

1. KERNEL LINUX & MODUL

Kernel

Kernel adalah bagian dari sistem operasi yang menjembatani akses ke sumber daya sistem. Kernel bertanggung jawab untuk memungkinkan beberapa perangkat lunak untuk secara efektif berbagi perangkat keras dengan mengendalikan akses ke CPU, memori, disk I/O, dan jaringan.

Sistem operasi adalah kernel yang dilengkapi dengan perangkat lunak (aplikasi) yang memungkinkan pengguna untuk dapat melakukan sesuatu (seperti shell, compiler, editor teks, window manager, dll)

Peran utama kernel:

- Mengelola seluruh sumber daya perangkat keras: cpu, memory, I/O
- Menyediakan sebuah set portabel, arsitektur dan API yang independen terhadap hardware untuk memungkinkan aplikasi user-space dan pustaka untuk menggunakan sumber daya perangkat keras.
- Menangani akses bersamaan dan penggunaan sumber daya perangkat keras dari aplikasi yang berbeda.

Modul

Meskipun "monolitik," dalam arti bahwa seluruh kernel berjalan dalam satu ruang alamat, kernel Linux adalah modular, mendukung penyisipan (insertion) dan penghapusan (removal) kode secara dinamis dari dirinya sendiri pada saat runtime. Subrutin terkait, data, entry dan exit point dikelompokkan bersama kedalam sebuah image biner tunggal, sebuah obyek kernel loadable, yang disebut modul.

Dukungan untuk modul-modul memungkinkan sistem untuk hanya memiliki image kernel dasar yang minimal, dengan fitur-fitur opsional dan driver-driver yang disertakan melalui loadable, sebagai obyek yang terpisah.

Modul-modul juga memungkinkan penghapusan dan pemuatan ulang kode kernel, memfasilitasi debugging, dan memungkinkan untuk permintaan pemuatan driver-driver baru dalam menanggapi hot plugging perangkat baru

Manfaat modul:

- Module mempermudah pengembangan driver-driver tanpa reboot: load, test, unload, rebuild, load...
- Berguna untuk menjaga agar ukuran image kernel minimum.
- Berguna mengurangi waktu boot: kita tidak perlu menghabiskan waktu untuk inisialisasi device dan fitur-fitur kernel yang hanya kita butuhkan di waktu akan datang

Lab 1.1. Mengidentifikasi versi kernel

- Buka program terminal, gunakan perintah:
 - `uname -r`
 - `cat /proc/version`
- Lihatlah isi direktori /boot, temukan file dengan nama **vmlinuz-x.x.x**

Lab 1.2. Membangun kernel baru (build kernel) dengan dukungan NTFS

- Login sebagai root
- Atau switch sebagai user root , gunakan perintah : “su - “
- Periksa perlengkapan yang dibutuhkan, yaitu gcc, binutils, dan make.
 - # `rpm -qa | grep ^gcc`
 - # `rpm -qa | grep ^binutils`
 - # `rpm -qa | grep ^make`
- Download kode sumber kernel yang stabil dan terbaru (misal: 3.10.30) dari <http://www.kernel.org>
- Pindah ke direktori /usr/src
 - # `cd /usr/src`
- Ekstrak kode sumber kernel linux yang telah di download, seperti berikut:
 - # `tar -xJvf /<path-lokasi-file-source-kernel>/linux-3.10.30.tar.xz`
- Perhatikan apakah proses ekstrak berhasil , gunakan perintah ls, amati dengan perintah:
 - # `ls -l /usr/src`
- Kemudian masuk kedalam direktori hasil ekstrak.
 - # `cd /usr/src/linux-3.10.30`
- Salinlah file konfigurasi kernel linux saat ini.
 - # `cp /boot/config-x.x.x /usr/src/linux-3.10.30/.config`
- **Catatan:**
Ubah 'x.x.x' dengan versi kernel linux saat ini yang telah Anda identifikasi
- Lakukan konfigurasi kernel dengan mengetikkan perintah
 - # `make menuconfig`
 - agar mendukung file sistem NTFS, dengan memilih opsi File Systems → DOS/FAT/NT Filesystems → NTFS file systems support
- Simpan konfigurasi.
- Lakukan kompilasi (build) kernel, dengan mengetikkan perintah berikut:
 - # `time make`
 - atau jika Anda ingin menjalankan lebih dari satu job, untuk mengoptimalkan processor yang tersedia pada sistem Anda, misalnya sistem Anda terdeteksi memiliki lebih dari satu CPU Core : “cat /proc/cpuinfo” . Maka perintah kompilasi bisa seperti berikut:

```
# time make -j2
```

