang lebih detail mengenai hardware tersebut
- Masalah yang sering dihadapi adalah tipe produk hardware yang bermasalah tersebut tidak diketahui, atau terlalu umum
- Informasi yang harus kita punya adalah nama manufaktur dari hardware tersebut dan model chipset motherboardnya
- Device driver berkomunikasi dengan hardware lewat informasi tersebut.
- Sebagian besar device driver untuk hardware sudah menjadi bagian dari kernel dan disimpan di /lib/modules/<versikernel>

**Instalasi driver**

- Download driver dari site 3rd party atau dari repository debian
- Jika driver berupa packet binary, anda dapat menggunakan perintah apt-get install
- Jika driver berupaka packet source, maka anda harus melakukan kompilasi

**Direktori Penting untuk Hardware**

**1. Dikrektori dev**

Semua devais di Linux dapat diwakili dalam bentuk file. File yang dimaksud adalah file-file yang berada pada direktory /dev. File-file devais ini dibuat :

(1) saat instalasi, saat Linux mengenali devais-devais yang terpasang pada komputer

(2) lewat skrip di /dev/MAKEDEV untuk driver-driver hardware yang tidak standard.

| | Nama file | Keterangan |
|---|---|---|
| Block Device | /dev/fd0, /dev/fd1, ... | Drive floppy pertama, kedua, dst |
| | /dev/hda, /dev/hdb,/dev/hdc, /dev/hdd | Hardisk IDE pertama, hard disk kedua, hard disk ketiga,hard disk keempat, |
| | /dev/hda1 s.d. /dev/hda15 | Harddisk IDE pertama partisi 1 s.d. partisi 15 |
| | /dev/hdb1 s.d. /dev/hdb15 | Harddisk kedua partisi 1 s.d. partisi 15 |
| | /dev/hdc1 s.d. /dev/hdc15 | Harddisk ketiga partisi 1 s.d. partisi 15 |
| | /dev/hdd1 s.d. /dev/hdd15 | Harddisk keempat partisi 1 s.d. partisi 15 |
| | /dev/sda, /dev/sdb,... | Hardisk SCSI pertama, hard disk SCSI kedua,... |
| | /dev/sda1 s.d. /dev/sda2 | Hardisk SCSI pertama partisi 1 s.d. partisi 15 |
| | /dev/sdb1 s.d. /dev/sdb2 | Hardisk SCSI kedua partisi 1 s.d. partisi 15 |
| | /dev/cdrom | CDROM |
| | /dev/ht0 | Tape drive IDE pertama |

| | | |
|---|---|---|
| Character Device | /dev/mouse | Mouse |
| | /dev/psaux | PS/2 mouse port. |
| | /dev/modem | Modem |
| | /dev/fb0 | Framebuffer pertama. Framebuffer adalah adalah layer abstraksi antara software dan hardware |
| | /dev/mixer | Driver OSS (Open Sound System) |
| Serial Port | /dev/ttyS0 - /dev/ttyS3 | Interface serial 0 hingga 3 |
| | /dev/cua0 - /dev/cua3 | Interface serial 0 hingga 3 (untuk modem) |
| Pararel Port | /dev/lp0 - /dev/lp2 | Printer 0 hingga 2 |
| | /dev/pt0 | Pararel port tape device |
| Other | /dev/null | Data apapun (data bin) akan di"telan" |
| | /dev/zero | Menghasilkan keluaran bilangan dengan byte null |
| | /dev/tty1 - /dev/tty8 | Konsol virtual |

**2. Direktori /proc**

Direktori proc ini berfungsi menyediakan informasi tentang sistem secara real time. Direktori ini unik karena isinya tidak tersimpan dalam bentuk file melainkan disimpan dalam memory. Dibawah ini dijelaskan mengenai beberapa direktori proc beserta fungsinya.

