

MODUL KULIAH SISTEM OPERASI

Raissa Amanda Putri, S.Kom, M.TI

Universitas Islam
Negeri Sumatera
Utara Medan

Daftar Isi

1. VMWare dan Windows	4
1.1. VIRTUAL MACHINE.....	4
1.2. WINDOWS 7	4
1.3. INSTALASI SISTEM OPERASI PADA VMWARE	5
1.4. INSTALASI SISTEM OPERASI WINDOWS 7.....	9
1.5. DYNAMIC LINK LIBRARIES.....	12
1.6. TUGAS.....	13
2. INSTALASI DAN KONFIGURASI UBUNTU	14
2.1. INSTALASI UBUNTU.....	14
2.2. DISTRO LINUX	19
2.3. TUGAS.....	22
3. Command Run & Command Prompt	23
3.1. Command Run.....	23
3.2. Command Prompt	24
3.3. Latihan.....	25
4. Proses Penjadwalan di Prosesor	27
4.1. Deskripsi Penjadwalan Proses.....	27
4.2. Tipe Penjadwalan	27
4.3. Strategi penjadwalan	28
4.4. Algoritma Penjadwalan	28
4.5. FIFO (First In First Out).....	29
4.6. SJF (Shortest Job First)	30
4.7. HRN (Higest-ratio next).....	32
4.8. RR (Round robin).....	35
4.9. SRF (Shortest remaining first)	37
4.10. PS (priority schedulling).....	38

5.	SIKRONISASI	42
5.1.	Race Condition.....	43
5.2.	Critical Section.....	45
5.3.	Mutual Exclusion	47
5.4.	Semaphore.....	47
5.5.	Operasi Down.....	48
5.6.	Problem Klasik pada Sinkronisasi.....	48
5.7.	Problem Readers-Writers	48
5.8.	Problem Dining Philosopers.....	49
5.9.	Monitors	51
6.	Keamanan Sistem Operasi.....	53
6.1.	Microsoft Windows 2000.....	53
6.2.	File System Security.....	53
6.3.	User Accounts	54
6.4.	Password Security	55
6.5.	Account Lockout Policy.....	56
6.6.	Security Audits.....	56
6.7.	User Rights Assignment Policy.....	57
6.8.	Additional Security Options.....	58
6.9.	Active Directory.....	59
6.10.	Linux.....	60
6.11.	Boot Loader	60
6.12.	Password.....	60
6.13.	Limiting Use of The 'root' Account.....	61
6.14.	Services.....	61
6.15.	Firewall.....	62
7.	Antivirus Software.....	63

8. Manajemen File	64
8.1. Sasaran Manajemen File.	64
8.2. Fungsi Manajemen File.....	65
8.3. Arsitektur Pengelolaan File	66

1. VMWare dan Windows

1.1. VIRTUAL MACHINE

Virtual Machine (Indonesia : mesin virtual) pada mulanya didefinisikan oleh Gerard J. Popek dan Robert P. Goldberg pada tahun 1974 sebagai *sebuah duplikat yang efisien dan terisolasi dari suatu mesin asli*. Pada masa sekarang ini, virtual machine dapat mensimulasikan perangkat keras walaupun tidak ada perangkat keras aslinya sama sekali.

Virtual machine terdiri dari dua kategori besar, dipisahkan menurut cara penggunaan dan tingkat keterhubungannya dengan mesin-mesin aslinya. Sebuah System Virtual Machine adalah perangkat yang berupa platform sistem yang lengkap dan dapat menjalankan sebuah sistem operasi yang lengkap. Sebaliknya, Process Virtual Machine didesain untuk menjalankan sebuah program komputer tertentu (tunggal), yang berarti mesin virtual ini mendukung proses tertentu juga.

Disini akan dikenalkan bagaimana cara instalasi dan menggunakan suatu System Virtual Machine. Yaitu VMware Workstations 12 Player (VMware). VMware adalah salah satu System Virtual Machine yang dapat menjalankan suatu sistem operasi secara lengkap.

1.2. WINDOWS 7

Windows 7 adalah suatu sistem operasi yang banyak dipakai sampai saat ini karena kemudahan dalam pemakaiannya. Untuk melakukan penginstalan Windows 7 diperlukan ketelitian dan kesabaran dalam prosesnya karena memerlukan waktu yang lumayan lama. Ada beberapa jenis Windows 7 diantaranya Windows 7 Starter, Home Basic, Home Premium, Professional, Ultimate, dan Enterprise.

Penggunaan dan fitur- fitur bawaan dari Windows 7 cukup lengkap dan mudah dioperasikan, hal ini membuat Windows 7 masih tetap digemari dan masih layak konsumsi

bagi masyarakat banyak. Walaupun sudah terbit versi – versi Windows terbaru seperti Windows 8 ataupun Windows 10.

1.3. INSTALASI SISTEM OPERASI PADA VMWARE

1. Install VMware Workstation 12 Player.

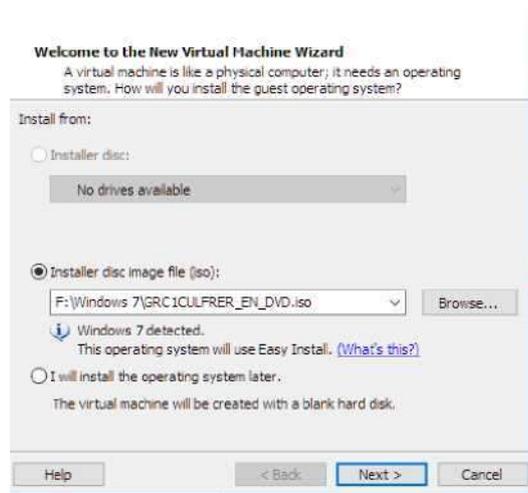


2. Pilih “Use VMware Workstation 12 Plater for free for non – commercial use” dan input alamat email.

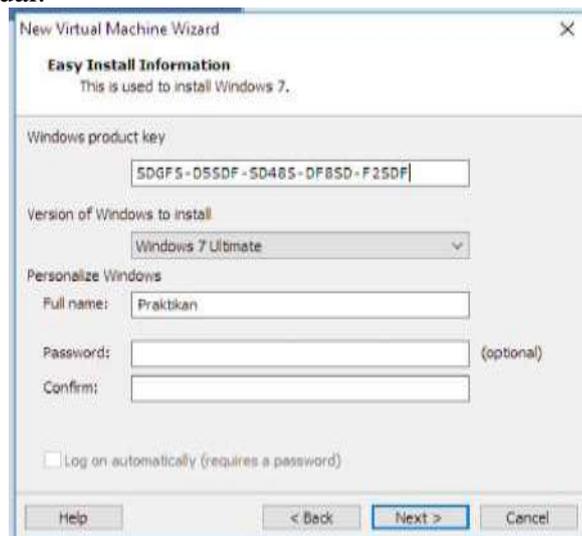


3. Pilih “Create a New Virtual Machine”.
4. Akan muncul windows kecil seperti gambar di bawah. Klik Browse... dan pilih file .iso sistem operasi yang akan diinstal.

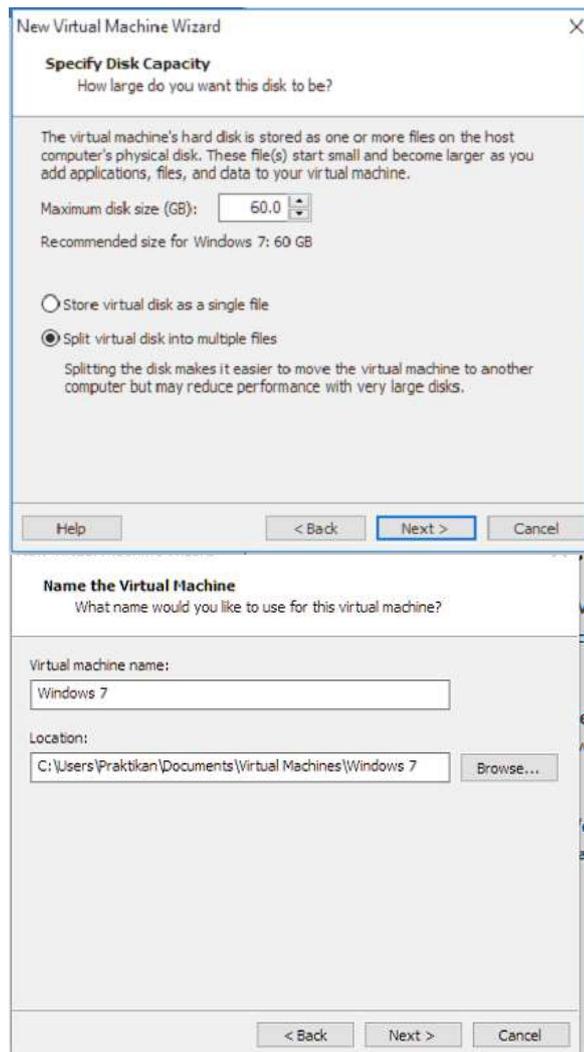
5. Jika berhasil, Sistem Operasi yang akan diinstal akan langsung terdeteksi.



6. Tuliskan serial number Windows dan username yang diinginkan. Dan pilih juga versi windows yang sesuai.



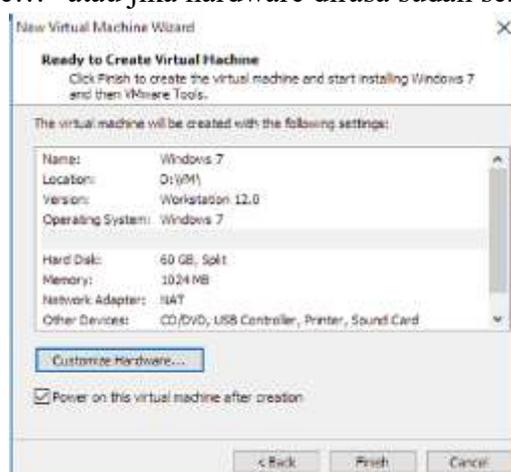
7. Tuliskan Nama Virtual Machine dan lokasi dimana sistem operasi akan di-install.



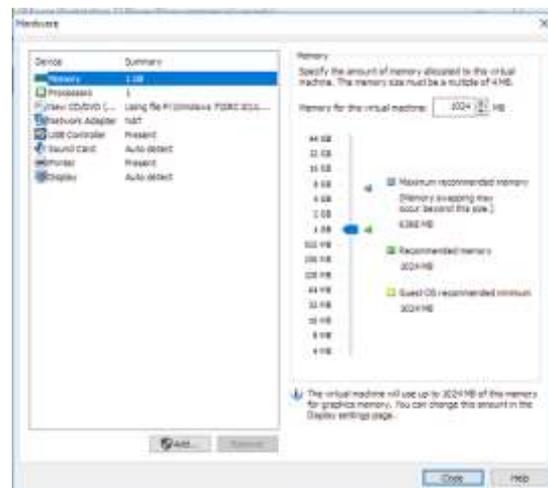
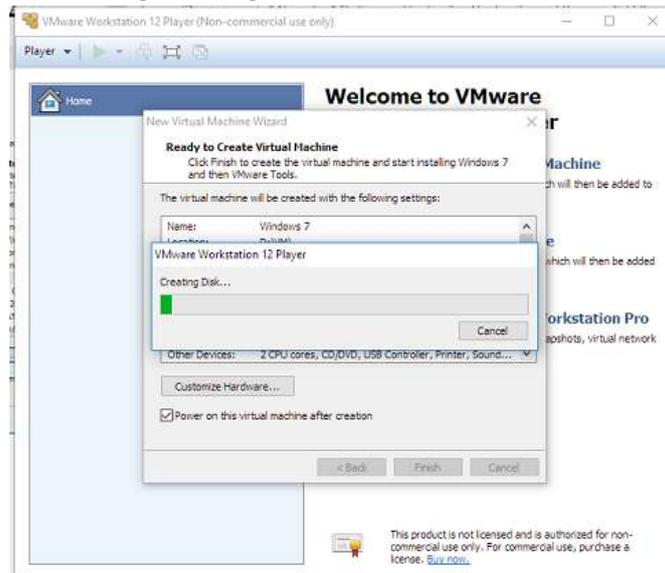
8. Sesuaikan ukuran Hard Disk untuk sistem operasi.

9. Dapat dilakukan perubahan hardware untuk virtual machine dengan men-klik

“Customize Hardware...” atau jika hardware dirasa sudah sesuai.



