Bab 7

MANAJEMEN PERANGKAT MASUKAN/KELUARAN

7.1. Pendahuluan

Manajemen perangkat masukan/keluaran merupakan aspek perancangan sistem operasi terluas dan kompleks karena sangat beragamnya perangkat dan aplikasinya.

Beberapa fungsi manajemen input/ouput (I/O) :

a. Mengirim perintah ke perangkat I/O agar menyediakan layanan.

b. Menangani interupsi perangkat I/O.

c. Menangani kesalahan perangkat I/O.

d. Menyediakan interface ke pemakai.

7.2. Klasifikasi perangkat I/O

Perangkat I/O dapat dikelompokkan berdasarkan :

a. Sifat aliran datanya, yang terbagi atas :

a.1 Perangkat berorientasi blok.

Yaitu menyimpan, menerima, dan mengirim informasi sebagai blok-blok berukuran tetap yang berukuran 128 sampai 1024 byte dan memiliki alamat tersendiri, sehingga memungkinkan membaca atau menulis blok-blok secara independen, yaitu dapat membaca atau menulis sembarang blok tanpa harus melewati blok-blok lain. Contoh : disk,tape,CD ROM, optical disk.

a.2 Perangkat berorientasi aliran karakter.

Yaitu perangkat yang menerima, dan mengirimkan aliran karakter tanpa membentuk suatu struktur blok. Contoh : terminal, line printer, pita kertas, kartu-kartu berlubang, interface jaringan, mouse.

b. Sasaran komunikasi, yang terbagi atas :.

b.1 Perangkat yang terbaca oleh manusia.

Perangkat yang digunakan untuk berkomunikasi dengan manusia.

Contoh : VDT (video display terminal) : monitor, keyboard, mouse.

b.2 Perangkat yang terbaca oleh mesin.

Perangkat yang digunakan untuk berkomunikasi dengan perangkat elektronik.

Contoh : Disk dan tape, sensor, controller.

b.3 Perangkat komunikasi.

Perangkat yang digunakan untuk komunikasi dengan perangkat jarak jauh.

Contoh : Modem.

Faktor-faktor yang membedakan antar perangkat :

- Kecepatan transmisi data (data rate).
- Jenis aplikasi yang digunakan.
- Tingkat kerumitan dalam pengendalian.
- Besarnya unit yang ditransfer.
- Representasi atau perwujudan data.
- Kondisi-kondisi kesalahan.

7.3. Teknik pemograman perangkat I/O

Terdapat 3 teknik, yaitu :

a. I/O terprogram atau polling system.

Ketika perangkat I/O menangani permintaan, perangkat men-set bit status di register status perangkat. Perangkat tidak memberitahu ke pemroses saat tugas telah selesai dilakukan sehingga pemroses harus selalu memeriksa register tersebut secara periodik dan melakukan tindakan berdasar status yang dibaca. Software pengendali perangkat (driver) dipemroses harus mentransfer data ke/dari pengendali. Driver mengekseksui perintah yang berkomunikasi dengan pengendali (adapter) di perangkat dan menunggui sampai operasi yang dilakukan perangkat selesai.

Driver berisi kumpulan instruksi :

a.1 Pengendalian.

Berfungsi mengaktifkan perangkat eksternal dan memberitahu yang perlu dilakukan. Contoh : unit tape magnetik diinstruksikan untuk kembali ke posisi awal, bergerak ke record berikut, dan sebagainya.

a.2 Pengujian.

Berfungsi memeriksa status perangkat keras berkaitan dengan perangkat I/O.

a.3 Pembacaan/penulisan

Berfungsi membaca/menulis untuk transfer data antara register pemroses dan perangkat eksternal.

Masalah utama I/O terprogram adalah pemroses diboroskan untuk menunggu dan menjagai operasi I/O. Diperlukan teknik lain untuk meningkatkan efisiensi pemroses.

b. I/O dikendalikan interupsi.

Teknik I/O dituntun interupsi mempunyai mekanisme kerja sebagai berikut :

- Pemroses memberi instruksi ke perangkat I/O kemudian melanjutkan melakukan pekerjaan lainnya.
- Perangkat I/O akan menginterupsi meminta layanan saat perangkat telah siap bertukar data dengan pemroses.
- Saat menerima interupsi perangkat keras (yang memberitahukan bahwa perangkat siap melakukan transfer), pemroses segera mengeksekusi transfer data.

Keunggulan :

- Pemroses tidak disibukkan menunggui dan menjaga perangkat I/O untuk memeriksa status perangkat.

Kelemahan :

- Rate transfer I/O dibatasi kecepatan menguji dan melayani operasi perangkat.
- Pemroses terikat ketat dalam mengelola transfer I/O. Sejumlah intruksi harus dieksekusi untuk tiap transfer I/O.

c. Dengan DMA (direct memory access).

DMA berfungsi membebaskan pemroses menunggui transfer data yang dilakukan perangkat I/O. Saat pemroses ingin membaca atau menulis data, pemroses memerintahkan DMA controller dengan mengirim informasi berikut :

- Perintah penulisan/pembacaan.
- Alamat perangkat I/O.
- Awal lokasi memori yang ditulis/dibaca.
- Jumlah word (byte) yang ditulis/dibaca.

Setelah mengirim informasi-informasi itu ke DMA controller, pemroses dapat melanjutkan kerja lain. Pemroses mendelegasikan operasi I/O ke DMA. DMA mentransfer seluruh data yang diminta ke/dari memori secara langsung tanpa melewati pemroses. Ketika transfer data selesai, DMA mengirim sinyal interupsi ke pemroses. Sehingga pemroses hanya dilibatkan pada awal dan akhir transfer data. Operasi transfer antara perangkat dan memori utama dilakukan sepenuhnya oleh DMA lepas dari pemroses dan hanya melakukan interupsi bila operasi telah selesai.

Keunggulan :

- Penghematan waktu pemroses.
- Peningkatan kinerja I/O.

7.4. Evolusi fungsi perangkat I/O

Sistem komputer mengalami peningkatan kompleksitas dan kecanggihan komponen-komponennya, yang sangat tampak pada fungsi-fungsi I/O sebagai berikut :

a. Pemroses mengendalikan perangkat I/O secara langsung.

Masih digunakan sampai saat ini untuk perangkat sederhana yang dikendalikan mikroprosessor sehingga menjadi perangkat berintelijen (inteligent device).

b. Pemroses dilengkapi pengendali I/O (I/O controller).

Pemroses menggunakan I/O terpogram tanpa interupsi, sehingga tak perlu memperhatikan rincian-rincian spesifik antarmuka perangkat.

c. Perangkat dilengkapi fasilitas interupsi.

Pemroses tidak perlu menghabiskan waktu menunggu selesainya operasi I/O, sehingga meningkatkan efisiensi pemroses.

d. I/O controller mengendalikan memori secara langsung lewat DMA.

Pengendali dapat memindahkan blok data ke/dari memori tanpa melibatkan pemroses kecuali diawal dan akhir transfer.

e. Pengendali I/O menjadi pemroses terpisah.

Pemroses pusat mengendalikan.memerintahkan pemroses khusus I/O untuk mengeksekusi program I/O di memori utama. Pemroses I/O mengambil dan mengeksekusi intruksi-intruksi ini tanpa intervensi pemroses pusat.

Dimungkinkan pemroses pusat menspesifikasikan barisan aktivitas I/O dan hanya diinterupsi ketika seluruh barisan intruksi diselesaikan.

f. Pengendali I/O mempunyai memori lokal sendiri.

