



PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI
DAN INFORMATIKA

Dr. Ir. Bana Handaga, M.T.
Aris Rakhmadi, S.T., M.T.
Nurul Kholisatul Ulya, S.Kom., M.T.
Ihsan Cahyo Utomo, S.Kom., M.Kom
Novendius Eka Saputra

SISTEM OPERASI

Modul Praktikum



Sistem Operasi

Modul Praktikum

Dr. Ir. Bana Handaga, M.T.

Aris Rakhmadi, S.T.,M.T.

Nurul Kholisatul Ulya, S.Kom.,M.T.

Ihsan Cahyo Utomo, S.Kom.,M.Kom

Novendius Eka Saputra



2019

Sistem Operasi

Modul Praktikum

Penulis:

Dr. Ir. Bana Handaga, M.T.

Aris Rakhmadi, S.T.,M.T.

Nurul Kholisatul Ulya, S.Kom.,M.T.

Ihsan Cahyo Utomo, S.Kom.,M.Kom

Novendius Eka Saputra

Editor:

Ihsan Cahyo Utomo, S.Kom.,M.Kom

Layouter & Cover:

Ali Himawan

ISBN: 978-602-361-237-6

Cetakan 1, September 2019

©2019 Hak cipta pada penulis dilindungi undang-undang

Penerbit

Muhammadiyah University Press

Universitas Muhammadiyah Surakarta

Gedung i Lantai 1

Jl. A Yani Pabelan Tromol Pos 1 Kartasura Surakarta 57102

Jawa Tengah - Indonesia

Telp: (0271) 717417 Eks. 2172

Email: mupress@ums.ac.id

KATA PENGANTAR

Alhamdulillah, puji dan syukur disampaikan ke hadirat Allah swt yang telah memberi kelonggaran dan kesempatan kepada penulis untuk menyusun buku “Modul Praktikum Sistem Operasi” ini. Buku petunjuk disusun berdasarkan kurikulum 2015 yang diberlakukan mulai semester gasal 2016/2017.

Buku petunjuk praktikum ini mengalami beberapa pengembangan sejak versi pertama yang digunakan dalam matakuliah “Sistem Operasi”. Petunjuk praktikum ini mengalami perbaikan di sana-sini terkait kesalahan ketik dan *update* pengetahuan. Pada tahun 2016, ditambahkan dua bab baru yang bersifat opsional untuk praktikum terakhir. Pada tahun 2018 dilakukan penambahan materi tentang remastering sistem operasi.

Bab-bab dalam buku ini dibagi menjadi dua bagian besar. bagian pertama tentang pengenalan sistem operasi windows dan bagian kedua pengenalan tentang sistem operasi linux dan cara membuatnya

Penulis berharap buku ini bermanfaat dan dapat digunakan secara maksimal dalam memahamkan mahasiswa tentang porses pembuatan sistem operasi. Tak lupa pula penulis mengucapkan terima kasih kepada berbagai pihak yang membantu dalam penyelesaian buku ini, mulai dari pimpinan Program Studi Informatika, sejawat dosen terutama yang pengajar paralel matakuliah “Sistem Operasi”, asisten dan staf. Kritik dan saran sangat diharapkan demi penyempuranaan konten maupun layout buku petunjuk praktikum ini.

Semoga Allah swt memberkati kita semua.

Surakarta, Januari 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
Modul 1	
Pengenalan Sistem Pengembangan OS dengan PC	
Simulator ‘Bochs’	1
Tujuan.....	1
Pendahuluan	1
Peralatan	2
Langkah Kerja	2
Tugas	21
Modul 2	
Mengenal Proses Pembuatan ‘DISK BOOT’	23
Tujuan.....	23
Pendahuluan	23
Peralatan	24
Langkah kerja	25
Tugas	33
Modul 3	
Mengenal Cara ‘Debugging’ Program Bootstrap-Loader	35
Tujuan.....	35
Pendahuluan	35
Peralatan	37
Langkah Kerja	37
Tugas	43
Modul 4	
Pengolahan File dan Directory Menggunakan MS-DOS/ Command Prompt di Windows	45

Tujuan.....	45
Pendahuluan	45
Peralatan	49
Langkah kerja.....	49
Tugas	55
Modul 5	
Pengenalan Sistem Operasi Linux	57
Tujuan.....	57
Pendahuluan	57
Petunjuk praktikum	64
Tugas	66
Modul 6	
Instalasi Sistem Operasi Linux, dan Menambah Aplikasi Linux	67
Pendahuluan	67
Petunjuk Praktikum	68
Menambah aplikasi.....	79
Tugas	80
Modul 7	
Managemen User dan Permission Direktori/File pada Linux.....	81
Tujuan.....	81
Pendahuluan	81
Petunjuk praktikum	85
Modul 8	
System Call	91
Tujuan.....	91
Pendahuluan	91
Peralatan.....	93
Langkah kerja.....	94

Modul 9

File System Call	103
Tujuan.....	103
Pendahuluan	103
Peralatan.....	104
Langkah Kerja	105
Hasil	109
Kesimpulan	109

Modul 10

Simulasi Command (Perintah)	111
Tujuan.....	111
Pendahuluan	111
Peralatan.....	112
Langkah Kerja	112
Hasil	117
Kesimpulan	117

Modul 11

Penjadwalan Proses dan Manajemen Memori (OSSim)	119
Tujuan.....	119
Pendahuluan	119
Peralatan.....	120
Langkah Kerja	120

Modul 12

Remastering	121
Pendahuluan	121
Tujuan.....	123
Peralatan	123

Modul 1

Pengenalan Sistem Pengembangan OS dengan PC Simulator ‘Bochs’

Tujuan

Mengenal lingkungan dan tools untuk pengembangan dasar-dasar sistem operasi

Pendahuluan

Modul ini merupakan pengantar yang keberhasilanya akan menentukan kelancaran proses praktikum pada modul-modul berikutnya. Pada praktikum ini akan dikenalkan beberapa hal penting yang terkait dengan PC dan dasar sistem operasi, serta beberapa tools alternatif pengembangan sistem operasi.

Beberapa hal penting di samping pengetahuan tentang sistem operasi yang diperlukan dalam pengembangan sistem operasi antara lain adalah menguasai bahasa pemrogramman assembly dan C dengan baik, kemudian memahami cara kerja PC dan dapat menggunakan PC dengan lancar. Selain itu juga diperlukan pengetahuan sistem angka digital yang memadai, seperti representasi angka biner dan hexa, konversi angka desimal ke biner atau hexa. Hal-hal di atas merupakan dasar yang perlu dipahami dengan baik sebelum mengembangkan sebuah sistem operasi.

Pada modul ini akan belajar cara menggunakan beberapa program bantu untuk pengembangan sistem operasi, dan belajar lebih dekat tentang proses ‘boot’ pada PC dan beberapa

hal yang terkait dengan hal tersebut. Praktikum akan dimulai dengan cara membuka direktori kerja, melihat file-file dalam direktori kerja, sekilas tentang ‘Makefile’, memahami ‘boot disk’, melihat data dalam bootsector

Peralatan

1. PC dengan sistem operasi Windows Xp
2. Program Simulator Boschs
3. Kompiler bahasa assembly ‘nasm’
4. Kompiler bahasa C.
5. program bantu seperti ‘make’, ‘debug’, ‘dd’, dan ‘tdump’
6. notepad

Semua proses praktikum dilakukan dalam mode kerja ‘Command Prompt’, hal ini dimaksudkan agar praktikan dapat memahami prosesnya secara lebih baik.

Langkah Kerja

Menuju ke direktori kerja.

- a. Jalankan program command prompt atau cmd.
- b. Masuk ke direktori kerja ‘C:\OS’, dengan perintah ‘cd os’ <ENTER>.
- c. Masukan perintah dir, untuk melihat isi direktori di dalam folder tersebut. Akan muncul seperti di tampilkan pada **gambar**

```
C:\OS>dir
Volume in drive C is S3A4070D001
Volume Serial Number is A8E5-3D6F

Directory of C:\OS

14/12/2008  03:48    <DIR> .
14/12/2008  03:48    <DIR> ..
14/12/2008  03:43    <DIR> Bochs-2.3.5
14/12/2008  03:44    <DIR> Dev-CPP
14/12/2008  03:48    <DIR> Lab
14/12/2008  03:47      57 Setpath.bat
                           1 File(s)   57 bytes
                           5 Dir(s)  15.977.971.712 bytes free

C:\OS>
```

Gambar 1.1 Struktur direktori kerja

- d. Jalankan file setpath, untuk menjalankannya ketik 'setpath' tekan <ENTER>

File 'setpath.bat' digunakan untuk mengatur lingkungan kerja ('path') selama anda melakukan praktikum, anda harus menjalankan progam ini sebelum memulai setiap sesi praktikum anda, untuk menjalakannya ketik 'setpath' tekan <ENTER>. Perhatikan teks yang muncul di layar, seperti ditampilkan pada Gambar 1.1.

Untuk melihat script yang terdapat di dalam file 'setpath.bat' dapat digunakan perintah 'type setpath.bat', cobalah. 'PATH' dapat bersisi banyak daftar lokasi, di antara item lokasi di batasi dengan karakter ';' (titik koma). Variabel 'PATH' ini akan digunakan oleh windows untuk mencari lokasi file yang dipanggil oleh 'user'. Urutan pencarian dimulai dari daftar lokasi yang paling awal.



Gambar 1.2 Pengaturan lingkungan kerja (path).

Untuk pengaturan ‘path’ seperti ditampilkan pada gambar 1.2 ., maka ketika user memasukan perintah (seperti memangil program) pertama kali yang dilakukan Windows adalah mencari file program di dalam direktori kerja saat itu, jika di sana tidak ditemukan file yang dimaksud, selanjutnya Windows akan meneruskan pencarian file di lokasi yang terdaftar pada variabel ‘PATH’ dimulai dari lokasi ‘C:\OS\Dev-Cpp\bin’ jika file belum ditemukan pencarian dilanjutkan ke lokasi ‘C:\Windows’, pencarian terus dilakukan sampai ke lokasi terakhir ‘C:\Windows\system32’. Jika sampai lokasi terakhir file tidak juga ditemukan maka windows akan manmpulkan informasi bahwa file yang dimaksud tidak ada di dalam sistem. Sebaliknya jika file sudah ditemukan maka windows akan menghentikan pencarian di lokasi terakhir ditemukannya file yang dimaksud.

Selain file pengatur lingkungan kerja, pada direktori kerja anda juga terdapat tiga folder yaitu folder ‘Bochs-2.3.5’, merupakan lokasi dari program ‘PC-Simulator’ yang akan digunakan untuk membuat PC-virtual, untuk menjalankan program ‘bootstrap loader’ dan program-program yang anda buat selama menjalankan praktikum.

Folder ‘Dev-Cpp’ berisi file program kompiler ‘nasm’ dan ‘gcc’ yang dapat digunakan untuk mengkompilasi program yang ditulis dalam bahasa c, c++ maupun bahasa assembly, selain itu pada folder ini juga terdapat beberapa program bantu, seperti ‘dd.exe’ untuk menyalin byte data dari satu file ke file yang lainnya. File-file kerja selama melakukan praktikum diletakan di bawah folder ‘Lab’.

Melihat isi direktori kerja

Untuk masuk didirektori kerja untuk modul ini pertama adalah

- a. Jalankan command prompt
- b. Masuk ke direktori kerja pada ‘C :\OS\LAB\LAB1’.
- c. Cobalah untuk membuka file tersebut dengan perintah berikut, dari ‘COMMAND PROMPT’, ketikkan ‘Notepad boot.asm’ dan tekan <ENTER>. Saat ini sebaiknya anda tidak melakukan modifikasi terhadap program tersebut. Tutuplah kembali program ‘notepad’-nya. Source code prototype program bootloader disimpan dalam file ‘boot.asm’ sedangkan untuk source code prototype program kernel disimpan dalam file ‘kernel.asm’ listing lengkap keduanya bisa dibaca di lampiran. Sebaiknya anda memahami jalannya kedua program sebelum melakukan praktikum. Kedua program ditulis dalam bahasa assembly, anda dapat membukanya dengan menggunakan program aplikasi teks-editor sembarang, termasuk ‘Notepad.exe’ milik windows juga dapat digunakan.
- d. Selain kedua file source code ada sebuah file ‘image floppy’ dengan nama file ‘floppya.img’ file ini yang akan digunakan untuk menyimpan hasil kompilasi kedua source code, kemudian digunakan sebagai ‘boot disk’ pada PC-simulator ‘Bochs’. Sebagai kompiler digunakan program ‘nasm.exe’ yang terletak dalam subdirektori ‘C:\OS\Dev-Cpp\bin’ dan pada direktori tersebut juga ada program ‘dd.exe’ untuk memindahkan byte data dari satu file ke file yang lain, program ‘make.exe’ untuk mempercepat proses kompilasi. Pada saat ini kita akan menggunakan ketiga program tersebut yang dikemas dalam script ‘Makefile’ sehingga proses kompilasi menjadi lebih cepat, cukup dengan memanggil ‘make’. Script ‘makefile’ berupa file dengan format teks yang

dapat anda sunting dengan menggunakan program aplikasi teks-editor.

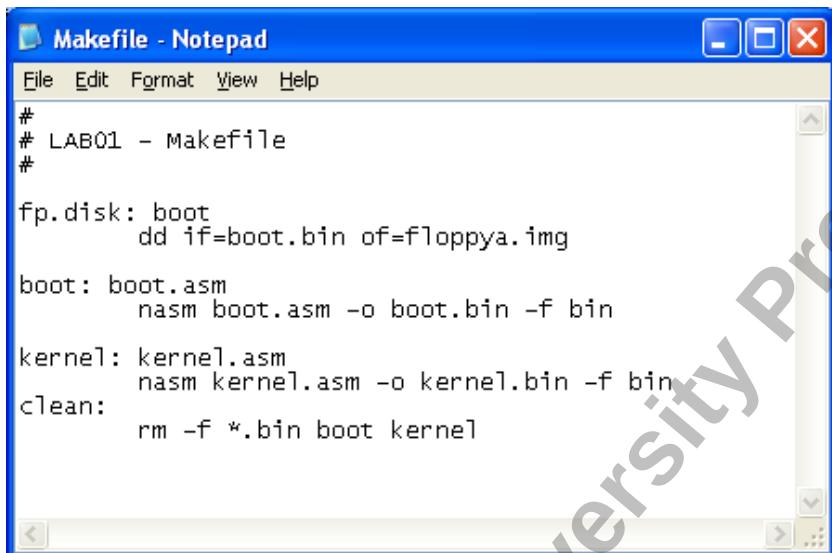
Sekilas tentang Makefile

‘Makefile’ adalah sebuah file teks yang berisi kumpulan perintah ‘Command Prompt’ mirip seperti BAT file, namun tidak dapat dijalankan secara langsung. File ini digunakan oleh program aplikasi ‘make.exe’ untuk mengendalikan proses kompilasi, lingking dan lain-lain, program ‘make.exe’ akan lebih terasa manfaatnya ketika source code program yang dibuat terdiri dari banyak file, proses kompilasi dan yang lain dapat dilakukan secara otomatis, sebagai kontrolnya adalah script yang terdapat pada ‘Makefile’. Cara ini banyak digunakan oleh para pengguna sistem operasi ‘linux’.

Seakarang bukalah file ‘Makefile’, dari ‘Command Prompt’ untuk mengetahui script makefile dengan cara: bukalah direktori kerja anda ‘C:\OS\LAB\LAB1’ selanjutnya ketik ‘Notepad M’ tekan tombol ‘TAB’ sehingga muncul ‘Notepad Makefile’ dan tekan <ENTER>. Jika langkah anda benar akan ditampilkan windows ‘Notepad’ dengan file ‘Makefile’ yang siap disunting seperti ditampilkan pada gambar 1.3

Tanda karakter '#' digunakan untuk menyisipkan baris komentar atau keterangan, karakter ini harus diletakkan pada kolom pertama, anda dapat menambahkan keterangan sebanyak yang anda inginkan.

Label yang berakhir dengan ‘:’, seperti ‘fp.disk:’ dan ‘boot:’ adalah merupakan kontrol proses (mirip menu) yang dapat dipilih saat menjalankan program ‘make.exe’. Anda dapat mengatur proses dengan menggunakan menu ini.



The screenshot shows a Windows-style Notepad window titled 'Makefile - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The main content area contains the following Makefile script:

```
#  
# LAB01 - Makefile  
  
fp.disk: boot  
        dd if=boot.bin of=floppya.img  
  
boot: boot.asm  
        nasm boot.asm -o boot.bin -f bin  
  
kernel: kernel.asm  
        nasm kernel.asm -o kernel.bin -f bin  
clean:  
        rm -f *.bin boot kernel
```

Gambar 1.3 Menyunting file 'Makefile'

Perhatikan baris menu ‘fp.disk: boot’, baris ini memerintahkan pada komputer untuk mengerjakan baris perintah di bawah menu tersebut, tetapi setelah baris perintah yang terdapat di bawah menu ‘boot:’ dikerjakan. Jadi jika dilihat menu ‘fp.disk’ maka pertama kali ‘make.exe’ akan mengerjakan baris perintah yang terdapat di bawah label menu ‘boot:’ yaitu ‘nasm boot.asm -o boot.bin -f bin’, baris ini merupakan perintah untuk melakukan proses kompilasi program assembly dengan menggunakan program kompiler ‘nasm’. Sebagai source code adalah file ‘boot.asm’ kemudian hasil kompilasi akan dituliskan dalam output file (‘-o’) dengan nama ‘boot.bin’. sedangkan format (‘-f’) file output adalah dipilih dengan format ‘bin’. Program hasil kompilasi ini (‘boot.bin’) tidak dapat dijalankan secara langsung dari ‘Command Prompt’ karena tidak memiliki format ‘header’ yang dapat dimengerti oleh OS. Selanjutnya ‘make.exe’ akan melanjutkan pekerjaannya untuk memproses perintah yang terletak di bawah menu ‘fp.disk:boot’ yaitu

'dd if=boot.bin of=floppya.img' ini adalah perintah untuk menyalin byte data dari file 'boot.ini' de dalam file 'floppya.img' pada lokasi sektor nomor '0', sebanyak '512' byte. Program aplikasi 'dd.exe' merupakan program bawaan linux yang di kemas ulang sehingga dapat dijalankan di windows, program bantu ini juga bersifat open source, banyak variasi yang dapat dilakukan oleh 'dd' tetapi tidak di bahas dalam modul ini anda dapat mencarinya di internet lewat yahoo atau google dan masukkan kata kunci 'dd command', cara ini juga berlaku untuk tema program yang lainnya.

File 'Makefile' akan di baca secara otomatis saat program 'make' dijalankan, menu dipilih saat menjalankan program make misalnya 'make fp.disk' berarti kita akan melakukan kompilasi hadap source code program 'boot.asm', sebagai outputnya file 'boot.bin' dan isinya disalin ke dalam bootsector file image floppy 'floppya.img'.

Sekarang bukalah 'Command Prompt' dan buka direktori kerja 'LAB1' selanjutnya ketik 'make fp.disk', jika tidak ada kesalahan akan ditampilkan informasi seperti pada gambar .

```
C:\OS\Lab\LAB1>make fp.disk
make boot.asm -o bootsector.iso
dd if=boot.bin of=floppya.iso
rawwrite dd for windows version 0.5.
written by John Newbiggin <jn@it.swin.edu.au>
This program is covered by the GPL. See
1+0 records in
1+0 records out
C:\OS\Lab\LAB1>
```

Dua proses yang di kerjakan oleh 'Make'. Baris di bawahnya merupakan hasil dari perintah 'dd''

Gambar 1.4 Menggunakan perintah Make

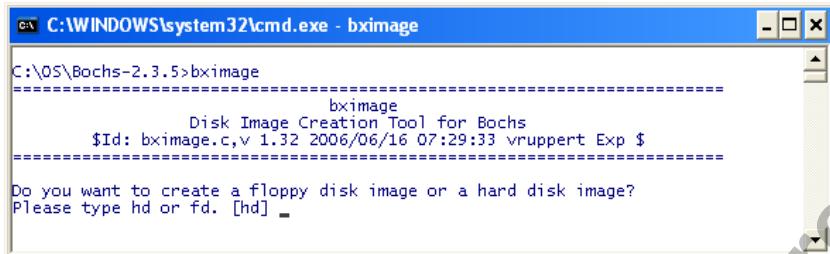
Periksa hasil kompilasi dengan memasukan perintah 'dir', sekarang pada direktori kerja terdapat tambahan file baru, yaitu 'boot.bin' dan isinya sudah disalin kedalam bootsector 'floppya.img'. Anda juga dapat melakukannya secara manual dari 'Command Prompt', pertama jalankan

perintah baris satu yaitu ‘nasm boot.asm -o boot.bin -f bin’, jika tidak ada berita kesalahan lanjutkan dengan perintah ‘dd if=boot.bin of=floppya.img’. Selanjutnya silahkan di pelajari menu-menu lain dalam ‘makefile’ proses ini akan dilakukan berulang kali dalam praktikum-praktikum sealnjutnya. Perintah ‘rm ...’ pada baris paling bawah dalam ‘Makefile’ di atas, digunakan untuk menghapus file-file yang sudah tidak terpakai. Daftar nama file yang di hapus diletakkan di belakang tanda (flag) ‘-f’ , kemudian diantara nama-nama file dipisahkan dengan ‘spasi’, anda dapat mencobanya dengan perintah ‘make clean’, periksa isi file dalam direktori kerja, sekarang ‘boot.bin’ sudah terhapus.

Mengenal ‘BOOT DISK’

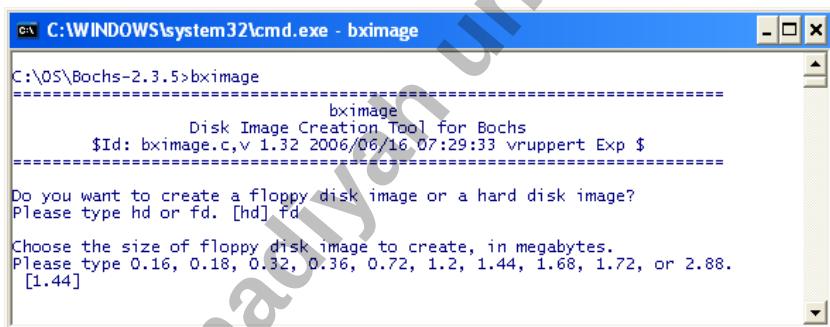
File ‘floppya.img’ adalah file image (bukan gambar), sebuah file yang isinya diformat seperti format floppy disk kapasitas 1.44M yang akan digunakan sebagai ‘bootdisk’ pada PC-Simulator, anda juga dapat memindahkan isi file image ini ke dalam floppy disk sebenarnya dengan menggunakan program bantu ‘rawrite’ atau yang lainnya. File image ini dapat di buat dengan menggunakan progam bantu bawaan ‘Bochs’ yaitu ‘bximage’ atau menggunakan program aplikasi ‘Virtual PC’ milik Windows (free juga) atau dengan program ‘Magiciso’. Sekarang kita coba membuat file image floppy baru dengan menggunakan program aplikasi ‘bximage.exe’, lakukan urutan perintah berikut.

- a. Hapuslah file ‘floppya.img’ jika sudah ada pada direktori kerja anda, dari ‘Command Prompt’ (lakukan dari direktori kerja) ketik ‘**del floppya.img /P**’ lanjutkan dengan tekan ‘Y’ dan <ENTER>. Pastikan bahwa file sudah benar- benar terhapus dengan perintah ‘dir’. Selanjutnya panggil ‘**bximage**’ sehingga ditampilkan window seperti pada gambar 1.5.