10. Sesuaikan hardware dengan keinginan.

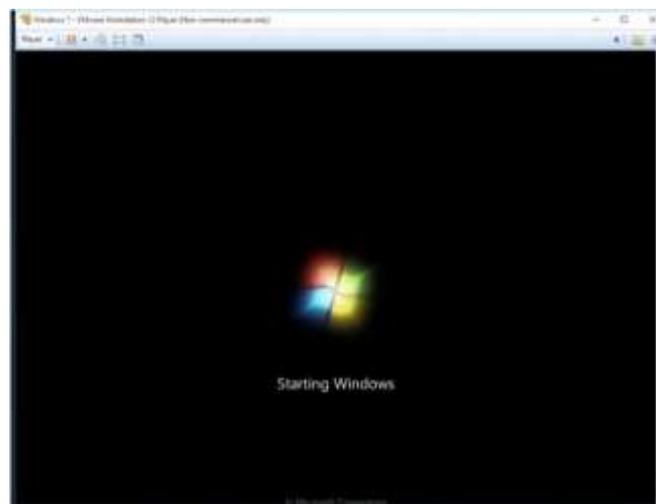


11. Tunggu proses instalasi selesai.

12. Sistem operasi siap dipakai.

1.4. INSTALASI SISTEM OPERASI WINDOWS 7

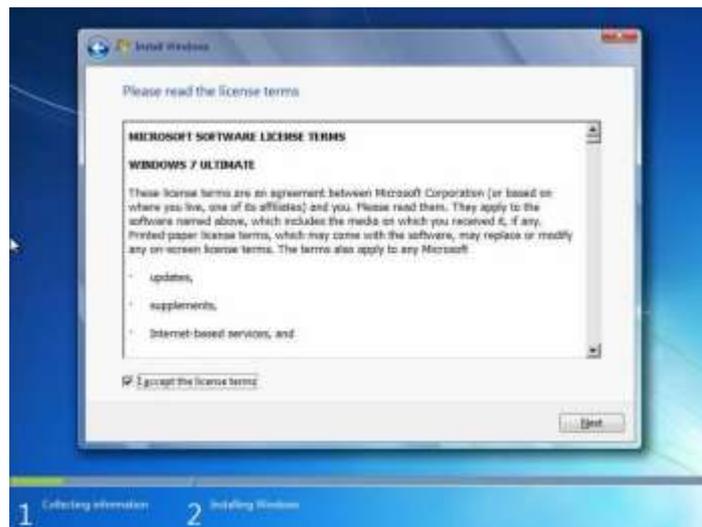
1. Mulai instalasi windows 7, terserah dari CD, DVD, atau Flashdisk.



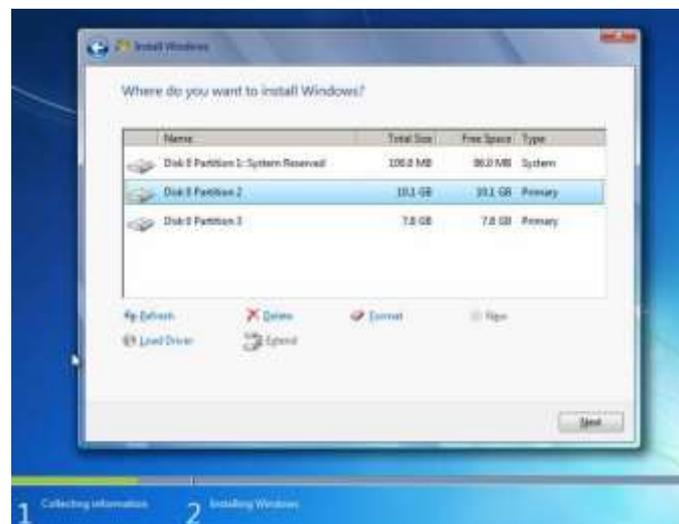
2. Kemudian pilih bahasa yang akan dipakai.



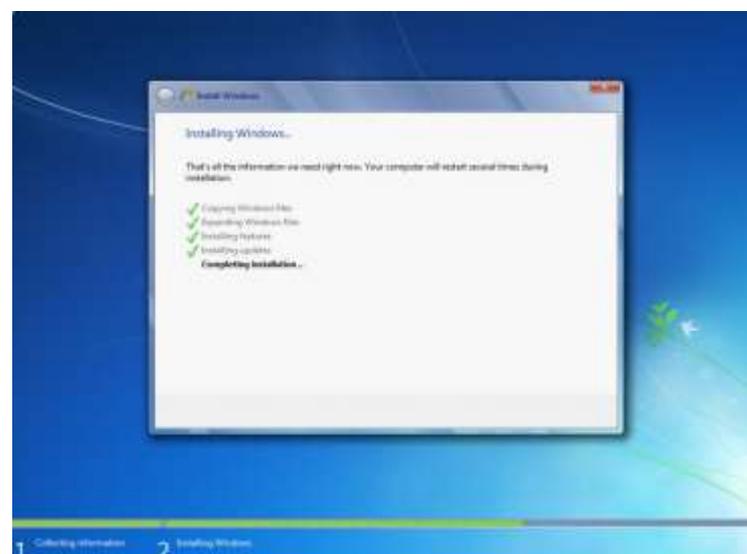
3. Masukkan serial number windows.



4. Buat partisi Harddisk dan tentukan drive mana, sistem operasi akan di-instal.



5. Tunggu Proses instalasi hingga selesai. Komputer mungkin akan melakukan restart beberapa kali.



6. Jika melihat proses ini, berarti Windows akan melakukan setup untuk penggunaan pertama.



7. Isikan nama user dan nama komputer.



8. Windows 7 siap digunakan.



1.5. DYNAMIC LINK LIBRARIES

Semua proses dari Windows 7, disimpan dalam bentuk dynamic link libraries (.dll). Segala fungsi di Windows dijalankan dalam fungsi – fungsi kecil dalam file dll. File dll sendiri terdapat di C:\Windows\System32 untuk sistem operasi 32bit atau C:\Windows\SysWOW64 untuk sistem operasi 64bit.

File .dll berisi kode dan data yang dapat dipakai lebih dari satu proses pada sat yang sama. Sebagai contoh, Comdlg32.dll berfungsi untuk membuka dialog box suatu fungsi. Jadi setiap program dapat menggunakan fungsi dll ini. Fungsionalitas yagn tersimpan dalam dll ini adalah implementasi dialog box.

Untuk Windows 7 sendiri, terdapat lebih dari 3000 file dll yang siap dipakai untuk melakukan prosesnya masing – masing. Disini terdapat beberapa file dll yang penting dan

<i>File name</i>	Fungsi
<i>COMDLG32</i>	Mengontrol dialog box
<i>GDI32</i>	Berisi grafik seperti teks dan font
<i>KERNEL32</i>	Berisi manajemen memori dan proses
<i>USER32</i>	Berisi user interface dan interaksi antar windows
<i>SHELL32</i>	Berisi Windows Shell Common
<i>MSVCRT</i>	Berisi kode program agar program dapat di-run dalam Microsoft Visual C++ dengan benar
<i>NTDLL</i>	Mengontrol messages, timing, threading, dan sinkronisasi antar program di dalam kernel-mode
<i>ADVAPI32</i>	Berisi Windows 32 Base API
<i>SETUPAPI</i>	Mengontrol instalasi suatu program
<i>CRYPT32</i>	Berisi modul Certificate dan Cryptographic Messaging
<i>WMP</i>	Mengontrol Windows Media Player

fungsinya:

<i>WMPLOC</i>	Berisi Resources untuk Windows Media Player
<i>MSVBVM60</i>	Berisi Visual Basic Virtual Machine
<i>IEFRAME</i>	Mengontrol jalannya Internet Explorer
<i>PROPSYS</i>	Berisi System Properties
<i>COMRES</i>	Berisi Resources COM+
<i>DISPLAY</i>	Menampilkan Control Panel
<i>EXPLORERFRAME</i>	Mengontrol Frame dari Windows Explorer
<i>DINPUT8</i>	Mengontrol Direct Input pada sistem operasi
<i>D3DX10</i>	Mengontrol DirectX 10 pada Windows

1.6. TUGAS

1. Buatlah tutorial cara instalasi Windows 8 atau 10 boleh dalam bentuk laporan dengan screenshot ataupun video.
2. Buatlah tabel yang berisi minimal 20 file dll dan fungsinya selain yang sudah ada di atas.

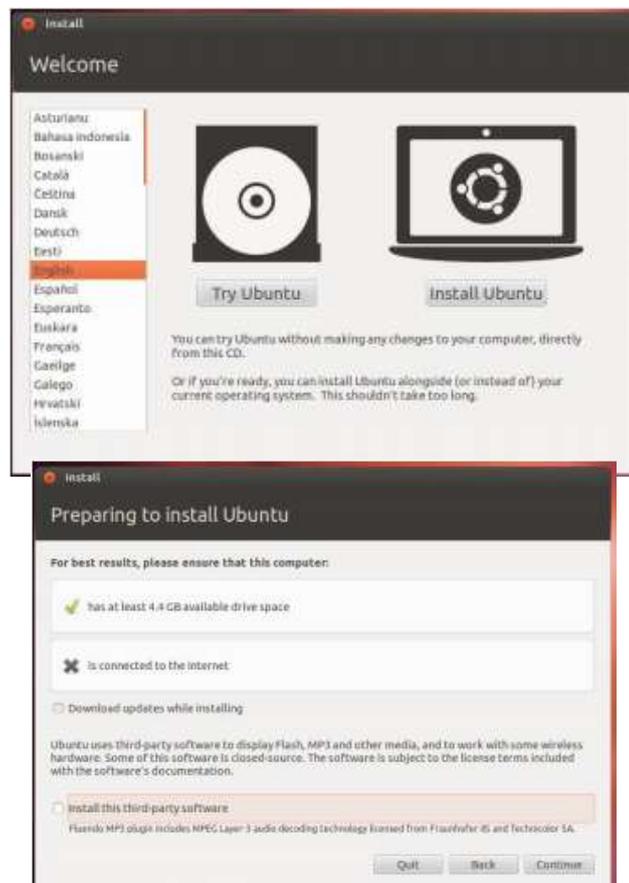
2. INSTALASI DAN KONFIGURASI UBUNTU

2.1. INSTALASI UBUNTU

1. Tampilan awal ketika akan menginstall Ubuntu versi 12.04 adalah seperti pada screenshot di bawah ini :



2. Tunggu (sekitar 1 menit) hingga layar berubah seperti pada gambar di bawah. Ketika tampilan di bawah ini telah muncul, pilih bahasa yang sesuai dengan keinginan anda (pemilihan bahasa akan mempengaruhi petunjuk instalasi ubuntu selanjutnya). Di sini



misalkan kami memilih bahasa Inggris, lalu klik Install Ubuntu untuk menginstall ke dalam komputer anda.

3. Ketika gambar seperti di bawah ini muncul, ada baiknya membaca terlebih dahulu kebutuhan dasar yang dibutuhkan dalam menginstall Ubuntu seperti misalnya memiliki drive space sebesar 4.4 GB. Untuk melanjutkan klik “Continue”.
4. Selanjutnya akan muncul 2 buah opsi seperti gambar di bawah ini. Opsi Erase disk and Install Ubuntu memiliki dampak menghapus seluruh data yang sebelumnya tersimpan dalam HDD komputer anda lalu menginstall Ubuntu dalam keadaan HDD sudah kosong (Data anda sebelumnya sudah hilang). Untuk keamanan data sebaiknya pilih opsi

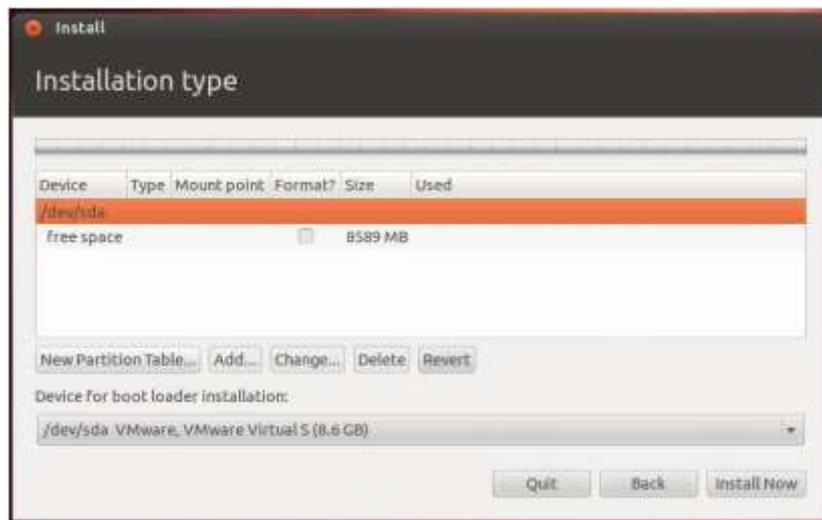


Something else untuk melakukan setting manual mengenai apa yang akan anda lakukan terhadap komputer anda selanjutnya.