Perangkat I/O dapat dikendalikan dengan keterlibatan pemroses pusat yang minimum.

Arsitektur ini untuk pengendalian komunikasi dengan terminal-terminal interaktif. Pemroses I/O mengambil alih kebanyakan tugas yang melibatkan pengendalian terminal.

Evolusi bertujuan meminimalkan keterlibatan pemroses pusat, sehingga pemroses tidak disibukkan dengan tugas I/O dan dapat meningkatkan kinerja sistem.

7.5. Prinsip manajemen perangkat I/O

Terdapat dua sasaran perancangan I/O, yaitu :

a. Efisiensi.

Aspek penting karena operasi I/O sering menimbulkan bottleneck.

b. Generalitas (device independence).

Manajemen perangkat I/O selain berkaitan dengan simplisitas dan bebas kesalahan, juga menangani perangkat secara seragam baik dari cara proses memandang maupun cara sistem operasi mengelola perangkat dan operasi I/O.

Software diorganisasikan berlapis. Lapisan bawah berurusan menyembunyikan kerumitan perangkat keras untuk lapisan-lapisan lebih atas. Lapisan lebih atas berurusan memberi antar muka yang bagus, bersih, nyaman dan seragam ke pemakai.

Masalah-masalah manajemen I/O adalah :

a. Penamaan yang seragam (uniform naming).
   Nama berkas atau perangkat adalah string atau integer, tidak bergantung pada perangkat sama sekali.
b. Penanganan kesalahan (error handling).
   Umumnya penanganan kesalahan ditangani sedekat mungkin dengan perangkat keras.
c. Transfer sinkron vs asinkron.
   Kebanyakan I/O adalah asinkron. Pemroses mulai transfer dan mengabaikan untuk melakukan kerja lain sampai interupsi tiba. Program pemakai sangat lebih mudah ditulis jika operasi I/O berorientasi blok. Setelah perintah read, program kemudian ditunda secara otomatis sampai data tersedia di buffer.
d. Sharable vs dedicated.
   Beberapa perangk dapat dipakai bersama seperti disk, tapi ada juga perangkat yang hanya satu pemakai yang dibolehkan memakai pada satu saat.
   Contoh : printer.

7.6. Hirarki manajemen perangkat I/O

Hirarki manajemen perangkat I/O :

a. Interrupt handler.
   Interupsi harus disembunyikan agar tidak terlihat rutin berikutnya.
   Device driver di blocked saat perintah I/O diberikan dan menunggu interupsi. Ketika interupsi terjadi, prosedur penanganan interupsi bekerja agar device driver keluar dari state blocked.

b. Device drivers.
   Semua kode bergantung perangkat ditempatkan di device driver. Tiap device driver menangani satu tipe (kelas) perangkat dan bertugas menerima permintaan

abstrak perangkat lunak device independent diatasnya dan melakukan layanan permintaan.

Mekanisme kerja device driver :

- Menerjemahkan perintah abstrak menjadi perintah konkret.
- Setelah ditentukan perintah yang harus diberikan ke pengendali, device driver mulai menulis ke register-register pengendali perangkat.
- Setelah operasi selesai dilakukan perangkat, device driver memeriksa status kesalahan yang terjadi.
- Jika berjalan baik, device driver melewatkan data ke perangkat lunak device independent.
- Kemudian device driver melaporkan status operasinya ke pemanggil.

c. Perangkat lunak device independent.

Bertujuan membentuk fungsi-fungsi I/O yang berlaku untuk semua perangkat dan memberi antarmuka seragam ke perangkat lunak tingkat pemakai.

Fungsi-fungsi lain yang dilakukan :

- Sebagai interface seragam untuk seluruh device driver.
- Penamaan perangkat.
- Proteksi perangkat.
- Memberi ukuran blok perangkat agar bersifat device independent.
- Melakukan buffering.
- Alokasi penyimpanan pada block devices.
- Alokasi dan pelepasan dedicated devices.
- Pelaporan kesalahan.

d. Perangkat lunak level pemakai.

Kebanyakan perangkat lunak I/O terdapat di sistem operasi. Satu bagian kecil berisi pustaka-pustaka yang dikaitkan pada program pemakai dan berjalan diluar kernel. System calls I/O umumnya dibuat sebagai prosedur-prosedur pustaka. Kumpulan prosedur pustaka I/O merupakan bagian sistem I/O. Tidak semua perangkat lunak I/O level pemakai berupa prosedur-prosedur pustaka. Kategori penting adalah sistem spooling. Spooling adalah cara khusus berurusan dengan perangkat I/O yang harus didedikasikan pada sistem multiprogramming.

7.7. Buffering I/O

Buffering adalah melembutkan lonjakan-lonjakan kebutuhan pengaksesan I/O, sehingga meningkatkan efisiensi dan kinerja sistem operasi.

Terdapat beragam cara buffering, antar lain :

a. Single buffering.

Merupakan teknik paling sederhana. Ketika proses memberi perintah untuk perangkat I/O, sistem operasi menyediakan buffer memori utama sistem untuk operasi.

Untuk perangkat berorientasi blok.

Transfer masukan dibuat ke buffer sistem. Ketika transfer selesai, proses memindahkan blok ke ruang pemakai dan segera meminta blok lain.

Teknik ini disebut reading ahead atau anticipated input. Teknik ini dilakukan dengan harapan blok akan segera diperlukan. Untuk banyak tipe komputasi, asumsi ini berlaku. Hanya di akhir pemrosesan maka blok yang dibaca tidak diperlukan.

Keunggulan :

Pendekatan in umumnya meningkatkan kecepatan dibanding tanpa buffering.

Proses pemakai dapat memproses blok data sementara blok berikutnya sedang dibaca. Sistem operasi dapat menswap keluar proses karena operasi masukan berada di memori sistem bukan memori proses pemakai.

Kelemahan :

- Merumitkan sistem operasi karena harus mencatat pemberian buffer-buffer sistem ke proses pemakai.
- Logika swapping juga dipengaruhi. Jika operasi I/O melibatkan disk untuk swapping, maka membuat antrian penulisan ke disk yang sama yang digunakan untuk swap out proses. Untuk menswap proses dan melepas memori utama tidak dapat dimulai sampai operasi I/O selesai, dimana waktu swapping ke disk tidak bagus untuk dilaksanakan.

Buffering keluaran serupa buffering masukan. Ketika data transmisi, data lebih dulu dikopi dari ruang pemakai ke buffer sistem. Proses pengirim menjadi bebas untuk melanjutkan eksekusi berikutnya atau di swap ke disk jika perlu.

Untuk perangkat berorientasi aliran karakter.

Single buffering dapat diterapkan dengan dua mode, yaitu :

- o Mode line at a time.

  Cocok untuk terminal mode gulung (scroll terminal atau dumb terminal).

  Masukan pemakai adalah satu baris per waktu dengan enter menandai akhir baris. Keluaran terminal juga serupa, yaitu satu baris per waktu.

  Contoh mode ini adalah printer.

  Buffer digunakan untuk menyimpan satu baris tunggal. Proses pemakai ditunda selama masukan, menunggu kedatangan satu baris seluruhnya.

  Untuk keluaran, proses pemakai menempatkan satu baris keluaran pada buffer dan melanjutkan pemrosesan. Proses tidak perlu suspend kecuali bila baris kedua dikirim sebelum buffer dikosongkan.

- o Mode byte at a time.