| **Direktori** | **Keterangan** |
|---|---|
| /proc/cpuinfo | Infomasi mengenai prosesor, termasuk diantaranya : tipe, vendor, model dan performansinya |
| /proc/devices | Informasi mengenai list device driver yang sedang berjalan di kernel |
| /proc/dma | Informasi mengenai kanal DMA yang sedang dipakai |
| /proc/filesystems | Informasi mengenai filesistem |
| /proc/interrupts | Informasi mengenai interupt mana yang dipakai dan berapa yang tersedia |
| /proc/ioports | Informasi mengenai port I/O yang sedang dipakai |
| /proc/kcore | Informasi mengenai memory fisik |
| /proc/kmsg | Informasi mengenai kernel yang dihasilkan syslog |
| /proc/ksyms | Tabel simbol untuk kernel |
| /proc/loadavg | Informasi mengenai beban rata-rata system (load average) yang menunjukkan kinerja system secara real |
| /proc/meminfo | Informasi mengenai penggunaan memori baik memory fisik dan swap |
| /proc/modules | Informasi mengenai modul-modul kernel diloading saat itu |
| /proc/net | Informasi status protokol network |
| /proc/stat | Informasi statistik dari systen, misalnya jumlah page fault ketika sistem dibooting |
| /proc/uptime | Informasi penggunaan waktu sistem been up. |
| /proc/version | Informasi tentang versi kernel |

**3. Filesystem Sysfs dandirektori /sys**

Sysfs adalah sistem file virtual yang dipakai oleh kernel Linux untuk mengekspor informasi mengenai kernel hingga proses yang berjalan pada user space. Filesistem tersebut mulai digunakan pada kernel 2.6, dan merupakan pengembangan dari sistem file ramfs pada kernel 2.4. Filesistem sysfs ini merupakan virtual in-memory filesystem yang dimount di direktory /sys.

| Subdirektory /sys | Fungsi |
|---|---|
| /block | Contains known block devices |
| /bus | Contains all registered buses. |
| /class | Contains Devices |
| /device | All devices known by the kernel organised by the bus that they connect to |
| /firmware | Contains firmware files for some devices |
| /fs | Contains files to control filesystems |
| /kernel | Various kernel related files |
| /module | Loaded kernel modules. Each module is represented by a directory of the same name. |
| /power | Various files to handle power state of system |

# PERCOBAAN

## Percobaan 1. Melihat informasi hardware lewat direktori /proc

1. Menunjukkan ukuran RAM fisik yang dapat dipakai sistem

```
#grep MemTotal /proc/meminfo
```

2. Menunjukkan tipe CPU yang dipakai

```
# cat /proc/cpuinfo
```

3. Menunjukkan semua partisi disk

```
#cat /proc/partitions
```

4. Menunjukkan list nomor interrupt IRQ yang sedang digunakan oleh driver

```
#cat /proc/interrupts
```

5. Me

```
# cat /proc/devices
```

## Percobaan 2. Melihat informasi hardware lewat perintah ls*

1. Listing hardware berikut settingnya secara komprehensif

```
#lshw> hardware.txt
```

Untuk format yang berupa tabel yang lebih singkat

```
#lshw -short
```

Untuk mencari hardware tertentu, dapat digunakan opsi class. Ada beberapa class yang dipakai, yaitu : address, bridge, bus, communication, disk, display, generic, input, memory, multimedia, network, power, printer, processor, storage, system, tape,volume

```
#lshw -class memory
```

Menampilkan hardware tertentu dalam bentuk tabel

```
#lshw -short -class memory
```

Menampilkan beberapa hardware bersama-sama dalam bentuk tabel

```
# lshw -short -class disk -class storage -class volume
```

2. Melisting semua hardware yang terkoneksi lewat USB

```
#lsusb
```

3. Melisting semua hardware yang terkoneksi lewat bus PCI

```
#lspci
```

Output lspci :

**03:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTS5209 PCI Express Card Reader (rev 01)**

terdiri atas 3 bagian yaitu : no bus slot PCI, nama slot PCI, nama devais dan nama manufakturer.