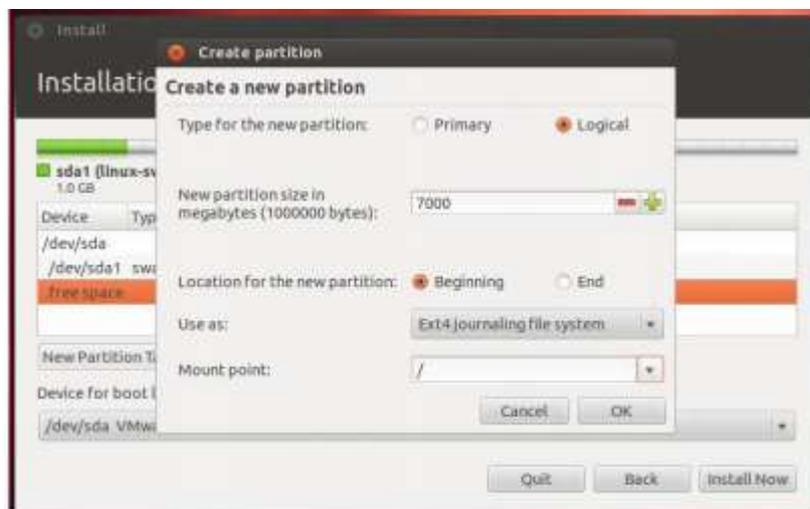
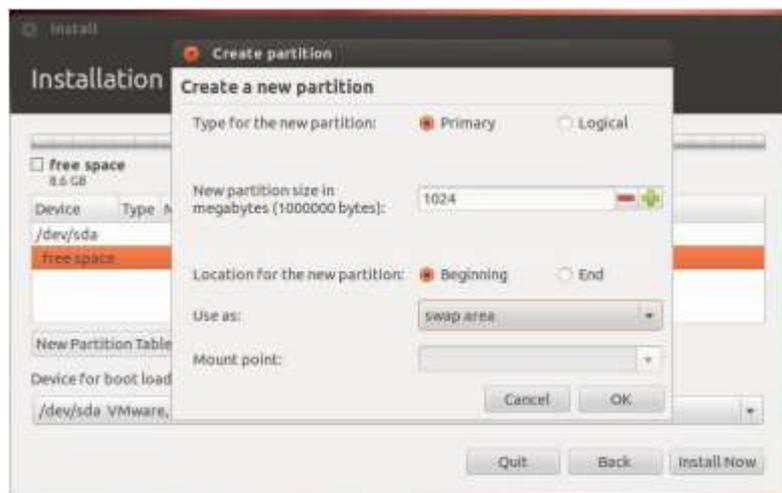
5. Pilih /dev/sda klik new partition table
6. Ketika muncul command box seperti di bawah ini klik saja continue



7. Pilih dan double klik free space



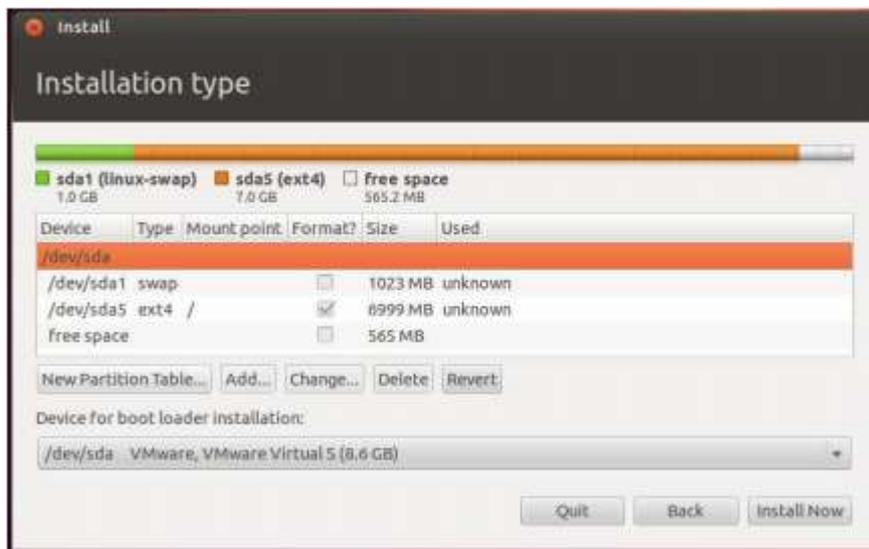
8. Edit value pada partition size menjadi sebesar 2x RAM anda (RAM kami sebesar 512 MB maka pada Partition Size kami masukkan value 1024 MB), edit kolom Use as menjadi swap area> klik OK



9. Selanjutnya, double klik free space yang tersisa, edit partition size menjadi 7000 mb, lalu ubah mount point menjadi “ / ”(root) > klik OK



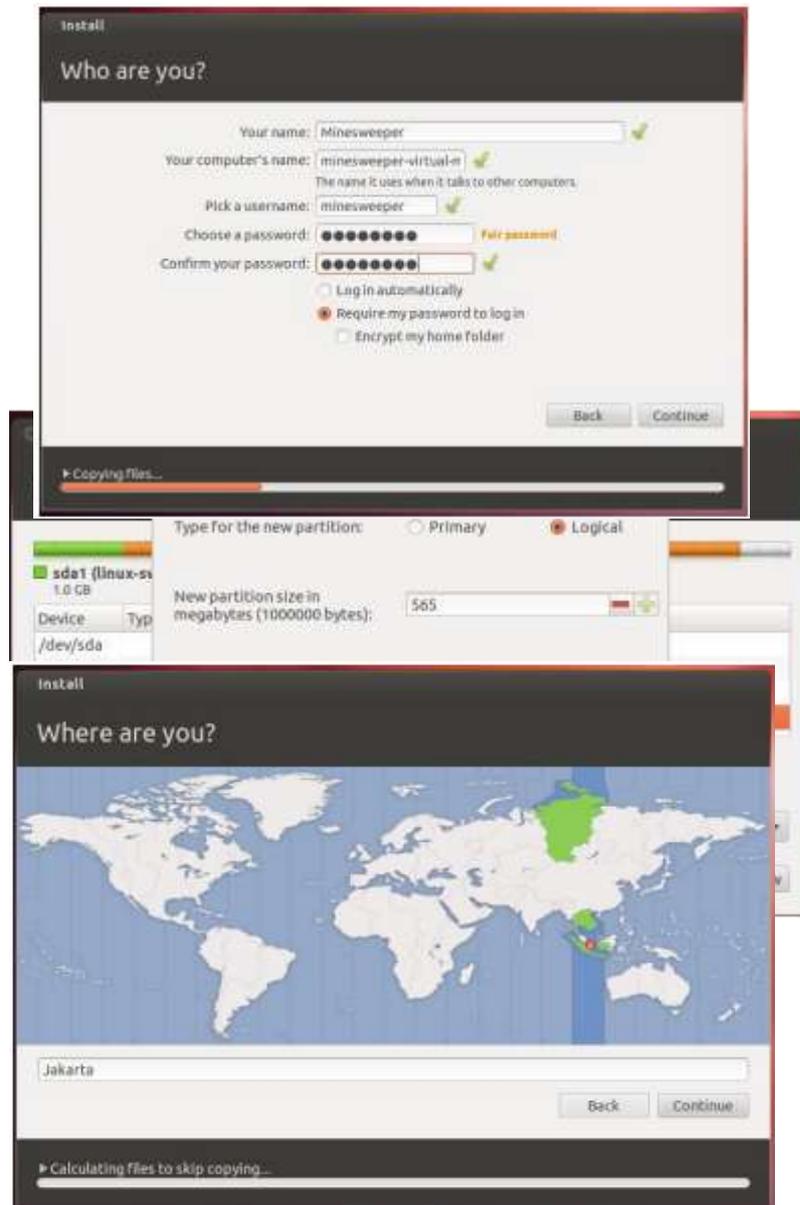
10. Pilih dan double klik free space sekali lagi untuk mendayagunakan sisa kapasitas HDD yang tersisa.



11. Ubah mount point menjadi “/home”

12. Setelah kira-kira sudah tidak ada sisa kapasitas HDD yang terbuang, langsung saja klik

Install Now



13. Pilih region sebagai acuan waktu GMT di komputer anda (kami memilih Jakarta, Indonesia sebagai contohnya) lalu klik Continue

14. Pada bagian Keyboard layout pilih default saja (english) lalu klik Continue



15. Masukkan data diri anda sebelum melakukan finalisasi instalisasi Ubuntu 12.0.4 anda, pastikan anda masukkan password juga sebagai tambahan sekuritas di komputer anda lalusebagai proses terakhir klik Continue

16. Sekarang proses Installasi Ubuntu 12.0.4 telah dimulai, anda tinggal menunggu sampai



prosesnya berakhir dan selamat menikmati Ubuntu 12.0.4 di komputer anda!

2.2. DISTRO LINUX

Distro Linux (singkatan dari distribusi Linux) adalah sebutan untuk sistem operasi komputer dan aplikasinya, merupakan keluarga Unix yang menggunakan kernel Linux.

Distribusi Linux bisa berupa perangkat lunak bebas dan bisa juga berupa perangkat lunak komersial seperti Red Hat Enterprise, SuSE, dan lain-lain.

Ada banyak distribusi atau distro Linux yang telah muncul. Beberapa bertahan dan menjadi distro besar, bahkan sampai menghasilkan distro turunan, contohnya distro Debian GNU/Linux. Distro ini telah menghasilkan puluhan distro turunan, antara lain Ubuntu, Knoppix, Xandros, DSL, dan sebagainya.

Berikut macam-macam distro Linux:

- Arch Linux, merupakan distribusi jenis rolling release yang ditargetkan pada pengguna Linux yang sudah berpengalaman, Arch Linux dikelola oleh komunitas.
 - Manjaro, adalah distribusi turunan Arch Linux yang cepat, ramah pengguna dan berorientasi-desktop
- Debian, distribusi ini dikelola oleh sukarelawan di komunitas. Debian merupakan distribusi tua yang masih ada hingga saat ini dan banyak menghasilkan turunan berupa distribusi baru. Berikut ini merupakan contoh distribusi populer yang diturunkan dari Debian.
 - Canaima, adalah sebuah proyek sosial-teknologi terbuka. Canaima dibangun dan dikembangkan di Venezuela berdasarkan Debian.
 - Knoppix, distribusi pertama yang menggunakan Live CD untuk menjalankan sistem dari removable media tanpa harus melakukan instalasi ke hard disk.
 - Linux Mint Debian Edition (LMDE), yang didasarkan langsung dari distribusi Debian.
 - Ubuntu, merupakan distribusi yang paling populer yang berasal dari Debian, dikembangkan oleh perusahaan Canonical Ltd.

- Backtrack, distribusi yang dikembangkan oleh tim Offensive Security yang dibangun khusus untuk tujuan digital forensik dan uji keamanan.
- Linux Mint, distribusi Linux berbasis Debian (Linux Mint Debian Edition) dan Ubuntu.
- Lubuntu, varian Ubuntu yang menggunakan antarmuka desktop LXDE (untuk komputer dengan spesifikasi rendah)
- Xubuntu, varian Ubuntu yang menggunakan antarmuka desktop XFCE (untuk komputer dengan spesifikasi rendah)
- Kubuntu, varian Ubuntu yang menggunakan antarmuka desktop KDE
- Edubuntu, varian Ubuntu khusus untuk edukasi atau pembelajaran.
- BlackBuntu, varian Ubuntu khusus untuk Penetrasi jaringan, keamanan, digital forensic.
- Sably, Varian ubuntu cocok bagi muslim(lengkap dengan software yang islami)
- Elementary OS, distribusi turunan Ubuntu yang dikembangkan oleh elementary LLC. Elementary OS menggunakan antarmuka desktop Pantheon
 - BlankOn, merupakan distribusi yang dikembangkan di Indonesia. BlankOn menggunakan antarmuka desktop yang disebut Manokwari dan tersedia dalam beberapa bahasa daerah lokal di Indonesia.
- Fedora, distribusi komunitas yang disponsori oleh perusahaan Amerika, RedHat.
 - Red Hat Enterprise Linux, distribusi yang dikelola dan dikomersialkan oleh RedHat.
 - CentOS, distribusi komunitas yang dibangun dari kode sumber yang sama oleh RedHat.

- Oracle Enterprise Linux, merupakan turunan dari Red Hat Enterprise Linux, dipelihara dan komersial didukung oleh Oracle.
 - IGOS Nusantara, dikembangkan oleh Lembaga Ilmu Pengetahuan Indonesia. Dibangun berdasarkan Red Hat Enterprise Linux.
 - Mandriva, merupakan distribusi komersial turunan RedHat yang populer di beberapa negara Eropa dan Brazil. Distribusi Mandriva dikelola oleh perusahaan Prancis dengan nama yang sama, Mandriva.
 - Mageia, merupakan distribusi komunitas yang merupakan percabangan dari Mandriva pada tahun 2010.
 - PCLinuxOS, turunan dari Mandriva yang mengkhususkan kepada penggunaan komputer desktop.
- Slackware, merupakan distribusi pertama yang lahir pada tahun 1993 dan merupakan salah satu distribusi Linux tertua saat ini.
 - SUSE Linux Enterprise, dibangun berdasarkan OpenSUSE dan dikelola oleh perusahaan SUSE.
 - OpenSUSE, distribusi komunitas yang disponsori oleh perusahaan Jerman, SUSE.

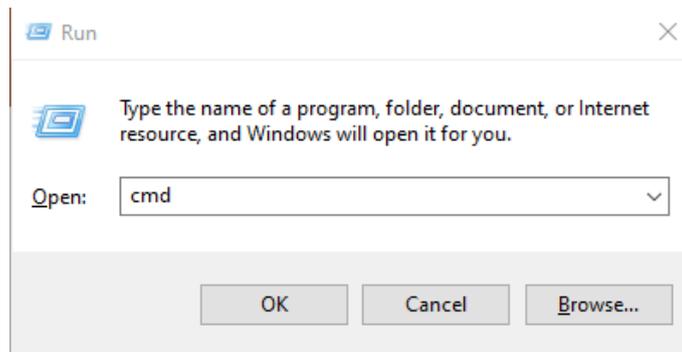
2.3. TUGAS

1. Pilih salah satu distro linux yang ada saat ini lalu tulis kelebihan dan kekurangan distro linux tersebut.
2. Buatlah tutorial untuk instalasi distro linux tersebut.