  Operasi ini cocok untuk terminal mode form, dimana tiap ketikan adalah penting dan untuk peripheral lain seperti sensor dan pengendali.

b. Double buffering.

Peningkatan dapat dibuat dengan dua buffer sistem.Proses dapat ditransfer ke/dari satu buffer sementara sistem operasi mengosongkan (atau mengisi) buffer lain. Teknik ini disebut double buffering atau buffer swapping.

Double buffering menjamin proses tidak menunggu operasi I/O. Peningkatan ini harus dibayar dengan peningkatan kompleksitas. Untuk berorientasi aliran karakter, double buffering mempunyai 2 mode alternatif, yaitu :

- o Mode line at a time.

  Proses pemakai tidak perlu ditunda untuk I/O kecuali proses secepatnya mengosongkan buffer ganda.

- o Mode byte at a time.

  Buffer ganda tidak memberi keunggulan berarti atas buffer tunggal.

  Double buffering mengikuti model producer-consumer.

c. Circular buffering.

Seharusnya melembutkan aliran data antara perangkat I/O dan proses. Jika kinerja proses tertentu menjadi fokus kita, maka kita ingin agar operasi I/O mengikuti proses. Double buffering tidak mencukupi jika proses melakukan

operasi I/O yang berturutan dengan cepat. Masalah sering dapat dihindari denga menggunakan lebih dari dua buffer.

Ketika lebih dari dua buffer yang digunakan, kumpulan buffer itu sendiri diacu sebagai circulat buffer. Tiap buffer individu adalah satu unit di circular buffer.

Perangkat keras dan parameter kinerja disk

Disk diorganisasikan menjadi silinder-silinder dengan tiap permukaan terdapat head yang ditumpuk secara vertical. Track terbagi menjadi sektor-sektor.

Waktu yang dibutuhkan untuk membaca dan menulis disk dipengaruhi oleh :

o Waktu pencarian (seek time).

Merupakan faktor yang dominan. Waktu yang diperlukan untuk sampai ke posisi track yang dituju, yaitu : $S = S_c + di$, dimana :

S_c : adalah waktu penyalaan awal (initial startup time).

d : adalah waktu yang bergerak antar-antar track.

i : adalah jarak yang ditempuh (dalam ukuran ruang antar track).

Untuk track terdekat, $S_1 = S_c + d$ lebih kecil dibanding waktu yang diperlukan untuk satu putaran.Untuk memudahkan perhitungan maka dipakai s rata-rata,yaitu :

$$S = \sum_{i=1}^{j-1} S_i \, pdi,$$

S_i : adalah waktu tempuh untuk jarak ke-1.

Pdi : adalah probabilitas menempuh jarak ke-1.

Seek time rata-rata biasanya diinformasikan oleh pabrik pembuat.

o Waktu rotasi (rotational latency).

Waktu yang diperlukan mekanisme akses mencapai blok yang diinginkan.

Rumus untuk mendapatkan r adalah :

R=1/2*((60*1000)/rpm).

Rpm atau jumlah putaran permenit, biasa diinformasikan oleh pabrik pembuat.

o Waktu transfer (t).

Tergantung pada kecepatan rotasi dan kepadatan rekaman. Transfer rate (t) adalah kecepatan transfer data sesaat, data ini diberikan oleh pembuat. Maka dapat dihitung :

> Waktu transfer per rekord (TR, record transfer time).

TR (waktu untuk transfer rekord dengan panjang rekord, R), yaitu :

TR=R/t.

> Waktu transfer per blok (btt).

Bit (block transfer time,waktu yang diperlukan untuk transfer 1 blok),

yaitu : btt=B/t.

> Bulk transfer time (t').

Didalam kasus pembacaan/penulisan secara sekuens besar maka harus melewati gap dan daerah-daerah bukan data. Pada akhir tiap silinder, seek akan terjadi dan selama seek time, tidak ada data yang ditransfer.

Untuk keperluan didefinisikan bulk transfer time (t'), yaitu :

t'=(R)/(((R+W)/t)+s')

dimana :

R : adalah ukuran rekord.

W : adalah ruang yang disiakan.

s' : adalah seek time untuk sekuen.

t : adalah transfer mode.

Algoritma penjadwalan disk

Pada sistem multiprogramming, banyak proses yang melakukan permintaan membaca dan menulis rekord-rekord disk. Proses-proses membuat permintaan-permintaan lebih cepat dibanding yang dapat dilayani disk, membentuk antrian permintaan layanan disk. Diperlukan penjadwalan disk agar memperoleh kinerja yang optimal.

Terdapat dua tipe penjadwalan disk, yaitu :

1. Optimasi seek.
2. Optimasi rotasi (rotational latency).

Karena waktu seek lebih tinggi satu orde dibanding waktu rotasi, maka kebanyakan algoritma penjadwalan berkonsentrasi meminimumkan seek kumpulan atau antrian permintaan layanan disk. Meminimumkan latency biasanya berdampak kecil pada kinerja seluruh sistem.

Penjadwalan disk melibatkan pemeriksaan terhadap permintaan-permintaan yang belum dilayani untuk menentukan cara paling efisien melayani permintaan-permintaan. Penjadwal disk memeriksa hubungan posisi diantara permintaan-permintaan. Antrian permintaan disusun kembali sehingga permintaan-permintaan akan dilayani dengan pergerakan mekanis minimum.

Beberapa kriteria penjadwalan disk, yaitu :

- Throughput, yaitu berusaha memaksimumkan.
- Waktu tanggap rata-rata, nilai ini diusahakan minimum.
- Variansi waktu tanggap, diusahakan minimum.

Beberapa algoritma penjadwalan disk, antara lain :

- First come first serve (FCFS).

  Disk driver melayani satu permintaan sesuai urutan kedatangannya, merupakan metode yang adil. Saat rate permintaan sangat berat, FCFS dapat menghasilkan waktu tunggu sangat panjang. Dengan FCFS, sangat sedikit usaha optimasi waktu seek. FCFS dapat menyebabkan banyak waktu untuk seek silinder yang paling dalam ke silinder paling luar.

  Ketika permintaan-permintaan terdistribusi seragam pada permukaan-permukaan disk, penjadwalan FCFS menghasilkan pola seek yang acak. FCFS mengabaikan keterhubungan posisi diantara permintaan-permintaan yang menunggu di antrian. FCFS tidak membuat upaya optimasi pola seek. FCFS dapat diterima ketika beban disk masih ringan, tetapi begitu beban tumbuh cenderung menjenuhi perangkat dan menyebabkan waktu tanggap membesar.

- Shortest seek first (SSF).

  Algoritma ini melayani permintaan seek track terdekat dari track dimana head berada.

  Kekurangan : lengan disk akan berkutat ditengah disk. Permintaan di daerah ekstrim (pinggir) akan menunggu sampai fluktuasi statistik menyebabkan tidak

ada permintaan track-track tengah. Terdapat konflik antara meminimalkan waktu tanggao dengan fairness (adil).

- Elevator (SCAN).

  Yaitu head bergerak searah sampai tidak ada permintaan ke arah itu, kemudian berbalik arah. Diperlukan bit tambahan untuk mencatat arah gerak head. Kebaikan : batas atas jumlah gerak adalah tetap yaitu dua kali jumlah silinder.

- Elevator dimodifikasi (C-SCAN).

  Lengan head hanya bergerak searah, setiap kali mencapai silinder tertinggi, maka head akan bergerak ke silinder terendah dan dilanjutkan terus head bergerak searah. Ada kontroller yang dapat mengetahui pada track mana ia berada, dengan ini dapat dibuat optimasi untuk mencari sektor yang ada pada track tersebut.