Gambar 1.5 Membuat file image floppy tahap-1

- b. Ada dua pilihan file image yaitu [hd] untuk membuat harddisk image atau [fd] untuk membuat floppy image. Kita akan membuat floppy image (karena ukuran filenya lebih kecil), selanjutnya ketikan '**fd**' dan tekan <ENTER> sehingga muncul tahapan berikutnya seperti pada gambar 1.6..



Gambar 1.6 Membuat file image floppy tahap-2

```
C:\OS\Bochs-2.3.5>bximage
=====
          bximage
          Disk Image Creation Tool for Bochs
          $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44]
I will create a floppy image with
cyl=80
heads=2
sectors per track=18
total sectors=2880
total bytes=1474560

What should I name the image?
[a.img] -
```

Gambar 1.7 Membuat file image floppy tahap-3.

```
C:\OS\Bochs-2.3.5>bximage
=====
          bximage
          Disk Image Creation Tool for Bochs
          $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44]
I will create a floppy image with
cyl=80
heads=2
sectors per track=18
total sectors=2880
total bytes=1474560

What should I name the image?
[a.img] floppya.img

Writing: [] Done.

I wrote 1474560 bytes to floppya.img.

The following line should appear in your bochsrc:
floppya: image="floppya.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)

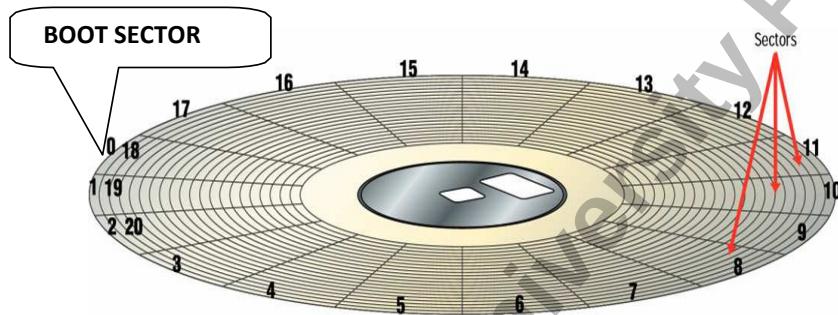
Press any key to continue
```

Gambar 1.8 Membuat file image floppy tahap-akhir.

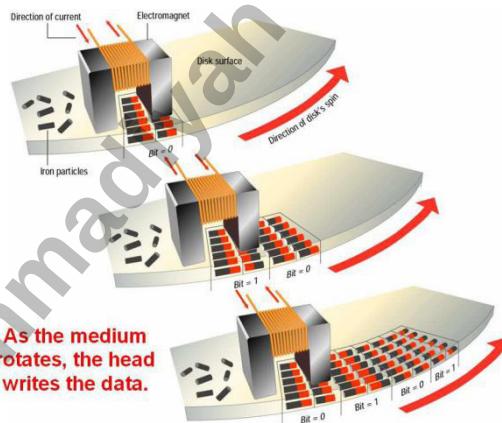
- c. Ada beberapa tipe (kapasitas) yang ditawarkan, pilih tipe yang paling banyak digunakan saat ini yaitu tipe floppy dengan kapasitas ‘1.44MB’, ditunjukan oleh angka

[1.44]. Selanjutnya tekan <ENTER>, untuk memilih tipe lain, masukan angka yang menunjukkan kapasitas yang anda inginkan (lihat gambar di atas) dan <ENTER>. Pada windows akan di tampilkan berita seperti terlihat pada gambar 1.8. .

Formatted Disk



Gambar 1.9 Format floppy disk FAT12



Gambar 1.10 Teknik penyimpanan data pada piringan DISK MAGNETIK.

- d. Terakhir anda diminta untuk memberikan nama file, ketikan 'floppya.img' dan <ENTER>, untuk saat ini jangan memberikan nama yang lain karena setting

konfigurasi pada ‘Bochs’ sudah diatur menggunakan nama tersebut, jadi pastikan bahwa nama file imagnya adalah ‘floppya.img’, pastikan keberadaan file image tersebut dengan perintah ‘dir’, sudah adakah? Jika belum, anda harus mengulangi proses dari awal dengan memanggil program ‘bximage’.

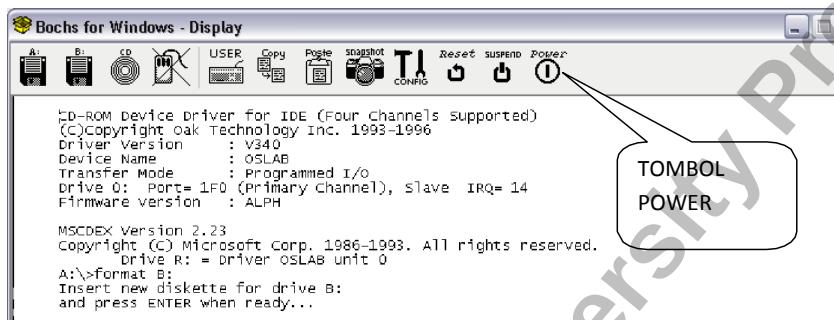
File image floppy ini memiliki tipe Format FAT12, yang memiliki 80 cylinder, 2 head (atas dan bawah), 18 sektor tiap track, jumlah sektor 2880, jumlah byte data tiap sektor 512 byte sehingga total byte $2880 \times 512 = 1.474.560$ byte (1,44M). Format floppy secara fisik dapat dilihat pada gambar 1.0, periksa dengan data di atas, samakah? Sistem yang serupa juga digunakan pada harddisk tetapi dengan variasi jumlah cylinder, head dan sector per track (disingkat CHS) yang berbeda. Cylinder adalah jumlah track (garis melingkar) pada setiap permukaan piringan, jumlah permukaan piringan ada 2 (atas dan bawah, 2 head) sehingga pada floppy 1.44M setiap permukaan terdapat 40 cylinder. Apakah anda dapat melihat track pada floppy yang sebenarnya, tanpa alat bantu? Bayangkan ukuran track yang terdapat pada harddisk dengan kapasitas 300G....dapatkah anda menghitung ukuran fisik (ketebalan garis) tiap track?

Perlu diingat bahwa File image floppy ‘floppya.img’ berada dalam direktori Bochs-2.3.5. Dan selanjutnya yang berkenaan dengan file ini kita harus merujuknya ke lokasi yang tepat.

File image floppy ‘floppya.img’ yang dibuat dengan langkah di atas belum dapat digunakan, karena belum di ‘FORMAT’, seperti pada floppy sebenarnya jika belum di format floppy tidak dapat dibaca. Langkah -langkah Untuk memformat ‘floppya.img’ adalah:

- a. jalankan PC-Simulator dari ‘Command Prompt’ dengan perintah ‘**DosFp**’,

- b. pada konfigurasi PC-Simulator file ‘floppya.img’ terpasang pada ‘drive B:’. Selanjutnya dari prompt ‘A:>’ ketikan ‘Format B:’<ENTER> [2x], seperti ditampilkan pada gambar 1.11.



Gambar 1.11 Bochs PC-Simulator

- c. Tutup kembali PC-Simulator dengan klik pada tombol power, di bagian menu atas-kanan. Sekarang ‘floppya.img’ sudah terformat dan dapat digunakan untuk menyimpan data, namun belum dapat digunakan untuk ‘booting’, karena pada ‘bootsector’ belum diisi program ‘bootloader’. Untuk lebih detail mengenai pembahasan ini ada dalam Modul kedua.

Melihat data dalam boot sector

Untuk membuat sistem operasi (OS – Operating system) salah satu tahap terpenting adalah menyusun program bootstrap-loader (akan dipraktikkan mulai modul praktikum kedua). Ukuran program ini tidak besar, hanya 512 byte dan tersimpan dalam lokasi sektor 0 (track 0 dan cylender 0) disebut BOOT-SECTOR, penomoran dimulai dari ‘0’. Seperti terlihat pada tabel 1.1. Data pada lokasi ini hanya dibaca oleh PC pada saat proses ‘booting’ setelah itu PC tidak akan pernah membaca lagi, sampai Power OFF.

Tabel 1.1 Struktur data BOOTSECTOR

Offset (Hex)	Data
0000-0002	Jump ke lokasi Boot Code
0003-003D	Parameter DISK
003E-01FD	Boot Code (program boot)
01FE-01FF	Signature (ID bootable)

Struktur data dalam Bootsector seperti ditunjukan pada Tiga byte pertama harus berupa kode program untuk meloncat ('jump') ke posisi 0x003E (hex). Karena data pertama bootsector akan diletakan pada alamat 0x7C00 pada memori kerja komputer (RAM) (simbol '**0x**' adalah simbol untuk menunjukkan format angka hex, banyak digunakan dalam bahasa C atau Assembly, jadi $0x7C00 = 7C00$ hexadecimal), maka instruksi pertama adalah 'jump 0x7C3E' dimana pada area tersebut tersimpan program bootstrap-loader. Lokasi data offset 0x0003-0x003D sebanyak 59 byte digunakan untuk menerangkan format disk, jumlah sektor per track, jumlah FAT (*Field Allocation Table*) kode penunjuk urutan lokasi cluster (sektor) data dari sebuah file dan sebagainya (selanjutnya dapat anda lihat pada source code 'boot.asm' bagian atas). 'Boot code' atau inti dari program 'bootstrap loader' diletakan pada lokasi byte mulai nomor 0x003E sampai dengan 0x01FD sebanyak 448 byte.

Dua byte terakhir berupa byte '**Signature**' atau ID bahwa sebuah disk bersifat 'Bootable', yaitu data '**0x55**' pada lokasi offset 510, sedangkan data '**0xAA**' terletak pada lokasi offset 511, nomor lokasi data dimulai dari '0' jadi total lokasi data dari 0 s/d 511 ada 512 byte. BIOS akan mengindifikasi byte 'Signature' untuk menentukan apakah sebuah disk bersifat 'Bootable' atau tidak. 0x55AA berarti disk bersifat '**bootable**'.

Untuk memeriksa data bootsector pada file image floppy dapat digunakan program ‘debug’ yang terdapat pada window dibantu program ‘dd.exe’, lakukan urutan perintah berikut ini. Copy 512 byte data bootsektor ke dalam sebuah file terpisah, caranya dari ‘Command Prompt’ ketik ‘dd if=floppya.img of=boots.bin count=1’ maksud dari perintah ini adalah menyalin byte data dari file ‘floppya.img’ ke dalam file ‘boots.bin’ sebanyak satu sektor mulai dari sektor 0. Hasilnya akan tersimpan dalam file bernama ‘boots.bin’ dengan ukuran data sebanyak 512 byte, ini merupakan data dalam bootsector pada file image ‘floppya.img’. Untuk melihat isinya gunakan program ‘debug’ dengan cara sebagai berikut.

Jalankan ‘Command Prompt’ dan ketik ‘debug boots.bin’ dan <ENTER>, tampilan windows seperti pada gambar 1.12.. (Perlu diketahui untuk mengakhiri debug ketikkan ‘quit’ kemudian enter)



Gambar 1.12 Tampilan windows saat memulai program ‘debug’

Maksud perintah ini adalah memindah data dalam file ‘boots.bin’ ke dalam memori kerja ‘debug’ mulai dari alamat ‘0000:0100’. Debug hanya dapat bekerja dengan ukuran data/file maksimum sebesar 64KB, jadi dengan debug kita tidak bisa membuka file image floppy yang berukuran 1,4MB. file ‘boots.bin’ berukuran 512 B (atau 0x01FF) dan oleh debug data ini akan diletakkan di memori mulai dari alamat CS:0100 sampai dengan alamat CS:02FF. Simbol CS merupakan simbol register ‘CODE SEGMENT’, nilai CS tergantung pada kondisi memori komputer saat itu.

Untuk melihat perintah-perintah yang disediakan dalam ‘debug’ ketikan karakter ‘?’ Untuk melihat isi data file ‘boots.bin’ ketikkan perintah ‘D CS:0100 02FF’ dan tekan <ENTER>. Maksud perintah ini adalah manampilkan isi memori mulai CS:0100 sebanyak 512 byte dalam bentuk hex, seperti terlihat pada gambar 1.14.. Angka pada kolom paling kiri menunjukkan lokasi alamat memori yang digunakan untuk menampung data ‘boots.bin’. Register CS menunjuk segmen alamat ‘13EA’, 16 kolom angka di sampingnya adalah data ‘boots.bin’ yang ditampilkan dalam format hex dikelompokan per byte, 2 angka hexa = 1 byte. Kelompok karakter pada kolom paling kanan adalah data ‘boots.bin’ yang ditampilkan sebagai ‘karakter’, karena data ini merupakan program maka karakter yang terlihat terkesan tidak beraturan, bahkan ada data yang tidak dapat ditampilkan sebagai karakter sehingga hanya muncul tanda ‘...’. Dapatkah anda mengidentifikasi ‘boot signature’ 0x55AA, tunjukkan posisi alamatnya..!

```
C:\OS\Lab1>debug boots.bin
-D cs:0100 02FF
13EA:0100  EB 3C 90 4D 53 57 49 4E-34 2E 31 00 02 01 01 00 .<.MSWIN4.1.....
13EA:0110  02 E0 00 40 0B F0 09 00-12 00 02 00 00 00 00 00 ...@.....
13EA:0120  00 00 00 00 00 00 29 EB-0C 6E 45 42 20 20 20 20 .....).n'FB
13EA:0130  20 20 20 20 20 20 46 41-54 31 32 20 20 20 33 C9   FAT12 3.
13EA:0140  8E D1 BC FC 7B 16 07 BD-78 00 C5 76 00 1E 56 16 ...,{....x..v..V.
13EA:0150  55 BF 22 05 89 7E 00 89-4E 02 B1 0B FC F3 A4 06 U..~..N.....
13EA:0160  1F B0 00 7C C6 45 FE OF-38 4E 24 7D 20 8B C1 99 ...I.E..8N$}...
13EA:0170  E8 7E 01 83 EB JA 66 A1-1C 7C 66 38 07 8A 57 FC ~...:f..!F;..w.
13EA:0180  75 06 80 CA 02 88 56 02-80 C3 10 73 ED 33 C9 FE u....V...s.3.
13EA:0190  06 D8 7D 8A 46 10 98 F7-66 16 03 46 1C 13 56 1E ..}.F...f..F..V.
13EA:01A0  03 46 0E 13 D1 8B 76 11-60 89 46 FC 89 56 FE B8 .F...V..F..V..
13EA:01B0  20 00 F7 E6 8B 5E 0B 03-C3 48 E7 F3 01 46 FC 11 ....A..H..F..V.
13EA:01C0  4E FE 61 BF 00 07 EB 28-01 72 3E 38 2D 74 17 60 N.a....(r>8-t..
13EA:01D0  B1 0E BE D8 7D F3 A6 61-74 3D 4E 74 09 83 C7 20 .....}..at=NT..
13EA:01E0  3B FB 72 E7 EB DD FE 0E-D8 7D 7B A7 BE 7F 7D AC ;.r.....}{...}.
13EA:01F0  98 03 F0 AC 98 40 74 OC-48 74 13 B4 0E BB 07 00 .....@.Ht. .....
13EA:0200  CD 10 EB EF BE 82 7D EB-E6 BE 80 7D EB E1 CD 16 .....}..}....
13EA:0210  5E 1E 66 8F 04 CD 19 BE-81 7D 8B 7D 1A 80 45 FE ^..F.....}..}..E.
13EA:0220  8A 4E 0D F7 E1 03 46 FC-13 56 FE B1 04 E8 C2 00 .N....F..V....
13EA:0230  72 D7 EA 00 02 70 00 52-50 06 53 6A 01 DA 10 91 r....p.RP.Sj{j..
13EA:0240  8B 46 18 A2 26 05 96 92-33 D2 F7 F6 91 F7 F6 42 .F..&...3.....
13EA:0250  87 CA F7 76 1A 8A F2 8A-E8 C0 CC 02 0A CC B8 01 ...V.....
13EA:0260  02 80 7E 02 0E 75 04 B4-42 8B F4 8A 56 24 CD 13 ..~..u..B..V$..
13EA:0270  61 61 72 0A 40 75 01 42-03 5E 0B 49 75 77 C3 03 aar:@.B.A.Iu..
13EA:0280  18 05 27 0D 0A 49 6E 76-61 6C 69 64 20 73 79 73 ...Invalid sys
13EA:0290  74 65 6D 20 64 69 73 6B-FF 0D 0A 44 69 73 6B 20 tem disk...Disk
13EA:02A0  49 2F 4F 20 65 72 72 6F-72 FF 0D 0A 52 65 70 6C I/O error...Repl
13EA:02B0  61 63 65 20 74 68 65 20-64 69 73 6B 2C 20 61 6E ace the disk, an
13EA:02C0  64 20 74 68 65 6E 20 70-72 65 73 73 20 61 6E 79 d then press any
13EA:02D0  20 68 65 79 0D 0A 00 00-49 4F 20 20 20 20 20 20 key...IO
13EA:02E0  53 59 53 4D 53 44 4F 53-20 20 20 53 59 53 7F 01 SYSMSDOS  SYS..
13EA:02F0  00 41 BB 00 07 60 66 6A-00 E9 3B FF 00 00 55 AA .A...`fj;...U..
```

Gambar 1.13 data dalam bootsector file floppya.img

Cara lain dapat digunakan program ‘tdump.exe’ milik borland, dari ‘Command Prompt’ ketikan ‘**tdump boots.bin**’ <ENTER> deretan data yang serupa dengan gambar 11 akan ditampilkan, seperti ditampilkan pada gambar 1.14. Perbedaan terletak pada kolom paling kiri, dengan ‘tdump’ kolom paling kiri menunjukkan posisi relatif setiap byte data dimulai dari posisi ‘000000’.

```

C:\OS\Lab\LAB1>tdump boots.bin
Turbo Dump Version 5.0.16.12 Copyright (c) 1988, 2000 Inprise Corporation
Display of File BOOTS.BIN

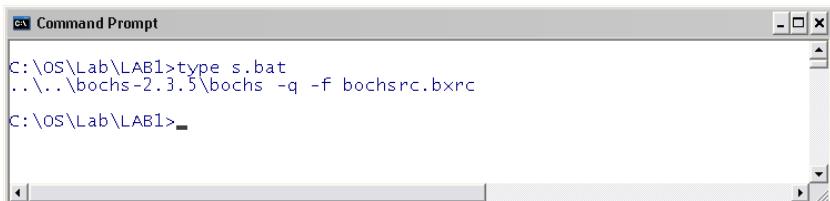
000000: EB 3C 90 4D 53 57 49 4E 34 2E 31 00 02 01 01 00 .<.MSWIN4.1.....
000010: 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 .@....;
000020: 00 00 00 00 00 29 EB 0C 6E 27 46 42 20 20 20 .).n;FB
000030: 20 20 20 20 20 20 46 41 54 31 32 20 20 20 33 C9 .FAT12 3.
000040: 8E D1 BC FC 7B 16 07 BD 78 00 C5 76 00 1E 56 16 .x..v..v.
000050: 55 BF 22 05 89 7E 00 89 4E 02 B1 0B FC F3 A4 06 U."~.N.....
000060: 1F BD 00 7C C6 45 FE 0F 38 4E 24 7D 20 8B C1 99 .|E..8N$}...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC .:...:f;..|w.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u..V..s.3..
000090: 06 D8 70 8A 46 10 98 F7 66 16 03 46 1C 13 56 1e .}.F..F.F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F..v..F..V..
0000B0: 20 00 F7 E6 8B 5E 0B 03 C3 48 F7 F3 01 46 FC 11 .A..H..F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a...(.r>8-t..
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 .).at=nt.|.
0000E0: 3B FB 72 E7 EB DD FE 0E D8 7D 7B A7 BE 7F 7D AC ;r.....}{.}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E BB 07 00 .....@t.Ht. .....
000100: CD 10 EB EF BE 82 7D EB E6 B8 7D EB E1 CD 16 .}.}.}.
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 88 7D 1A 8D 45 FE A.F.....}.E..
000120: 8A 4E 00 D7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N..F..V. .....
000130: 72 D7 EA 00 27 70 00 52 50 06 53 6A 01 6A 10 91 r...p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F.&..3....B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC B8 01 ...v. .....
000160: 02 80 7E 02 0E 75 04 B4 42 8E F4 8A 56 24 CD 13 ..~.u.B..V$..
000170: 61 61 72 0A 40 75 01 42 03 0E 49 75 77 C3 03 aar.@U.B.^,Iuw..
000180: 18 01 27 0D 49 6E 76 61 6C 65 64 20 73 79 73 ...'.Invalid sys
000190: 74 65 60 20 64 69 73 68 FF 0D 0A 44 69 73 6B 20 tem disk..Disk
0001A0: 49 2F 4F 20 65 72 72 6F 72 FF 0D 0A 52 65 70 6C I/O error...Repl
0001B0: 61 63 65 20 74 68 65 20 64 69 73 6B 2C 20 61 6E ace the disk, an
0001C0: 64 20 74 68 65 6E 20 70 72 65 73 73 20 61 6E 79 d them press any
0001D0: 20 68 65 79 0D 0A 00 00 49 4F 20 20 20 20 20 key....IO
0001E0: 53 59 53 4D 53 44 4F 53 20 20 20 53 59 53 7F 01 SYMSDOS SYS..
0001F0: 00 41 BB 00 07 60 66 6A 00 E9 3B FF 00 00 55 AA .A...`fj...;...u.

```

Gambar 1.14 Melihat data bootsector dengan program ‘tdump.exe’

‘Boot’ PC-Simulator dengan file image ‘floppya.img’

Salah satu file dalam direktori kerja adalah file ‘s.bat’, berisi dua baris perintah untuk memanggil PC-Simulator ‘Bochs’. Lihat isi file ‘s.bat’ dengan perintah ‘**type s.bat**’ <ENTER>, hasilnya terlihat seperti pada .



```
C:\OS\Lab\LAB1>type s.bat
..\..\bochs-2.3.5\bochs -q -f bochssrc.bxrc
C:\OS\Lab\LAB1>-
```

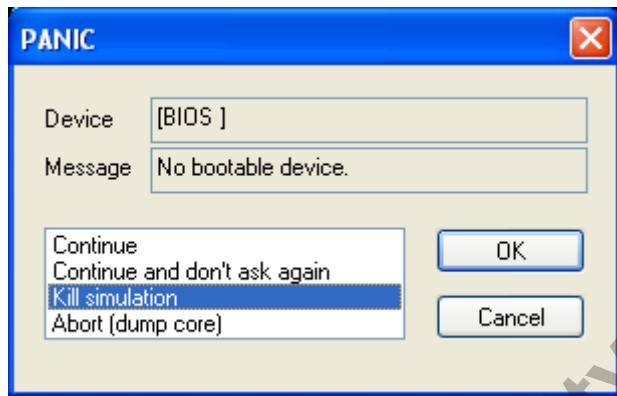
Gambar 1.15 Isi file ‘s.bat’

Sebuah baris perintah untuk memanggil ‘Bochs’ (PC-Simulator), dengan file konfigurasi ‘bochssrc.bxrc’. File konfigurasi ini berupa file teks yang dapat di sunting dengan menggunakan ‘notepad’, digunakan untuk mengatur konfigurasi PC yang akan disimulasi seperti file ROM BIOS dan ROM VGA yang digunakan, kapasitas ‘RAM’, perangkat-perangkat pendukung seperti floppy, harddisk dan lain-lain. Saat ini kita belum perlu merubah konfigurasi tersebut biarkan apa adanya. Pada konfigurasi ini ‘floppya.img’ di pasang pada ‘DRIVE A:’ dan digunakan sebagai ‘boot disk’.