3. Command Run & Command Prompt

3.1. Command Run

Pengertian Run adalah suatu Tools pada Windows yang bertujuan untuk memanggil suatu Program atau Aplikasi dengan menggunakan Perintah atau Comment. Cara membuka Run ada



beberapa cara Klik Start → Run, atau dengan mengetik Logo Windows pada Keyboard + R.

Perintah pada Run

Perintah	Keterangan
logoff	Log off from Windows
calc	Launch Calculator
cmd	Command Prompt
control	Control Panel
compmgmt.msc	Computer Management
comexp.msc	Component Services
devmgmt.msc	Device Manager
dxdiag	DirectX Diagnostic Tool
cleanmgr	Disk Cleanup
dfrgui	Disk Defragmenter
diskmgmt.msc	Disk Management
utilman	Ease of Access Center
eventvwr.msc	Event Viewer
iexplore	Launch Internet Explorer
gpedit.msc	Local Group Policy Editor
secpol.msc	Local Security Policy
lusermgr.msc	Local Users and Groups
magnify	Launch Magnifier
narrator	Launch Narrator
notepad	Launch Notepad
regedit	Registry Editor
mspaint	Launch Paint
powercfg.cpl	Power Options
appwiz.cpl	Programs and Features
mstsc	Remote Desktop Connection
snippingtool	Snipping Tool
msconfig	System Configuration
sysedit	System Configuration Editor
msinfo32	System Information
taskmgr	Task Manager
explorer	Launch Windows Explorer
firewall.cpl	Windows Firewall
powershell	Windows Powershell
write	Launch WordPad

3.2. Command Prompt



Perintah-perintah yang dapat dijalankan pada Command Prompt dapat dilihat dengan cara membuka Command Prompt lalu menuliskan “help” dan menekan tombol Enter. Sedangkan untuk mendapatkan informasi lebih spesifik mengenai sebuah perintah, dapat mengetik “help

Perintah	Keterangan
cd	Displays the name of or changes the current directory
chdir	Displays the name of or changes the current directory
chkdsk	Checks a disk and displays a status report
cls	Clears the screen
cmd	Starts a new instance of the Windows command interpreter
comp	Compares the contents of two files or sets of files
copy	Copies one or more files to another location
date	Displays or sets the date
del	Deletes one or more files
dir	Displays a list of files and subdirectories in a directory
echo	Displays messages, or turns command echoing on or off
erase	Deletes one or more files
exit	Quits the CMD.EXE program (command interpreter)
help	Provides help information for windows commands
md	Creates a directory
mkdir	Creates a directory
move	Moves one or more files from one directory to another directory
print	Prints a text file
rd	Removes a directory
ren	Renames a file or files
rename	Renames a file or files
rmdir	Removes a directory
shutdown	Allows proper local or remote shutdown of machine
start	Starts a separate window to run a specified program or command
systeminfo	Displays machine specific properties and configuration
tasklist	Displays all currently running tasks including services
taskkill	Kill or stop a running process or application
time	Displays or sets the system time
type	Displays the contents of a text file

command” dimana command adalah perintah yang ingin kita dapatkan informasinya. Berikut ini adalah beberapa contoh perintah pada Command Prompt beserta fungsinya.

3.3. Latihan

1. Buatlah folder dengan nama folder TES_CMD

```
MD TES_CMD
```

2. Di dalam folder TES_CMD tambahkan 3 folder dengan nama TES_1, TES_2, dan TES_3

```
CD TES_CMD
```

```
MD TES_1
```

```
MD TES_2
```

```
MD TES_3
```

3. Tambahkan file Berkas.txt ke dalam folder TES_1 dengan isi pesan Nama Saya adalah ...(isi dengan nama Anda)

```
CD TES_1
```

```
ECHO NAMA SAYA ADALAH RAISSA>BERKAS.TXT
```

4. Tampilkan isi dari file Berkas.txt

```
TYPE BERKAS.TXT
```

5. Copy file Berkas.txt ke folder TES_2 dan TES_3

```
COPY BERKAS.TXT C:\USERS\RAISSA\TES_CMD\TES_2
```

```
COPY BERKAS.TXT C:\USERS\RAISSA\TES_CMD\TES_3
```

6. Pindahkan file Berkas.txt yang ada di folder TES_3 ke folder TES_CMD

```
CD C:\USERS\RAISSA\TES_CMD\TES_3
```

```
MOVE BERKAS.TXT C:\USERS\RAISSA\TES_CMD
```

7. Hapus file Berkas.txt yang ada di folder TES_1

```
CD C:\USERS\RAISSA\TES_CMD\TES_1
```

DEL BERKAS.TXT

8. Hapus folder TES_1

CD C:\USERS\RAISSA\TES_CMD

RD TES_1

9. Lihat isi dari folder TES_CMD

DIR

10. Ganti nama file Berkas.txt di folder TES_CMD menjadi BerkasBaru.txt

REN BERKAS.TXT BERKASBARU.TXT

4. Proses Penjadwalan di Prosesor

4.1. Deskripsi Penjadwalan Proses

Penjadwalan merupakan kumpulan kebijaksanaan dan mekanisme di sistem operasi yang berkaitan dengan urutan kerja yang sering dilakukan di sistem komputer. Tujuan untuk memutuskan proses mana yang harus berjalan dan kapan serta berapa lama waktu yang diperlukan.

Kriteria yang digunakan untuk optimasi kinerja penjadwalan proses :

- a. Adil (fairness): setiap proses mendapat jatah waktu yang sama.
- b. Efisiensi: menjaga agar pemroses dalam keadaan sibuk sehingga efisiensinya maksimum.
- c. Waktu tanggap (response time): meminimalkan waktu tanggap. Waktu tanggap ada 2 yaitu waktu tanggap interaktif (terminal response time) dan waktu tanggap pada sistem nyata (event response time). Waktu tanggap interaktif adalah waktu yang dihabiskan dari saat karakter perintah terakhir berinteraksi sampai instruksi pertama rutin layanan dieksekusi. Waktu tanggap pada sistem nyata adalah waktu saat kejadian sampai instruksi pertama rutin layanan dieksekusi.
- d. Turn around time: meminimalkan turn around time. Turn around time adalah waktu yang dihabiskan dari saat program masuk ke sistem sampai proses diselesaikan sistem. Dengan kata lain $\text{turn around time} = \text{waktu eksekusi} + \text{waktu tunggu}$.
- e. Throughput: memaksimalkan jumlah job yang diproses per satuan waktu. Throughput adalah jumlah kerja yang dapat diselesaikan persatuan waktu.

4.2. Tipe Penjadwalan

Terdapat 3 tipe penjadwalan pada sistem operasi yang kompleks :

- a. Penjadwalan jangka pendek (short term scheduler) bertugas menjadwalkan alokasi pemroses diantara proses – proses ready di memori utama. Sasarannya memaksimalkan kinerja untuk memenuhi satu kumpulan kriteria.
- b. Penjadwalan jangka menengah (medium term scheduler) bertugas menangani proses – proses swapping, mengendalikan transisi dari suspended to ready proses-proses swapping. Swapping adalah pemindahan proses yang tertunda dari memori utama ke memori sekunder.
- c. Penjadwalan jangka panjang (long term scheduler) bertugas terhadap antrian batch dengan sasaran memberi keseimbangan job – job campuran.

4.3. Strategi penjadwalan

Terdapat 2 strategi penjadwalan :

- a. Penjadwalan nonpreemptive (run-to-completion) : apabila suatu proses telah diberi jatah maka tidak dapat diambilalih proses lain sampai proses tersebut selesai.
- b. Penjadwalan preemptive : proses lain dapat mengambilalih jatah waktu proses lain. Ini sangat berguna pada sistem dimana proses memerlukan tanggapan dari pemroses lebih cepat. Misalnya pada sistem waktu nyata, kehilangan interupsi dapat berakibat fatal atau pada sistem interaktif time sharing, penjadwalan preemptive untuk menjamin waktu tanggap yang memadai.

4.4. Algoritma Penjadwalan

Algoritma yang sering digunakan pada setragi non-preemptive :

- FIFO (First-in, first out) atau FCFS (first come, first serve)
- SJF (Shortest Job First)
- HRN (Higest-ratio next)

Algoritma yang sering digunakan dalam strategi preemptive:

- RR (Round robin)
- SRF (Shortest remaining first)
- PS (priority scheduling) – preemptive dan non-preemptive

4.5. FIFO (First In First Out)

Penjadwalan paling sederhana, tidak berprioritas, proses diberi jatah berdasarkan waktu datangnya dan dijalankan sampai proses selesai. FIFO biasanya dikombinasikan dengan skema lainnya. Pada FIFO, proses yang tiba lebih dahulu akan dilayani lebih dahulu.

Contoh Kasus:

Dianggap semua proses sebagai satu kumpulan proses serentak dan hanya dilayani oleh satu prosesor. Table berikut menggambarkan dari beberapa proses yang akan dilayani oleh satu prosesor.

Nama Proses	Saat Tiba	Lama Proses
P1	0	5
P2	1	3
P3	2	8
P4	3	6

Penyelesaian : buat Gantt Card lalu hitung hitung rata-rata lama waktu tanggap untuk semua proses.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	5	0	5	5	0
P2	1	3	5	8	7	4
P3	2	8	8	16	14	6
P4	3	6	16	22	19	13
Jumlah					45	23
Rata-rata					11.25	5.75

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P1	P1	P2	P2	P2	P3	P4	P4							

19	20	21	22
P4	P4	P4	P4

4.6. SJF (Shortest Job First)

Tak berprioritas, jarang digunakan, menggunakan asumsi waktu jalan proses sudah diketahui, menjadwalkan proses dengan waktu terpendek dijalankan lebih dahulu, efisiensi tinggi dan time around rendah. Masalah yang terjadi : tidak dapat mengetahui ukuran job saat job masuk dan untuk proses yang datang tidak bersamaan menggunakan penetapan dinamis.

Contoh kasus:

Nama Proses	Saat Tiba	Lama Proses
P1	0	5
P2	1	3
P3	2	8
P4	3	6

Penyelesaian:

- Dimulai dari saat 0, proses terpendek yaitu P2 belum tiba, jadi P1 dikerjakan, rampung pada saat 5
- Proses terpendek (P2) sudah tiba, maka yang dikerjakan adalah P2 dimulai pada saat 5 rampung pada saat 8.
- Setelah P2 rampung, P4 yang urutan kedua sudah tiba sehingga dpt dikerjakan oleh processor. Mulai saat 8 dan rampung saat 14.
- P3 urutan akhir sudah tiba dan dikerjakan, mulai saat 14 dan rampung saat 22.

Di bawah ini dibuat barisan saat dan daftar proses.

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	5	0	5	5	0
P2	1	3	5	8	7	4
P3	2	8	14	22	20	12
P4	3	6	8	14	11	5
Jumlah					43	21
Rata-rata					10.75	5.25

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P1	P1	P2	P2	P2	P4	P4	P4	P4	P4	P4	P3	P3	P3	P3

19	20	21	22
P3	P3	P3	P3

4.7. HRN (Higest-ratio next)

Mendahulukan proses pendek, namun prioritas proses panjang akan turut meningkat melalui peningkatan rasio penaltinya. Pada suatu saat, prioritas proses panjang yang lama tertunda akan dapat menyusul prioritas proses pendek.

Rumus Ratio Pinalti:

$$R_p = \frac{(s+t)}{t}$$

dimana

t = Lama Proses.

s = Lama Tunggu (antri, wkt sia-sia)

Contoh kasus:

Nama Proses	Saat Tiba	Lama Proses
P1	0	4
P2	1	2
P3	2	5
P4	3	8
P5	4	4

Penyelesaian:

- Pada saat 0, hanya proses P1 yang tiba, P1 dikerjakan rampung saat 4.
- Pada saat itu, semua proses sudah tiba, sehingga nilai ratio pinalti mereka dapat dihitung.

Nama Proses	Tiba Selama	Ratio Penalti
P2	3	$(3+2)/2 = 2.5$
P3	2	$(2+5)/5 = 1.2$
P4	1	$(1+8)/8 = 1.125$
P5	0	$(0+4)/4 = 1$

- Tampak disini proses P2 mempunyai nilai ratio pinalti tertinggi yaitu 2.5, maka pada saat 4 proses P2 dikerjakan rampung pada saat 6.
- Pada saat 6 dihitung lagi ratio penalti

Nama Proses	Tiba Selama	Ratio Penalti
P3	4	$(4+5)/5 = 1.8$
P4	3	$(3+8)/8 = 1.375$
P5	2	$(2+4)/4 = 1.5$

- Tampak disini proses P3 mempunyai nilai ratio pinalti tertinggi yaitu 1.8, maka pada saat 6 proses P3 dikerjakan rampung pada saat 11.
- Pada saat 11 dihitung lagi ratio penalti

Nama Proses	Tiba Selama	Ratio Penalti
P4	3	$(8+8)/8 = 2$
P5	2	$(7+4)/4 = 2.75$

- Tampak disini proses P5 mempunyai nilai ratio pinalti tertinggi yaitu 2.75, maka pada saat 11 proses P5 dikerjakan rampung pada saat 15.
- Pada saat 15 hanya P4 yang belum dikerjakan, maka P4 dikerjakan selesai pada saat 23.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P1	P2	P2	P3	P3	P3	P3	P3	P5	P5	P5	P5	P4	P4	P4

19	20	21	22	23
P4	P4	P4	P4	P4

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	4	0	4	4	0
P2	1	2	4	6	5	3
P3	2	5	6	11	9	4
P4	3	8	15	23	20	12
P5	4	4	11	15	11	7

Jumlah	49	26
Rata-rata	9.8	5.2

4.8. RR (Round robin)

Penjadwalan ini berlangsung tanpa prioritas, tetapi dengan pre-empsti. Pada penjadwalan ini, secara bergiliran berdasarkan antrian (tanpa prioritas), prosesor melayani sejenak setiap proses. Secara berturut-turut, proses yang telah dilayani prosesor dan belum rampung akan kembali ke akhir antrian yang ada pada saat itu, sehingga penggiliran ini berputar seperti gelang. Hanya proses yang rampung yang meninggalkan prosesor dan antrian itu.