- N-step scan.

  Lengan disk bergerak maju mundur seperti algoritma SCAN, tapi dengan semua permintaan yang tiba selama menyapu dalam satu arah dikumpulkan dulu dan disusun kembali agar layanan optimal selama penyapuan balik.

- Exchenbach scheme.

  Pergerakan lengan disk sirkular seperti C-SCAN, tapi dengan beberapa kekecualian penting setiap silinder dilayani tepat satu track informasi baik terdapat permintaan atau tidak untuk silinder itu. Permintaan-permintaan disusun untuk layanan dalam silinder itu untuk mendapatkan keunggulan posisi secara rotasi (agar dapat diterapkan optimasi rotasi), tapi jika terdapat dua permintaan dengan sektor-sektor yang overlap dalam satu silinder, hanya satu permintaan yang dilayani pada satu kesempatan.

Penanganan masalah operasi disk

Beberapa tipe kesalahan saat operasi disk dikategorikan sebagai berikut :

o Programming error.

Kesalahan disebabkan programming. Driver memerintahkan mencari track, membaca sektor, menggunakan head atau mentransfer ke atau dari memori

yang tak ada. Biasanya tiap controller memeriksa parameter sehingga tidak melakukan operasi yang tak valid. Kesalahan ini seharusnya tidak pernah ada.

o Transient checksum error.

Kesalahan disebabkan adanya debu diantara head dengan permukaan disk.

Untuk mengeliminasi kesalahan ini maka dilakukan pengulangan operasi pada disk.

o Permanent checksum error.

Kesalahan disebabkan kerusakan disk.

o Seek error.

Kesalahan ini ditanggulangi dengan mengkalibrasi disk supaya berfungsi kembali.

o Controller error.

Kesalahan ini ditanggulangi dengan menukar pengendali yang salah dengan pengendali yang baru.

o Track at time caching.

Kontroller mempunyai memori untuk menyimpan informasi track dimana ia berada, permintaan pembacaan blok track dilakukan tanpa pergerakan mekanik.

b. Clock

Perangkat keras clock.

Komputer dilengkapi dengan RTC (real time clock). Tipe perangkat clock, terdiri dari :

- Clock yang ditimbulkan impulse tegangan listrik.
  Clock in menginterupsi 50-60 interupt tiap detik sesuai dengan frekuensi listrik.
- Programmable interval timer (PIT).
  Clock ini terdiri dari crystal oscilator, counter, dan holding register.
  Dua keunggulan PIT, yaitu :
  - Mempunyai akurasi tinggi.
  - Frekuensi interupsi dapat diatur secara perangkat lunak.
    Dengan crystal oscilator 2 MHz, menggunakan 16 bit holding register, interupsi yang terjadi dapat diatur antara 1 ms sampai 65.536 ms.

PIT biasa digunakan sebagai :

- Waktu sistem.

- Pembangkit band rate.
- Penghitung kejadian.
- Pembangkit musik.
- Dan diberagam aplikasi yang memerlukan pewaktuan.

Ketika digunakan untuk pewaktuan PIT menghasilkan interupsi secara periodik. PIT bekerja dengan menghitung pulsa eksternal yang diberikan crystal oscilator. Keluaran PIT berupa pulsa yang diteruskan secara langsung ke IRQm(Interupt Request) sehingga menimbulkan interupsi ke pemroses. Periode waktu antara dua interupsi timer berturutan dapat diprogram dengan memasukkan nilai ke holding register.

Interval interupsi mempunyai rumus sebagai berikut :
Interval = (periode clock) x (nilai holding register).

Contoh :
Dikehendaki interval pewaktuan setiap 10 ms.
Frekuensi crystal oscilator adalah 2 MHz.

Berapa nilai yang harus dimasukkan ke holding register ?
Perhitungan :
Periode clock = 1/(2x106) = 0.5 x 10-6 = 0,5mms.
Nilai yang harus diberikan ke holding register = (10x10-3)/(0.5x10-6)=20x103.
Agar PIT menimbulkan interupsi dengan waktu interval 10 ms, maka holding register diset dengan nilai 20.000.

Metode pemrograman PIT.
Terdapat dua mode pemograman PIT, yaitu :
1. One shot mode.
   Setiap kali PIT diinisialisasi maka dikopikan nilai holding register ke counter.Counter diturunkan setiap terjadi pulsa crystal oscilator.
   Ketika counter bernilai 0, PIT membuat interupsi ke pemroses dan berhenti. PIT menunggu diinisialisasi secara eksplisit oleh perangkat lunak. Mode ini hanya untuk menghasilkan satu kejadian tunggal, diperlukan ketika clock diaktifkan berdasarkan kejadian.

2. Square wave mode.

Sesudah counter mencapai 0 maka menyebabkan interupsi ke pemroses.

Holding register dikopikan secara otomatis ke counter dan seluruh proses diulangi lagi sampai tak berhingga. Periode ini disebut clock ticks. Mode ini untuk menghasilkan kejadian-kejadian interupsi timer secara periodik, dilakukan secara otomatis tanpa melibatkan pemroses (perangkat lunak untuk inisialisasi kembali). Biasanya chip berisi dua atau tiga PIT independen dan mempunyai banyak option pemrograman (seperti menghitung keatas, pematian interupsi, dan sebagainya).

Perangkat lunak clock

Beberapa fungsi clock disistem operasi, antara lain :

1. Mengelola waktu dan tanggal (waktu nyata).

Tekniknya adalah counter dinaikkan setiap terjadi clock tick.

Teknik ini bermasalah karena keterbatasan jumlah bit counter.

Counter berukuran 32 bit akan overflow setelah 2 tahun bila clock ratenya bernilai 60Hz, solusinya adalah :

- Menggunakan counter 64 bit.
- Waktu dihitung dalam detik bukan dalam clock tick.
- Waktu dihitung relatif dengan saat komputer dihidupkan.

2. Mencegah proses berjalan lebih dari waktu yang ditetapkan.

Setiap kali proses dimulai, penjadwal inisialisasi counter dalam hitungan clock ticks. Setiap kali terjadi clock ticks, counter diturunkan. Saat counter mencapai 0 maka penjadwal mengalihkan pemroses ke proses lain.

3. Menghitung penggunaan pemroses (CPU).

Bila dikehendaki penghitungan dengan akurasi tinggi maka dilakukan dengan menggunakan timer kedua. Timer kedua terpisah dari timer sistem utama. Begitu proses dimulai, timer diaktifkan, saat proses berhenti maka timer dibaca. Timer menunjukkan lama waktu yang telah digunakan proses. Akurasi rendah dapat diperoleh dengan mengelola pointer ke tabel proses dan counter global.

4. Menangani system call alarm yang dibuat proses pemakai.
   Mensimulasi banyak clock dengan membuat senarai semua permintaan clock, terurut berdasar waktu. Isinya adalah jumlah clock ticks setelah signal proses sebelumnya.

5. Mengerjakan profiling, monitoring dan pengumpulan statistik.
   Untuk membuat data statistik kegiatan komputer.

c. RAM Disk.

Adalah perangkat disk yang disimulasikan pada memori akses acak (RAM).

RAM disk sepenuhnya mengeliminasi waktu tunda yang disebabkan pergerakan mekanis dalam seek dan rotasi. Kegunaannya untuk aplikasi yang memerlukan kinerja disk yang tinggi. Perangkat blok mempunyai dua perintah, yaitu membaca dan menulis blok. Normalnya blok-blok disimpan di disk berputar yang memerlukan mekanisme fisik.