Selanjutnya masukan perintah ‘**s**’ <ENTER>, akan ditampilkan windows ‘Bochs for windows – display’ yang sedang melakukan proses ‘booting’ namun tidak berhasil karena tidak menemukan diskboot, seperti ditampilkan pada teks di bawah

```
Booting from Floppy...
Invalid system disk
Replace the disk, and then press any key
```

Atau tampil seperti berikut:



'Floppya.img' belum terisi 'system file' atau 'kernel' sehingga proses boot gagal. Sekarang kita akan format 'floppya.img' dan menambahkan 'system file' ke dalamnya. Panggil '**DosFp**' <ENTER>.

Pada windows 'Bochs' masukan perintah '**A:>format B:/S**' <ENTER>, selesaikan proses format dan berikan nama label disk jika perlu. Jika sudah selesai, untuk memastikan bahwa floppy pada drive 'B:' terisi dengan 'system file', periksa dengan perintah berikut '**A:>dir B:**' <ENTER>, jika tidak ada kesalahan pada windows 'Bochs' akan ditampilkan berita berikut:

```
A:>dir B:  
  
Volume in drive B is DOS  
Volume Serial Number is 3256-1404  
Directory of B:\  
COMMAND.COM 94,292 05-05-03 21:22  
1 file(s) 94,292 bytes  
0 dir(s) 1,235,968 bytes free
```

```
A:>
```

Matikan PC-Simulator (klik tombol Power Off), sekarang kita coba untuk menggunakan 'floppya.img' sebagai 'boot disk' lagi, ketik '**S**' <ENTER>. Jika tidak ada kesalahan akan

ditampilkan teks seperti berikut:

```
Booting from Floppy... Starting BootCD.....
```

```
Microsoft (R) MS-DOS 7.1  
(C) Copyright Microsoft Corp 1981-1999.
```

```
A:\>
```

Pada baris kedua terdapat teks ‘starting BootCD’ Ini disebabkan oleh pembuatan boot disk awal menggunakan sistem yang berasal dari CD. Namun proses boot sebenarnya saat ini berasal dari ‘Floppya.img’. Pada modul selanjutnya kita akan belajar untuk membuat ‘bootloader’ dan ‘system file’ ini.

Tugas

1. Apa yang dimaksud dengan kode ‘ASCII’, buatlah tabel kode ASCII lengkap cukup kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary dan hexadesimal serta karakter dan simbol yang dikodekan.
2. Carilah daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap (dari buku referensi atau internet). Daftar perintah ini dapat digunakan sebagai pedoman untuk memahami program ‘boot.asm’ dan ‘kernel.asm’.

©Muhammadiyah university Press

Modul 2

Mengenal Proses Pembuatan ‘DISK BOOT’

Tujuan

Memahami langkah-langkah pembuatan ‘disk boot’ (floppy) pada PC.

Pendahuluan

Proses boot adalah proses yang pertama kali terjadi dalam sistem PC. Ketika tombol ‘Power’ di tekan, mikroprosesor memasuki keadaan ‘reset’ dan pada saat itu semua register di dalam mikroprosesor berada dalam keadaan kosong. Selanjutnya kode program pada memori yang terletak pada alamat **0xFFFF** di ‘load’ (dipindahkan) ke dalam ‘Code Segmen’ dan dieksekusi. Program yang terletak pada alamat **0xFFFF** ini biasanya disimpan dalam sebuah chip ROM dan dikenal dengan nama *Basic Input Output System* (BIOS). Program yang pertama kali dijalankan oleh BIOS adalah memeriksa memori diteruskan dengan memeriksa semua perangkat bantu yang terhubung pada PC, seperti serial port, floppy dan sebagainya. Jika tidak ditemukan kesalahan fatal pada proses ini, BIOS melanjutkan prosesnya dengan mencari file sistem operasi (*Operating System* – OS) atau ‘Kernel’. Urutan pencarian OS sesuai dengan pengaturan prioritas, dapat dimulai dari floppy disk, hardisk, CDROM, bootrom dan seterusnya. Pada sistem operasi lama seperti MSDOS, semua proses ini dilakukan dalam mode real (*real-mode*), sedangkan pada OS modern salah satu proses yang dilakukan oleh ‘kernel’ adalah merubah mode kerja dari ‘*real-mode*’ ke

dalam mode kerja ‘*protect-mode*’. Pada modul praktikum ini akan dipelajari OS yang selalu bekerja pada mode-kerja ‘real-mode’. Mode kerja ‘*protect-mode*’ akan di bahas pada modul praktikum selanjutnya.

BIOS tidak memindahkan seluruh file ‘kernel’ ke dalam memori kerja (RAM), hanya sebagian kecil program saja yang dipindahkan, yaitu program yang terletak pada sektor pertama pada floppy disk atau harddisk disebut ‘**BOOT SECTOR**’. Ukuran program pada boot-sector hanya 512 byte dan dua byte terakhir berisi data 0xAA55, disebut ‘*boot-signature*’. Jika kode ‘*boot-signature*’ ini tidak ada maka disk dianggap tidak bersifat ‘bootable’ dan BIOS akan mencari media booting yang lainnya. Potongan program sepanjang 512 byte pada boot-sector ini disebut ‘**BOOTSTRAP LOADER**’. BIOS memindahkan program bootstrap-loader ke dalam RAM (memori kerja) dengan alamat 0x7C00 kemudian mengeksekusinya. Inilah yang disebut dengan proses boot atau sering disebut booting. Sebelum melakukan praktikum sebaiknya anda memahami proses ini dengan baik.

Pada praktikum ini anda akan mencoba untuk membuat program bootstrap loader sederhana pada sebuah disk. Semua percobaan akan dilakukan dengan menggunakan program PC-Simulator open source dengan nama ‘Bochs’ untuk memahami lebih jauh tentang ‘Bochs’ anda dapat mengunjungi alamat web-site ini www.bochs.sourceforge.net

Peralatan

Sebuah PC dengan sistem operasi Windows Xp yang didalamnya terdapat program PC-Simulator ‘Bochs’, kompiler bahasa assembly ‘nasm’, kompiler bahasa C dari ‘Dev-Cpp’ (gcc versi windows) dan beberapa program bantu seperti ‘make’, ‘debug’, ‘dd’, dan ‘tdump’. Sebagai ‘editor-text’ untuk

memodifikasi program digunakan program ‘Notepad’ yang terdapat pada sistem operasi Windows Xp.

Daftar perintah bahasa assembly mesin intel x86 (dari tugas modul-1), sebagai alat bantu.

Langkah kerja

1. Buka ‘Command Prompt’, atur ‘path’ dan pergi ke direktori kerja. Klik ‘Start | run’ ketik ‘cmd’ <ENTER>, pada windows ‘Command Prompt’ ketik ‘CD OS’, dan jalankan perintah ‘setpath’, terakhir ketik ‘cd LAB/LAB2’ dilanjutkan dengan perintah ‘DIR’. Prosedure ini adalah standar prosedure awal yang harus anda lakukan sebelum melakukan praktikum di setiap modul. Direktori kerja akan tampak seperti pada gambar berikut.

```
C:\OS\Lab\LAB2>dir
 Volume in drive C is S3A4070D001
 Volume Serial Number is A8E5-3D6F

 Directory of C:\OS\Lab\LAB2

15/12/2008  13:54    <DIR>    .
15/12/2008  13:54    <DIR>    ..
15/12/2008  13:26           1.625 bochssrc.bxrc
15/12/2008  13:48           15.914 boot.asm
15/12/2008  13:52              78 dosfp.bat
14/12/2008  11:45           7.966 kernel.asm
15/12/2008  13:51             228 Makefile
15/12/2008  12:20           44 s.bat
                           6 File(s)   25.855 bytes
                           2 Dir(s)  15.831.035.904 bytes free

c:\OS\Lab\LAB2>
```

Gambar 2.1 Direktori kerja LAB2

2. Menyiapkan file ‘floppya.img’. Ingat namanya harus ‘floppya.img’. Jalankan ‘bxImage’, selanjutnya jawablah pertanyaan-pertanyaan yang muncul dengan urutan berikut ‘fd’, ‘1.44’ dan ‘floppya.img’. Lihat proses ini pada modul-1 jika perlu. Pastikan hasil file ‘floppya.

img', dengan memasukan perintah 'dir', hasilnya seperti pada gambar berikut

```
Command Prompt
Volume Serial Number is A8E5-3D6F
Directory of C:\OS\Lab\LAB2
15/12/2008 16:26 <DIR> .
15/12/2008 16:26 <DIR> ..
15/12/2008 16:18 1.625 bochssrc.bxrc
15/12/2008 16:05 15.916 boot.asm
15/12/2008 13:52 78 dosfp.bat
15/12/2008 16:05 1.474.560 floppya.img
14/12/2008 11:45 7.966 kernel.asm
15/12/2008 16:21 228 Makefile
15/12/2008 12:20 44 s.bat
               7 File(s)   1.500.417 bytes
               2 Dir(s)  15.819.517.952 bytes free
C:\OS\Lab\LAB2>
```

Gambar 2.2 Memastikan adanya file 'floppya.img'

3. Mem-format 'floppya.img' dan mengisinya dengan sistem operasi DOS versi 7. Jalankan perintah berikut: 'dosfp'. Pindah ke windows 'Bochs' (PC Simulator), klik menu gambar floppy disk nomor dua dari kanan, seperti pada gambar berikut



4. Selanjutnya atur lokasi file image sehingga menunjuk ke file 'floppya.img' yang terdapat pada direktori kerja 'LAB2', klik tombol 'BROWSE' dan atur sehingga seperti tampak pada gambar di bawah, dan lanjutkan dengan klik 'OK'.



5. Dari prompt 'A:>' ketikan perintah 'A:>Format B: /S' selesaikan prosesnya. Jika tidak ada kesalahan maka 'floppya.img' sekarang dapat digunakan untuk proses booting. Tutup windows 'Bochs', klik menu POWER. Pada window 'Command Prompt' akan tampak bekas akatifitas 'Bochs' seperti pada gambar berikut:

```

C:\OS\Lab\LAB2>cd "..\..\Bochs-2.3.5\dos"
C:\OS\Bochs-2.3.5\dos>..\bochs -q F bochsrc2.txt
000000000001[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc2.txt
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochsout.txt
# In bx_win32_gui c:>exit(void)
=====
Bochs is exiting with the following message:
[WGUI ] POWER button turned off.
=====
C:\OS\Bochs-2.3.5\dos>cd "C:\os\Lab\LAB2"
C:\OS\Lab\LAB2>
  
```

6. BOOT PC-simulator dengan file 'floppya.img'. Pindah ke direktori kerja pada window 'Command Prompt' dan jalankan perintah 'S' <ENTER>. Sekarang PC- Simulator akan melakukan proses boot dengan disk boot yang berasal dari file 'floppya.img' yang diletakkan pada

'drive A:', dan proses boot telah berjalan dengan lancar. Pastikan dengan menekan tombol gambar floppy yang tidak di silang (paling kiri). Tutup kembali PC-Simulator, klik tombol POWER.



7. Kompilasi source code 'boot.asm' dan memindah hasilnya ke bootsector 'floppya.img'. Pindah ke direktori kerja 'LAB2' ketikan perintah 'cls' untuk membersihkan layar. Selanjutnya jalankan perintah 'make fp.disk' <ENTER> jika langkah ini berhasil, pada layar akan ditampilkan teks berikut:

A screenshot of a Windows Command Prompt window titled 'Command Prompt'. The text inside the window is:

```
C:\OS\Lab\LAB2>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.iso
rawwrite dd for windows version 0.5.
Written by John Newbegin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out
C:\OS\Lab\LAB2>
```

8. Boot PC Simulator dengan program bootstraploader yang baru. Jalankan PC- Simulator ketik 'S' <ENTER>. Tampilan pada PC-Simulator sekarang adalah sebagai berikut:

```
Booting from Floppy...
Loading kernel ver 0.01
.
.
.
ERROR : Press Any Key to Reboot
```

Bootloader telah berubah, sekarang telah menggunakan program bootstrap-loader yang berasal kompilasi file ‘boot.asm’ yaitu file ‘boot.bin’ yang telah disalin kedalam file ‘floppya.img’ tetapi proses boot gagal karena tidak ada file ‘KERNEL.BIN’ pada ‘floppya.img’. Kita akan buat file ‘KERNEL.BIN’ pada tahap berikutnya. Sekarang kita coba merubah tampilan program bootstrap-loader, tutup PC-Simultor, klik tombol POWER pada PC-Simultor.

9. Menyunting file ‘boot.asm’, ketikan ‘notepad boot.asm’ <ENTER>, cari teks ‘Loading kernel’ kelompok baris bawah, pada windows ‘Notepad’, tekan tombol ‘CTRL+F’ masukan kata kunci ‘Loading kernel’, <ENTER>, klik ‘CANCEL’. Tampilan pada Notepad tampak seperti gambar berikut:

```
boot.asm - Notepad
File Edit Format View Help
ImageName db "KERNEL BIN"
=====
;----- Teks yang akan ditampilkan saat mulai proses
;----- BOOT : Loading Kernel ver 0.01
;----- Di awali dan diakhiri dengan tanda :
;----- -- 0xD akhir baris
;----- -- 0xA baris baru
;----- -- 0x0 Karakter 'NULL' pembatas dengan data di bawahnya
;----- -----
msgLoading db 0xD, 0xA, "Loading kernel ver 0.01", 0xD, 0xA, 0x00
msgCRLF db 0xD, 0xA, 0x00
msgProgress db ".", 0x00
```

Sekarang sunting baris dibelakang variabel ‘msgLoading’ sehingga menjadi seperti berikut

```
'msgLoading    db    0x0D, 0x0A, "Belajar membuat
BOOTSTRAP-LOADER", 0x0D, 0x0A, 0x00'
```

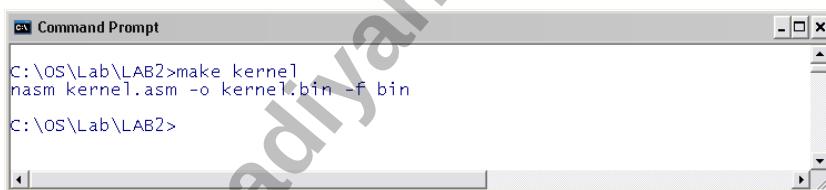
Buatlah sehingga menjadi satu baris. Anda juga dapat mengganti kata-kata ‘ERROR: Press Any Key to Reboot’ dengan cara yang sama, yang didimpan pada variabel ‘msgFailure’.

Simpan file ‘boot.asm’, tekan ‘CTRL+S’ pada notepad dan pindah ke window ‘Command Prompt’ direktori

kerja, ulangi proses kompilasi file ‘boot.asm’, jalankan perintah ‘**Make fp.disk**’ <ENTER>, setelah proses kompilasi, jalanan PC-Simulator ‘s’<ENTER>. Sekarang teks yang ditampilkan di layar adalah seperti yang anda tuliskan pada file ‘boot.asm’ yaitu seperti berikut:

```
Booting from Floppy...  
  
Belajar membuat BOOTSTRAP-LOADER  
.....  
ERROR : Press Any Key to Reboot
```

10. Menyiapkan file ‘KERNEL.BIN’: Prototype source code program kernel di simpan pada file ‘kernel.asm’, periksa keberadannya pada direktori kerja anda. Selanjutnya lakukan proses kompilasi untuk menhasilkan file ‘KERNEL.BIN’, huruf besar atau kecil pada nama file tidak berpengaruh. Jalankan perintah berikut: ‘**make kernel**’ pada layar tampak seperti gambar berikut:

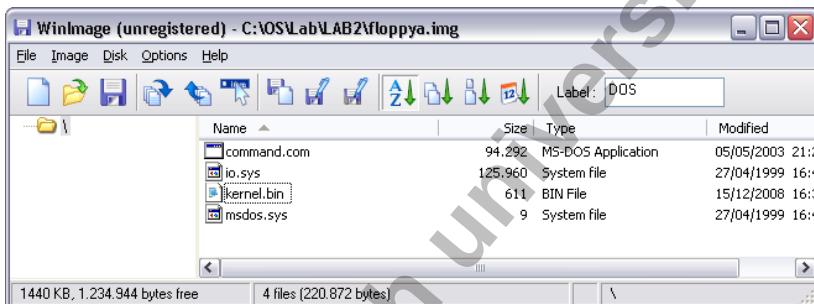


```
Command Prompt  
C:\OS\Lab\LAB2>make kernel  
nasm kernel.asm -o kernel.bin -f bin  
C:\OS\Lab\LAB2>
```

Jika tidak ada berita kesalahan, berarti proses kompilasi telah berhasil dan pada direktori kerja anda tredapat tambahan file baru yaitu ‘kernel.bin’, periksalah dengan perintah ‘dir’.

11. Memindahkan file ‘kernel.bin’ ke dalam file image ‘floppya.img’. Proses ini sedikit rumit, seperti proses kompresi sebuah file ke dalam file ‘zip’. Untuk melakukan hal ini kita gunakan program shareware namanya ‘winimage’. Bukalah direktori kerja ‘C:\OS\LAB\LAB2’ dengan menggunakan ‘Windows

‘Explorer’ kemudian dobel klik pada file ‘floppya.img’, atau panggil dari program winimage klik ‘Start|All Programs|Winimage|winimage’ selanjutnya klik menu ‘open’ dan cari file ‘floppya.img’ pada direktori kerja. Untuk memasukan file ‘kernel.bin’, pada window ‘WinImage’ klik menu ‘image|inject’ cari file ‘kernel.bin’ pada direktori kerja dan OK. Selanjutnya lakukan proses penyimpanan ‘File|Save’ dan keluar dari ‘WinImage’. Hasil akhir seperti di tampilkan pada gambar berikut:



12. Selanjutnya siap melakukan proses boot pada PC Simulator dengan menggunakan ‘floppya.img’ yang sudah diberi tambahan file ‘kernel.bin’. Kembali ke Windows ‘Command Prompt’ direktori kerja, jalankan PC-Simulator dengan memasukan perintah ‘S’<ENTER>. Jika tidak ada kesalahan pada proses sebelumnya maka pada layar PC Simulator akan ditampilkan teks seperti berikut ini:

```
Booting from Floppy...

Belajar membuat BOOTSTRAP-LOADER
.....
.....
.

Welcome to MY KERNEL Ver 0.01
MY KERNEL$>
```

13. Memodifikasi file ‘kernel.asm’ : pada bagian berikut ini kita akan mencoba melakukan modifikasi kecil pada file ‘kernel.asm’. Jalankan perintah berikut ‘Notepad kernel.asm’ <ENTER>. Cari teks ‘Welcome to MY KERNEL ...’ dari windows ‘notepad’, tekan tombol ‘CTRL+F’ ketikan ‘Welcom to MY KERNEL’ dan tekan <ENTER> dan klik ‘CANCLE’. Teks tersebut disimpan pada variabel ‘strWelcomeMsg’, gantikan isinya dengan ‘Belajar membuat KERNEL’:

```
strWelcomeMsg db "Belajar membuat KERNEL", 0x00
```

Simpan file ‘kernel.asm’, tekan ‘CTRL+S’, tutup windows ‘Notepad’ dan kembali ke ‘Command Prompt’ direktori kerja.

Ulangi proses kompilasi, jalankan perintah ‘make kernel’<ENTER>, selanjutnya ulangi proses yang dilakukanp ada nomor 11 (memindahkan file ‘kernel.bin’ ke dalam file ‘floppya.img’ dengan ‘WinImage’).

Terakhir jalankan PC Simulator, ‘s’<ENTER>. Jika langkah yang anda lakukan tidak ada kesalahan maka pada layar PC Simulator akan ditampilkan teks seperti berikut:

```
Booting from Floppy...
```

```
Belajar membuat BOOTSTRAP-LOADER
```

```
.....
```

```
.....
```

```
..
```

```
Belajar membuat KERNEL
```

```
MY KERNEL$>
```

14. Proses diatas merupakan proses pengembangan sistem operasi yang paling dasar, setiap sistim operasi, seperti windows, linux, unix, mac dan lain-lain, akan melakukan tahapan proses tersebut. Anda juga dapat

mengembangkan ‘embedded system’ dengan cara yang serupa.

Untuk memastikan apakah ‘floppya.img’ benar-benar dapat digunakan pada PC yang sebenarnya, anda dapat memindah isi ‘floppya.img’ ke dalam floppy sebenarnya dengan program ‘WinImage’ (sudah tentu pada PC anda harus ada FDD Drive), jalankan program ‘WinImage’ buka file ‘floppya.img’ dan klik menu ‘disk|Write disk’. Selanjutnya lakukan proses ‘BOOT’ pada PC anda dengan floppy tersebut, jika pada langkah anda tidak ada kesalahan maka pada PC akan ditampilkan teks seperti pada PC Simulator, silahkan mencoba?

Pada modul berikutnya akan kita akan belajar cara menjelak tahapan proses (*trace* atau *debugging*) program ‘bootstraploader’ dan ‘kernel’.

Tugas

1. Pelajari cara kerja program ‘boot.asm’ buatlah algoritma dari program tersebut dalam bentuk flowchart. Untuk memudahkan dalam memahami proses boot buatlah dua jenis algoritma, pertama buat alogritma yang bersifat global dan kedua buat algoritma yang bersifat lebih detail.
2. Lakukan hal yang sama untuk program ‘kernel.asm’.

Hint: Gunakan buku referensi atau akses internet sebagai panduan

©Muhammadiyah university Press

Modul 3

Mengenal Cara ‘Debugging’ Program Bootstrap-Loader

Tujuan

Memahami cara melakukan proses ‘debugging’ program bootstrap-loader dan kernel dengan menggunakan program PC-Simultor ‘Bochs’.

Pendahuluan

Pada tahapan pengembangan program selalu dihadapi munculnya kesalahan pada program, terdapat dua jenis kesalahan yang biasa terjadi pada proses program, yaitu kesalahan syntax dan kesalahan logika.

Kesalahan syntax atau bentuk adalah kesalahan yang terjadi pada source code yang dapat mengakibatkan gagalnya proses kompilasi. Hal ini disebabkan oleh adanya beberapa struktur penulisan dan/atau istilah yang tidak di mengerti oleh kompiler (penerjemah bahasa pemrograman dari bahasa tingkat tinggi menjadi bahasa mesin). Kesalahan ini juga dapat disebut sebagai kesalahan penulisan bahasa pemrogramman. Kesalahan jenis ini sangat mudah di betulkan karena terlihat jelas. Biasanya kompiler yang baik akan menunjukkan letak kesalahannya dengan memberikan informasi mengenai nomor baris pada source code yang menyebabkan terjadinya kesalahan.

Jenis kesalahan kedua adalah kesalahan logika, kesalahan ini lebih sulit dideteksi, karena secara syntak sudah betul namun setelah program dijalankan, hasil akhirnya tidak sesuai dengan apa yang diharapkan oleh programmer, terdapat

kesalahan alur pikir atau logika di dalam program. Cara paling mudah untuk mendeteksi jenis kesalahan ini adalah dengan menggunakan fasilitas pembantu yang disebut ‘debugger’, hampir setiap bahasa pemrogramman menyediakan fasilitas debugger. Namun dalam pengembangan sistem operasi ini kita tidak dapat menggunakan fasilitas debugger yang disediakan oleh kompiler. Karena PC harus di reset untuk menjalankan program bootstrap-loader, dan jika PC di reset semua yang tersimpan dalam RAM juga akan di reset. Diperlukan cara tersendiri untuk melakukan debugging pada program bootstrap-loader dan kernel.

Salah satu cara melakukan debugging pada sistim ini adalah dengan menggunakan PC-Simulator yang menyediakan fasilitas debugging seperti ‘Bochs’. Pada modul ini akan dipelajari cara melakukan proses ‘debugging’ program bootstrap loader dengan PC-Simulator. Untuk melakukan proses ini sebaiknya anda sudah memahami alur logika atau cara kerja program ‘boot.asm’ dan ‘kernel.asm’.