Contoh kasus:

Nama Proses	Saat Tiba	Lama Proses
P1	0	7
P2	0	5
P3	0	8
P4	0	2
P5	0	6
Kuantum waktu : 3		

Penyelesaian:

- Pada saat 0, proses P1 dikerjakan,pada saat 3 atau pada saat akhir kuantum, pelaksanaan proses P1 dihentikan.
- Melalui pre-empsti proses P1 dikeluarkan dari prosesor. Pada saat itu proses P2 dikerjakan. Pada saat 6 (pada akhir kuantum) waktu proses P2 dikeluarkan dari prosesor.

- Pada saat itu juga proses P3 dikerjakan. Pada saat 9 waktu proses P3 dikeluarkan dari prosesor.
- Pada saat itu proses P4 dikerjakan, karena lama proses P4 = 2 satuan waktu (kurang dari kuantum waktu), maka proses P4 rampung pada saat 11.
- Pada saat 11, proses P5 dikerjakan. Pada saat 14 (sisa akhir kuantum waktu), proses P5
- dikeluarkan.
- Pada saat 14, sisa proses P1 dikerjakan lagi, namun pada saat 17 proses P1 dikeluarkan lagi.
- Pada saat 17 ini, sisa proses P2 dikerjakan dan pada saat 19, proses P2 rampung.
- Pada saat 19, sisa proses P3 dikerjakan.
- Pada saat 22, waktu proses P3 dikeluarkan lagi. Karena P4 sudah rampung, maka pada saat 22 sisa P5 dikerjakan.
- Pada saat 25 proses P5 rampung.
- Pada saat 25, sisa proses P1 dikerjakan, pada saat 26 proses P1 rampung.
- Pada saat 26 itu, sisa proses P3 dikerjakan dan rampung saat 28.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P2	P2	P2	P3	P3	P3	P4	P4	P5	P5	P5	P1	P1	P1	P2

19	20	21	22	23	24	25	26	27	28
P2	P3	P3	P3	P5	P5	P5	P1	P3	P3

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	7	0	26	26	0
P2	0	5	3	19	19	3
P3	0	8	6	28	28	6
P4	0	2	9	11	11	9
P5	0	6	11	25	25	11
Jumlah					109	29
Rata-rata					21.8	5.8

4.9. SRF (Shortest remaining first)

Merupakan penjadwalan dengan prioritas dan dengan pre-emptive. Prioritas didasarkan pada pendeknya sisa proses. Makin pendek sisa proses makin tinggi prioritasnya.

Digunakan 2 langkah melihat pelaksanaan penjadwalan ini :

- Setiap kali, diperhatikan saat proses tiba atau saat proses rampung.
- Menghitung lama proses dari semua proses yang ada pada saat itu

Nama Proses	Saat Tiba	Lama Proses
P1	0	5
P2	1	3
P3	2	8
P4	3	6

Penyelesaian : buat Gannt Card lalu hitung hitung rata-rata lama waktu tanggap untuk semua proses.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P2	P2	P2	P1	P1	P1	P1	P4	P4	P4	P4	P4	P4	P3	P3	P3	P3

19	20	21	22
P3	P3	P3	P3

Nama Proses	Saat Tiba	Lama Proses	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	5	0	8	8	0
P2	1	3	1	4	3	0
P3	2	8	14	22	20	12
P4	3	6	8	14	11	5
Jumlah					42	17
Rata-rata					10.5	4.25

4.10. PS (priority scheduling)

Penjadwalan Priority dapat dilakukan dengan cara preemptive dan non-preemptive dan salah satu algoritma penjadwalan yang paling sering dipakai di batch system. Setiap proses diberi prioritas. Proses dengan prioritas tertinggi akan dikerjakan pertama kali. Proses dengan prioritas yang sama akan dijalankan dengan prinsip FIFO. Prioritas dapat ditentukan berdasarkan kebutuhan memori, kebutuhan waktu, atau kebutuhan resources yang lain.

Contoh Kasus:

- a. Non-Preemptive

Nama Proses	Saat Tiba	Lama Proses	Prioritas
P1	0	5	1
P2	1	3	2
P3	2	8	1
P4	3	6	3

Penyelesaian:

- Berdasarkan prioritas, P4 yang paling tinggi, namun belum tiba pada saat 0. Pada saat 0 hanya P1 yang sudah tiba, maka P1 dikerjakan, selesai pada saat 5.
- Pada saat 5, P4 yang prioritasnya paling tinggi sudah tiba, maka P4 dikerjakan, selesai pada saat 11.
- Prioritas tertinggi kedua adalah P2, maka pada saat 11 P2 dikerjakan selesai pada saat 14.
- Pada saat 14 P3 dikerjakan selesai pada saat 22.

Nama Proses	Saat Tiba	Lama Proses	Prioritas	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	5	1	0	5	5	0
P2	1	3	2	11	14	13	10
P3	2	8	1	14	22	20	12
P4	3	6	3	5	11	8	2
Jumlah						46	24
Rata-rata						11.5	6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P1	P1	P4	P4	P4	P4	P4	P4	P2	P2	P2	P3	P3	P3	P3

19	20	21	22
P3	P3	P3	P3

b. Preemptive

Nama Proses	Saat Tiba	Lama Proses	Prioritas
P1	0	5	1
P2	1	3	2
P3	2	8	1
P4	3	6	3

Penyelesaian:

- Berdasarkan prioritas, P4 yang paling tinggi, namun belum tiba pada saat 0. Pada saat 0 hanya P1 yang sudah tiba, maka P1 dikerjakan.
- Pada saat 3 P4 dengan prioritas yang lebih tinggi sudah tiba, maka P1 dijeda P4 dikerjakan selesai pada saat 9.
- Pada saat 9, P2 dengan prioritas tertinggi kedua dikerjakan, selesai pada saat 12.
- Pada saat 12, sisa proses P1 dikerjakan selesai pada saat 14.
- Pada saat 14, P3 dikerjakan selesai pada saat 22

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	P1	P1	P4	P4	P4	P4	P4	P4	P2	P2	P2	P1	P1	P3	P3	P3	P3

19	20	21	22
P3	P3	P3	P3

Nama Proses	Saat Tiba	Lama Proses	Prioritas	Saat Mulai	Saat Rampung	Lama Tanggap	Lama Tunggu
P1	0	5	1	0	14	14	0
P2	1	3	2	9	12	11	8
P3	2	8	1	14	22	20	12
P4	3	6	3	3	9	6	0
Jumlah						51	20
Rata-rata						12.75	5

5. SIKRONISASI

Dua proses berbagi sebuah buffer dengan ukuran yang tetap. Salah satunya produser, meletakkan informasi ke buffer yang lainnya. Konsumen mengambil informasi dari buffer. Ini juga dapat digeneralisasi untuk masalah yang memiliki m buah produser dan n buah konsumen, tetapi kita hanya akan memfokuskan kasus dengan satu produser dan satu konsumen karena diasumsikan dapat menyederhanakan solusi.

Masalah akan timbul ketika produser ingin menaruh barang yang baru tetapi buffer sudah penuh. Solusi untuk produser adalah istirahat (sleep) dan akan dibangunkan ketika konsumen telah mengambil satu atau lebih barang dari buffer. Biasanya jika konsumen ingin mengambil barang dari buffer dan melihat bahwa buffer sedang kosong, maka konsumen istirahat (sleep) sampai produser meletakkan barang pada buffer dan membangunkan (wake up) konsumen.

Pendekatan seperti ini terdengar cukup sederhana, tetapi hal ini dapat menggiring kita ke jenis masalah yang sama seperti race condition dengan spooler direktori. Untuk mengetahui jumlah barang di buffer, kita membutuhkan sebuah variabel kita namakan count. Jika jumlah maksimum dari barang yang dapat ditampung buffer adalah N , kode produser pertama kali akan mencoba untuk mengetahui apakah nilai count sama dengan nilai N . Jika itu terjadi maka produser akan istirahat (sleep), tetapi jika nilai count tidak sama dengan N , produser akan terus menambahkan barang dan menaikkan nilai count.

Sekarang mari kita kembali ke permasalahan race condition. Ini dapat terjadi karena akses ke count tidak dipaksakan. Situasi seperti itu mungkin dapat terjadi. Buffer sedang kosong dan konsumen baru saja membaca count untuk melihat apakah count bernilai 0. Pada saat itu, penjadual memutuskan untuk mengentikan proses konsumen sementara dan menjalankan produser. Produser memasukkan barang ke buffer, menaikkan nilai count, dan

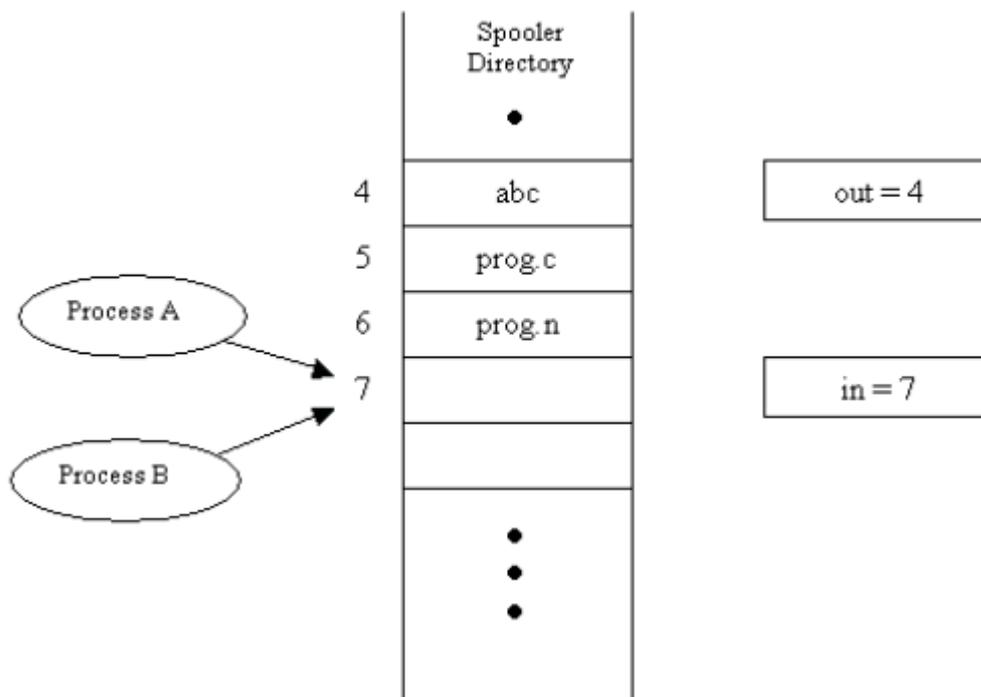
memberitahukan bahwa count sekarang bernilai 1. Pemikiran bahwa count baru saja bernilai 0 sehingga konsumen harus istirahat (sleep). Produsen memanggil fungsi wake up untuk membangkitkan konsumen. Sayangnya, konsumen secara logika belum istirahat. Jadi sinyal untuk membangkitkan konsumen, tidak dapat ditangkap oleh konsumen. Ketika konsumen bekerja berikutnya, konsumen akan memeriksa nilai count yang dibaca sebelumnya, dan mendapatkan nilai 0, kemudian konsumen istirahat (sleep) lagi. Cepat atau lambat produsen akan mengisi buffer dan juga istirahat (sleep). Keduanya akan istirahat selamanya.

Inti permasalahannya disini adalah pesan untuk membangkitkan sebuah proses tidak tersampaikan. Jika pesan/ sinyal ini tersampaikan dengan baik, segalanya akan berjalan lancar.