Gagasannya adalah meniru perangkat dengan mengalokasikan terlebih satu bagian memori utama untuk menyimpan blok-blok data.

Keunggulan :

Berkecepatan tinggi karena pengaksesan sesaat (tidak ada waktu tunda seek dan rotational latency), sangat baik untuk menyimpan program atau data yang sering diakses. Memori utama dibagi menjadi n blok berukuran sama, bergantung banyak memori yang dialokasikan. Ketika driver untuk RAM disk menerima perintah membaca atau menulis suatu blok, driver tinggal menghitung dimana lokasi memori tempat blok berada kemudian membaca atau menuliskannya.

Bab 8

MANAJEMEN FILE

8.1. Sasaran dan fungsi sistem manajemen file

File mempunyai sifat sebagai berikut :

a. Persistence.

Informasi dapat bertahan meski proses yang membangkitkannya berakhir atau meskipun catu daya dihilangkan. Dengan properti ini maka file dapat digunakan untuk menjaga hasil-hasil yang diperoleh dari suatu proses dapat digunakaaan di masa datang.

b. Size.

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

c. Sharability.

File dapat digunakan banyak proses mengakses informasi secara kongkuren.

8.2. 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. Sistem pun menyediakan pengendalian terhadap aset penting ini.

Beberapa sasaran sistem file adalah sebagai berikut :

a. Memenuhi kebutuhan manajemen data bagi pemakai.

b. Menjamin data pada file adalah valid.

c. Optimasi kinerja.
d. Menyediakan dukungan masukan/keluaran beragam tipe perangkat penyimpan.
e. Meminimalkan atau mengeliminasi potensi kehilangan atau perusakan data.
f. Menyediakan sekumpulan rutin interface masukan/keluaran.
g. Menyediakan dukungan masukan/keluaran banyak pemakai di sistem multiuser.

Memenuhi kebutuan manajemen data bagi pemakai.
Kebutuhan manajemen data bagi pemakai, yaitu kemampuan melakukan operasi-operasi berikut :
a. Retrieve all, yaitu menampilkan seluruh rekord data.
b. Retrieve one, yaitu menampilkan seluruh satu rekord data tertentu.
c. Retrieve next, yaitu menampilkan satu rekord data berikutnya.
d. Retrieve previous, yaitu menampilkan satu rekord data sebelumnya.
e. Insert one, yaitu menyisipkan satu rekord data.
f. Delete one, yaitu menghapus satu rekord data tertentu.
g. Update one, yaitu memperbarui satu rekord data tertentu.
h. Update few, yaitu memperbarui beberapa rekord data tertentu yang satu kriteria.

Optimasi kerja.
o Menurut sistem, yaitu meningkatkan jumlah througput keseluruhan.
o Menurut pemakai, yaitu cepatnya waktu tanggap.

8.3. Fungsi manajemen file

Beberapaa fungsi yang diharapkan dari pengelolaan file adalah :
a. Penciptaan, modifikasi dan penghapusan file.
b. Mekanisme pemakaian file secara bersama.
Menyediakan beragam tipe pengaksesan terkendali, seperti :
- Read access (pengendalian terhadap akses membaca).
- Write access (pengendalian terhadap akses memodifikasi).
- Execute access (pengendalian terhadap akses menjalankan program).
- Atau beragam kombinasi lain.

c. Kemampuan backup dan recovery untuk mencegah kehilangan karena kecelakaan atau dari upaya penghancuran informasi.

d. Pemakai dapat mengacu file dengan nama simbolik bukan menggunakan penamaan yang mengacu perangkat keras.

e. Pada lingkungan sensitif dikehendaki informasi tersimpan amana dan rahasia.

Lingkungan ini, seperti :

- Electronic fund transfer system.
- Criminal record system.
- Medical record system.
- Dan sebagainya.

f. Sistem file harus menyediakan interface user-friendly.

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

- Pandangan secara logik bukan pandangan secara fisik terhadap data.
- Fungsi yang dapat dilakukan terhadap data.

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

8.4. Arsitektur pengelolaan file

Pengelolaan file, biasanya terdiri dari :

1. Sistem akses.

   Berkaitan dengan bagaimana cara data yang disimpan pada file diakses.

2. Manajemen file.

   Berkaitan dengan penyediaan mekanisme operasi pada file seperti :

   - Penyimpanan.
   - Pengacuan.
   - Pemakaian bersama.
   - Pengamanan.

3. Manajemen ruang penyimpan.

   Berkaitan dengan alokasi ruang untuk file di perangkat penyimpan.

4. Mekanisme integritas file.

   Berkaitan dengan jaminan informasi pada file tak terkorupsi.

Program dapat mengakses file di sistem melalui sistem manajemen basisdata (DBMS) ataupun secara langsung melalui fasilitas yang disediakan sistem operasi. Umumnya, sistem operasi menyediakan :

- Manajemen file.
- Manajemen penyimpanan file.
- Mekanisme integritas.

DBMS umumnya memuat bagian berikut :

- Database enginei, diantaranya mekanisme integritas.
- Sistem akses.

DBMS menggunakan fasilitas yang disediakan sistem operasi untuk memberikan layanan-layanannya. Mekanisme integritas merupakan masalah yang dilakukan baik di tingkat sistem operasi maupun di DBMS. Hanya sistem operasi tertentu, yaitu sistem operasi yang dikhususkan untuk basisdata yang secara langsung menyatakan sistem akses di sistem operasi agar diperoleh kinerja yang lebih bagus. Kebanyakan sistem operasi hanya menyediakan fasilitas pengelolaan umum yang akan digunakan perangkat lunak aplikasi diatasnya.

Pengelolaan file melibatkan banyak subsistem penting, yaitu :

o Manajemen perangkat I/O di sistem operasi.

Device driver.

Merupakan lapisan terbawah, berkomunikasi dengan perangkat secara langsung, bertanggungjawab memulai operasi I/O dan memproses penyelesaian permintaan I/O. Pada operasi file, perangkat yang biasa dipakai adalah disk atau tape.

Device driver merupakan bagian manajemen I/O.

> Sistem file di sistem operasi.

Sistem file dasar.

Merupakan interface utama dengan perangkat keras. Lapisan ini berurusan dengan blok-blok data yang dipertukarkan antara sistem dengan disk dan tape. Lapisan ini berfungsi dalam penempatan blok-blok data diperangkat penyimpanan sekunder dan buffering blok-blok data itu di memori utama. Lapisan ini tidak berkaitan dengan isi data atau struktur file. Sistem file dasar merupakan bagian sistem operasi.

Abstraksi file dan direktori.

Sistem file memberikan abstraksi ke pemakai berupa file/direktori.

Pemakai yaitu manusia ataupun proses tidak lagi berkaitan dengan blok-blok data melainkan beroperasi terhadap abstraksi file dan / atau direktori.

Operasi-operasi terhadap file dan direktori.

Kumpulan system call dan / atua pustaka untuk manipulasi file dan direktori.

Sistem akses dan/atau sistem manajemen basisdata.

Metode akses merupakan lapisan terakhir. Lapisan ini menyediakan interface standar antara aplikasi-aplikasi dan sistem file serta

perangkat yang menyimpan data. Metode-metode pengaksesan yang berbeda merefleksikan struktur file berbeda dan cara-cara pengaksesan dan pemrosesan yang berbeda.