Prototipe program ‘boot.asm’ oleh BIOS akan di letakkan pada alamat ‘0000:07C00’, selanjutnya BIOS akan menyerahkan tugasnya pada program ‘boot.asm’, dimulai dengan mengatur register segmen, kemudian mengidentifikasi format disk boot untuk menentukan lokasi sektor yang digunakan untuk menyimpan data root direktori (daftar nama direktori dan nama file). Selanjutnya mencari nama file yang akan digunakan sebagai kernel (pada modul ini namanya ‘KERNEL.BIN’), dan memindahkan isi file kernel pada lokasi memori ‘0100:0000’ kemudian menjalankan program kernel mulai dari alamat tersebut.

Prototipe program ‘kernel.asm’ relatif lebih sederhana yaitu menjalankan program ‘shell’ sangat sederhana, menunggu perintah yang dimasukan user dari ‘keyboard’, jika ada perintah yang dimasukan user, selanjutnya shell mengidentifikasi perintah dan menjalankan operasi sesuai

dengan perintah user. Saat ini hanya perintah ‘ver’ dan ‘exit’ yang dimengerti oleh ‘shell’ yang dijalankan dari ‘kernel.bin’.

Peralatan

Sebuah PC dengan sistem operasi Windows Xp yang didalamnya terdapat program PC-Simulator ‘Bochs’, kompiler bahasa assembly ‘nasm’, kompiler bahasa C dari ‘Dev-Cpp’ (gcc versi windows) dan beberapa program bantu seperti ‘make’, ‘debug’, ‘dd’, dan ‘tdump’. Sebagai ‘editor-text’ untuk memodifikasi program digunakan program ‘Notepad’ yang terdapat pada sistem operasi Windows Xp.

Dierktori kerja ‘C:\OS\LAB\LAB3’

Langkah Kerja

1. ‘Start | run’ ketik ‘cmd’ lanjutkan dengan ‘CD OS’, ‘setpath’ dan ‘cd LAB/LAB3’.
2. File sudah di siapkan, tugas anda adalah melakukan proses ‘debugging’. Program ‘Bochs’ yang diaktifkan adalah program versi debug yaitu ‘Bochsdbg’, lihatlah pada perintah yang tersimpan pada file ‘s.bat’, ketika ‘type s.bat’.
3. Mulai melakukan ‘debugging’: masukan perintah ‘S’ <ENTER>. Layar pada PC-Simulator akan terlihat gelap, tidak ada aktifitas, tidak ada kesalahan disana tetapi jalannya program dihentikan oleh ‘Bochs’ menunggu masukan dari user. Pindahlah ke window ‘Command Prompt’ yang sekarang muncul tampilan seperti pada gambar berikut:

The screenshot shows a Windows console window titled "Bochs for Windows - Console". The command entered is "C:\OS\LAB\LAB3>> bochs-2.3.5\bochsrc -q -f bochsrc.bxrc". The output shows the Bochs x86 Emulator version 2.3.5 starting up, reading configuration from bochsrc.bxrc, installing the win32 module, and using log file bochs.log. It then displays assembly code at address F000:FFF0:

```
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
00000000000000000000000000000000i[ ] reading configuration from bochsrc.bxrc
00000000000000000000000000000000i[ ] installing win32 module as the Bochs GUI
00000000000000000000000000000000i[ ] using log file bochs.log
Next at t=0
(0) [0xfffffff] f000:ffff0 (unk. ctxt): jmp far f000:e05b ; ea5be000f0
<bochs:1>
```

A callout bubble labeled "CLOCK" points to the assembly instruction "jmp far f000:e05b".

4. Kondisi pada gambar di atas menjelaskan kondisi PC pada mode 'Real-Mode' yang sedang akan menjalankan program yang pertama kali (0), yaitu program yang terdapat pada alamat 'F000:FFF0', lokasi alamat tersebut pada PC merupakan alamat paling atas yang dapat diakses oleh PC dan biasanya digunakan untuk menyimpan alamat ROM BIOS. Selanjutnya PC akan memanggil program yang terdapat pada BIOS. Pertama kali yang dilakukan BIOS adalah loncat ke alamat yang lebih rendah yaitu alamat f000:e05b terlihat pada layar sebagai baris perintah 'jmp far f000:e05b' kode bahasa mesinnya adalah 'ea5be000f0' akan lebih mudah dimengerti jika dipisahkan dalam bentuk seperti ini 'ea 5be0 00f0', Kode 'ea' : adalah kode untuk perintah 'jmp far' sedangkan data 8 byte berikutnya merupakan lokasi alamat yang ditunjuk, 4 byte paling akhir adalah menunjukkan alamat segmen yang dituju dengan susunan dibalik '00f0' menjadi 'f000'. Sedangkan 4 byte di tengah adalah alamat penunjuk (IP), susunannya juga dibalik '5be0' menjadi 'e05b' inilah aturan penyimpanan data pada PC IBM compatible. Lokasi program yang akan dijalankan oleh PC pada mode kerja 'Real-Mode' adalah lokasi yang ditunjuk oleh kombinasi register CS:IP. Anda dapat melihat isi register CS dan IP dengan perintah 'r'. Sekarang ketikan 'r' <ENTER> akan ditampilkan teks berikut :

```

000000000001[      ] reading configuration from bochssrc.bxrc
000000000001[      ] installing win32 module as the Bochs GUI
000000000001[      ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:ffff0 (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
<bochs:> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:0000F20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8: 0x00000000:00000000 r9: 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:>

```

5. Selanjutnya kita suruh PC untuk mengeksekusi perintah tersebut, ketikan ‘s’<ENTER> kemudian lanjutkan dengan perintah ‘r’ <ENTER>. Pada layar akan ditampilkan terks berikut:

```

rip: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:> s
CLOCK pertama (1)
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax      ; 31c0
<bochs:> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:0000F20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8: 0x00000000:00000000 r9: 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:>

```

6. Jika anda ingin tahu tahapan detail yang dilakukan oleh PC, jalankan dengan perintah ‘s’<ENTER> secara berulang. Tetapi dengan cara ini, untuk sampai pada proses bootstrap mungkin tidak akan selesai dalam satu hari. Cara yang efektif adalah kita biarkan PC melanjutkan pekerjaannya, kemudian kita suruh berhenti saat PC mulai memasuki tahapan proses BOOT, yang dimulai pada alamat 0000:7C00. Hal ini dapat dilakukan dengan cara memberikan sebuah sinyal berhenti yang disebut ‘break point’.

Masukan perintah berikut ‘vb 0:0x7C00’ <ENTER>

Maksud perintah ini adalah membuat titik pemberhentian (halte) pada alamat 0000:7C000. Selanjutnya kita perintahkan PC untuk melanjutkan pekerjaannya sekarang, yaitu melanjutkan program yang terdapat pada BIOS untuk memeriksa RAM dan peralatan lainnya.

Masukkan perintah ‘c’<ENTER>

Maksud perintah ini adalah teruskan (Continue) prosesnya sampai ke titik pemberhentian. Dalam sekejap PC sudah sampai pada pemberhentian yang kita buat di atas yaitu pada alamat 0000:7C00, pada layar akan tampak teks seperti berikut.

The screenshot shows a window titled 'Bochs for Windows - Console'. The assembly code displayed is:

```
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8: 0x00000000:00000000 r9: 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:000e05b
eflags: 0x00000002
IOPL=0 id via vif ac vm rf nt df if tf sf zf af pf cf
<bochs:4> vb 0:0x7C00
<bochs:5> c
(2089918400) Breakpoint 51119118, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

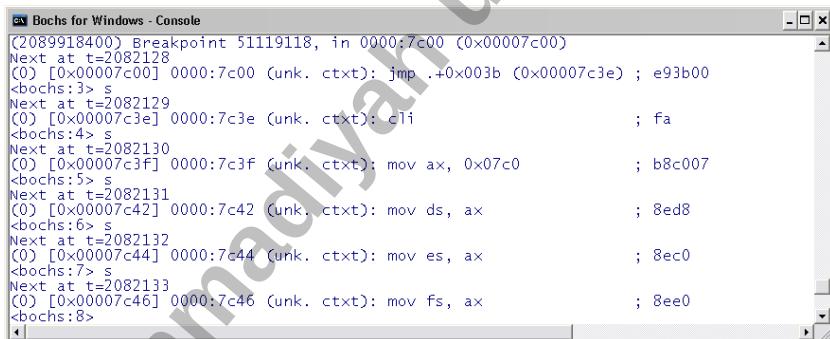
A callout bubble points to the text 'Jumlah CLOCK dihitung sejak POWER ON'.

Sedangkan pada PC Simulator mulai ditampilkan teks yang berakhir dengan tulisan berikut (periksalah):

Booting from Floppy...

7. Sekarang PC mulai memasuki tahapan ‘BOOTSTRAP-LOADER’, untuk sampai pada tahap ini PC sudah menghabiskan clock sebanyak ‘2082128’ clock (lihat gambar di atas). Termasuk langkah memindahkan data bootsector ke dalam memori pada alamat 0000:7C00. Program yang akan dieksekusi selanjutnya adalah program yang berasal dari BOOTSECTOR pada diskboot.

Jika digunakan ‘floppya.img’ sebagai disk-boot maka sekarang PC akan mulai menjalankan program ‘boot.asm’. Selanjutnya bandingkan 10 instruksi berikutnya yang akan dieksekusi oleh PC dengan program yang terdapat pada ‘boot.asm’, lakukan dengan cara menjalankan PC langkah demi langkah (debugging) dengan perintah ‘s’ <ENTER>, tulislah setiap teks yang ditampilkan pada setiap langkah. Perhatikan semua perubahan yang terjadi. Jika terjadi kesalahan, anda dapat menghentikan proses dengan perintah ‘q’<ENTER> dan mengulangi proses dari awal atau klik tombol POWER pada PC Simulator. Untuk melihat daftar menu yang disediakan dapat digunakan perintah ‘h’<ENTER>. Untuk memerintahkan PC agar melanjutkan prosesnya masukan ‘c’<ENTER>. Empat instruksi selanjutnya akan tampak seperti pada gambar berikut:

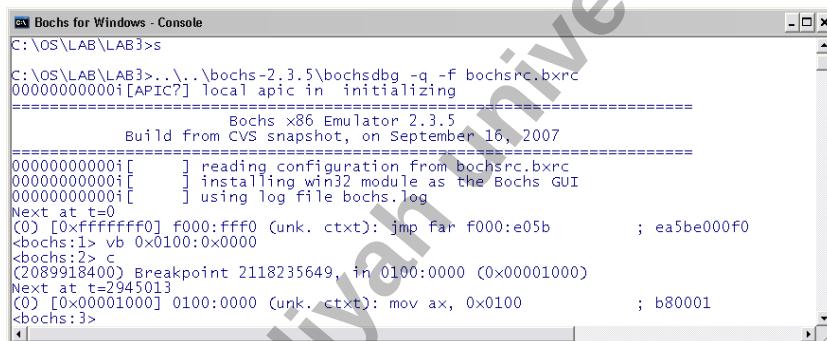


The screenshot shows a Windows console window titled 'Bochs for Windows - Console'. It displays assembly code being executed by the Bochs emulator. The code consists of several 'jmp' instructions with various offsets and labels like '_bochs:3', '_bochs:4', etc. The assembly mnemonics and addresses are color-coded. The window has standard Windows controls (minimize, maximize, close) and a scroll bar.

```
(C:\089918400) Breakpoint 51119118, in 0000:7c00 (0x00007c00)
Next at t=2082128
<bochs:3> s
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:4> s
Next at t=2082129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:5> s
Next at t=2082130
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:6> s
Next at t=2082131
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:7> s
Next at t=2082132
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:8> s
Next at t=2082133
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
```

8. Selanjutnya anda dapat memerintahkan PC Simulator untuk melanjutkan pekerjaannya. Anda juga dapat menambahkan ‘break-point’ yang lain (maksimal 7). Tahapan penting berikutnya adalah ketika PC Simulator mulai menjalankan program ‘kernel.bin’, hal ini terjadi pada alamat ‘0100:0000’.
9. Menghentikan PC Simulator pada saat akan menjalankan program ‘kernel.bin’: Mulailah dari awal, hentikan

'debugging' sebelumnya dengan memasukan perintah 'q'<ENTER>, jika PC Simulator sedang bekerja, hentikan dengan menekan tombol 'CTRL+C' kemudian 'q'<ENTER>. Selanjutnya mulai dari awal, ketik 's'<ENTER> (dari 'Command Prompt'). Kemudian buatlah break-point, masukan perintah 'vb 0x0100:0x0000' untuk mengehentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin'. Selanjutnya perintahkan PC untuk melanjutkan perkerjaan, 'c'<ENTER>. Jika langkah anda sesuai dengan petunjuk di atas maka pada layar 'debugging' akan di tampilkan teks berikut:



The screenshot shows a Windows console window titled 'Bochs for Windows - Console'. The command entered was 'bochs-2.3.5\bochsrc.bxrc'. The output shows the Bochs x86 Emulator starting up, reading configuration from bochsrc.bxrc, installing the win32 module as the GUI, and using log file bochs.log. It then displays assembly code at address 0x0000000000f0, showing a jump instruction (jmp far) to address 0x00000000e05b. A breakpoint is set at address 0x000000001000, which is triggered at t=2945013. The assembly code at this address is mov ax, 0x0100. The console window has a standard Windows title bar and scroll bars.

10. Selanjutnya teruskan langkah PC Simulator step-by-step minimal sebanyak 10x, ketik 's'<ENTER>, step berikutnya dapat dilakukan dengan cara menekan tombol <ENTER> secara langsung. Perhatikan dan catat setiap perubahan teks yang ditampilkan. Bandingkan dengan source-code pada program 'kernel.asm'.
11. Langkah-langkah di atas adalah merupakan salah satu cara yang banyak di gunakan para pengembang perangkat lunak, termasuk pengembangan sistem operasi. Jika anda tertarik sebaiknya anda membiasakan dengan kegiatan di atas.

Tugas

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.
2. Baca buku referensi, jelaskan perbedaan antara mode kerja ‘Real-Mode’ dan mode kerja ‘Protect-Mode’ pada PC IBM Compatible.

©Muhammadiyah university Press

Modul 4

Pengolahan File dan Directory Menggunakan MS-DOS/Command Prompt di Windows

Tujuan

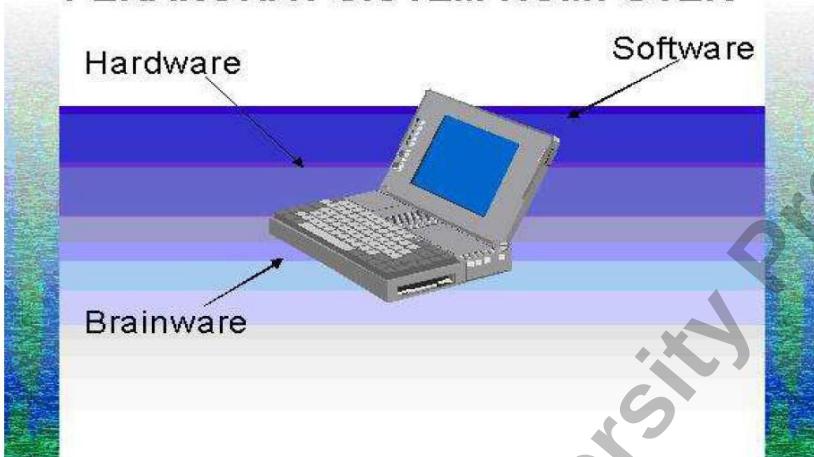
Mengenal perintah-perintah pengolahan file dan directory menggunakan command prompt pada windows xp

Pendahuluan

MS-DOS adalah singkatan dari Microsoft Disk Operating System, yaitu Sistem Operasi berbasis baris-perintah (command-line) yang digunakan pada PC. Seperti sistem operasi lain contohnya OS/2, ia menterjemahkan input dari keyboard menjadi pekerjaan yang dapat dilakukan oleh komputer, ia juga dapat menangani pekerjaan seperti input dan output pada disket atau harddisk, dukungan video, kontrol keyboard , dan banyak lagi fungsi-fungsi internal lainnya yang berkaitan dengan eksekusi sebuah program dan pemeliharaan file.

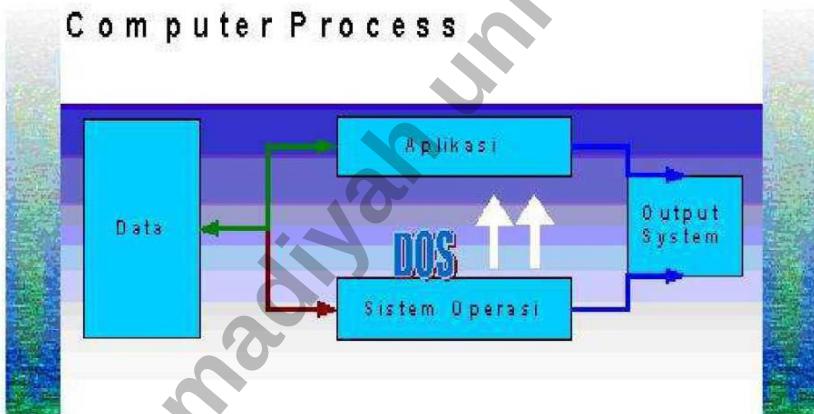
DOS menempati posisi sebagai operating system yang menggunakan CUI(Character User Interface). Dalam hal ini DOS yang saat ini telah tergantikan/diperbarui dengan adanya Microsoft Windows versi 9x, 2k, dan sebagainya yang berbasiskan GUI (Grafical user Interface).

PERANGKAT SISTEM KOMPUTER



Gambar 4.1 perangkat sistem komputer

Computer Process



Gambar 4.2 proses komputer

Beberapa Fungsi dari Operating system (DOS) adalah:

1. Mengorganisasikan atau mengendalikan kegiatan computer
2. Mengatur Memori
3. Mengatur proses input dan output data
4. Menegment file
5. Management directory

Perintah MS-DOS diketikkan dalam sebuah jendela yang disebut Command Prompt Window. Untuk keluar dari MS-DOS, ketik exit dalam jendela tersebut yaitu pada kursor yang berkedipkedip. MS-DOS Mode adalah sebuah shell dimana lingkungan MS-DOS di-emulasikan dalam Sistem Operasi 32-bit, seperti Windows. Program berbasis MS-DOS dapat berjalan di Windows dan biasanya ia membuat sebuah file yang disebut Program Information File (PIF) yang muncul sebagai shortcut di desktop anda.

Dalam pengoperasian DOS terdapat Command-command/perintah yang dikelompokkan dalam 2 kelompok yaitu:

1. Internal Command

Adalah perintah yang tidak lagi membutuhkan file khusus,karena semua instruksi internal sudah ditampung dalam file command.com

2. External Command

Untuk mempermudah mampelajari fasilitas DOS maka tiap perintah sudah terdapat file Help. Untuk menjalankannya bisa digunakan perintah seperti contoh berikut:

A:\>Copy/?

Atau

A:\>help copy

Perintah-Perintah Internal MS-DOS/Command Prompt dalam pengolahan file dan directory

Berikut ini adalah daftar perintah-perintah command prompt yang ada pada MS-DOS mode di Windows XP. Untuk informasi lebih spesifik mengenai suatu perintah, ketik HELP nama perintah di jendela Command Prompt.

ASSOC	Menampilkan atau mengubah asosiasi ekstensi file.
CD/CHDIR	Menampilkan nama atau mengubah direktori sekarang.
CLS	Menghapus layar jendela command prompt.
CMD	Menjalankan interpreter perintah Windows yang baru.
COPY	Menyalin satu atau beberapa file ke lokasi lain.
DEL/ERASE	Menghapus satu atau beberapa file.
DIR	Menampilkan daftar file dan subdirektori dalam sebuah direktori.
EXIT	Keluar dari program CMD.EXE dan menutup jendela command prompt.
HELP	Menyediakan informasi bantuan untuk perintah-perintah Windows.
MD/MKDIR	Membuat direktori.
MOVE	Memindahkan satu atau beberapa file dari satu direktori ke direktori yang lain.
PROMPT	Mengubah command prompt Windows.
RD/RMDIR	Menghapus direktori.
REN/RENAME	Mengubah nama file.
REPLACE	Mengganti file.
TREE	Menampilkan secara grafis struktur direktori dari sebuah drive atau path.
TYPE	Menampilkan isi dari sebuah file teks.
VER	Menampilkan versi Windows yang anda gunakan.
XCOPY	Menyalin file serta pohon direktori.

Untuk membuka jendela Command Prompt, klik Start, pilih All Programs, pilih Accessories, dan kemudian klik Command Prompt

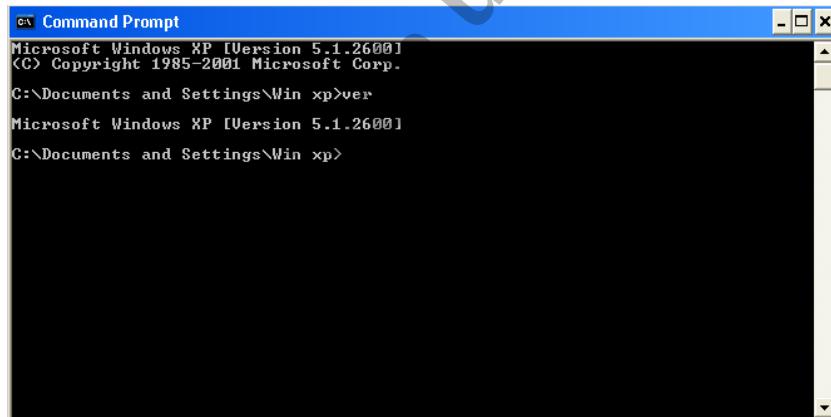
Peralatan

Sebuah PC dengan sistem operasi windows xp Modul sistem operasi

Langkah kerja

Percobaan pertama : Melihat versi MS-Dos (ver)

1. Klik start - all programs - accessories - command prompt sehingga muncul command prompt
2. Ketik perintah ver untuk melihat versi command prompt yang kita gunakan



Percobaan kedua : melihat, masuk, keluar dari directory dan keluar dari command prompt (dir, cd, cd.., exit)

3. ketik perintah cls untuk membersihkan tulisan pada layar
4. ketik perintah dir untuk melihat isi dari directory yang ada pada komputer kita

C:\ Command Prompt

```
C:\Documents and Settings\Win xp>dir
Volume in drive C is systemxp
Volume Serial Number is 0027-C4A9

Directory of C:\Documents and Settings\Win xp

08/28/2009  10:37 AM    <DIR>          .
08/28/2009  10:37 AM    <DIR>          ..
08/29/2009  03:46 PM    <DIR>          Desktop
05/19/2009  01:48 PM    <DIR>          Favorites
08/20/2009  02:47 PM    <DIR>          0_file_dari_ftp_pada_pc1.txt.txt
08/26/2009  12:06 PM    <DIR>          kel12
08/31/2009  02:55 PM    <DIR>          My Documents
09/01/2009  08:48 AM    2,883,584 NTUSER.DAT
08/20/2009  02:58 PM    <DIR>          0_pc1.txt
05/19/2009  08:36 PM    <DIR>          Start Menu
05/20/2009  03:15 PM    <DIR>          WINDOWS
        4 File(s)      2,883,584 bytes
        7 Dir(s)   62,307,606,528 bytes free

C:\Documents and Settings\Win xp>_
```

5. Ketik perintah `cd..` untuk keluar dari directory Win xp dan ketik perintah `cd..` untuk keluar dari directory document and settings sehingga masuk pada drive C
6. Ketik perintah `dir` untuk melihat isi directory yang ada pada drive C. Kemudian ketik perintah `cd windows` untuk masuk pada directory windows.
7. ketik perintah `dir/p` untuk menampilkan nama file perhalaman, kemudian tekan spasi untuk menampilkan halaman berikutnya dan tekan spasi untuk halaman berikutnya lagi dan seterusnya sampai di akhir halaman sesuai dengan banyaknya file dalam directory tersebut.

C:\WINDOWS\system32\cmd.exe - dir /p