5.1. Race Condition

Race Condition adalah situasi di mana beberapa proses mengakses dan memanipulasi data bersama pada saat bersamaan. Nilai akhir dari data bersama tersebut tergantung pada proses yang terakhir selesai. Untuk mencegah race condition, proses - proses yang berjalan bersamaan harus disinkronisasi. Dalam beberapa sistem operasi, proses-proses yang berjalan bersamaan mungkin untuk membagi beberapa penyimpanan umum, masing-masing dapat melakukan proses baca (read) dan proses tulis (write). Penyimpanan bersama (shared storage) mungkin berada di memori utama atau berupa sebuah berkas bersama, lokasi dari memori bersama tidak merubah kealamian dari komunikasi atau masalah yang muncul. Untuk mengetahui bagaimana komunikasi antar proses bekerja, mari kita simak sebuah contoh sederhana, sebuah print spooler. Ketika sebuah proses ingin mencetak sebuah berkas, proses tersebut memasukkan nama berkas ke dalam sebuah spooler direktori yang khusus. Proses yang lain, printer daemon, secara periodik memeriksa untuk mengetahui jika ada banyak berkas yang akan dicetak, dan jika ada berkas yang sudah dicetak dihilangkan nama berkasnya dari direktori.

Bayangkan bahwa spooler direktori memiliki slot dengan jumlah yang sangat besar, diberi nomor 0, 1, 2, 3, 4,... masing-masing dapat memuat sebuah nama berkas. Juga bayangkan bahwa ada dua variabel bersama, out, penunjuk berkas berikutnya untuk dicetak, dan in, menunjuk slot kosong di direktori. Dua variabel tersebut dapat menangani sebuah two-word berkas untuk semua proses. Dengan segera, slot 0, 1, 2, 3 kosong (berkas telah selesai dicetak), dan slot 4, 5, 6 sedang terisi (berisi nama dari berkas yang antri untuk dicetak). Lebih atau kurang secara bersamaan, proses A dan B, mereka memutuskan untuk antri untuk sebuah berkas untuk dicetak.



Dalam Murphy's Law kasus tersebut dapat terjadi. Proses A membaca in dan menyimpan nilai "7" di sebuah variabel lokal yang disebut next_free_slot. Sebuah clock interrupt terjadi dan CPU memutuskan bahwa proses A berjalan cukup lama, sehingga digantikan oleh proses B. Proses B juga membaca in, dan juga mengambil nilai 7, sehingga menyimpan nama berkas

di slot nomor 7 dan memperbaharui nilai in menjadi 8. Maka proses mati dan melakukan hal lain.

Akhirnya proses A berjalan lagi, dimulai dari tempat di mana proses tersebut mati. Hal ini terlihat dalam `next_free_slot`, ditemukan nilai 7 di sana, dan menulis nama berkas di slot nomor 7, menghapus nama berkas yang baru saja diletakkan oleh proses B. Kemudian proses A menghitung `next_free_slot + 1`, yang nilainya 8 dan memperbaharui nilai in menjadi 8. Direktori spooler sekarang secara internal konsisten, sehingga printer daemon tidak akan memberitahukan apa pun yang terjadi, tetapi proses B tidak akan mengambil output apa pun. Situasi seperti ini, dimana dua atau lebih proses melakukan proses reading atau writing beberapa shared data dan hasilnya bergantung pada ketepatan berjalan disebut race condition.

5.2. Critical Section

Bagaimana menghindari race conditions? Kunci untuk mencegah masalah ini dan di situasi yang lain yang melibatkan shared memori, shared berkas, and shared sumber daya yang lain adalah menemukan beberapa jalan untuk mencegah lebih dari satu proses untuk melakukan proses writing dan reading kepada shared data pada saat yang sama.

Dengan kata lain kita membutuhkan mutual exclusion, sebuah jalan yang menjamin jika sebuah proses sedang menggunakan shared berkas, proses lain dikeluarkan dari pekerjaan yang sama. Kesulitan yang terjadi karena proses 2 mulai menggunakan variabel bersama sebelum proses 1 menyelesaikan tugasnya.

Masalah menghindari race conditions dapat juga diformulasikan secara abstrak. Bagian dari waktu, sebuah proses sedang sibuk melakukan perhitungan internal dan hal lain yang tidak menggiring ke kondisi race conditions. Bagaimana pun setiap kali sebuah proses mengakses shared memory atau shared berkas atau melakukan sesuatu yang kritis akan menggiring kepada

race conditions. Bagian dari program dimana shared memory diakses disebut Critical Section atau Critical Region.

Walau pun dapat mencegah race conditions, tapi tidak cukup untuk melakukan kerjasama antar proses secara paralel dengan baik dan efisien dalam menggunakan shared data. Kita butuh 4 kondisi agar menghasilkan solusi yang baik:

1. Tidak ada dua proses secara bersamaan masuk ke dalam critical section.
2. Tidak ada asumsi mengenai kecepatan atau jumlah cpu.
3. Tidak ada proses yang berjalan di luar critical section yang dapat memblok proses lain.
4. Tidak ada proses yang menunggu selamanya untuk masuk critical section.

Critical Section adalah sebuah segmen kode di mana sebuah proses yang mana sumber daya bersama diakses. Terdiri dari:

- Entry Section: kode yang digunakan untuk masuk ke dalam critical section
- Critical Section: Kode di mana hanya ada satu proses yang dapat dieksekusi pada satu waktu
- Exit Section: akhir dari critical section, mengizinkan proses lain
- Remainder Section: kode istirahat setelah masuk ke critical section

Solusi yang diberikan harus memuaskan permintaan berikut:

- Mutual exclusion
- Deadlock free
- Starvation free

Pendekatan yang mungkin untuk solusi proses sinkronisasi

- i. Solusi Perangkat lunak (Software solution)

- Tanpa Sinkronisasi.
 - Dengan Sinkronisasi.
 - Low-level primitives: semaphore
 - High-level primitives: monitors
- ii. Solusi Piranti Keras (Hardware solution)

5.3. Mutual Exclusion

Tiga kondisi untuk menentukan mutual Exclusion

1. Tidak ada dua proses yang pada saat bersamaan berada di critical region.
2. Tidak ada proses yang berjalan diluar critical region yang bisa menghambat proses lain
3. Tidak ada proses yang tidak bisa masuk ke critical region

5.4. Semaphore

Jika kita ingin dapat melakukan proses tulis lebih rumit kita membutuhkan sebuah bahasa untuk melakukannya. Kita akhirnya mendefinisikan semaphore yang kita asumsikan sebagai sebuah operasi atomik.

Semaphore adalah pendekatan yang diajukan oleh Dijkstra, dengan prinsip bahwa dua proses atau lebih dapat bekerja sama dengan menggunakan penanda-penanda sederhana. Seperti proses dapat dipaksa berhenti pada suatu saat, sampai proses mendapatkan penanda tertentu itu. Sembarang kebutuhan koordinasi kompleks dapat dipenuhi dengan struktur penanda yang cocok untuk kebutuhan itu. Variabel khusus untuk penanda ini disebut semaphore.

Semaphore mempunyai dua sifat, yaitu:

1. Semaphore dapat diinisialisasi dengan nilai non-negatif.
2. Terdapat dua operasi terhadap semaphore, yaitu Down dan Up. Usulan asli yang

disampaikan Dijkstra adalah operasi P dan V.

5.5. Operasi Down

Operasi ini menurunkan nilai semaphore, jika nilai semaphore menjadi non-positif maka proses yang mengeksekusinya diblocked. Operasi Down adalah atomic, tak dapat diinterupsi sebelum diselesaikan. Menurunkan nilai, memeriksa nilai, menempatkan proses pada antrian dan memblok sebagai instruksi tunggal. Sejak dimulai, tak ada proses lain yang dapat mengakses semaphore sampai operasi selesai atau diblocked.

5.6. Problem Klasik pada Sinkronisasi

Ada tiga hal yang selalu menjadi masalah pada proses sinkronisasi:

1. Problem Bounded buffer.
2. Problem Reader and Writer.
3. Problem Dining Philosophers.

5.7. Problem Readers-Writers

Problem lain yang terkenal adalah readers-writer problem yang memodelkan proses yang mengakses database. Sebagai contoh sebuah sistem pemesanan sebuah perusahaan penerbangan, dimana banyak proses berkompetisi berharap untuk membaca (read) dan menulis (write). Hal ini dapat diterima bahwa banyak proses membaca database pada saat yang sama, tetapi jika suatu proses sedang menulis database, tidak boleh ada proses lain yang mengakses database tersebut, termasuk membaca database tersebut.

Dalam solusi ini, pertama-tama pembaca mengakses database kemudian melakukan DOWN pada semaphore db.. Langkah selanjutnya readers hanya menaikkan nilai sebuah counter. Hasil dari pembaca nilai counter diturunkan dan nilai terakhir dilakukan UP pada semaphore, mengizinkan memblok writer.

Misalkan selama sebuah reader menggunakan database, reader lain terus berdatangan. Karena ada dua reader pada saat bersamaan bukanlah sebuah masalah, maka reader yang kedua diterima, reader yang ketiga juga dapat diterima jika terus berdatangan reader-reader baru.

Sekarang misalkan writer berdatangan terus menerus. Writer tidak dapat diterima ke database karena writer hanya bisa mengakses data ke database secara eksklusif, jadi writer ditangguhkan. Nanti penambahan reader akan menunjukkan peningkatan. Selama paling tidak ada satu reader yang aktif, reader berikutnya jika datang akan diterima.

Sebagai konsekuensi dari strategi ini, selama terdapat suplai reader yang terus menerus, mereka akan dilayani segera sesuai kedatangan mereka. Writer akan ditunda sampai tidak ada reader lagi. Jika sebuah reader baru tiba, katakan, setiap dua detik, dan masing-masing reader mendapatkan lima detik untuk melakukan tugasnya, writer tidak akan pernah mendapatkan kesempatan.

Untuk mencegah situasi seperti itu, program dapat ditulis agak sedikit berbeda: Ketika reader tiba dan writer menunggu, reader ditunda dibelakang writer yang justru diterima dengan segera. Dengan cara ini, writer tidak harus menunggu reader yang sedang aktif menyelesaikan pekerjaannya, tapi tidak perlu menunggu reader lain yang datang berturut-turut setelah itu.

5.8. Problem Dining Philosophers

Pada tahun 1965, Dijkstra menyelesaikan sebuah masalah sinkronisasi yang beliau sebut dengan dining philosophers problem. Dining philosophers dapat diuraikan sebagai berikut: Lima orang filosof duduk mengelilingi sebuah meja bundar. Masing-masing filosof mempunyai sepiring spageti. Spageti-spageti tersebut sangat licin dan membutuhkan dua garpu untuk memakannya. Diantara sepiring spageti terdapat satu garpu. Kehidupan para filosof terdiri dari dua periode, yaitu makan atau berpikir. Ketika seorang filosof lapar, dia berusaha untuk mendapatkan garpu kiri dan garpu kanan sekaligus. Jika sukses dalam mengambil dua

garpu, filosof tersebut makan untuk sementara waktu, kemudian meletakkan kedua garpu dan melanjutkan berpikir.

Pertanyaan kuncinya adalah, dapatkah anda menulis program untuk masing-masing filosof yang melakukan apa yang harus mereka lakukan dan tidak pernah mengalami kebuntuan.

Prosedur take-fork menunggu sampai garpu-garpu yang sesuai didapatkan dan kemudian menggunakannya. Sayangnya dari solusi ini ternyata salah. Seharusnya lima orang filosof mengambil garpu kirinya secara bersamaan. Tidak akan mungkin mereka mengambil garpu kanan mereka, dan akan terjadi deadlock.

Kita dapat memodifikasi program sehingga setelah mengambil garpu kiri, program memeriksa apakah garpu kanan memungkinkan untuk diambil. Jika garpu kanan tidak mungkin diambil, filosof tersebut meletakkan kembali garpu kirinya, menunggu untuk beberapa waktu, kemudian mengulangi proses yang sama. Usulan tersebut juga salah, walau pun dengan alasan yang berbeda. Dengan sedikit nasib buruk, semua filosof dapat memulai algoritma secara bersamaan, mengambil garpu kiri mereka, melihat garpu kanan mereka yang tidak mungkin untuk diambil, meletakkan kembali garpu kiri mereka, menunggu, mengambil garpu kiri mereka lagi secara bersamaan, dan begitu seterusnya.

Situasi seperti ini dimana semua program terus berjalan secara tidak terbatas tetapi tidak ada perubahan/kemajuan yang dihasilkan disebut starvation. Sekarang anda dapat berpikir "jika filosof dapat saja menunggu sebuah waktu acak sebagai pengganti waktu yang sama setelah tidak dapat mengambil garpu kiri dan kanan, kesempatan bahwa segala sesuatu akan berlanjut dalam kemandegan untuk beberapa jam adalah sangat kecil." Pemikiran seperti itu adalah benar, tapi beberapa aplikasi mengirimkan sebuah solusi yang selalu bekerja dan tidak ada kesalahan tidak seperti hsk nomor acak yang selalu berubah.

Sebelum mulai mengambil garpu, seorang filosof melakukan DOWN di mutex. Setelah menggantikan garpu dia harus melakukan UP di mutex. Dari segi teori, solusi ini cukup memadai. Dari segi praktek, solusi ini tetap memiliki masalah. Hanya ada satu filosof yang dapat makan spageti dalam berbagai kesempatan. Dengan lima buah garpu, seharusnya kita bisa menyaksikan dua orang filosof makan spageti pada saat bersamaan.