Untuk melihat informasi yang lebih detail :

```
#lspci -v
```

Untuk

```
#lspci -vv
```

VGA card

```
$ lspci -nnk | grep VGA -A1
```

Sound Card

```
$lspci -v | grep -A7 -i "audio"
```

NIC card

```
$ lspci -nnk | grep net -A2
```

4. Menampilkan informasi detail mengenai cpu

```
#lscpu
```

5. Menampilan seluruh block device

```
#lsblk
```

6.

```
# lsb_release -a
```

**Percobaan 3. Melihat informasi hardware tertentu**

## CPU

Menunjukkan informasi mengenai cpu, tipe socket, vendor dan beberapa flag lainnya.

```
#dmidecode -t 4
```

## Memory

```
#free -m
#top
#htop
#vmstat
#dmidecode -t 17
```

## Hard Disk

1. Disk Usage (du) : Mengestimasi pemakaian disk

```
#du
```

2. Disk Free (df) : Mengestimasi % penggunaan disk

```
#df -h
```

3. hdparm : menunjukkan informasi mengenai disk sda

```
#hdparm
```

4. Melihat partisi yang dimount

```
#mount
```

5. Melihat ukuran hard disk yang dipakai

```
# fdisk -l | grep '^Disk /dev/'
```

## Percobaan 4. Mengecek Informasi Umum

1. Text based

```
#hwinfo
```

Melihat informasi dalam bentuk tabel

```
#hwinfo --short
```

2.GUI based

Install terlebih dahulu aplikasi dibawah dengan apt-get install jika belum ada

```
#hardinfo
#lshw-gtk
#sysinfo
```

## Percobaan 5. Melihat Loadable Modul dan Dependensi Modul

1. Melihat seluruh modul yang diload di OS

```
#lsmod
#cat /proc/modules
```

2. Melihat klasifikasi semua module tersebut. Untuk melihat versi kernel, gunakan perintah uname -r. Mari kita lihat is direktori modul device driver.

```
#uname -r
#cd /lib/modules/<versikernel>/kernel/drivers
#ls -al
```

Anda akan melihat sekumpulan direktory dimana seluruh modul diklompokkan sesuai dengan fungsinya seperti fs, sound, net, dll. didalam direktori. File module sendiri ditandai dengan ekstensi .ko

3. Melihat info modul

Perintah modinfo memberikan informasi mengenai modul, pembuat modul, dependensi, tipe lisensi dan parameter modul.

```
#modinfo usbcore
```

4. Menghapus dan Memasukkan modul

Sebelumnya cek apakah modul tersebut sudah ada atau belum dan bagaimana dependensinya. Jika ada dependensinya, maka anda harus menghapus juga modul tersebut. Gunakan modul ppdev pada percobaan ini.

```
#lsmod | grep usbcore
```

Hapus module tersebut dengan perintah rmmod. Cek dg lsmod, apakah modul tersebut masih ada atau tidak.

```
#rmmod  usbcore
#lsmod | grep usbcore
```

Berhasilkah ? Apa error yang diberikan ?

Jika gagal, ini disebabkan usbcore digunakan oleh 5 modul yang lain yaitu, btusb, uhci_hcd, ehci_hcd, ehci_pci dan usbhid. Masalah ini disebut juga sebagai masalah dependensi. Untuk mengatasinya, hapus kelima modul tersebut terlebih dahulu. Jika terjadi error, maka hapus modul yang lainnya terlebih dahulu. Misal, untuk menghapus ehci_hcd, anda harus menghapus ehci_pci terlebih dahulu.

Berikutnya hapus kembali modul usbcore. Berhasilkah ?

```
#rmmod  usbcore
#lsmod | grep usbcore
```

Kemudian masukkan kembali modul tersebu dengan modprobe. Cek kembali keberadaan modul dengan lsmod

```
#modprobe -a  usbcore
#lsmod | grep  usbcore
```

Lakukan hal yang sama untuk kelima modul yang lain yaitu, btusb, uhci_hcd, ehci_hcd, ehci_pci dan usbhid.