```
Volume in drive C is systemxp
Volume Serial Number is 0027-C4A9

Directory of C:\WINDOWS

09/01/2009  08:14 AM    <DIR>          .
09/01/2009  08:14 AM    <DIR>          ..
09/01/2009  08:14 AM          0 0.log
05/19/2009  08:17 PM    <DIR>          addins
05/03/2005  05:43 PM    69,632 Alcctr.exe
05/04/2006  03:26 PM    2,808,832 alcwzrd.exe
05/19/2009  08:20 PM    <DIR>          AppPatch
07/31/2008  02:09 AM    38 avisplitter.ini
09/01/2004  07:00 AM    1,272 Blue Lace 16.bmp
09/01/2004  07:00 AM    82,944 clock.avi
09/01/2004  07:00 AM    17,062 Coffee Bean.bmp
08/31/2009  03:21 PM    4,344 COM+.log
08/31/2009  02:34 PM    70,451 comsetup.log
05/19/2009  08:17 PM    <DIR>          Config
05/19/2009  08:17 PM    <DIR>          Connection Wizard
05/19/2009  01:45 PM          0 control.ini
05/19/2009  03:33 PM    <DIR>          Core1
07/06/1999  01:09 PM    28,252 corelpf.lrs
05/19/2009  01:42 PM    <DIR>          Cursors

Press any key to continue . . .
```

8. Ketik perintah `dir /w` menampilkan file secara mendatar. Kemudian ketik perintah `dir /a` untuk menampilkan semua file terutama file yang terhidden.
9. ketik perintah `exit` untuk keluar dari command prompt

Percobaan ketiga : berubah nama file (ren, rename)

10. Buka windows explorer kemudian masuk ke drive C. Buat folder baru dan beri nama “files”. Masuk ke folder files kemudian arahkan kursor ke daerah yang kosong, klik kanan – new – micorosoft word document. Beri nama dengan “file1”.
11. ulangi langkah 1 dan langkah 5 untuk masuk ke command prompt dan berada pada drive C.
12. ketik `dir` kemudian ketik `cd files` untuk masuk ke directory files. Ketik `dir` lagi untuk melihat isi directory files.
13. ketik perintah `ren file1.doc file.doc` untuk mengubah nama file.
14. ketik perintah `rename file.doc file2.doc` untuk mengubah nama file.
15. ketik perintah `dir` untuk melihat directory files kemudian lihat perubahan file1.doc menjadi file.doc
16. Sekarang lihatlah lewat windows explorer dan buka folder files, pastikan bahwa nama file1.doc telah berubah menjadi file2.doc

Percobaan keempat : copy file dan menghapus file (copy, del, xcopy)

17. Ketik perintah `cd ..` untuk keluar dari directory files.
18. buka windows explorer kemudian masuk ke drive C. buatlah folder baru dengan mengarahkan kursor ke daerah yang kosong kemudian klik kanan – new – folder, beri nama dengan “files2”.
19. Kembali ke command prompt.

20. ketik perintah `copy C:\files C:\files2` untuk mengcopy isi folder files ke folder files2.
21. buka windows explorer untuk memastikan isi folder files telah tercopy ke folder files2.
22. kembali ke command prompt
23. ketik perintah `cd files` untuk masuk ke directory files
24. ketik perintah `copy c:\files\file.doc d:\` untuk mengcopy file.doc ke drive D.
25. ketik perintah `D:` untuk pindah ke drive D kemudian ketik perintah `dir` untuk melihat isi drive D dan pastikan bahwa file.doc telah tercopy.
26. ketik `C:` untuk kembali ke directory awal kemudian ketik perintah `cd..` untuk masuk drive C.
27. ketik `del files` untuk menghapus file yang terdapat pada directory files kemudian tekan yes
28. ketik perintah `cd documents and settings` untuk masuk ke folder documents and settings
29. ketik perintah `cd win xp` untuk masuk ke folder win xp
30. ketik perintah `xcopy desktop D:baru\ /s/e` untuk mengcopy isi file kemudian membuat folder baru dan meletakkan isi copyan ke folder baru yang terdapat pada directory files ke drive d. Kemudian tekan a



A screenshot of a Windows Command Prompt window titled "cmd C:\WINDOWS\system32\cmd.exe". The window shows the command `xcopy desktop D:baru\ /s/e` being run. The user is prompted with "Overwrite D:\baru\Adobe Photoshop CS.Ink {Yes/No/All}? a". The command lists various files being copied from the desktop to the D:\baru\ directory, including "Adobe Photoshop CS.Ink", "CorelDRAW 12.Ink", "DIGITALQURAN.Ink", "Linguist.Ink", "Microsoft Office Excel 2003.Ink", "Microsoft Office PowerPoint 2003.Ink", "Microsoft Office Word 2003.Ink", "nIRC.Ink", "Nero StartSmart.Ink", "Remote Desktop Connection.Ink", "Skype.Ink", and "XAMPP Control Panel.Ink". The output concludes with "13 File(s) copied". The command prompt then returns to the "C:\Documents and Settings\Win xp>" prompt.

31. ketik perintah `d`: untuk masuk ke drive D
32. ketik perintah `dir` untuk melihat directory
33. ketik perintah `cd baru`
34. ketik perintah `dir` untuk melihat isi directory baru

Percobaan kelima: membuat dan menghapus directory (`md`, `rd`, `mkdir`, `rmdir`)

35. ketik perintah `c`: untuk kembali ke folder awal
36. ketik perintah `cd..` kemudian `cd..` lagi untuk berada pada drive C
37. ketik perintah `md baru` untuk membuat directory/folder yang bernama “baru”
38. ketik perintah `dir` untuk melihat isi directory files
39. ketik perintah `rd baru` untuk menghapus directory baru
40. ketik perintah `dir` untuk melihat directory files sekarang
41. ketik perintah `mkdir baru1` untuk membuat directory/folder yang bernama “baru1”
42. ketik perintah `dir` untuk melihat isi dari directory files
43. ketik perintah `rmdir baru1` untuk melihat isi directory files
44. ketik perintah `dir` untuk melihat isi dari directory files
45. ketik perintah `md \komputer` untuk membuat folder bernama komputer didalam drive C
46. ketik perintah `dir` untuk mengecek folder yang telah dibuat
47. ketik perintah `md \files\film` untuk membuat folder bernama film didalam folder files.
48. ketik `cd files` untuk masuk ke dalam directory files kemudian ketik perintah `dir` untuk melihat isi dari directory files.

C:\WINDOWS\system32\cmd.exe

```
09/01/2009  08:14 AM    <DIR>          WINDOWS
08/06/2009  12:00 AM    <DIR>          xampp
05/19/2009  01:58 PM           140  vServer.txt
                           8 File(s)   18,324 bytes
                          13 Dir(s)  62,299,095,040 bytes free

C:>md \komputer
C:>md \files\film
C:>cd files
C:\files>dir
 Volume in drive C is systemXp
 Volume Serial Number is 0027-C4A9

 Directory of C:\files

09/01/2009  04:14 PM    <DIR>          .
09/01/2009  04:14 PM    <DIR>          film
09/01/2009  04:14 PM           0  bytes
                           0 File(s)   0 bytes
                          3 Dir(s)  62,299,095,040 bytes free

C:\files>
```

49. ketik perintah `exit` untuk keluar dari command prompt
50. buka windows explorer untuk melihat hasilnya.

Percobaan keenam : memindahkan file dan menampilkan isi folder secara terstruktur (move, tree)

51. Ulangi langkah 1 dan 5 sampai berapa pada drive c.
52. buka windows explorer kemudian masuk ke `C:\files\baru`. Arahkan kursor pada daerah yang kosong kemudian klik kanan - new - micorosoft word document. Beri nama dengan “`pindah.doc`”
53. Arahkan kursor pada daerah yang kosong lagi kemudian klik kanan - new - text document. Beri nama dengan “`pindah.txt`”
54. kembali ke command prompt
55. ketik perintah `move c:\files\baru\pindah.doc c:\files2` untuk memindah file `pindah.doc` ke folder `files2`
56. ketik perintah `move c:\files\baru\pindah.txt c:\files2` untuk memindah file `pindah.txt` ke folder `files2`
57. ketik perintah `cd files2` untuk masuk ke directory `files2`
58. ketik perintah `dir` untuk melihat isi dari directory

59. sekarang buka windows explorer untuk melihat hasilnya
60. kembali ke command prompt
61. ketik perintah `cd..` untuk keluar dari directory files2
62. ketik `tree c:\files` untuk melihat isi folder files secara terstruktur
63. ketik `tree c:\windows` untuk melihat seluruh isi folder windows secara terstruktur

The screenshot shows a Windows Command Prompt window titled "cmd C:\WINDOWS\system32\cmd.exe". The window displays the output of the "tree" command, which lists the directory structure of the C:\Windows\ directory. The structure includes sub-directories like "CatRoot2", "Com", "config", "dhcpc", "Directx", "Dinput", "drivers", "Avg", "disdn", "etc", and "DRUSTORE". Under "config", there are "systemprofile" and "Start Menu" sub-folders. "Start Menu" contains "Desktop", "Favorites", "My Documents", and "Programs". "Programs" contains "Accessories", which further contains "Accessibility" and "Entertainment". The output ends with a file entry: "965g_0A955058A1827C9EAF48AAB47287245C6F5G49C".

Tugas

- Buatlah sebuah directory/folder yang bernama multimedia di drive D
- Copykan 1 buah file musik (.mp3), gambar (.jpg), document (.doc), teks (.txt) ke dalam folder multimedia
- Tuliskan langkah-langkahnya ke dalam laporan praktikum

©Muhammadiyah university Press

Modul 5

Pengenalan Sistem Operasi Linux

Tujuan

1. Mengenal sistem operasi Linux
2. Memahami proses login/logout pada sistem operasi Linux
3. Memahami perintah-perintah dasar Linux
4. Memahami teknik mencari bantuan pada sistem operasi Linux.

Pendahuluan

Sebuah komputer dapat dioperasikan apabila dalam komputer tersebut terdapat sistem operasi yang kompatibel. Saat ini banyak sistem operasi yang dapat berjalan di PC seperti DOS, Windows, Linux, dan Unix.

Pada awal perkembangannya sekitar akhir 1960 sistem operasi masih belum memiliki arsitektur sistem yang standard, sampai pada awal tahun 1969 lahir sistem operasi yang bernama UNICS (Uniplexed Information and Computer System) yang dikembangkan oleh Ken Thompson seorang peneliti di BELL Lab. Ia membangun sistem operasi dengan bahasa pemrograman assembly pada komputer PDP7 yang disebut UNICS. Model dari UNICS ini masih mengikuti pola pada sistem operasi UNIX yaitu pada model perintah pendek dari UNIX, ls, cp, rm, mv, dll.

Pada tahun 1973 Ken Thompson bersama Dennis Ritchie seorang pembuat C compiler, mengembangkan sistem

operasi UNIX kernel dengan bahasa pemrograman C, dan pada tahun 1974 lahirlah sistem operasi UNIX generasi ke-5. pada tahun 1978 UNIX kembali dirilis untuk edisi yang ke-7. Pada generasi ke-7 ini UNIX pecah menjadi dua cabang yaitu SYSV (System 5) dan BSD (Berkeley Software Distribution). SYSV dikembangkan oleh BELL AT&T Lab sebagai paket sistem operasi UNIX komersial, sedangkan BSD dikembangkan oleh Ken Thompson bersama dengan para mahasiswanya dalam sebuah penelitian di universitas Berkeley. Generasi baru dari SYSV adalah SYSVR4(System 5 Release 4). Pada dasar antara SYSV dan BSD memiliki kesamaan yaitu keduanya sama berasal dari satu rumpun UNIX namun dari keduanya terdapat beberapa sedikit perbedaan seperti pada tabel 1 berikut.

<u>Feature</u>	<u>Typical SYSV</u>	<u>Typical BSD</u>
kernel name	/unix	/vmunix
boot init	/etc/rc.d directories	/etc/rc.* files
mounted FS	/etc/mnttab	/etc/mtab
default shell	sh, ksh	csh, tcsh
FS block size	512 bytes->2K	4K->8K
print subsystem	lp, lpstat, cancel	lpr, lpq, lprm
echo command (no new line)	echo "\c"	echo -n
ps command	ps -fae	ps -aux
multiple wait syscalls	poll	select
memory access syscalls	memset, memcpy	bzero, bcopy

Linux merupakan kombinasi turunan dan SYSV dan BSD, pada tahun 1991 Linus B Torvalds, seorang mahasiswa di universitas Helsinki mengembangkan sebuah sistem operasi bernama Linux. Linux mengadopsi standard spesifikasi UNIX yaitu POSIX. (Portable Operating System Interface). Linux dirilis dalam bentuk open source, dimana sifat alamiah open source Linux adalah semua kode kernel Linux didistribusikan dan dapat ditambah atau dimodifikasi oleh siapa saja.

Arsitektur Sistem Operasi Linux

Seperti halnya semua sistem operasi pada umumnya Linux memiliki konsep sama dalam hal arsitektur. Perbedaanya adalah bentuk di setiap komponen pada arsitektur tersebut. Komponen pada arsitektur sistem operasi Linux adalah sebagai berikut :

1. Kernel

Kernel adalah bagian terpenting dari sebuah sistem operasi. Pada sistem operasi kernel berfungsi untuk mengendalikan semua perangkat keras yang ada di PC seperti CPU(Central Processing Unit), Graphic Cards, Sound Card, Perangkat USB, Harddisk, dll. Selain itu juga kernel berfungsi untuk menjembatani antara aplikasi yang berjalan dengan perangkat keras yang diakses oleh aplikasi tersebut. Sebagai contoh aplikasi text editor (vi,notepad,emacs) membutuhkan resources layar monitor sebagai tampilan bagi penggunanya, dalam kasus ini maka aplikasi tersebut akan dihubungkan oleh kernel melalui device driver VGA Card untuk mengeluarkan data menuju layar monitor. Selain dari aplikasi kernel juga menjembatai perintah yang ditulis oleh pengguna melalui shell atau GUI. Kernel Linux merupakan turunan dari Unix BSD dan SYSV, dan model dari kernel Linux mengacu pada spesifikasi POSIX(Portable Operating System IX).

Sistem kerja dari kernel adalah ketika PC melakukan booting pertama kali maka file biner dari kernel akan dipindah menuju memory dan akan menjalankan semua fungsi manajemen resources pada PC tersebut. Setiap PC memiliki jenis perangkat keras yang berbeda-beda, sehingga kernel akan menginisialisasi perangkat tersebut dengan module device driver yang berbeda. File biner kernel linux dapat dilihat pada direktori “/boot/vmlinuz” sedangkan source kode dapat dilihat di “/usr/src/linux”. Sampai saat ini (Agustus 2009) kernel linux stable sudah keluar dengan versi 2.6.30.5. Untuk

lebih detail mengenai perkembangan generasi kernel dapat dilihat pada situs <http://www.kernel.org/>.

2. Shells/GUI

Shell merupakan antarmuka linux kernel dengan penggunanya. Di dalam shell pengguna dapat menuliskan perintah yang kemudian akan dieksekusi langsung oleh kernel. Perintah-perintah yang dapat dituliskan merupakan perintah yang berhubungan dengan pengelolaan sistem operasi, misalnya melihat isi file yang tersimpan dalam sebuah direktori, membuat file baru, atau melihat isi dari sebuah file. Selain operasi file shell juga dapat digunakan untuk operasi I/O atau operasi penanganan pengguna.

Pada sistem operasi linux ada dua jenis shell atau penghubung antara pengguna dengan sistem operasi yang pertama adalah text-based shell dan yang kedua adalah GUI(Graphic User Interface). Beda dari keduanya adalah cara penyajiannya kepada pengguna, jenis text-based adalah antarmuka paling sederhana dimana pengguna hanya dapat berinteraksi melalui keyboard saja, sedangkan GUI memungkinkan pengguna berinteraksi dengan sistem operasi melalui keyboard saja, mouse dan perangkat lainnya dapat digunakan. Selain itu tampilan pada GUI sudah lebih modern yaitu berbasiskan tampilan grafis.

3. Utilitas sistem

Setiap sistem operasi pada dasarnya mengacu pada sebuah spesifikasi tertentu seperti halnya Unix pada POSIX.2 dimana pada spesifikasi ini dijelaskan konsep implementasi utilitas sistem untuk membantu pengguna berinteraksi dengan sistem operasi.

Utilitas sistem dalam sebuah sistem operasi berfungsi sebagai pengaturan sumber daya yang sudah diinisialisasi oleh sistem operasi pada saat proses booting awal dilakukan. Fungsi monitoring, fungsi pengaktifan dan penon-aktifan

sebuah service dalam sistem operasi merupakan pekerjaan yang dapat dilakukan dengan menggunakan utilitas sistem yang ada dalam sebuah sistem operasi seperti Linux. Beberapa contoh utilitas sistem dalam Linux adalah ls, man, fdisk, grep, awk, sed, cp, mv, more, dan lain sebagainya. Seperti halnya dalam sistem operasi Windows beberapa utilitas misalnya control panel, dir, cd, dan lain-lain.

Selain itu juga utilitas sistem juga termasuk program yang bersifat server, dimana program ini akan berjalan secara background (tidak tampak), seperti misalnya utilitas untuk menangani koneksi dari jarak jauh dengan telnet atau ssh, kedua layanan ini merupakan utilitas sistem yang bersifat server dan dijalankan secara background. contoh lain dari server program dalam Linux adalah lpd untuk mengangani proses printer, httpd sebagai web server, crond untuk menjalankan tugas administratif sistem operasi secara reguler. Utilitas sistem yang bersifat server dan dijalankan secara backgroung dalam Linux disebut DAEMON(Disk And Execution MONitor). daemon diaktifkan pada saat proses booting sistem operasi, dan akan berjalan secara terus menerus. Program ini akan memantau event yang terjadi dan apabila sesuai dengan yang dikehendaki maka daemon akan menjalankan sesuatu. Sebagai contoh program daemon telnet server. Daemon ini akan melakukan pemantauan pada semua koneksi yang datang dari luar PC melalui jaringan pada nomer port 23. jika ada koneksi dari luar yang menghubungi nomer tersebut maka telnet server akan menerima koneksi tersebut dan melayani proses yang akan dilakukan.

4. Aplikasi

Aplikasi merupakan level tertinggi pada sistem operasi. Dimana pada applikasi ini pengguna sistem operasi dapat menggunakan applikasi berdasarkan kebutuhannya. Misalnya aplikasi untuk menulis, pada sistem operasi Linux dapat dilakukan dengan menggunakan bantuan aplikasi

Open Office, sedangkan untuk melakukan editing gambar dapat menggunakan GIMP. Karena aplikasi merupakan level tertinggi dari sebuah sistem operasi maka setiap aplikasi dibentuk dengan cara berbeda untuk masing-masing sistem operasi dan hanya bisa berjalan pada sistem operasi tempat dimana aplikasi tersebut dibangun. Misalnya MS Office tidak dapat berjalan pada sistem operasi Linux, kecuali pembangun MS Office mengeluarkan distribusi paket MS Office for Linux.

Distribusi Linux yang beredar saat ini, selain kernel juga menyertakan paket-paket applikasi yang dapat dimanfaatkan oleh pengguna, namun tidak semua aplikasi disertakan, tergantung sifat dari aplikasi tersebut komersial atau open source. Sementara ini yang bersifat open source saja yang disertakan. Contoh aplikasi open source GCC, G++, Xfix, Latex, dll.

Memulai bekerja dengan Linux

Untuk memulai bekerja pada sistem operasi Linux hal yang paling awal dilakukan adalah proses login. Pada proses ini Linux akan meminta pengguna memasukkan username dan password. Jika berhasil maka pengguna telah berada dalam lingkungan sistem operasi Linux.

Login dan Logout

Ada dua jenis login yang dapat dilakukan dalam Linux. Yang pertama berbasis text atau lebih dikenal dengan istilah TTY, yang kedua berbasis GUI. Pada login berbasis text user akan diminta memasukkan username dan password seperti berikut ini :

 Login : <username> tekan ‘enter’
 Password : <password>

Setelah berhasil memasukkan dengan benar username dan password maka akan muncul prompt di monitor dengan ciri terdapat karakter ‘\$’.