Metode-metode pengaksesan yang paling terkenal, antara lain :

- File pile (pile file).
- File sekuen (sequential file).
- File sekuen (index-sequential file).
- File berindeks majemuk (multiple-indexed file).
- File ber-hash (hashed file).
- File multiring (multiring file).

8.5. Sistem file

Konsep terpenting dari pengelolaan file disistem operasi adalah :

a. File.

Abstraksi penyimpanan dan pengambilan informasi di disk. Abstraksi ini membuat pemakai tidak dibebani rincian cara dan letak penyimpanan informasi, serta mekanisme kerja perangkat penyimpan data.

Terdapat beragam pandangan mengenai file, yaitu :

- Pandangan pemakai.

Terhadap file pemakai berkepentingan memahami berikut :

- Penamaan untuk file.

Pemakai mengacu file dengan nama simbolik. Tiap file di sistem harus mempunyai nama unik agar tidak ambigu. Penamaan file dengan nama direktori tempat file memberi nama unik. Tidak diperbolehkan nama file yang sama di satu direktori.

Penamaan file berbeda sesuai sistem. Terdapat dua pendekatan, yaitu :

- Sistem yang case sensitive.
  Sistem membedakan antara huruf kecil dan huruf kapital.
- Sistem yang case insensitive.
  Sistem tidak membedakan antara huruf kecil dan huruf kapital.
  Saat ini, penamaan cenderung dapat menggunakan nama file panjang karena deskriptif.
- Tipe file.
  Terdapat tiga tipe file di sistem operasi, yaitu :
- Reguler.
  File berisi informasi, terdiri dari file ASCII dan biner.
  File ASCII berisi baris teks. File biner adalah file yang bukan file ASCII. Untuk file biner eksekusi (exe) mempunyai struktur internal yang hanya diketahui sistem operasi. Untuk file biner hasil program aplikasi, struktur internalnya hanya diketahui program aplikasi yang menggunakan file tersebut.
- Direktori.
  File direktori merupakan file yang dimiliki sistem untuk mengelola struktur sistem file. File direktori merupakan file berisi informasi-informasi mengenai file-file yang termasuk dalam direktori itu.
- Spesial.
  File spesial merupakan nama logik perangkat I/O. Perangkat I/O dapat dipandang sebagai file. Pemakai dihindarkan dari kerumitan operasi perangkat I/O.

File in terbagi dua, yaitu :

File spesial karakter.

Berhubungan dengan perangkat I/O aliran karakter. File ini memodelkan perangkat I/O seperti :

Ø Terminal.

Ø Printer.

Ø Port jaringan.

Ø Modem.

Ø Dan alat-alat yang bukan penyimpan sekunder.

File spesial blok.

Berhubungan dengan perangkat I/O sebagai kumpulan blok-blok data (berorientasi blok).

- Atribut file.

Informasi tambahan mengenai file untuk memperjelas dan membatasi operasi-operasi yang dapat diterapkan dan dipergunakan untuk pengelolaan file.

Tabel berikut menunjukkan atribut-atribut di file.

Tabel 8.1 : Atribut-Atribut File

Field	Deskripsi
Protection	Siapa yang dapat mengakses file dan dengan cara apa.
Password	Password yang diperlukan untuk mengakses file.
Creator	ID orang yang menciptakan file.
Owner	Pemilik saat itu.
Read only flag	0 untuk read/write, 1 untuk read-only.
Hidden flag	0 untuk normal, 1 untuk tidak ditampilkan pada listing.
System flag	0 untuk normal. 1 untuk file sistem.
Archive flag	0 telah dibackup, 1 untuk perlu dibackup.
ASCII/binary flag	0 untuk file ASCII, 1 untuk file biner.
Random access flag	0 untuk sequential access only, 1 untuk random access.
Temporary flag	0 untuk normal, 1 untuk dihapus saat keluar (exit).
Lock flag	0 untuk tak terkunci, 1 untuk terkunci.
Record length	Jumlah byte pada satu record.
Key position	Offset kunci pada masing-masing record.
Key length	Jumlah byte dari field kunci.
Creation time	Tanggal dan waktu file diciptakan.
Time of last access	Tanggal dan waktu file diakses terakhir kali.
Time of last change	Tanggal dan waktu file diubah terakhir kali.
Current size	Jumlah byte dalam file.
Maksimum size	Ukuran maksimum file boleh tumbuh.

- Perintah-perintah untuk manipulasi file.

  Merupakan perintah yang dapat diberikan pemakai dibaris perintah ke shell (command interpreter). Perintah-perintah tersebut dapat dikategorikan menjadi :

  - Perintah penciptaan file.
  - Perintah penghapusan file.
  - Perintah pengkopian file.
  - Perintah penggantian nama.
  - Perintah manipulasi yang lain.

- Pandangan pemogram.

  Selain perlu memahami sebagai pemakai, pemrogram perlu memahami :

  - Operasi-operasi terhadap file.

    Beragam operasi dapat diterapkan pada file, seperti operasi-operasi berikut :

Tabel 8.2 : Operasi-Operasi pada File

Operasi	Deskripsi
Create	Menciptakan berkas.
Delete	Menghapus berkas.
Open	Membuka berkas untuk penyiapan proses selanjutnya.
Close	Menutup berkas untuk menyimpan semua informasi ke berkas dan mendealokasikan sumber daya yang digunakan.
Read	Membaca data pada berkas.
Write Append	Memodifikasi data pada berkas, yaitu pada posisi yang ditunjuk. Menambah data pada berkas, merupakan operasi write yang lebih spesifik, yaitu di akhir berkas.
Seek	Mencari lokasi tertentu, hanya berlaku untuk berkas akses acak.
Get attributes	Membaca atribut-atribut berkas.
Set attributes	Menuliskan (memodifikasi) atribut-atribut berkas.
Rename	Mengganti nama berkas

  - Pandangan perancang sistem.

    Implementasi pengelolaan file.

b. Direktori.

Berisi informasi mengenai file. Kebanyakan informasi berkaitan dengan penyimpan. Direktori adalah file, dimiliki sistem operasi dan dapat diakses dengan rutin-rutin di sistem operasi. Meski beberapa informasi direktori tersedia

bagi pemakai atau aplikasi, informasi itu umumnya disediakan secara tidak langsung. Pemakai tidak dapat mengakses direktori secara langsung meski dalam mode read-only.

Pandangan pemakai.

Direktori menyediakan pemetaan nama file ke file. Informasi terpenting pada direktori adalah berkaitan dengan penyimpanan, termasuk lokasi dan ukuran penyimpanan file. Pada sistem bersama (shared system), informasi yang penting adalah informasi pengendalian akses file. Satu pemakai adalah pemilik file yang dapat memberi wewenang pengaksesan ke pemakai-pemakai lain.

Aturan penamaan direktori mengikuti aturan penamaan file karena direktori merupakan file yang khusus.

Beberapa konsep penting :

- Hirarki direktori.

  Kebanyakan sistem menggunakan hirarki direktori atau berstruktur pohon.

  Terdapat satu direktori master (root) yang didalamnya dapat terdapat subdirektori-subdirektori. Subdirektori dapat membuat subdirektori-subdirektori berikutnya, demikian seterusnya. Penamaan direktori sama aturannya dengan penamaan file karena direktori adalah file yang mempunyai arti khusus. Direktori diimplentasi dengan file.

- Jalur pengaksesan (path name).

  Bila sistem file diorganisasikan dengan pohon direktori, maka diperlukan cara menspesifikasikan nama file. Masalah penamaan file diselesaikan dengan penamaan absolut dan penamaan file relatif.