**Percobaan 6. Kompilasi modul eksternal dan menambahkannya ke kernel**

Pada percobaan ini, anda akan membuat modul baru, yaitu modul : helloworld1 dan melakukan kompilasi terhadap modul tersebut. Setelah itu, anda akan menambahkan modul tersebut ke kernel Linux anda. Modul Helloworld ini bertugas menuliskan "HelloWorld 1" ke kernel ketika diload dan menuliskan "Goodbye World 1", ketika diunloading dari kernel .

1. Install dulu linux-headers

```
#apt-get install linux-headers-$(uname -r)
```

2. Buat dua file dibawah dan letakkan pada direktori /home/<user>/hello

File hello-1.c

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#define DRIVER_AUTHOR "Peter Jay Salzman <p@dirac.org>"
#define DRIVER_DESC   "A sample driver"

MODULE_LICENSE("GPL");
MODULE_AUTHOR(DRIVER_AUTHOR);
MODULE_DESCRIPTION(DRIVER_DESC);
MODULE_SUPPORTED_DEVICE("testdevice");

static int __init hello_init(void)
{
        printk(KERN_INFO "Hello world-1\n");
        return 0;
}

Static void __exit cleanup_exit (void)
{
        printk(KERN_INFO "Goodbye world-1\n");
}

module_init(hello_init);
module_exit(cleanup_exit);
```

File Makefile

```
obj-m += hello-1.o
all:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

3. Make

```
#cd /home/<user>/hello
#make
```

4. Lihat ke direktory hello. Anda akan mendapatkan beberapa file : hello-1.o, hello-1.ko, hello-1.mod.c, hello-1.mod.o, modules.order , Module.symvers

```
#ls
```

5. Anda telah berhasil mengkompilasi file hello-1.c. Mari kita lihat lebih jauh dengan modinfo. Anda akan melihat filename, depend dan vermagic

```
#modinfo hello-1.ko
```

6. Masukkan modul hello ke kernel

```
#insmod  hello-1.ko
```

7. Cek apakah modul hello sudah berhasil masuk ke kernel lewat /proc/modules. Anda dapat juga melakukannya lewat lsmod.

```
#cat /proc/modules | grep hello*
#lsmod | grep hello*
```

8. Cek juga lewat /var/log/messages.

```
#cat /var/log/messages | grep hello*
```

9. Cek juga lewat /var/log/kern.log. Apakah kalimat Hello World sudah ter-print ?

```
#cat /var/log/kern.log |  grep hello*
```

10. Coba lakukan remove module

```
#rmmod hello-1.ko
```

11. Cek juga lewat /var/log/kern.log. Apakah kalimat Goodbye sudah ter-print ?

```
#cat /var/log/kern.log |  grep Goodbye*
```

12. Reboot system dan cek apakah module hello sudah ter-loading secara otomatis ketika booting ? Tidak ada. Artinya : Module harus diloading dulu kedalam kernel

```
#reboot
#cat /proc/modules | grep hello*
```

## Daftar Pustaka


1. https://www.evoluso.com/my/knowledgebase/article/77/16-commands-to-check-hardware-information-on-linux/
2. https://linuxconfig.org/getting-know-a-hardware-on-your-linux-box
3. http://haifux.org/lectures/86-sil/kernel-modules-drivers/kernel-modules-drivers.html
4. https://www.linux.com/news/udev-introduction-device-management-modern-linux-system
5. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/Reference_Guide/s1-proc-directories.html
6. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/Reference_Guide/s1-proc-sysctl.html
7. http://www.crashcourse.ca/introduction-linux-kernel-programming/lesson-1-building-and-running-new-linux-kernel
8. https://www.quora.com/Whats-the-best-way-to-learn-device-driver-development-on-Linux