```
mrbee@froodo:~$
```

Arti dari tulisan tersebut adalah “mrbee” merupakan username sedangkan @ merupakan simbol tempat atau lokasi dan “froodo” merupakan nama host atau komputer, sehingga jika diartikan user “mrbee” login pada host bernama “froodo”.

Setelah proses login dilalui maka proses selanjutnya adalah bergantung pada keinginan pengguna. Aturan mainnya sangat sederhana hanya cukup menuliskan perintah-perintah yang diterima oleh Linux kemudian tekan ‘enter’, jika perintah yang dituliskan benar maka sistem akan merespon keluaran perintah tersebut. Sebagai contoh :

```
mrbee@froodo:~$ls 'enter'
```

Perintah ‘ls’ akan menampilkan seluruh isi dari direktori saat itu. Setelah selesai memainkan perintah yang bisa dijalankan dalam lingkungan sistem operasi Linux, jika diinginkan untuk keluar dari shell maka perintah yang dapat digunakan adalah “logout” atau bisa juga dengan tekan “ctrl-d”. Jika perintah ini benar maka sistem akan keluar dan tampilan monitor akan menjadi :

Login :

Maka secara umum operasi perintah dalam Linux dapat disimpulkan ke dalam bentuk umum

```
$<perintah> <option> "enter"
```

Petunjuk praktikum

1. Nyalakan komputer dan pilih system operasi Linux yang tersedia
2. Tunggu proses booting selesai yaitu pada saat keluar permintaan untuk memasukkan username dan password. Masukkan username kemudian tekan enter.
3. Buka Applications - Accessories - Terminal. Untuk menggunakan command line.
4. Jika menggunakan ubuntu login root dengan menggunakan “sudo su” kemudian ‘enter’, setelah itu masukkan password milik user admin (tanyakan kepada asisten praktikum).

Setelah proses selesai dan berada dalam shell, tuliskan perintah-perintah berikut ini. Perintah harus dijalankan kemudian analisis atau maknai respon yang muncul pada layar monitor. Contoh :

Daftar perintah :

echo halo dunia 'enter'
date 'enter'
hostname 'enter'
arch 'enter'
uname -a 'enter'
dmesg more 'enter' (tekan 'q' untuk keluar)
uptime 'enter'
whoami 'enter'
who 'enter'
id 'enter'
last 'enter'
finger 'enter'

```
w 'enter'  
top 'enter' (tekan 'q' untuk keluar)  
echo $SHELL 'enter'  
echo {con,pre}{sent,fer}{s,ed} 'enter'  
man ls 'enter' (tekan 'q' untuk keluar)  
man who 'enter' (tekan 'q' untuk keluar)  
who can tell me about linux 'enter'  
last 'enter'  
clear 'enter'  
fdisk -l 'enter'  
users 'enter'  
cat /etc/fstab 'enter'  
cal 2000 'enter'  
cal 9 1752 'enter'  
cal 10 2007 'enter'  
bc -l 'enter' (tulis "quit" untuk  
mengakhiri)  
echo 5+4 | bc -l 'enter'  
yes please 'enter' (tekan 'ctrl-c' untuk  
keluar)  
pwd 'enter'  
history 'enter'  
tail -f /var/log/message 'enter' (tekan  
ctrl-c untuk keluar)  
lsmod 'enter'  
ps -axu 'enter'  
lspci 'enter'  
free 'enter'  
cat /proc/cpuinfo 'enter'  
finger root 'enter'  
reboot 'enter'
```

```
halt 'enter'
```

5. Catat versi kernel yang digunakan saat itu dengan menggunakan perintah “uname -a” kemudian tekan ‘enter’.

Tugas

1. Jelaskan distro linux yang ada saat ini (minimal 5).
2. Jelaskan 20 perintah yang sama di antara masing-masing distro.
3. Jelaskan maksud perintah ‘init 0’, ‘init 1’, ‘init 2’, ‘init 3’, ‘init 4’, ‘init 5’, dan ‘init 6’.
4. Jelaskan maksud dari perintah ‘quota’

Modul 6

Instalasi Sistem Operasi Linux, dan Menambah Aplikasi Linux

Pendahuluan

Linux Ubuntu 10.04 LTS memiliki dua varian yang dapat digunakan sebagai desktop dan server. Perbedaan yang paling menonjol dari varian Ubuntu 10.04 LTS desktop dan server adalah pada versi desktop memiliki GUI (grafik user interface). Ubuntu Desktop memiliki antar muka yang menarik untuk para pengguna. Sedangkan pada versi Ubuntu LTS server tidak memiliki GUI, sehingga antar muka bagi pengguna hanya berupa shell atau layar terminal.

Alasan yang mendasar mengapa Ubuntu 10.04 LTS tidak memiliki GUI yaitu dikarenakan sebuah GUI membutuhkan memori pemroses yang cukup besar dan dapat menyita atau mengurangi kemampuan fungsi komputer. Dengan tidak adanya GUI pada komputer server, diharapkan kegiatan pemrosesan menjadi lebih cepat. Dalam melakukan instalasi sistem operasi linux ubuntu 10.04 dibutuhkan beberapa komponen agar kegiatan instalasi dapat berjalan sukses. Komponen yang mendasar dalam persiapan instalasi Ubuntu 10.04 LTS Desktop adalah dengan mempersiapkan spesifikasi komputer yang memadai. Ubuntu 10.04 LTS Desktop membutuhkan spesifikasi komputer sebagai berikut:

1. Hardisk 10 GB (ATA, SATA/SCSCI)
2. Processor Pentium 4
3. Memori 512 MB

4. VGA (16 bit, 32 bit, 64 bit)
5. Mouse dan Keyboard (USB/PS2)
6. Monitor (CRT/LCD)
7. CD/DVD ROM
8. CD Ubuntu 10.04 LTS Desktop

Untuk mendapatkan CD Ubuntu 10.04 LTS Desktop dapat mengakses di internet dengan memilih folder Ubuntu dan mendownload file bertipe .iso dengan nama Ubuntu_10.04_LTS/Desktop.iso dengan ukuran file sekitar 600 MB, atau langsung mengunjungi situs resmi www.ubuntu.com. Cara yang lebih mudah untuk mendapat CD Ubuntu yaitu dengan membeli pada toko online yang ada di internet.

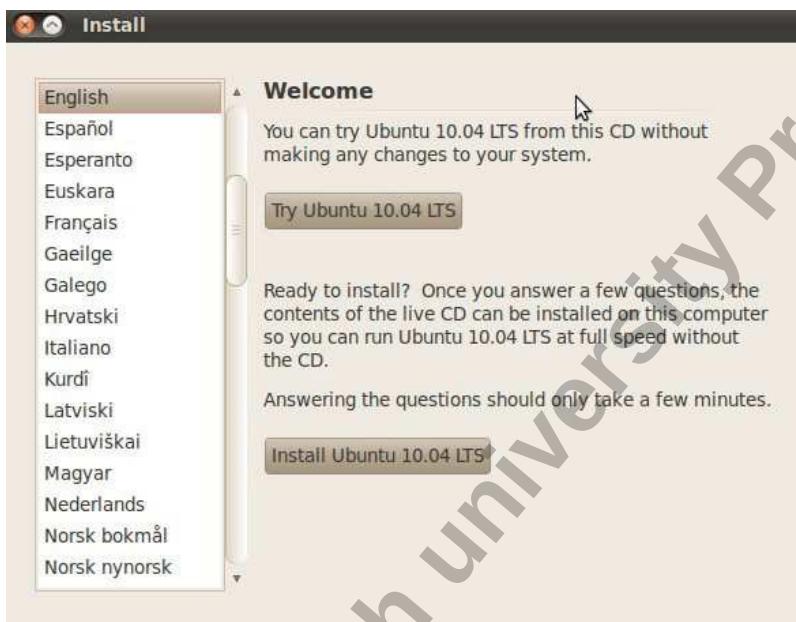
Petunjuk Praktikum

Melakukan instalasi Ubuntu 10.04 LTS Desktop dapat dimulai dengan mengubah struktur booting pada BIOS, dengan menjadikan CD/DVD ROM sebagai perangkat keras utama yang akan dibaca ketika pertama kali komputer dijalankan. Komputer akan membaca CD Ubuntu 10.04 LTS Desktop kemudian melakukan proses pembacaan awal seperti yang ditunjukkan pada gambar 6.1.



Gambar 6.1 Proses awal pembacaan CD Ubuntu

Komputer membutuhkan beberapa menit untuk melakukan proses pembacaan CD Ubuntu 10.04 LTS Desktop.



Gambar 6.2 Gambar 2.2 Live CD dan Install

Proses selanjutnya adalah live cd dan install, seperti yang ditunjukkan pada gambar 6.2 dimana terdapat dua pilihan. Pilihan pertama dimaksutkan untuk penggunaan Ubuntu 10.04 LTS Desktop dengan live cd. Istilah live cd artinya pengguna dapat menjalankan sistem operasi Ubuntu 10.04 LTS Desktop dengan menggunakan cd installer. Pilihan yang kedua adalah pengguna dapat melanjutkan instalasi Ubuntu 10.04 LTS Desktop secara permanen pada PC (personal computer). Lanjutkan dengan menekan tombol Install Ubuntu 10.04 LTS.



Gambar 6.3 Pemilihan Wilayah dan Waktu

Tahapan selanjutnya yaitu pemilihan wilayah instalasi dan waktu yang akan digunakan. Untuk memilih wilayah dan waktu dapat menggunakan mouse dan mengarahkan pada peta kemudian menandainya.



Gambar 6.4 Pemilihan Keyboard

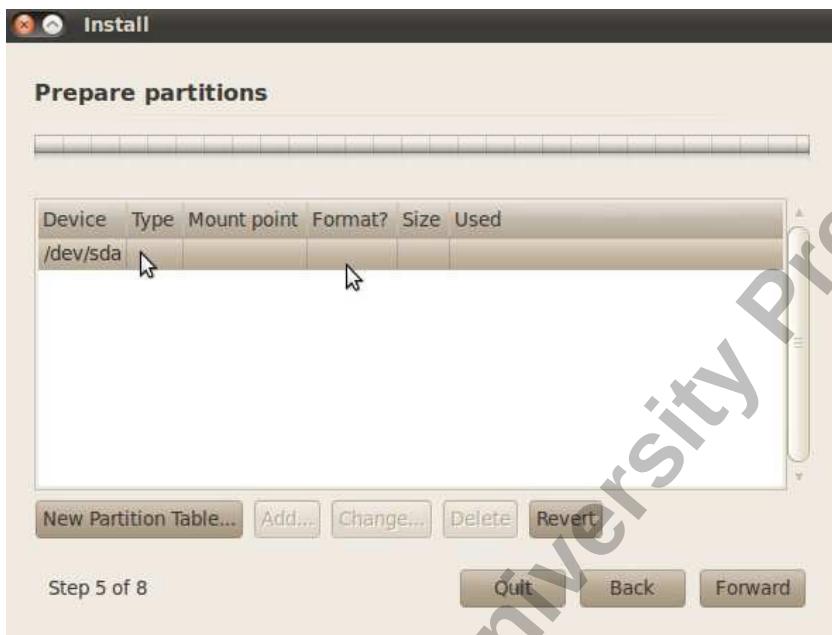
Halaman selanjutnya adalah pemilihan keyboard yang akan digunakan, untuk wilayah Indonesia disarankan menggunakan layout USA. Lanjutkan dengan menekan tombol Forward.



Gambar 6.5 Partisi Hard Disk Manual

Bagian terpenting dari instalasi Ubuntu 10.04 LTS Desktop adalah pada bagian partisi, bagi para pemula ini adalah tahapan yang menyulitkan. Untuk proses pembelajaran diharapkan untuk memilih bagian Specify partitions manually (advanced), lanjutkan dengan menekan tombol Forward. Pembagian hard disk secara manual untuk Ubuntu 10.04 LTS Desktop seperti dijelaskan pada contoh pembagian hard disk berikut ini.

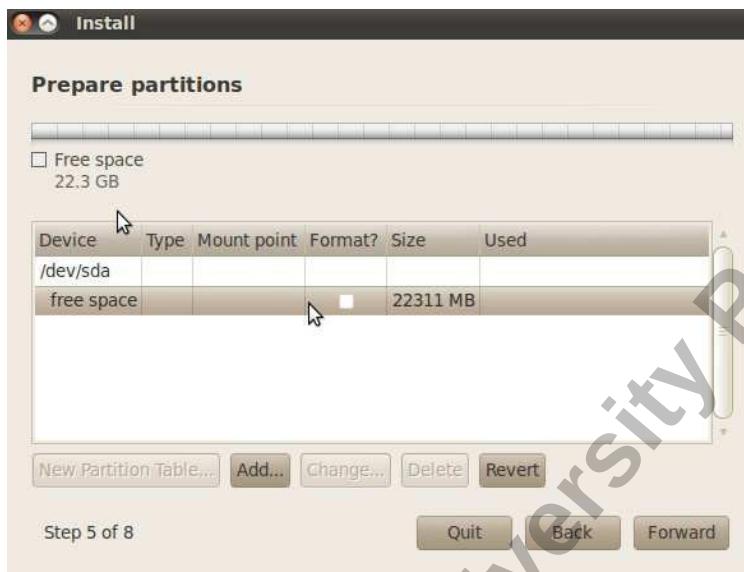
- kapasitas hard disk yang dimiliki 22 GB
- **swap** 1 GB (1024 MB) dengan perhitungan jumlah swap = 2 x RAM
- file sistem **/root** = 21 GB (21504 MB)



Gambar 6.6 Persiapan Pembagian Hard Disk

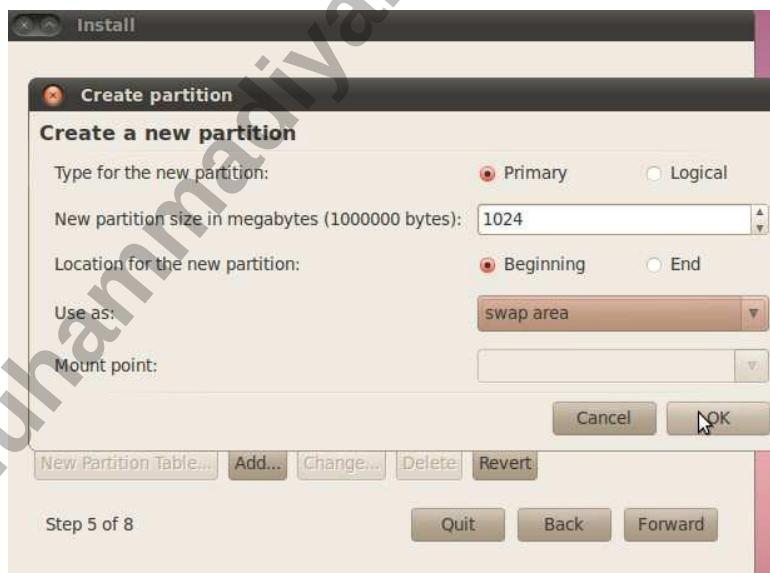
Perhatikan gambar 6.6, halaman tersebut menampilkan perangkat keras dengan nama **/dev/sda**. Penamaan hard disk dalam sistem operasi linux berbeda dengan sistem operasi lain seperti windows. Hard disk yang diberinama /dev/sda artinya hard disk tersebut diletakkan pada direktori /dev dan bernama sda. Semua perangkat keras dalam sistem operasi linux akan tersimpan pada direktori /dev. Makna dari sda adalah hard disk bertipe SATA. Jika hard disk bertipe ATA maka penamaan hard disk akan menjadi hda.

Arahkan kursor mouse pada /dev/sda kemudian lanjutkan dengan menekan tombol **New Partition Table**.



Gambar 6.7 Format Hard Disk

Memulai pembagian partisi dengan mengarahkan kursor pada **free space** lanjutkan dengan menekan tombol **Add**.



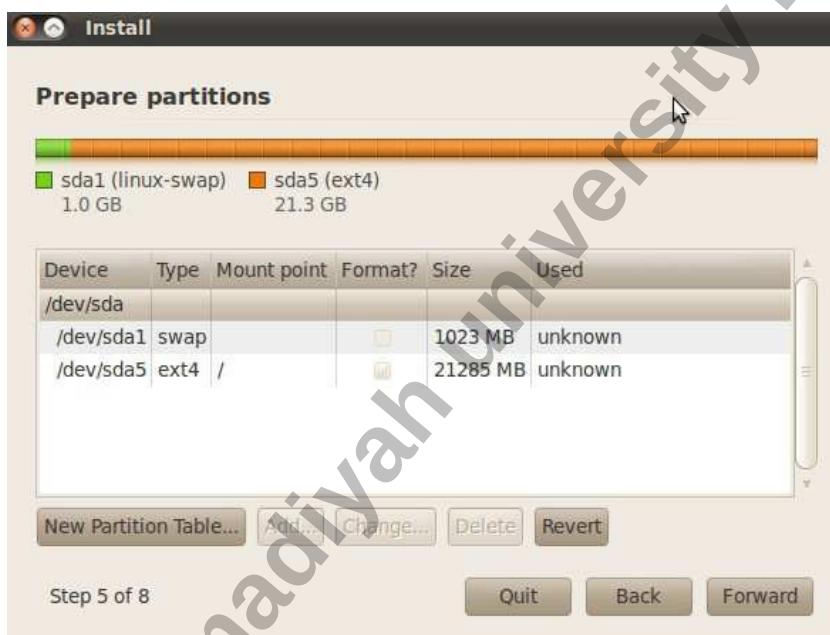
Gambar 6.8 Membuat Swap

Swap adalah memori virtual yang digunakan sistem operasi linux untuk menyimpan proses. Memori swap atau sering disebut swap tanpa kata memori, tidak berbeda dengan memori utama yang sering disebut dengan RAM. Ukuran swap biasanya didapat dari rumus ($\text{swap} = 2 \times \text{RAM}$), jika $\text{RAM} = 512 \text{ MB}$ maka ukuran swap : $2 \times 512 = 1024 \text{ MB}/1 \text{ GB}$. Swap digunakan linux apabila proses yang dilakukan sangat berat sehingga membutuhkan memori tambahan. Bagaimana bila RAM yang dimiliki sebesar 8 GB, haruskah membuat swap sebesar 16 GB sesuai dengan rumus yang telah dibuat diatas. Ada beberapa pendapat mengatakan batas ukuran swap adalah 2 GB, sehingga bila ukuran RAM sangat besar ada kemungkinan linux tidak membutuhkan swap. Tekan OK untuk melanjutkan tahap selanjutnya.



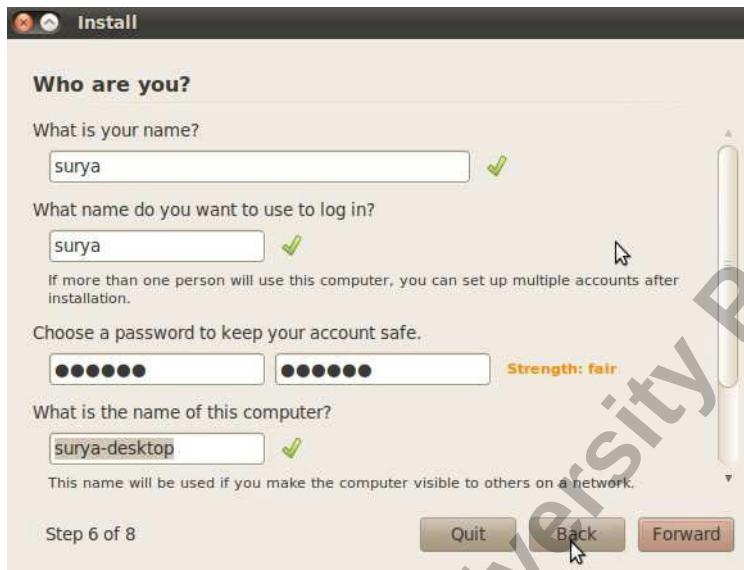
Gambar 6.9 Membuat File System

Proses pembagian hard disk yang terakhir adalah menjadikan sisa ruang hard disk sebagai file system. Pada gambar 2.9 dapat dilihat ukuran sebesar **21287 MB** dengan tipe file **Ext4 journaling file system**. Perhatikan juga pada bagian Mount point terdapat simbol “/” yang mengindikasikan sebagai tempat penyimpanan sistem operasi linux Ubuntu 10.04 LTS Desktop. Lanjutkan dengan menekan tombol OK.



Gambar 6.10 Format Hard Disk

Saat ini telah terbentuk dua bagian pada hard disk yang ditunjukkan pada gambar 6.10 di atas, terlihat dua tipe file yaitu **/dev/sda1 swap** dan **/dev/sda5 ext4**. Lanjutkan dengan menekan tombol Forward.



Gambar 6.11 Membuat User Sistem

Proses selanjutnya adalah membuat user sistem yang akan digunakan untuk login pertama kali pada sistem operasi linux. Ada tiga user yang diterapkan pada sistem operasi linux khususnya pada Ubuntu 10.04 LTS Desktop yaitu:

1. user sistem: memiliki kemampuan untuk merubah sistem konfigurasi secara menyeluruh
2. user administrator atau root: memiliki kemampuan penuh terhadap sistem linux, dengan kemampuan mengendalikan semua user dan konfigurasi sistem
3. user biasa: hanya memiliki kemampuan untuk mengelola desktop milik sendiri, tanpa ada kemampuan untuk merubah sistem konfigurasi.

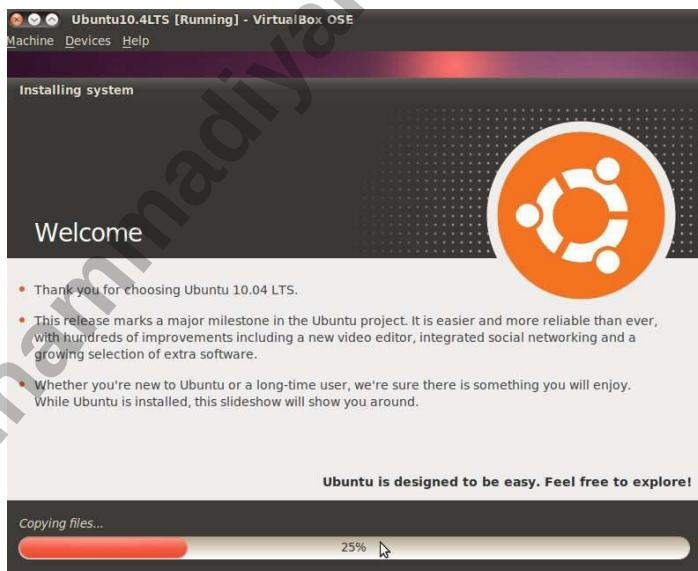
Pada contoh di atas nama user adalah surya, user name login surya dan password login 6 karakter. Batasan panjang karakter adalah minimal 6 karakter.

Pada gambar 6.12 menunjukkan halaman konfirmasi, dimana isi dari halaman tersebut berupa hasil keseluruhan

dari konfigurasi yang telah dilakukan. Lanjutkan dengan menekan tombol Install.



Gambar 6.12 Halaman Konfirmasi



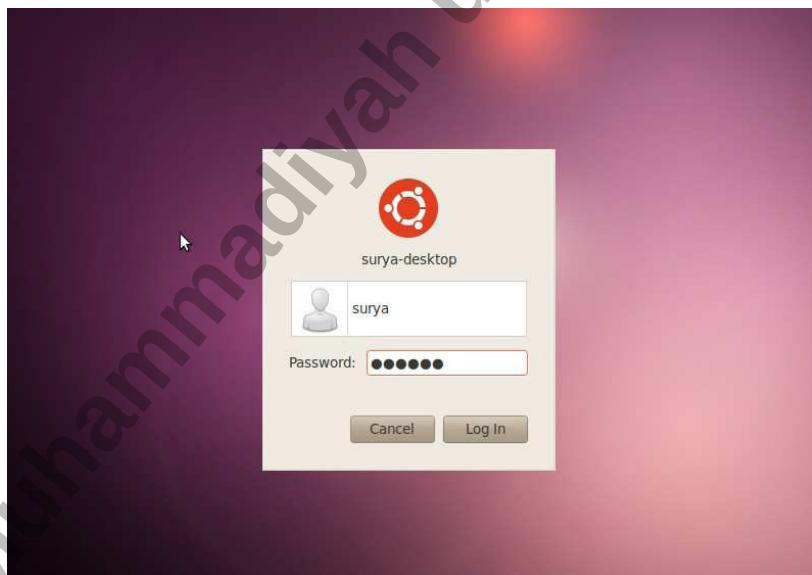
Gambar 6.13 Install Sistem

Proses install sistem dilakukan untuk menuliskan semua berkas dan membuat direktori pada /dev/sda5. Butuh waktu 10 sampa 30 menit, bergantung pada spesifikasi komputer yang digunakan.



Gambar 6.14 Install Sukses dan Restart Komputer

Tekan tombol Restart Now, kemudian cd install Ubuntu 10.04 LTS Desktop akan keluar dari CD ROM drive.



Gambar 6.15 Halaman Login Utama

Selesai sudah proses installasi Ubuntu 10.04 LTS Desktop. Tekan nama user dan masukkan password, kemudian tekan tombol Log In.

Menambah aplikasi

Kemampuan Ubuntu 10.04 LTS Desktop salah satunya adalah kemudahan dalam penambahan aplikasi yang disediakan oleh pengembang Ubuntu yaitu perusahaan Canonical Ltd asal Afrika yang didirikan oleh Mark Shuttleworth pada tahun 2004. Nama Ubuntu diambil dari ideologi di Afrika Selatan, kata Ubuntu berasal dari bahasa kuno Afrika yang artinya “rasa perikemanusiaan terhadap sesama manusia”.

Untuk menambah aplikasi pada Ubuntu 10.04 LTS Desktop, dapat menggunakan perangkat bantu yakni **Ubuntu Software Center** yang terletak pada menu Applications. Tetapi harus menjalankan perintah **sudo apt-get update** pada terminal.



Tugas

Menggunakan laptop masing-masing instalkan 10 aplikasi yang berbeda pada linux, dan sebutkan fungsinya.

No	Aplikasi	Tampilan	Fungsi
1			
2			
3			
4			
5			
6			
7			

Modul 7

Managemen User dan Permission Direktori/File pada Linux

Tujuan

- Peserta Praktikum Operasi dapat melakukan manajemen user dan quota.
- Peserta Praktikum Operasi dapat mengenal permission dan ownership file/direktori
- Mengetahui permission dan ownership pada tiap objek file atau direktori pada Linux

Pendahuluan

Manajemen User

1. Membuat User Baru

Membuat sebuah account user yang baru sangatlah mudah. Untuk membuat user baru dari command line, kita dapat menjalankan perintah `useradd`. Sebagai contoh, untuk membuat sebuah user yang bernama Hana, kita dapat menjalankan perintah sbb:

```
$ useradd hana
```

Untuk memberikan password kepada user baru yang telah dibuat dengan mengetikkan perintah `passwd`. Contoh:

```
$ passwd hana
```

Setelah mengetikkan perintah di atas, maka kita akan diminta untuk memasukkan password yang kita inginkan dua kali. Ingat, untuk mengganti password user jangan sampai lupa menambahkan nama user.

Dalam default system, ketika user baru ditambahkan, maka user baru tersebut akan mempunyai home directory di bawah /home. Sebagai contoh, ketika kita mengeksekusi perintah di atas (`useradd hana`), maka user yang bernama hana akan memiliki home directory `/home/hana`.

Untuk mengubah default system ini, kita dapat memberikan option-option lain setelah perintah `useradd`. Misalnya kita menginginkan agar si hana diberikan home directory di `/www/hana`, maka kita dapat memberikan option `-d` setelah perintah `useradd`. Contoh:

```
$ useradd -d /www/hana
```

Sebagai catatan, `useradd` hanya membuat direktori akhir, bukan semua path. Sehingga bila kita memberikan perintah `useradd -d /www/hana`, maka `useradd` akan menciptakan direktori `hana` apabila telah ada direktori yg bernama `/www`. Program `useradd` akan menciptakan suatu group bagi setiap user yg ditambahkan. Setiap user akan tergabung dalam group-group tertentu. Program `useradd` akan memberikan nama group yang sama seperti nama user yang dibuat. Misalnya ketika kita membuat user baru dengan perintah `useradd hana`, maka si hana akan berada dalam group `hana`. Untuk mengubah default ini, maka kita dapat memberikan parameter `-g` setelah perintah `useradd`. Contoh:

```
$ useradd hana -g users
```

Hal tersebut akan membuat user `hana` tergabung dalam group `users`.

2. Membuat group baru

Untuk membuat group baru kita dapat mengetikkan perintah groupadd. Contoh:

```
$ groupadd hacker
```

Perintah ini akan menambah group baru yang bernama hacker di /etc/group dan pada /etc/gshadow jika kita menggunakan shadow password.

3. Memodifikasi account user yang sudah ada

a. Mengubah password

Seorang system administrator akan sering berhubungan dengan masalah user yang lupa akan passwordnya. Perintah di bawah ini akan mengganti password user yang lama dengan password yang baru:

```
$ passwd
```

Misalkan si hana lupa akan passwordnya atau ingin mengubah passwordnya, maka si hana atau system administrator yang bersangkutan dapat mengubahnya dengan mengetikkan perintah:

```
$ passwd hana
```

b. Mengubah home directory

Untuk mengubah home directory dari user yang sudah ada, kita dapat mengetikkan perintah: usermod -d

Contoh:

```
$ usermod -d /home2/hana hana
```

Jika ternyata home directory hana yang lama telah berisi file-file kepunyaan hana, maka kita dapat memindahkan home directory hana beserta file-filenya dengan option -m. Contoh:

```
$ usermod -d -m /home2/hana hana
```

c. Menghapus account user

Untuk menghapus account user yang sudah ada, kita dapat menggunakan perintah userdel. Contoh:

```
$ userdel hana
```

Perintah tersebut akan menghapus user yang bernama hana. Jika kita ingin menghapus home directory user beserta isi file atau direktori yang berada di bawahnya, maka kita dapat memberikan option -r setelah userdel.

Mengenal Permission direktori/file pada Linux

Pada unix (juga Linux), setiap file dan direktori dapat ditentukan hak aksesnya. Hak yang dapat ditentukan : read (r=4), write (w=2), dan execute (x=1). Anda dapat menentukan hak akses setiap direktori dan file sesuai dengan yang diinginkan dengan menggunakan perintah chmod. Melihat informasi yang dimiliki oleh file dan/atau direktori dapat menggunakan perintah :

```
$ ls -l
```

Berdasarkan tampilan yang dihasilkan, bagian kolom pertama akan terbagi dalam 10 sub kolom.

- Sub kolom 1 menyatakan tipe file. - adalah file, d adalah direktori, l adalah link atau shortcut ke file/direktori.

- Sub kolom 2,3,4 mewakili hak akses rwx untuk pemilik file/direktori.
- Sub kolom 5,6,7 mewakili hak akses rwx untuk group/kelompok pemilik file/direktori.
- Sub kolom 8,9,10 mewakili hak akses rwx untuk semuanya (everyone) selain pengguna dan group.
- Pemilik dari file/direktori dilihat pada kolom tiga dan group dilihat pada kolom empat.

Petunjuk praktikum

1. Nyalakan komputer dan pilih system operasi Linux yang tersedia
2. Tunggu proses booting selesai yaitu pada saat keluar permintaan untuk memasukkan username dan password. Masukkan username kemudian tekan enter.
3. Buka Applications – Accessories – Terminal. Untuk menggunakan command line.
4. Jika menggunakan ubuntu login root dengan menggunakan “sudo su” kemudian ‘enter’, setelah itu masukkan password milik user admin.(tanyakan kepada asisten praktikum)
5. Setelah proses selesai dan berada dalam shell, tuliskan perintah-perintah berikut ini. Perintah harus dijalankan kemudian analisis atau maknai respon yang muncul pada layar monitor. Contoh :

Daftar perintah :

Praktikum 1:

Kerjakan perintah berikut:

1. Buat user baru: # useradd user<nim anda>
2. Masukkan pasword yang diminta sebanyak dua kali:

Enter new UNIX password:

Retype new UNIX password:

3. Jika sudah berhasil akan muncul pesan

passwd: password updated successfully

Dilanjutkan dengan memasukkan user information untuk user yang anda buat, masukkanlah data-data yang diminta.

Contoh tampilan:

```
Changing the user information for
user<nim anda> Enter the new value, or
press ENTER for the default
```

Full Name []:

Room Number []:

Work Phone []:

Home Phone []:

Other []:

Is the information correct? [Y/n] Y

Masukkanlah data-data anda.

4. Keluarlah dari shell dengan mengetikkan exit kemudian tutup jendela shell
5. Keluarlah dari gnome dengan mengklik tombol Log Out
6. Setelah masuk ke jendela login, cobalah untuk masuk menggunakan user dan pasword yang baru saja anda buat.

Setelah berhasil login, lanjutkan dengan langkah praktikum selanjutnya. Apa bila belum berhasil, anda harus mengulangi langkah paraktikum ini terlebih dahulu.

Praktikum 2:

Kerjakan perintah berikut :

1. Buat file latihan : \$ touch latihan
2. Ketikkan perintah : \$ chmod 666 latihan
3. Lihat hasilnya dengan : \$ ls -l (perhatikan izin aksesnya)
4. Ketikkan perintah : \$ chmod 640 latihan
5. Lihat hasilnya dengan : \$ ls -l (perhatikan izin aksesnya)
6. Ketikkan perintah : \$ chmod 111 latihan
7. Lihat hasilnya dengan : \$ ls -l (perhatikan izin aksesnya)
8. Ketikkan perintah : \$ chmod 222 latihan
9. Lihat hasilnya dengan : \$ ls -l (perhatikan izin aksesnya)
10. Ketikkan perintah : \$ chmod 333 latihan

Angka konfigurasi yang dapat dikombinasikan adalah angka 0-7.

Praktikum 3:

Hak akses juga dapat ditambahkan, selain diubah seperti cara di atas. Untuk melakukan perubahan hak akses, kita menggunakan operand +x, +r, +w,-r, -w, dan -x.

Kerjakan perintah berikut :

1. \$ chmod 000 latihan (tidak memberikan hak akses)
2. \$ ls -l latihan
3. \$ chmod +r latihan (menambahkan hak akses read)

4. \$ ls -l latihan
5. \$ chmod +w latihan (menambahkan hak akses write)
6. \$ ls -l latihan
7. \$ chmod +x latihan (menambahkan hak akses execute)
8. \$ ls -l latihan
9. \$ chmod -x latihan (menghilangkan hak akses execute)
10. \$ ls -l latihan
11. \$ chmod -w latihan (menghilangkan hak akses write)
12. \$ ls -l latihan
13. \$ chmod -r latihan (menghilangkan hak akses read)
14. \$ ls -l latihan

Perintah +operand/-operand pada chmod akan mengubah hak akses sesuai dengan default set. Kita akan membuat file skrip shell untuk dieksekusi di dalam direktori /bin. Disini kita akan melakukan perubahan hak akses agar file tersebut bisa dieksekusi.

Praktikum 4

Jalankan perintah berikut :

1. Masuklah ke dalam direktori /bin : \$ cd /bin
2. Buat file bernama info.sh : \$ vi info<nim anda>.sh
3. Ketik isinya (jangan lupa menekan tombol i dulu) sebagai berikut (perhatikan besar/kecil hurufnya) :

```
#!/bin/sh
WAKTU="Tanggal dan jam saat ini : \c"
JMLUSER="Jumlah user : \c" AKU="Status
personal : \c"
echo -e "$WAKTU" date
echo -e "$JMLUSER" who | wc -l
echo -e "$AKU" whoami
exit 0
```

4. Simpan dan keluar dengan menekan ESC, dan mengetik :wq <ENTER>
5. Ketikkan perintah loginfo.sh : \$ info<nim anda>.sh
6. Ubah tipenya menjadi file yang bisa dieksekusi : \$ chmod 777 info<nim anda>.sh
7. Ketikkan perintah info.sh : \$ info<nim anda>.sh
8. Apa hasilnya?

©Muhammadiyah university Press

Modul 8

System Call

Tujuan

1. Membuat sebuah ‘child process’ baru menggunakan system call ‘fork’.
2. Menghentikan sementara (block) proses parent sampai dengan proses child selesai, menggunakan perintah system call ‘wait’.
3. Loading sebuah program yang dapat dieksekusi dalam sebuah child proses dengan menggunakan perintah system call ‘exec’.
4. Menampilkan status sebuah file menggunakan system call ‘stat’.
5. Untuk menampilkan isi direktori menggunakan perintah system call ‘readdir’.

Pendahuluan

Dalam bidang komputer, istilah ‘system call’ adalah satu cara bagaimana sebuah aplikasi meminta layanan dari kernel (bagian utama) sistem operasi. Beberapa contoh antara lain layanan yang terkait dengan akses hardware seperti membaca hard disk drive, membuat dan mengeksekusi proses baru, dan berkomunikasi dengan layanan terpadu kernel seperti ‘process scheduling’. System call menyediakan antarmuka yang sangat penting antara proses dan sistem operasi. Beberapa fungsi penting dalam system call yang akan dibahas dalam praktikum ini antara lain adalah fork(), getpid(), getppid(), wait(), execl(), exit(), stat(), opendir(),

readdir(), dan closedir().

Keterangan untuk setiap fungsi system call diberikan dalam bagian berikut ini:

fork()

System call fork ini digunakan untuk membuat sebuah proses baru disebut ‘child process’.

(a) return value berupa angka NOL (0) jika proses merupakan ‘child process’, (b) return value berupa bilangan NEGATIF jika pemutuan proses baru TIDAK berhasil. (c) Jika proses berupa ‘parent process’ maka return value berupa sebuah angka positif (> 0). ‘Child process’ adalah sebuah proses tiruan yang sama persis dengan ‘Parent process’. Kedua jenis proses, ‘parent’ dan ‘child’, akan mengeksekusi secara terus menerus semua perintah yang diberikan setelah pemanggilan fungsi fork. ‘child’ dapat memulai eksekusi sebelum proses ‘parent’ dan sebaliknya.

getpid() dan getppid()

System call ‘getpid’ akan memberikan sebuah return value berupa proses ID dari proses yang dipanggil. Sedangkan ‘getppid’ akan memberikan sebuah return value berupa ‘parent’ proses ID dari proses yang dipanggil.

wait()

System call ‘wait’ jika dipanggil akan menyebabkan ‘parent’ proses akan diblok (dihentikan) sampai dengan ‘child’ proses berakhir. Ketika proses ‘child’ berakhir, kernel akan memberitahu kepada proses ‘parent’ dengan cara mengirimkan sinyal ‘SIGCHLD’ kepada proses parent. Jika wait tidak digunakan maka ‘parent’ proses akan berakhir lebih dulu dan akan meninggalkan sebuah ‘sampah’ (berasal dari child proses).

exec1()

Merupakan salah satu anggota kelompok fungsi ‘exec’, anggota yang lainnya adalah execv, execle, execve, execlp, execvp. Fungsi ini digunakan oleh ‘child’ proses untuk ‘loading’ sebuah program dan mengeksekusinya. System call ‘exec1’ memerlukan ‘path’, nama program dan sebuah pointer null.

exit()

System call ‘exit’ digunakan untuk menghentikan sebuah proses, baik secara dihentikan secara normal atau tidak normal.

stat()

System call stat digunakan untuk memperoleh informasi mengenai sebuah file dan strukturnya.

opendir() , readdir() dan closedir()

System call ‘opendir’ digunakan untuk membuka sebuah direktori, (a) akan menghasilkan sebuah POINTER yang menunjuk ke posisi direktori yang pertama. (b) Pointer akan bernilai NULL jika terjadi sebuah kesalahan (error).

System call ‘readdir’ digunakan untuk membaca direktori sebagai struktur ‘dirent’ (a) Akan menghasilkan sebuah pointer yang menunjuk pada entry direktori selanjutnya. (b) Pointer akan bernilai NULL jika terjadi error atau bertemu dengan tanda ‘END-OF- FILE’

System call ‘closedir’ digunakan untuk menutup direktori hanya dilakukan oleh kernel.

Peralatan

- Software: Sistem operasi Linux, dilengkapi dengan kompiler gcc, dan editor text (nano atau vi)

- Hardware: PC desktop/laptop
- koneksi internet (untuk akses compiler-online)

Langkah kerja

Gunakan aplikasi ‘nano’ atau ‘vi’ atau teks editor yang lain untuk mengedit kode program berikut, Selanjutnya untuk melakukan kompilasi dapat dilakukan dengan perintah berikut:

```
$ gcc 'nama_file.c'
```

Jika tidak ada kesalahan maka akan dihasilkan sebuah program bernama ‘a.out’, dan untuk menjalankan program tersebut dapat dilakukan dengan cara berikut:

```
$ ./a.out
```

Jika pada PC anda tidak tersedia compiler ‘gcc’ dapat digunakan fasilitas online compiler yang disediakan oleh link berikut:

http://www.tutorialspoint.com/compile_c_online.php

1. Membuat sebuah ‘child process’ (proses baru) dengan menggunakan system call ‘fork’.

Membuat program dengan algoritma sebagai berikut: (contoh program diberikan pada bagian berikutnya).

- a. Deklarasi sebuah variabel x yang akan diakses bersama antara child proses dan parent proses.
- b. Membuat sebuah child proses menggunakan system call fork.
- c. Jika return value bernilai -1, tampilkan teks ‘Pembuatan proses GAGAL’, dilanjutkan dengan keluar program dengan perintah system call ‘exit’.

- d. Jika return value sama dengan 0 (NOL), Tampilkan teks ‘Child Process’, tampilkan ID proses dari child proses menggunakan perintah system call ‘getpid’, tampilkan nilai x, dan tampilkan ID proses parent dengan perintah system call ‘getppid’.
- e. Untuk nilai return value yang lainnya, tampilkan teks ‘Parent process’, tampilkan ID dari parent proses menggunakan perintah system call getpid, tampilkan nilai x, dan tampilkan ID dari proses shell menggunakan perintah system call getppid.
- f. Stop

Kode program:

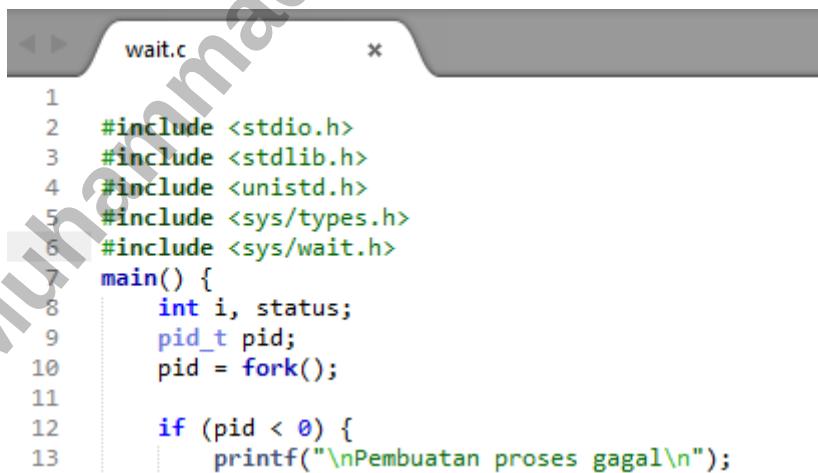
```
fork.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 main()
6 {
7     pid_t pid;
8     int x = 5;
9     pid = fork();
10    x++;
11    if (pid < 0)
12    {
13        printf("Process creation error"); exit(-1);
14    }
15    else if (pid == 0)
16    {
17        printf("Child process:");
18        printf("\nProcess id is %d", getpid());
19        printf("\nValue of x is %d", x);
20        printf("\nProcess id of parent is %d\n\n", getppid());
21    }
22    else
23    {
24        printf("\nParent process:");
25        printf("\nProcess id is %d", getpid());
26        printf("\nValue of x is %d", x);
27        printf("\nProcess id of shell is %d\n", getppid());
28    }
}
```

2. Menghentikan sementara (block) proses parent sampai dengan proses child selesai, menggunakan perintah system call ‘wait’.

Membuat program dengan algoritma sebagai berikut, contoh program diberikan pada bagian berikutnya.

- Membuat sebuah child proses menggunakan system call ‘fork’.
- Jika return value bernilai -1, selanjutnya tampilkan teks ‘pembuatan proses gagal’, dan keluar program dengan menggunakan perintah system call ‘exit’.
- Jika return value berupa angka positif (> 0), ‘pause’ hentikan sementara ‘parent’ proses tunggu sampai child proses berakhir dengan menggunakan perintah system call ‘wait’. Tampilkan teks ‘Parent starts’, selanjutnya tampilkan nomor genap mulai dari 0 s/d 10, terakhir tampilkan teks ‘Parent end’.
- Jika return value bernilai 0 (NOL), tampilkan teks ‘Child start’, tampilkan nomor ganjil mulai dari 0 s/d 10, selanjutnya tampilkan teks ‘child ends’
- Stop

Kode program:



```
wait.c *  
1  
2 #include <stdio.h>  
3 #include <stdlib.h>  
4 #include <unistd.h>  
5 #include <sys/types.h>  
6 #include <sys/wait.h>  
7 main() {  
8     int i, status;  
9     pid_t pid;  
10    pid = fork();  
11  
12    if (pid < 0) {  
13        printf("\nPembuatan proses gagal\n");  
14    } else {  
15        if (pid == 0) {  
16            // Child process logic  
17            // Example: Print odd numbers from 0 to 10  
18            for (i = 1; i <= 10; i++) {  
19                if (i % 2 != 0) {  
20                    printf("Child start: %d\n", i);  
21                    sleep(1);  
22                    printf("Child ends: %d\n", i);  
23                }  
24            }  
25        } else {  
26            // Parent process logic  
27            // Example: Print even numbers from 0 to 10  
28            for (i = 0; i <= 10; i++) {  
29                if (i % 2 == 0) {  
30                    printf("Parent starts: %d\n", i);  
31                    sleep(1);  
32                    printf("Parent ends: %d\n", i);  
33                }  
34            }  
35        }  
36    }  
37}
```

```

14         exit(-1);
15     }
16     else if(pid > 0)
17     {
18         wait(NULL);
19         printf ("\nParent starts\nNomor Genap:");
20         for (i=2;i<=10;i+=2)
21             printf ("%d",i);
22         printf("\nParent ends\n");
23     }
24     else if (pid == 0)
25     {
26         printf ("Child starts\nNomor Ganjil:");
27         for (i=1;i<10;i+=2)
28             printf ("%d",i);
29         printf ("\nChild ends\n");
30     }
31 }
```

Contoh eksekusi program di atas secara online:

The screenshot shows a web-based C compiler and executor interface. The code in the editor is:

```

1 #include <cs50.h>
2 #include <cslib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/wait.h>
6
7 main()
8 {
9     int i, status;
10    pid_t pid;
11    pid = fork();
12    if (pid == 0)
13        printf("\nPembuatan proses gagal\n");
14        exit(-1);
15    else if(pid > 0)
16    {
17 }
```

The terminal output shows the execution and results:

```

sh-4.3$ gcc -o main *.c
sh-4.3$ ./main
Child starts
Nomor Genap: 1 3 5 7 9
Child ends
Parent starts
Nomor Ganjil: 2 4 6 8 10
Parent ends
sh-4.3$
```

3. Loading program yang dapat dieksekusi dalam sebuah ‘child’ proses menggunakan perintah system call ‘exec’. Membuat program dengan algoritma sebagai berikut: (contoh program diberikan pada bagian berikutnya).

- a. Jika terdapat 3 argumen dalam command-line berhenti (stop).
- b. Membuat child proses dengan perintah system call ‘fork’
- c. Jika return value adalah -1, selanjutnya tampilkan teks ‘Pembuatan proses Gagal’, dan keluar program dengan perintah system call exit.
- d. Jika return value >0 (positif), selanjutnya hentikan parent-proses sementara hingga child-proses berakhir dengan menggunakan perintah system call wait. Tampilkan teks ‘Child berakhir’, dan hentikan parent-proses.
- e. Jika return value sama dengan 0 (NOL), selanjutnya tampilkan teks ‘Child starts’, load program dari lokasi yang diberikan dalam ‘path’ ke dalam child-proses, menggunakan perintah system call ‘exec’. Jika return value dari perintah ‘exec’ adalah bilangan negatif, tampilkan error yang terjadi dan stop. Hentikan child- proses.
- f. Stop

Contoh cara mengkompilasi dan menjalankan program

```
$ gcc exec.c  
$ ./a.out /bin/ls ls
```

Kode Program:

```
exec.c x
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 main(int argc, char*argv[])
6 {
7     pid_t pid;
8     int i;
9
10    if (argc != 3)
11    {
12        printf("\nInsufficient arguments to load program");
13        printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
14    }
15
16    switch(pid = fork())
17    {
18        case -1:
19            printf("Fork failed");
20            exit(-1);
21        case 0:
22            printf("Child process\n");
23            i = execl(argv[1], argv[2], 0);
24            if (i < 0)
25            {
26                printf("%s program not loaded using exec system call\n", argv[2]);
27                exit(-1);
28            }
29        default:
30            wait(NULL);
31            printf("Child Terminated\n");
32            exit(0);
33    }
34 }
```

4. Menampilkan status file menggunakan perintah system call ‘stat’

Membuat program dengan algoritma sebagai berikut (contoh code ada di bagian berikutnya):

- Gunakan ‘nama file’ yang diberikan melalui argumen dalam perintah command-line.
- Jika ‘nama-file’ tidak ada maka stop disini (keluar program)
- Panggil system call ‘stat’ pada ‘nama-file’ tersebut yang akan mengembalikan sebuah struktur.
- Tampilkan informasi mengenai st_uid, st_blksize, st_block, st_size, st_nlink, etc.

- e. Ubah waktu dalam st_time, st_mtime dengan menggunakan fungsi ctime.
- f. Bandingkan st_mode dengan konstanta mode seperti S_IRUSR, S_IWGRP, S_IXOTH dan tampilkan informasi mengenai ‘file-permissions’.
- g. Stop

Kode Program:

```

stat.c * 

1 #include <stdio.h>
2 #include <sys/stat.h>
3 #include <stdlib.h>
4 #include <time.h>
5 int main(int argc, char*argv[])
6 {
7     struct stat
8     file; int n;
9     if (argc != 2)
10    {
11        printf("Usage: ./a.out <filename>\n"); exit(-1);
12    }
13    if ((n = stat(argv[1], &file)) == -1)
14    {
15        perror(argv[1]);
16        exit(-1);
17    }
18    printf("User id : %d\n", file.st_uid);
19    printf("Group id : %d\n", file.st_gid);
20    printf("Block size : %d\n", file.st_blksize);
21    printf("Blocks allocated : %d\n", file.st_blocks);
22    printf("Inode no. : %d\n", file.st_ino);
23    printf("Last accessed : %s", ctime(&(file.st_atime)));
24    printf("Last modified : %s", ctime(&(file.st_mtime)));
25    printf("File size : %d bytes\n", file.st_size);
26    printf("No. of links : %d\n", file.st_nlink);
27    printf("Permissions : ");
28    printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
29    printf( (file.st_mode & S_IRUSR) ? "r" : "-");
30    printf( (file.st_mode & S_IWUSR) ? "w" : "-");
31    printf( (file.st_mode & S_IXUSR) ? "x" : "-");
32    printf( (file.st_mode & S_IRGRP) ? "r" : "-");
33    printf( (file.st_mode & S_IWGRP) ? "w" : "-");
34    printf( (file.st_mode & S_IXGRP) ? "x" : "-");
35    printf( (file.st_mode & S_IROTH) ? "r" : "-");
36    printf( (file.st_mode & S_IWOTH) ? "w" : "-");
37    printf( (file.st_mode & S_IXOTH) ? "x" : "-");
38    printf("\n");
39    if(file.st_mode & S_IFREG)
40        printf("File type : Regular\n");
41    if(file.st_mode & S_IFDIR)
42        printf("File type : Directory\n");
}

```

5. Menampilkan isi direktori menggunakan perintah system call ‘readdir’

Membuat program dengan algoritma sebagai berikut (contoh code ada di bagian berikutnya):

- a. Gunakan ‘nama-direktori’ yang diberikan sebagai argumen pada command-line.
- b. Jika direktori tidak ditemukan stop, keluar program
- c. Buka direktori menggunakan perintah system call ‘opendir’ yang akan menghasilkan sebuah struktur.
- d. Baca direktori menggunakan perintah system call ‘readdir’ yang juga akan menghasilkan struktur data.
- e. Tampilkan d_name (nama direktori)
- f. Akhiri pembacaan direktori dengan perintah system call ‘closedir’.
- g. Stop

Kode Program:

```
dirlist.c
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <stdlib.h>
4 main(int argc, char *argv[]){
5     struct dirent *dptr;
6     DIR *dname;
7
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <dirname>\n");
11         exit(-1);
12     }
13     if((dname = opendir(argv[1])) == NULL)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     while(dptr=readdir(dname))
19         printf("%s\n", dptr->d_name);
20
21     closedir(dname);
22 }
```

©Muhammadiyah university Press

Modul 9

File System Call

Tujuan

1. Membuat program untuk membuat sebuah file dan menuliskan isinya.
2. Membaca file dan menampilkan isinya
3. Menambahkan isi sebuah file

Pendahuluan

Terdapat beberapa system call yang berkaitan dengan pembuatan dan pembacaan file, yaitu open, creat, read, write, dan close. Pada bagian ini akan dijelaskan cara membuat dan membaca file dengan menggunakan perintah yang tersedia dalam dalam file-system-call. Penjelasan dari perintah-perintah tersebut adalah sebagai berikut.

open()

Digunakan untuk membuat file yang sudah dibuat sebelumnya (ada dalam sistem) untuk dibaca atau ditulis atau dapat juga digunakan untuk membuat sebuah file baru. Fungsi ini memiliki return value berupa angka NEGATIF jika terjadi kesalahan. Pilihan (opsi) yang disarankan ketika menggunakan fungsi open antara lain adalah O_RDONLY, O_WRONLY, dan O_RDWR, selain itu juga tersedia opsi O_APPEND, O_CREAT, O_TRUNC dan lain-lain. Gabungan beberapa opsi dapat dilakukan dengan menggunakan operasi digital OR. Sedangkan MODE (pada parameter ketiga) digunakan untuk menentukan hak akses.

creat()

Digunakan untuk membuat file baru dan membuka file tersebut untuk ditulis. Fungsinya dapat digantikan oleh fungsi open() dengan opsi O_WRONLY|O_CREAT|O_TRUNC **read()**

Membaca sejumlah byte tertentu dari sebuah file atau terminal. Jika proses pembacaan berhasil, fungsi tersebut akan memberi informasi (melalui return value) jumlah byte yang berhasil dibaca. Pembacaan berikutnya akan dimulai dari data selanjutnya sampai akhirnya bertemu dengan tanda END-OF-FILE yang menghasilkan return value bernilai 0 (NOL).

write()

Menulis sejumlah byte pada sebuah file. Setelah proses penulisan berhasil, penulisan selanjutnya akan dimulai dari posisi berikutnya. Jika memory tidak cukup akan menghasilkan error.

close()

Mentup status file yang terbuka (sedang dibuka oleh fungsi creat atau open).

Peralatan

- Software: Sistem operasi Linux, dilengkapi dengan kompiler gcc, dan editor text (nano atau vi)
- Hardware: PC desktop/laptop
- koneksi internet (untuk akses compiler-online)

Langkah Kerja

Membuat sebuah file dan menuliskan data di dalamnya.

Membuat program dengan algoritma sebagai berikut (Contoh program diberikan pada bagian berikutnya):

1. Deklarasi variabel untuk buffer ‘buf’ untuk menyimpan 100 byte data.
2. Gunakan nama file yang diberikan sebagai argumen pada command-line
3. Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O_CREAT dan O_TRUNC.
4. Periksa ‘file-descriptor’, jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.
5. Baca input dari console sampai user menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukan melalui console dan menyimpannya ke dalam variabel ‘buf’ menggunakan perintah system-call read(). Memindahkan isi variabel ‘buf’ ke dalam file menggunakan perintah ‘write’.
6. Menutup file dengan menggunakan fungsi ‘close’
7. Stop

Kode Program:

```
fcreate.c      *
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <fcntl.h>
5
6 main(int argc, char *argv[]) {
7     int fd, n, len;
8     char buf[100];
9     if (argc != 2)
10    {
```

```

11     printf("Usage: ./a.out <filename>\n");
12     exit(-1);
13 }
14 fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0644);
15 if(fd < 0)
16 {
17     printf("File creation problem\n");
18     exit(-1);
19 }
20
21 printf("Tekan Ctrl+D di akhir baris:\n");
22
23 while((n = read(0, buf, sizeof(buf))) > 0)
24 {
25     len = strlen(buf);
26     write(fd, buf, len);
27 }
28
29 close(fd);
30 }
```

Compilasi program dengan perintah \$gcc -o fcreate fcreate.c , selanjutnya jika proses compilasi berhasil jalankan program dengan memanggil nama file \$./fcreate test.txt, ketikan text yang akan disimpan dalam file, tekan Ctrl + D untuk menandai baris baru, dan tekan Ctrl + C jika sudah selesai.

Membaca sebuah file dan menampilkan isinya di layar.

Membuat kode program dengan algoritm sebagai berikut:

1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
2. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.
3. Buka file untuk dibaca menggunakan perintah ‘open’ dengan opsi O_RDONLY.
4. Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop
5. Baca isi file per 100 byte data menggunakan perintah

‘read’ sampai ketemu dengan tanda akhir file, ‘END-OF-FILE’.

6. Tutup file menggunakan perintah ‘close’
7. Stop



```
fread.c *  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 #include <fcntl.h>  
4 main(int argc, char *argv[]) {  
5     int fd,i;  
6     char buf[100];  
7     if (argc < 2)  
8     {  
9         printf("Usage: ./a.out <filename>\n");  
10        exit(-1);  
11    }  
12  
13    fd = open(argv[1], O_RDONLY);  
14    if(fd == -1)  
15    {  
16        printf("%s file does not exist\n", argv[1]);  
17        exit(-1);  
18    }  
19  
20    printf("Isi dari file %s adalah : \n", argv[1]);  
21    while(read(fd, buf, sizeof(buf)) > 0)  
22        printf("%s", buf);  
23  
24    close(fd);  
25 }
```

Menambah isi file.

Membuat kode program dengan algoritma sebagai berikut:

1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
2. Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.

3. Buka file di atas dengan menggunakan perintah open dengan opsi O_APPEND.
4. Periksa file-descriptor, jika nilainya berupa angka negatif, stop program
5. Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel ‘buf’ menggunakan perintah read. Selanjutnya tuliskan isi variabel ‘buf’ ke dalam file menggunakan perintah ‘write’.
6. Tutup file dengan menggunakan fungsi ‘close’.

```
fappend.c
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <fcntl.h>
5 main(int argc, char *argv[])
6 {
7     int fd, n, len;
8     char buf[100];
9     if (argc != 2)
10    {
11        printf("Usage: ./a.out <filename>\n"); exit(-1);
12    }
13    fd = open(argv[1], O_APPEND|O_WRONLY|O_CREAT, 0644);
14    if (fd < 0)
15    {
16        perror(argv[1]);
17        exit(-1);
18    }
19    while((n = read(0, buf, sizeof(buf))) > 0)
20    {
21        len = strlen(buf);
22        write(fd, buf, len);
23    }
24    close(fd);
25 }
```

Hasil

Telah dibuat program untuk membaca, menulis dan menambah isi sebuah file.

Kesimpulan

File System call merupakan kelompok fungsi system-call yang berkaitan dengan proses pembuatan, pembacaan dan penulisan file.

©Muhammadiyah university Press

Modul 10

Simulasi Command (Perintah)

Tujuan

Membuat program untuk mensimulasi beberapa perintah dalam sistem operasi linux menggunakan kelompok fungsi yang tersedia dalam system-call. Beberapa command yang akan disimulasi antara lain adalah

- a. Simulasi perintah ‘ls’
- b. Simulasi perintah ‘grep’
- c. Simulasi perintah ‘cp’
- d. Simulasi perintah ‘rm’

Pendahuluan

Fungsi-fungsi yang tersedia dalam system call dapat digunakan untuk mensimulasikan sebagian besar perintah yang terdapat dalam sistem operasi linux, seperti perintah ‘ls’, ‘grep’, ‘cp’, dan ‘rm’. Mensimulasi perintah dengan seluruh opsi yang disediakan dalam perintah tersebut merupakan pekerjaan yang sangat melelahkan. Tetapi membuat program untuk mensimulasi perintah dapat meningkatkan ketrampilan dalam membuat program dan memanfaatkan fasilitas dalam sistem operasi. Mensimulasi perintah juga dapat membantu dalam meng-customize perintah standar dalam sistem operasi. Secara umum perintah-perintah yang berhubungan dengan file I/O dapat disemulasikan. Pada modul ini akan dibuat program untuk mensimulasi empat buah perintah yaitu ‘ls’, ‘grep’, ‘cp’ dan ‘rm’.

Peralatan

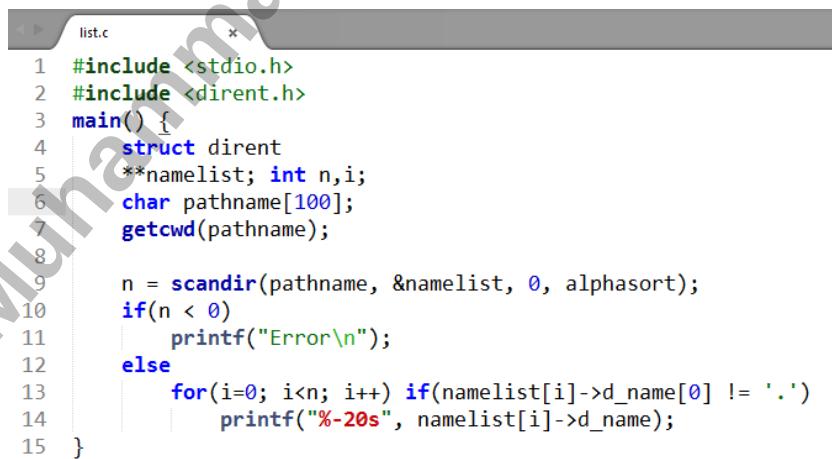
- Software: Sistem operasi Linux, dilengkapi dengan kompiler gcc, dan editor text (nano atau vi)
- Hardware: PC desktop/laptop
- koneksi internet (untuk akses compiler-online)

Langkah Kerja

Program untuk mensimulasikan perintah ‘ls’.

Membuat kode program dengan algoritm sebagai berikut:

1. Menyimpan ‘path’ dari direktori kerja saat ini menggunakan perintah system call ‘getcwd’
2. Membaca isi direktori dari path di atas menggunakan perintah system call ‘scandir’ dan mengurutkan hasil pembacaannya dan menyimpannya dalam sebuah variabel array.
3. Menampilkan nama direktori (dname) dan nama file didalamnya jika file atau direktori tersebut tidak memiliki properti ‘HIDE’.
4. Stop



```
list.c
1 #include <stdio.h>
2 #include <dirent.h>
3 main()
4 {
5     struct dirent
6     **namelist; int n,i;
7     char pathname[100];
8     getcwd(pathname,
9
10    n = scandir(pathname, &namelist, 0, alphasort);
11    if(n < 0)
12        printf("Error\n");
13    else
14        for(i=0; i<n; i++) if(namelist[i]->d_name[0] != '.')
15            printf("%-20s", namelist[i]->d_name); }
```

Program untuk mensimulasi perintah ‘grep’.

Membuat kode program dengan algoritm sebagai barikut:

1. Gunakan nama file yang diberikan dalam argumen command-line
2. Buka file dalam mode ‘read-only’ menggunakan perintah system call ‘open’
3. Jika file tidak ada, keluar program, stop
4. Misal panjang string yang dicari adalah n .
5. Baca file perbaris sampai akhir file (END-OF-FILE), untuk setiap baris lakukan hal-hal berikut: (a) Periksa untuk mencari string dalam baris tersebut dengan dalam range 1-n, 2-n+1, dan seterusnya, (b) Jika string ditemukan tampilan baris tersebut di layar.
6. Tutup file menggunakan perintah ‘close’.
7. Stop



```
mygrep.c *  
1 #include <stdio.h>  
2 #include <string.h>  
3 #include <stdlib.h>  
4 main(int argc,char *argv[]) {  
5     FILE *fd;  
6     char str[100];  
7     char c;  
8     int i, flag, j, m, k;  
9     char temp[30];  
10  
11    if (argc != 3)  
12    {  
13        printf("Usage: gcc mygrep.c -o mygrep\n");  
14        printf("Usage: ./mygrep <search_text> <filename>\n");  
15        exit(-1);  
16    }  
17  
18    fd = fopen(argv[2],"r");  
19    if(fd == NULL)  
20    {  
21        printf("%s is not exist\n",argv[2]);  
22        exit(-1);  
23    }  
24}
```

```

25     while(!feof(fd))
26     {
27         i = 0;
28         while(1)
29         {
30             c = fgetc(fd);
31             if(feof(fd))
32             {
33                 str[i++] = '\0'; break;
34             }
35             if(c == '\n')
36             {
37                 str[i++] = '\0'; break;
38             }
39             str[i++] = c;
40         }
41         if(strlen(str) >= strlen(argv[1]))
42         for(k=0; k<=strlen(str)-strlen(argv[1]); k++)
43         {
44             for(m=0; m<strlen(argv[1]); m++)
45             {
46                 temp[m] = str[k+m];
47                 temp[m] = '\0';
48                 if(strcmp(temp, argv[1]) == 0)
49                 {
50                     printf("%s\n", str);
51                     break;
52                 }
53             }
54         }
55     }

```

Program untuk mensimulasikan perintah ‘cp’.

Membuat kode program dengan algoritm sebagai berikut:

1. Gunakan nama file untuk sumber dan tujuan dari argumen yang diberikan dalam command line.
2. Deklarasi sebuah buffer berukuran 1 KB
3. Buka file sumber dalam mode ‘read-only’ menggunakan fungsi ‘open’
4. Jika file sumber tidak ditemukan, stop keluar dari program
5. Membuat file baru sebagai file target dengan menggunakan perintah ‘creat’.

6. Jika proses pembuatan file gagal, stop keluar dari program.
7. Proses penyalinan (copy) file dilakukan dengan cara berikut: (a) Membaca 1KB data dari file sumber dan menyimpan hasilnya dalam buffer menggunakan perintah ‘read’. (b) Menuliskan isi buffer dalam file target menggunakan perintah ‘write’. (c) Jika bertemu dengan kode ‘END-OF-FILE’ lanjut ke nomor 8, yang lain kembali ke perintah (a)
8. Tutup file sumber dan target menggunakan perintah ‘close’.
9. Stop

Kode Program

```
copy.c      *
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 #include <sys/stat.h>
5 #define SIZE 1024
6 main(int argc, char *argv[])
7 {
8     int src, dst, nread;
9     char buf[SIZE];
10    if (argc != 3)
11    {
12        printf("Usage: gcc copy.c -o copy\n");
13        printf("Usage: ./copy <filename> <newfile> \n");
14        exit(-1);
15    }
16    if ((src = open(argv[1], O_RDONLY)) == -1)
17    {
18        perror(argv[1]);
19        exit(-1);
20    }
21    if ((dst = creat(argv[2], 0644)) == -1)
22    {
23        perror(argv[2]);
24        exit(-1);
25    }
26    while ((nread = read(src, buf, SIZE)) > 0)
27    {
28        write(dst, buf, nread);
29    }
30    close(src);
31    close(dst);
32}
```

```

26     {
27         if (write(dst, buf, nread) == -1)
28         {
29             printf("can't write\n");
30             exit(-1);
31         }
32     }
33     close(src);
34     close(dst);
35 }

```

Program untuk mensimulasikan perintah ‘rm’.

Membuat kode program dengan algoritm sebagai berikut:

1. Gunakan nama file yang diberikan dalam argumen command line
2. Buka file dalam mode ‘read-only’ menggunakan perintah ‘read’
3. Jika file tidak ditemukan, stop keluar program
4. Tutup file menggunakan perintah ‘close’
5. Menghapus file menggunakan perintah ‘unlink’
6. Stop

Kode Program



```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 main(int argc, char* argv[]) {
5     int fd;
6     if (argc != 2)
7     {
8         printf("Usage: gcc del.c -o del\n");
9         printf("Usage: ./del <filename>\n");
10        exit(-1);
11    }
12    fd = open(argv[1], O_RDONLY);
13    if (fd != -1)
14    {
15        close(fd);
16        unlink(argv[1]);
17    } else
18        perror(argv[1]);
19 }

```

Hasil

Telah dibuat dan diperiksa empat program simulasi dengan menggunakan perintah dalam system-call, terdiri atas perintah ‘ls’, ‘grep’, ‘cp’ dan ‘rm’.

Kesimpulan

Perintah-perintah dalam system call, dapat digunakan untuk mensimulasi sebagian besar perintah dalam sistem operasi linux. Namun demikian mensimulasi sebuah perintah dengan semua opsi yang terdapat didalamnya merupakan pekerjaan yang melelahkan.

©Muhammadiyah university Press

Modul 11

Penjadwalan Proses dan Manajemen Memori (OSSim)

Tujuan

Melakukan simulasi untuk memahami proses penjadwalan dan memahami proses manajemen memori suatu sistem operasi

Pendahuluan

Pada praktikum ini akan dikenalkan beberapa hal penting yang terkait dengan algoritma penjadwalan proses dan proses manajemen memori. adapun tool yang akan digunakan yaitu OS Simulator atau disingkat dengan OSSim.



OSSim merupakan aplikasi yang mensimulasikan sistem operasi secara grafis yang berfokus pada empat komponen OS diantaranya penjadwalan proses, manajemen memori, manajemen file sistem dan penjadwalan disk.

Pada modul ini akan dipelajari beberapa macam algoritma penjadwalan. Ada 4 algoritma yang akan dibahas yaitu:

- a. First-Come, First-Served (FCFS)
- b. Shortest-Job-First(SJF)
- c. Priority
- d. Round Robin (RR)

Algoritma penjadwalan terdiri dari preemptive dan non-preemptive. Algoritma non-preemptive dibuat agar jika ada proses yang masuk ke running state, maka proses itu tidak bisa diganggu sampai menyelesaikan prosesnya, sedangkan penjadwalan preemptive menggunakan prioritas dimana jadwal dapat mengganggu proses dengan prioritas rendah ketika proses dengan prioritas tinggi sedang running.

Sedangkan untuk manajemen memori

Peralatan

- Software: Sistem operasi windows, aplikasi OSSim
- Hardware: PC desktop/laptop

Langkah Kerja

Kegiatan 1. First-Come, First-Served (FCFS)

1. Buka program OSSim, selanjutnya pilih menu processes -> process scheduling
2. Selanjutnya pilihlah setting dan pilih algoritma **First-Come, First-Served (FCFS)**

Modul 12

REMASTERING

PENDAHULUAN

Istilah Remastering sendiri sebenarnya adalah proses membuat master baru untuk sebuah album, film, atau ciptaan lainnya dari hasil cipta yang sebelumnya sudah ada. seperti pada proses memindahkan rekaman musik yang berasal dari media analog menjadi rekaman digital (hal ini lebih dikenal pada industri musik dan film), namun seiring dengan waktu istilah remaster tidak saja menjadi milik industri film dan musik, seperti halnya penggunaan istilah Virus Biologis pada bidang medis dan penggunaan istilah Virus Komputer pada industri software. Penggunaan istilah remaster pada linux sendiri mulai dipopulerkan oleh Klaus Knopper sang pencipta Distro Linux LiveCD-Knoppix yang mana Knoppix sendiri merupakan hasil remaster dari Debian.



Gambar Lambang Knoppix Distro Live

Dalam bidang software remastering dapat diartikan sebagai sebuah proses pembungkusan ulang paket aplikasi pada sistem operasi dimana kita bisa menambah bahkan bisa juga mengurangi paket aplikasi yang disertakan. Bisa dikatakan bahwa remastering merupakan proses pembuatan sistem operasi baru dengan paket aplikasi yang berbeda dari sistem aslinya (default). Dengan remastering memungkinkan kita untuk menambah atau mengurangi paket aplikasi di sistem operasi yang ada dengan paket aplikasi yang baru.

Istilah lain yang biasanya digunakan adalah operating system deployment atau slipstream istilah ini biasanya dikenal dilingkungan windows, dimana sebuah Sistem windows yang telah ada ditambahkan driver ataupun software tambahan dibundel kembali menggunakan imaging software atau wim image.

Secara umum dapat diketahui bahwa tujuan dari remastering itu sendiri adalah membuat sebuah sistem operasi yang sesuai dengan kehendak pembuatnya, dalam hal ini bisa bertujuan khusus atau memang ditargetkan digunakan pada lingkungan tertentu. Hampir semua sistem operasi modern yang beredar sekarang seperti Windows XP, Vista, Seven, Ubuntu, Slackware, Debian dan sistem operasi modern lainnya dapat di remaster, tetapi dari sekian banyak Sistem operasi tadi kita tidak bisa sembarangan meremastering sendiri. karena dari sekian banyak Sistem operasi tersebut ada yang memiliki licensi dan peraturan (hak kepemilikan), baik sistem operasi itu sendiri maupun software yang digunakan dalam prosesnya. Salah satu dari sistem operasi tersebut yang dapat diremaster secara bebas tanpa terikat akan license atau diwajibkan membayar adalah sistem operasi yang menggunakan Kernel Linux dalam hal ini Slackware, Debian, Ubuntu, dll. Malah dalam banyak hal kita dapat dengan mudah meremaster sebuah distribusi GNU/Linux dibandingkan sistem operasi lainnya hal ini

dikarenakan tersedianya software bantu dan dokumentasi yang dapat diperoleh secara bebas.

TUJUAN

Mahasiswa mampu membuat sistem operasi baru

PERALATAN

3. Seperangkat PC
4. Sistem operasi opensource
5. Aplikasi remasterimg

©Muhammadiyah university Press



ISBN: 978-602-361-237-6

A standard linear barcode is positioned in the bottom right corner, corresponding to the ISBN number above it. The ISBN number "978-602-361-237-6" is printed below the barcode.