  Terdapat dua jalur, yaitu :

  a. Nama jalur absolut (absolute pathname).

  Nama jalur dari direktori root ke file, selalu dimulai dari direktori root da nakan bernilai unik.

  b. Nama jalur relatif (relative pathname).

  Jalur relatif terhadap direktori kerja/saat itu (working atau current director). Pemakai dapat menyatakan satu direktori sebagai current directory. Nama jalur yang tidak dimulai direktori root berarti relatif terhadap current directory.

- Perintah-perintah memanipulasi direktori.

  Meliputi perintah :

  - Pindah direktori.

- Penciptaan direktori.
- Penghapusan direktori, yang mensyaratkan :.
  - o Direktori tidak sedang digunakan.
  - o Direktori telah kosong.

Operasi pada direktori.

Beragam operasi dapat diterapkan pada direktori seperti pada file.

Tabel berikut menunjukkan operasi-operasi yang khusus beroperasi pada direktori, sebagai berikut :

Tabel 8.3 : Operasi-Operasi pada Direktori

Operasi	Deskripsi
Create	Menciptakan direktori.
Delete	Menghapus direktori.
Open directory	Membuka direktori untuk dibaca.
Close directory	Menutup direktori untuk mendealokasi sumber daya.
Read directory	Membaca isi direktori.
Rename	Mengganti nama direktori.
Link	Membuat link (tautan) terhadap suatu berkas, sehingga berkas dapat muncul sebagao anggota lebih dari satu direktori.
Unlink	Memutuskan link (tautan) terhadap suatu berkas.

c. Memanipulasi seluruh sistem file.

Terdapat perintah-perintah memanipulasi sistem file, antara lain :

o Pembentukan sistem file.

o Pemeriksaan sistem file.

o Pengkopian seluruh sistem file.

o Manipulasi lain.

8.6. Shared file

Adalah file yang tidak hanya diacu oleh satu direktori (pemakai), tapi juga oleh direktori-direktori (pemakai) lain. Sistem file tidak lagi berupa pohon melainkan directed acyclic graph (DAG).

Masalah-masalah yang terdapat pada shared file adalah sebagai berikut :

1. Metode implementasi shared file.
2. Metode pemberian hak akses pada shared file.
3. Metode pengendalian atau penanganan terhadap pengaksesan yang secara simultan dilakukan pemakai-pemakai yang mengacu file. Persoalan pengaksesan simultan in menyangkut integritas atau koherensi data.

8.7. Sistem akses file

Sistem akses merupakan pilihan, yaitu :

1. Dapat menjadi bagian dari sistem operasi, atau
2. Sistem operasi sama sekali tidak mempunyai komponen sistem akses.

Sistem operasi bertujuan umum (general purposes operating system) tidak mengimplementasikan sistem akses sebagai komponen sistem operasi, terserah sistem manajemen basis data yang dijalankan di sistem operasi untuk menangani sistem akses. Sistem operasi hanya memberikan pengelolaan sistem file dasar.

Sistem operasi tertentu (khusus) sering mengimplementasikan sistem akses sebagai bagiannya seperti sistem operasi mainframe untuk tujuan khusus. Implementasi sistem akses ditingkat sistem operasi untuk meningkatkan kinerja sistem manajemen basisdata.

Cara akses perangkat penyimpanan.

Perangkat penyimpanan berdasar disiplin pengaksesan dibagi dua, yaitu :

1. Perangkat akses sekuen (sequential access devices).
   Proses harus membaca semua byte atau rekord file secara berturutan mulai dari awal, tidak dapat meloncati dan membaca diluar urutan.
   Contoh : tape.
2. Perangkat akses acak (random access devices).
   Dimungkinkan dapat membaca byte atau rekord file di luar urutan, atau mengakses rekord berdasar kunci bukan posisinya.

Organisasi file.

Elemen pokok perancangan sistem akses adalah cara rekord-rekord diorganisasikan atau distrukturkan. Beberapa kriteria umum untuk pemilihan organisasi file adalah :

1. Redundansi yang kecil.
2. Pengaksesan yang cepat.
3. Kemudahan dalam memperbaharui.
4. Pemeliharaan yang sederhana.
5. Kehandalan yang tinggi.

Terdapat enam organisasi dasar, kebanyakan organisasi file sistem nyata termasuk salah satu atau kombinasi kategori-kategori ini. Enam organisasi atau pengaksesan dasar adalah sebagai berikut :

1. File pile (pile).
2. File sekuen (sequential file).
3. File sekuen berindeks (indexed-sequential file).
4. File berindeks majemuk (multiple indexed file).
5. File berhash (hashed or direct file).
6. File cincin (multiring file).

8.9. Implementasi sistem file

File berisi sekumpulan blok. Sistem manajemen file bertanggungjawab untuk alokasi blok-blok disk ke file. Dua hal penting yang harus ditangani adalah :

o Pencatatan ruang yang dialokasikan untuk file.
o Pencatatan ruang bebas yang tersedia di disk.

Sistem file meliputi :

a. Alokasi file.

Masalah pokok adalah pencatatan blok-blok yang digunakan file.

Beragam metode dapat digunakan, diantaranya :

> Alokasi berturutan/kontigu (contigous allocation).

Teknik ini merupakan skema alokasi paling sederhana, yaitu menyimpan file sebagai blok-blok data berturutan (kontigu) di disk.

Keunggulan :

>> Sederhana.

Metode ini sederhana dalam implementasi karena pencatatan dimana blok-blok file berada direduksi menjadi hanya mengingat alamat awal file dan panjang file, yaitu jumlah blok dari file.

>> Kinerjanya luar biasa bagus.

Karena seluruh file dapat dibaca dari disk dengan satu operasi.

Tak ada metode alokasi lain yang dapat menandingi kinerja pengaksesan.

Keunggulan ini diperoleh karena rekord-rekord yang secara logik berturutan biasanya juga saling berdekatan secara fisik.

Kelemahan :

>> Hanya bila ukuran maksimum diketahui pada saat file diciptakan.

Layak digunakan kecuali bila ukuran maksimum diketahui pada saat file diciptakan. Tanpa informasi itu, sistem operasi tidak mengetahui berapa banyak ruang disk yang digunakan untuk suatu file.

>> Terjadi fragmentasi disk.

Fragmentasi disk dapat dihasilkan metode alokasi ini, ruang yang disiakan seharusnya dapat digunakan.

Pada lingkungan dimana file berkembang dan mengkerut setiap saat, alokasi kontigu sulit diterapkan :

- o Pemakai biasanya berlebihan dalam memperkirakan ruang yang diperlukan, menyebabkan banyak pemborosan.
- o Ketika file berkembang lebih besar dari slot yang dialokasikan, file harus ditransfer ke lokasi baru dapat memuat. Pemindahan memerlukan usaha besar yang mengkonsumsi banyak waktu komputasi.

Meskipun teknik ini banyak kelemahan, tetapi sangat cocok untuk sistem yang memerlukan pengaksesan data didisk yang sangat cepat. Contohnya, sistem jaringan dan sistem waktu nyata (real time).

> Alokasi blok-blok file sebagai senarai berkait.

Metode kedua adalah mencatat blok-blok file dengan senarai berkait blok-blok didisk. Word pertama di blok data sebagai pointer ke blok berikutnya, sisanya untuk menyimpan data. Skema ini disebut rantai blok (block chaining) karena

blok pertama merantai blok kedua, blok kedua merantai blok ketiga, dan seterusnya. Blok sebelumnya merantai blok berikutnya. Direktori mencatat blok pertama file.