- Jika telah selesai proses kompilasi kernel, coba perhatikan berapa lama proses kompilasi tersebut ? Dan coba perhatikan didalam direktori source seharusnya akan ada file dengan nama **vmlinux** sebagai hasil dari proses kompilasi kernel
- Selanjutnya lakukan proses terakhir yaitu menginstal kernel baru dan modul modulnya ke direktori direktori yang sesuai dengan mengetikkan perintah berikut:
make modules_install
make install
- Jika berhasil , perhatikan pada direktori **/boot** adakah kernel **vmlinuz-3.10.30** ?
- Jika ada, maka lanjutkan reboot sistem agar menggunakan kernel versi terbaru hasil kompilasi (build)
- Periksa kernel saat ini setelah boot ke versi yang baru, ?

Lab 1.3. Membuat Modul kernel sederhana

Buatlah kode modul sederhana berikut ini, gunakan teks editor vi atau gedit, beri nama file kode modul ini dengan nama **sttnf.c** :

```
#include <linux/module.h>    // included for all kernel modules
#include <linux/kernel.h>    // included for KERN_INFO
#include <linux/init.h>      // included for __init and __exit macros
MODULE_LICENSE("GPL v.2");
MODULE_AUTHOR("Mahasiswa STT NF");
MODULE_DESCRIPTION("Contoh sebuah modul kernel sederhana Halo STT NF");
static int __init sttnf_init(void)
{
    printk(KERN_INFO "[*] STTNF: Halo, Modul sttnf mulai dimuat !.\n");
    return 0;
}
static void __exit sttnf_cleanup(void)
{
    printk(KERN_INFO "[*] STTNF: Halo, Modul sttnf dilepas !.\n");
}
module_init(sttnf_init);
module_exit(sttnf_cleanup);
```

- Selanjutnya , buatlah file dengan nama Makefile didalam direktori yang sama dengan file sttnf.c . Isi dari file Makefile tersebut seperti berikut ini:

```
obj-m := sttnf.o

KERNEL_DIR = /usr/src/kernels/${shell uname -r}

all:
    $(MAKE) -C $(KERNEL_DIR) SUBDIRS=$(PWD) modules
clean:
    rm -f *.o *.ko *.mod *.symvers *.order *.mod.c
```

- Kemudian simpan file Makefile, lanjutkan dengan melakukan kompilasi (build) modul yang

Anda buat, dengan menjalankan perintah berikut (pada direktori yang terdapat file Makefile dan sttnf.c):

```
# make all
```

- Jika proses kompilasi berhasil, lihatlah akan ada file image biner modul **sttnf.ko** pada direktori yang sama.

Lab 1.4. Menggunakan utilitas modul

- Tampilkan modul modul yang saat ini dimuat kedalam kernel linux, gunakan perintah berikut:
lsmod
- Periksalah apakah modul sttnf sudah terpasang atau belum. Gunakan perintah berikut:
lsmod |grep sttnf
- Jika modul sttnf belum terpasang(dimuat) cobalah muat sekarang juga. Gunakan perintah berikut sebagai user root:
insmod sttnf.ko
- Periksalah kembali apakah modul sttnf sudah terpasang atau belum. Gunakan perintah berikut:
#lsmod |grep sttnf
- Jika modul sttnf berhasil dimuat, lihatlah log kernel dengan perintah berikut sebagai user root:
dmesg
- Gunakan perintah modinfo untuk mendapatkan informasi mengenai modul sttnf.ko.
modinfo sttnf.ko
- Lepas (remove) modul sttnf dengan perintah berikut:
rmmod sttnf
- Jika modul sttnf berhasil dilepas, lihatlah log kernel dengan perintah berikut sebagai user root:
dmesg
- Anda dapat juga memeriksa apakah modul sttnf sudah dilepas dengan perintah berikut:
#lsmod |grep sttnf