Solusi yang diberikan diatas benar dan juga mengizinkan jumlah maksimum kegiatan paralel untuk sebuah jumlah filosof yang berubah-ubah ini menggunakan sebuah array, state, untuk merekam status seorang filosof apakah sedang makan (eating), berpikir (think), atau sedang lapar (hungry) karena sedang berusaha mengambil garpu. Seorang filosof hanya dapat berstatus makan (eating) jika tidak ada tetangganya yang sedang makan juga. Tetangga seorang filosof didefinisikan oleh LEFT dan RIGHT.

Dengan kata lain, jika $i = 2$, maka tetangga kirinya (LEFT) = 1 dan tetangga kanannya (RIGHT) = 3. Program ini menggunakan sebuah array dari semaphore yang lapar (hungry) dapat ditahan jika garpu kiri atau kanannya sedang dipakai tetangganya. Catatan bahwa masing-masing proses menjalankan prosedur filosof sebagai kode utama, tetapi prosedur yang lain seperti take-forks, dan test adalah prosedur biasa dan bukan proses-proses yang terpisah.

5.9. Monitors

Solusi sinkronisasi ini dikemukakan oleh Hoare pada tahun 1974. Monitor adalah kumpulan prosedur, variabel dan struktur data di satu modul atau paket khusus. Proses dapat memanggil prosedur-prosedur kapan pun diinginkan. Tapi proses tak dapat mengakses struktur data internal dalam monitor secara langsung. Hanya lewat prosedur-prosedur yang dideklarasikan minitor untuk mengakses struktur internal.

Properti-properti monitor adalah sebagai berikut:

1. Variabel-variabel data lokal, hanya dapat diakses oleh prosedur-prosedur dalam monitor dan tidak oleh prosedur di luar monitor.
2. Hanya satu proses yang dapat aktif di monitor pada satu saat. Kompilator harus mengimplementasi ini (mutual exclusion).
3. Terdapat cara agar proses yang tidak dapat berlangsung di-blocked. Menambahkan variabel-variabel kondisi, dengan dua operasi, yaitu Wait dan Signal.
4. Wait: Ketika prosedur monitor tidak dapat berlanjut (misal producer menemui buffer penuh) menyebabkan proses pemanggil diblocked dan mengizinkan proses lain masuk monitor.
5. Signal: Proses membangunkan partner-nya yang sedang diblocked dengan signal pada variabel kondisi yang sedang ditunggu partner-nya.
6. Versi Hoare: Setelah signal, membangunkan proses baru agar berjalan dan menunda proses lain.
- vii. Versi Brinch Hansen: Setelah melakukan signal, proses segera keluar dari monitor.

Dengan memaksakan disiplin hanya satu proses pada satu saat yang berjalan pada monitor, monitor menyediakan fasilitas mutual exclusion. Variabel-variabel data dalam monitor hanya dapat diakses oleh satu proses pada satu saat. Struktur data bersama dapat dilindungi dengan menempatkannya dalam monitor. Jika data pada monitor merepresentasikan sumber daya, maka monitor menyediakan fasilitas mutual exclusion dalam mengakses sumber daya itu.

6. Keamanan Sistem Operasi

Sejauh perangkat lunak yang paling penting pada komputer adalah sistem operasi. Dalam sebuah jaringan komputer multi-user, adalah penting untuk keamanan jaringan bahwa sistem operasi dengan kontrol keamanan yang tepat ditempatkan pada kedua server dan komputer klien. keamanan perangkat keras saja tidak efektif dalam jaringan jika, misalnya, Microsoft Windows 95 yang akan digunakan sebagai sistem operasi pada semua mesin. Versi sistem operasi Microsoft Windows 95, 98 dan ME dirancang untuk digunakan di rumah dan seharusnya tidak pernah digunakan dalam jaringan bisnis karena kurangnya stabilitas, kehandalan dan, di atas semua dalam hal ini, keamanan. Mayoritas bisnis menggunakan Microsoft dan UNIX berbasis sistem, tetapi untuk tujuan ilustrasi di sini, dua sistem operasi multi-user kami akan memeriksa keamanan-bijaksana adalah Microsoft Windows 2000 dan Linux.

6.1. Microsoft Windows 2000

Sistem operasi pada umumnya dan versi Microsoft Windows secara khusus terus-menerus dikaji. Di sini kita telah meninjau Windows 2000 sebagai sistem operasi untuk jaringan sebagai contoh dari jenis fitur keamanan yang dapat ditemukan di ini dan sistem operasi berbasis Windows modern.

6.2. File System Security

Salah satu langkah pertama dalam membangun keamanan jaringan dalam lingkungan Windows 2000 adalah untuk memastikan sistem file pada semua komputer yang aman. Windows 2000 menyediakan pilihan menggunakan FAT32 atau NTFS file sistem. FAT32 (32-bit genteng tabel alokasi) adalah sistem file yang digunakan oleh Windows 98 dan ME. Ia tidak memiliki fitur untuk mengontrol akses pengguna ke data dan dengan demikian tidak dianjurkan oleh Microsoft sebagai sistem file untuk digunakan dalam bisnis. NTFS (New Technology File System) adalah sistem file yang digunakan oleh sistem operasi Windows NT berbasis dan

sistem file yang direkomendasikan dalam semua situasi bisnis, tidak hanya untuk dukungan untuk berkas dan akses folder permissions tetapi juga karena diperlukan oleh Windows 2000 Server jika Active Directory akan dipasang. NTFS juga diperlukan untuk EFS (Encrypting File System), yang tersedia pada Windows 2000 dan keatas (Todd & Johnson 2001).

Encrypting File System, seperti namanya, memungkinkan setiap bagian (atau semua, jika diinginkan) dari sistem file yang akan dienkrpsi pada permintaan. NTFS mengontrol akses ke volume ketika sedang digunakan oleh sistem, tetapi tidak efektif ketika volume dihapus dan ditempatkan ke komputer lain. EFS adalah ukuran keamanan sangat penting dalam hal ini karena jika keamanan fisik ke dalam sistem komputer dilanggar, file dienkrpsi, membuat penemuan data aktual sangat sulit.

6.3. User Accounts

Menjadi sistem operasi multi-user, Windows 2000 memberikan administrator kemampuan untuk memberikan setiap pengguna dalam organisasi akun pengguna mereka sendiri sehingga pengguna dapat mengakses sistem komputer dengan identitasnya sendiri dan semua orang dari luar yang hampir terkunci. 'Akun' A pengguna berisi informasi tentang pengguna, terutama tiga hal penting:

- account 'username' yang digunakan oleh pengguna untuk memberitahu sistem yang mereka gunakan
- account 'password', yang merupakan kunci akses bahwa pengguna menggunakan untuk membuktikan identitas mereka ke sistem
- ID keamanan 37-karakter (SID), yang dimulai dengan karakter 'S' dan diikuti oleh 36 digit, diatur dalam kelompok dan dipisahkan oleh tanda hubung. SID adalah unik untuk account pengguna tertentu.

6.4. Password Security

Dalam semua sistem operasi Windows NT (termasuk Windows 2000), otentikasi pengguna telah diperlukan penggunaan username dan password; password dasarnya menjadi kunci untuk akun pengguna. password terenkripsi tidak pernah dikirim melalui jaringan saat log on ke sistem Windows 2000 (kecuali untuk layanan tidak aman seperti FTP dan Telnet). Semua informasi password disimpan sebagai hash password pada file Windows 2000, yang tidak dapat diakses oleh setiap pengguna (termasuk administrator) ketika sistem berjalan. Dalam sebuah jaringan bisnis yang aman menjalankan jaringan berbasis Windows 2000, sangat penting bahwa semua password pengguna tetap aman, terlepas dari hak akses pengguna. Windows 2000 kontrol kebijakan keamanan menyediakan opsi berikut untuk membantu memastikan bahwa password pengguna adalah aman:

- 'Menegakkan sejarah sandi.' Opsi ini memungkinkan administrator untuk menentukan jumlah password yang sistem ingat sehingga password tidak dapat digunakan kembali.
- 'Usia sandi maksimal.' Jumlah maksimum hari sebelum sistem memaksa pengguna untuk mengubah password.
- 'Panjang password minimal.' jumlah minimum yang diperlukan karakter dalam password pengguna.
- 'Password harus memenuhi persyaratan kompleksitas.' Jika kebijakan ini adalah pada, password harus memenuhi persyaratan sebagai berikut: tidak mengandung semua atau bagian dari nama akun pengguna; setidaknya enam karakter; dan berisi karakter dari setidaknya tiga dari empat kategori berikut:
 - karakter bahasa Inggris huruf besar (A sampai Z)
 - karakter huruf kecil Inggris (A sampai z)
 - basis 10 digit (0 sampai 9)
 - karakter non-alfanumerik (misalnya *, !, \$, #, %)

6.5. Account Lockout Policy

Sama seperti mesin teller otomatis menangkap kartu setelah tiga upaya mengatasi masuknya PIN yang salah. Windows 2000 dapat menonaktifkan akun pengguna setelah sejumlah tertentu upaya masuk sandi yang salah.

- 'Rekening durasi lockout.' Lamanya waktu account terkunci setelah jumlah usaha entri kata sandi yang salah mencapai ambang batas. Jika ini diatur ke 0, maka sisa-sisa akun terkunci tanpa batas waktu sampai seorang administrator membuka itu.
- 'Rekening ambang lockout.' Jumlah upaya sandi tidak valid sebelum account terkunci keluar.
- 'Reset account lockout kontra setelah ...' Lamanya waktu sebelum jumlah tercatat upaya entri kata sandi tidak valid ulang ke nol.

6.6. Security Audits

Audit keamanan pada dasarnya adalah log dari setiap tindakan yang pengguna lakukan dan tindakan komputer itu sendiri secara otomatis. Secara default, sistem log hanya dapat diakses oleh administrator. Windows 2000 memungkinkan administrator untuk mengaudit kegiatan sebagai berikut:

- Peristiwa account logon: ketika pengguna log on atau off di mana komputer mempertahankan audit adalah komputer yang memvalidasi username dan password
- manajemen account: ketika account pengguna yang dimodifikasi
- akses layanan direktori: benda diakses di Windows 2000 Active Directory
- Peristiwa account logon: ketika pengguna log on atau off, atau membuat koneksi jaringan
- akses objek: ketika pengguna mengakses sebuah 'objek' pada sistem (misalnya yang membutuhkan penggunaan layanan)

- perubahan kebijakan: ketika kebijakan audit dimodifikasi
- keistimewaan penggunaan: ketika pengguna memanfaatkan setiap hak-hak istimewa
- pelacakan proses: ketika pengguna berinteraksi dengan proses
- kejadian sistem: ketika sistem melakukan kegiatan yang biasanya akan ditempatkan di event log. kejadian sistem audit juga dapat digunakan bersama dengan pilihan keamanan 'Shut down sistem segera jika tidak dapat menulis sebuah acara untuk log keamanan' untuk mencegah serangan DOS, yang biasanya akan mengisi log keamanan agak cepat.
- Akses ke file dan folder juga dapat diaudit dengan mengubah file atau informasi keamanan folder. Ini mensyaratkan bahwa sistem ini menggunakan sistem file NTFS.

6.7. User Rights Assignment Policy

Pilihan kebijakan penetapan hak pengguna memungkinkan administrator untuk menentukan pengguna yang memiliki hak untuk melakukan tindakan yang lebih spesifik, selain yang diperbolehkan oleh kelompok yang pengguna tertentu milik. Perubahan yang dilakukan pada pilihan ini mengontrol apakah setiap pengguna tertentu dapat:

- mengakses sistem melalui jaringan dengan menggunakan username dan password mereka
- bertindak sebagai bagian dari sistem operasi, memberikan pengguna kemampuan untuk melakukan hampir tindakan apapun pada sistem, seolah-olah mereka menggunakan sistem berbasis Windows 95
- lokal log on ke sistem
- mematikan atau restart sistem (lokal atau jarak jauh)
- melakukan perawatan pada volume hard drive
- menginstal, uninstall dan memperbarui driver perangkat

- menyesuaikan penggunaan memori
- mengubah jam sistem
- mengambil kepemilikan file.

Beberapa pilihan di bagian ini (misalnya 'login secara lokal' atau 'mengakses komputer ini dari jaringan') juga memiliki 'menyangkal' pilihan yang sesuai. Jika pengguna ditempatkan baik di 'memungkinkan' dan 'menyangkal' pilihan, maka 'menyangkal' diutamakan.

6.8. Additional Security Options

Kontrol kebijakan keamanan juga menyediakan satu set opsi keamanan tidak tercakup di tempat lain. Pilihan ini meliputi:

- Mengganti nama administrator atau tamu account (yang juga dapat dilakukan dari dalam 'pengguna lokal dan kelompok' kontrol untuk server yang berdiri sendiri dan workstation atau dari dalam 'pengguna direktori aktif dan komputer' untuk kontroler domain dan anggota domain). Mengubah nama account administrator memberikan kesempatan untuk membuat account dummy yang disebut 'administrator' (tanpa hak akses), yang kemudian dapat dipantau untuk setiap aktivitas yang mencurigakan.
- mencegah pengguna dari mengakses sistem melalui jaringan
- memaksa pengguna log off atau pemutusan ketika jam logon diizinkan mereka berakhir
- mencegah sistem dari yang ditutup tanpa harus login
- mencegah username dari pengguna terakhir pada sistem yang ditampilkan dalam kotak 'username' ketika pengguna berikutnya log on
- apakah 'Semua orang' izin akses berlaku untuk pengguna anonim
- mencegah instalasi driver yang belum ditandatangani secara digital

6.9. Active Directory

Active Directory adalah nama yang diberikan ke layanan direktori dalam Windows 2000 yang diinstal ketika server 'dipromosikan' ke kontroler domain. Seperti pengontrol domain Windows NT, ini menyediakan sentralisasi keamanan sehingga account pengguna lokal pada setiap komputer client (yang dapat menimbulkan sebagai risiko keamanan) tidak diperlukan dan dapat dinonaktifkan. Active Directory juga memungkinkan penciptaan terpisah 'unit organisasi' untuk pengelompokan benda Active Directory, termasuk account pengguna, folder bersama dan printer.

Active Directory berisi fitur yang disebut 'operasi master'. Fitur ini memungkinkan Windows 2000 server yang menjalankan layanan direktori untuk mendelegasikan peran yang lain. Menggunakan fitur ini, adalah mungkin untuk membuat beberapa domain controller, masing-masing bertanggung jawab untuk melakukan tugas yang berbeda. Ini berarti bahwa jika satu server adalah korban serangan, yang lain dapat terus melakukan peran mereka dan mendukung bisnis. Peran korban server dapat segera diganti jika memiliki master operasi backup yang sesuai. Biasanya, ini semacam lingkungan akan sulit untuk mengelola; Namun, alat administrasi Active Directory memungkinkan setiap master operasi untuk diberikan dari satu konsol.

Fitur keamanan tambahan dari Active Directory termasuk protokol otentikasi Kerberos dan kemampuan untuk menjalankan protokol layanan direktori (LDAP) lebih Secure Sockets Layer (SSL) sehingga transaksi direktori dapat dilakukan melalui saluran aman (misalnya untuk digunakan lebih extranet atau untuk tujuan e-commerce). Beralih layanan direktori untuk 'mode asli' (yang tersedia hanya ketika semua kontroler domain adalah Windows 2000 Server) memperkuat keamanan untuk Windows 2000 tingkat di seluruh domain. Salah satu manfaat dari ini adalah bahwa pengguna anonim tidak dapat membaca informasi pada domain.

6.10. Linux

Linux adalah sistem operasi open-source, sehingga source code-nya tersedia secara bebas dalam domain publik. Patch untuk masalah dapat dibuat oleh organisasi, daripada menunggu orang lain untuk menemukan bug dan merilis patch. Sebagai kode sumber tersedia untuk umum, sistem operasi dapat dimasukkan melalui kriteria evaluasi keamanan bersama oleh organisasi itu sendiri, daripada harus menunggu sekelompok individu di luar perusahaan untuk melakukan evaluasi ini.

Linux juga merupakan sistem operasi multi-user yang, seperti Windows 2000, mendukung berkas dan akses folder permissions. Namun, karena Linux adalah sistem operasi yang sama sekali berbeda, prinsip-prinsip keamanan ini bekerja sedikit berbeda; Selanjutnya, ada daerah yang berbeda di mana keamanan dapat disesuaikan. Kita sekarang akan memeriksa sejumlah daerah di mana Linux memungkinkan keamanan diperkuat.

6.11. Boot Loader

Sandi melindungi Linux boot loader mencegah akses tidak sah ke mode single-user (di mana pengguna menjadi pengguna root ') dan mencegah pemuatan sistem operasi lain (jika menjalankan sistem dual-boot), yang mungkin kurang aman. Juga, jika GRUB(Grand Unified Bootloader,) adalah boot loader, proteksi password mencegah akses tidak sah ke konsol GRUB, di mana opsi konfigurasi dapat diubah. Mengamankan setiap boot loader ini membutuhkan akses root.

6.12. Password

Untuk keamanan password terbaik, Linux secara otomatis akan memilih MD5 dan shadow password pada saat instalasi. MD5 adalah metode enkripsi yang lebih aman, yang juga memungkinkan karakter non-alfanumerik di password. Jika opsi ini tidak dipilih, Linux menggunakan enkripsi DES, yang memungkinkan karakter hanya alpha-numerik dan enkripsi

56-bit. Password shadowing penting untuk keamanan; jika hash password yang tersisa di file password, yang dapat dibaca untuk semua orang dan rentan terhadap serangan brute force attack off-site. Jika password shadowing diaktifkan, hash password disimpan dalam file password shadow, yang dapat dibaca hanya oleh root.

PAM (Pluggable Authentication Module) bekerja dengan cara yang mirip dengan kebijakan sandi Windows 2000 dan disesuaikan sehingga administrator dapat menentukan persyaratan sandi. Saat pengguna mengubah password mereka menggunakan perintah `passwd`, cek `passwd` PAM untuk melihat apakah password sesuai dengan persyaratan yang ditentukan. Linux juga memiliki password penuaan kebijakan, yang memungkinkan administrator untuk menentukan jumlah maksimum waktu password dapat tetap digunakan pada account.

6.13. Limiting Use of The 'root' Account

Secara default, pengguna root 'hanya dapat mengakses sistem Linux lokal. File `/etc/security` mengontrol bahwa komputer account root dapat digunakan pada. Jika file ini kosong, user root tidak bisa login sama sekali (kecuali melalui SSH-Secure Shell)/ secara remote. Jika file ini tidak ada, user root dapat login dari mana saja, yang berpotensi tidak aman. pilihan lebih lanjut ada yang dirancang untuk membatasi penggunaan account root, seperti mengubah shell root dan mencegah login SSH.

6.14. Services

Untuk keamanan, sebagian besar layanan jaringan di Linux dimatikan secara default. Beberapa pengecualian termasuk 'lpd' (print server), 'portmap' (wajib), 'xinetd', 'sendmail' dan 'sshd' (secure shell - pengganti telnet).

Hal ini juga harus dicatat bahwa meskipun sshd adalah layanan inheren aman, harus terus up to date untuk membantu mencegah ancaman keamanan.

Sendmail adalah agen transfer pesan tidak aman dan dapat rentan terhadap sejumlah serangan. Namun, ada beberapa pilihan yang dapat digunakan untuk mengamankan itu:

- Konfigurasi sendmail dapat dimodifikasi untuk menentukan sejumlah koneksi bersamaan, layanan anak, dan header dan ukuran pesan. Ini tidak diatur secara default dan meninggalkan layanan sendmail terbuka untuk serangan DOS.
- Direktori spool email dapat disimpan pada volume non-NFS untuk melindungi privasi dan keamanan pesan email dari pengguna lain.
- Jika layanan sendmail berjalan pada mesin yang tidak mengharuskan pengguna untuk mengakses shell, maka shell akses dapat ditolak.
- Penggunaan pihak ketiga port scanner alat 'nmap' dapat digunakan untuk memonitor port terbuka, dan setiap port terbuka yang mencurigakan kemudian dapat diselidiki. Port terbuka menjalankan layanan diindikasikan sebagai 'tidak diketahui', meskipun tidak selalu menyeramkan, masih layak perhatian.

6.15. Firewall

Linux mencakup perangkat untuk menciptakan firewall sederhana. Alat konfigurasi tingkat keamanan dan GNOME(*GNU Network Object Model Environment*) lokkit adalah alat berbasis GUI yang membuat aturan 'iptables'. 'Iptables' adalah antarmuka untuk subsistem 'netfilter' dari Linux, yang menyediakan paket pilihan penyaringan. Baris perintah penggunaan 'iptables' menyediakan akses ke fungsi seperti yang menyamar IP, NAT(Network Address Translation) dan larangan IP, dan juga dapat digunakan untuk mengontrol akses jaringan untuk layanan tertentu. Ini dapat digunakan untuk mengamankan layanan 'portmap' dan layanan RPC(Remote Procedure Call) lain dengan mekanisme keamanan yang lemah.

Alat GUI memberikan konfigurasi tujuan umum biasanya direkomendasikan untuk digunakan di rumah, tapi yang terbaik set-up untuk sebuah bisnis adalah dengan menggunakan

alat baris perintah. Meskipun mereka lebih kompleks, mereka menyediakan fungsionalitas terbesar.

7. Antivirus Software

Meskipun hardware dapat memberikan keamanan dalam bentuk firewall, itu tidak memiliki kemungkinan teknis memberikan perlindungan terhadap virus. Virus berbahaya (atau hanya mengganggu) bagi perangkat lunak yang menambahkan diri ke file yang sah, program atau email, menunggu untuk diaktifkan oleh pengguna yang tidak curiga.

Setelah virus ganas diaktifkan, tujuannya biasanya untuk menghancurkan data sebanyak itu bisa dalam waktu singkat. Hardware gagal di sini terutama karena virus baru ditemukan setiap hari. Meskipun hardware dapat dirancang untuk mengambil virus yang ada, dalam beberapa hari perangkat keras lama akan kadaluarsa. Membuat hardware 'diupdate' juga menyajikan masalah, karena virus dapat dirancang untuk menyerang sirkuit diprogram dan membuat unit tidak berguna.

Virus jauh dari kemungkinan akan benar-benar diberantas, sehingga perangkat lunak telah dirancang untuk memindai file, program, email dan seterusnya untuk virus sebelum mereka menjadi aktif. Dua perusahaan terkenal yang mengembangkan perangkat lunak antivirus yang Symantec dan McAfee. Masing-masing perusahaan telah menghasilkan solusi antivirus untuk perusahaan, termasuk produk terpisah yang dirancang untuk sistem klien dan server. perusahaan antivirus yang terus menerus waspada terhadap virus baru dan secara rutin merilis update paket antivirus mereka. Pembaruan ini perlu diperoleh secara teratur (idealnya segera setelah mereka dilepaskan) dan dipasang sehingga penjaga software antivirus perusahaan terhadap virus baru.

8. Manajemen File

File mempunyai sifat sbb:

1. Persistence.

Informasi dapat bertahan meski proses yang membangkitkannya berakhir atau meskipun catu daya dihilangkan.

2. Size.

File umumnya berukuran besar. Memungkinkan menyimpan informasi yang sangat besar.

3. Sharability.

File dapat digunakan banyak proses untuk mengakses informasi secara kongkuren.

8.1. Sasaran Manajemen File.

- Pengelolaan file adalah kumpulan perangkat lunak sistem yang menyediakan layanan-layanan berhubungan dengan penggunaan file ke pemakai dan/atau aplikasi. Biasanya, satu-satunya cara pemakai atau aplikasi mengakses file adalah lewat sistem file.
- Pemakai atau pemrogram tidak perlu mengembangkan perangkat lunak khusus untuk mengakses data di tiap aplikasi.

Grosshan [GRO-86] menyarankan sasaran sistem file sbb:

1. Memenuhi kebutuhan manajemen data bagi pemakai.

Kebutuhan manajemen data bagi pemakai, yaitu kemampuan melakukan operasi-operasi.

2. Menjamin data pada file adalah valid.

3. Optimasi kinerja.
 - Menurut sistem yaitu : meningkatkan jumlah throughput keseluruhan.
 - Menurut pemakai yaitu : cepatnya waktu tanggap.
4. Menyediakan dukungan masukan/keluaran beragam tipe perangkat penyimpan.
5. Meminimalkan atau mengeliminasi potensi kehilangan atau perusakan data.
6. Menyediakan sekumpulan rutin interface masukan/keluaran.
7. Menyediakan dukungan masukan/keluaran banyak pemakai di sistem multiuser.

8.2. Fungsi Manajemen File.

Beberapa fungsi yang diharapkan dari pengelolaan file adalah:

1. Penciptaan, modifikasi dan penghapusan file.
2. Mekanisme pemakaian file secara bersama.
3. Kemampuan backup dan recovery untuk mencegah kehilangan karena kecelakaan atau dari upaya penghancuran informasi.
4. Pemakai dapat mengacu file dengan nama simbolik (symbolic name) bukan menggunakan penamaan yang mengacu perangkat fisik.
5. Pada lingkungan sensitif dikehendaki informasi tersimpan aman dan rahasia.

Lingkungan ini seperti :

- a. Electronic fund transfer system.
- b. Criminal record system.
- c. Medical record system.

Sistem file menyediakan enkripsi dan dekripsi untuk menjaga informasi hanya digunakan oleh pemakai yang diotorisasi saja.

6. Sistem file harus menyediakan interface user-friendly.

a. Pandangan secara logik (logical view) bukan pandangan secara fisik (physical view) terhadap data.

b. Fungsi yang dapat dilakukan terhadap data.

Pemakai tidak berkuat pada perangkat fisik dimana data disimpan, bentuk data harus diambil dari perangkat atau cara-cara fisik transfer data ke/dari perangkat- perangkat itu.

8.3. Arsitektur Pengelolaan File

Pengelolaan file terdiri dari :

1. Sistem akses.

Berkaitan dengan bagaimana cara pengaksesan data yang disimpan pada file.

2. Manajemen file.

Berkaitan dengan penyediaan mekanisme operasi pada file, seperti:

a. Penyimpanan

b. Pengacuan.

c. Pemakaian bersama.

d. Pengamanan.

3. Manajemen ruang penyimpanan.

Berkaitan dengan alokasi ruang untuk file di perangkat penyimpan.

4. Mekanisme integritas file.

Berkaitan dengan jaminan informasi pada file tak terkorupsi.