Keunggulan :

>> Setiap blok didisk dapat digunakan.

>> Tak ada ruang yang hilang karena fragmentasi eksternal.

>> Isian/elemen direktori cukup menyimpan alamat blok pertama file.

Kelemahan :

>> Pembacaan sekuen cukup merepotkan karena harus menelusuri blok satu per satu.

>> Blok data tidak lagi berukuran 2k, karena pointer memerlukan beberapa byte.

Masalah ini tidak fatal. Ukuran yang janggal (bukan berukuran 2k) kurang efisien karena program membaca dan menulis blok tidak dapat memanfaatkan sifat bilangan biner.

> Alokasi blok-blok sebagai senarai berkait menggunakan index (FAT).

Kelemahan alokasi senarai berkait dieliminasi dengna menghilangkan pointer di blok dan meletakkan sebagai tabel tersendiri di memori.

Seluruh blok tersedia untuk data. Skema ini disebut block oriented file mapping.

Tabel yang mencatat nomor blok data disebut FAT (File Allocation Table).

Keunggulan :

>> Pengaksesan acak lebih mudah.

Meski masih harus menelusuri rantai berkait untuk menemukan lokasi blok file, rantai blok seluruhnya di memori sehingga dapat dilakukan secara cepat tanpa membuat pengaksesan ke disk.

>> Direktori cukup menyimpan bilangan bulat nomor blok awal.

Blok awal ini digunakan untuk menemukan seluruh blok, tak peduli jumlah blok file itu. Direktori menunjuk blok pertama file dan FAT menunjukkan blok-blok file berikutnya.

Kelemahan :

>> Seluruh tabel (FAT) harus disimpan di memori.

Jika penyimpanan berukuran besar mengakibatkan tabel berukuran besar dan harus ditaruh di memori utama meskipun hanya satu file yang dibuka.

Penggunaan : MS-DOS menggunakan metode ini.

b. Pencatatan ruang disk yang bebas.

Dapat dilakukan dengan :

o Berurutan.

Cara ini mempunyai masalah bila file berkembang dan ruang berikutnya telah ditempati file lain.

o Fixed block.

Perkembangan file dapat diatasi tapi menentukan ukuran blok merupakan hal sulit.

Blok-blok bebas yang belum digunakan pada disk harus dicatat sehingga dapat dilakukan alokasi blok-blok ke file yang memerlukan. Teknik pencatatan blok-blok bebas dapat dilakukan dengan :

- Peta bit.
- Senarai berkait.

Peta bit memerlukan ruang pencatatan lebih kecil karena tiap blok hanya dipresentasikan 1 bit, sementara senarai berkait memerlukan 16 bit perblok. Senarai berkait lebih kecil dibanding peta bit hanya jika disk telah hampir penuh.

c. Shared file.

Adalah file yang tidak hanya diacu satu direktori, juga oleh direktori-direktori lain. Sistem manajemen file tidak lagi berupa pohon melainkan graph berarah tak melingkar (DAG=directed acyclic graph).

Shared file dapat diimplementasikan dengan tiga teknik, yaitu :

o. Membuat pengkopian.

File yang dipakai bersama dikopi ke masing-masing direktori pemakai.

<u>Keunggulan :</u>

> Sederhana.

> Terdapat redundansi, sehingga kerusakan satu kopian masih tersedia kopian-kopian lain.

<u>Kelemahan :</u>

> Perubahan yang dibuat satu pemakai tidak akan terlihat pemakai-pemakai lain. Kelemahan ini merupakan masalah fatal, memerlukan satu mekanisme penjagaan integritas yang rumit.

> Penggunaan ruang disk sangat besar karena duplikasi atau pengkopian file-file yang sama.

Penggunaan :

>> Dapat diterapkan pada sistem jaringan secara hati-hati agar memperkecil lalu lintas komunikasi. Penerapan terutama pada file-file acuan yang jarang diperbarui.

o. I-node.

Blok-blok disk file dipakai bersama tidak didaftarkan di direktori melainkain di struktur kecil diasosiasikan dengan file. Direktori pemakai-pemakai lain cukup menunjuk struktur itu. Struktur data kecil tersebut adalah i-node.

Keuntungan :

> Tidak terdapat banyak kopian.

> Modifikasi oleh satu pemakai akan terlihat pemakai lain.

> Tidak memerlukan mekanisme penjagaan integrasi yang rumit.

Kelemahan :

> Jika sistem menghapus file dan juga memberikan i-node maka direktori pemakai lain akan menunjuk i-node tidak absah.

> Jika i-node kemudian diberikan ke file lain maka akan menunjuk ke file yang baru, file yang tidak dimaksud.

Penggunaan :

> Pendekatan ini digunakan UNIX, disebut hard-link.

o. Symbolic link.

Sistem membuat file bertipe link (kaitan) berisi jalur yang dikaitkan di direktori. Ketika file bertipe link dibaca maka sistem operasi mengetahui bahwa file yang harus dibaca adalah nama file yang ditunjuk file tipe link. Sistem mencari direktori yang memuat i-node file itu.

Kelemahannya

Tidak terdapat pada teknik symbolic link karena hanya pemilik file yang mempunyai penunjuk ke i-node file.

Keuntungan :

> Symbolic linking dapat digunakan men-links file di mesin manapun.

> Bila symbolic linking dihilangkan, tidak menimbulkan efek apapun pada file.

Kelemahan :

> File yang berisi jalur haris dibaca. Jalur diparse dan diikuti komponen demi komponen sampai dicapai i-node. Semua aktivitas ini memerlukan sejumlah pengaksesan disk.

> Pemborosan lain adalah diperlukan satu i-node ekstra untuk setiap symbolic linking.

Penggunaan :

> Pendekatan ini digunakan UNIX, disebut symbolic link.

Pengaksesan pada shared file

Sistem manajemen file harus menyediakan alat bantu agar mengijinkan pemakaian file bersama pemakai-pemakai, menyediakan sejumlah pilihan teknik pengendalian pengaksesan file bersama. Biasanya, pemakai atau sekelompok pemakai diberi wewenang hak pengaksesan tertentu terhadap file itu.

Hak-hak akses sangat beragam. Berikut adalah daftar hak-hak pengaksesan ke pemakai terhadap file :

> None.

Pemakai tidak mengetahui keberadaan file. Batasan ini dilakukan dengan pemakai tidak diijinkan membaca direktori.

> Knowledge.

Pemakai dapat mengetahui keberadaan file dan pemiliknya. Bila perlu, pemakai dapat meminta peningkatan hak akses file dengan mengirim pesan ke pemilik file. Pemilik file dapat mengendalikan yaitu berkuasa penuh untuk mengubah hak akses ataupun tidak.

> Execution.

Pemakai dapat memuatkan file dan mengeksekusi program tapi tidak dapat mengkopinya. Program-program khusus sering hanya dapat diakses dengan batasan ini.

> Reading.

Pemakai dapat membaca file untuk tujuan tertentu, termasuk pengkopian dan eksekusi. Beberapa sistem dapat memaksakan pembedaan antara sekedar melihat dan mengkopi. Pada aktivitas pertama,isi file dapat ditampilkan ke pemakai tapi pemakai tidak mempunyai cara untuk mengkopinya.

> Appending.

Pemakai dapat menambah data ke file, sering hanya di akhir file.

Pemakai tidak dapat memodifikasi atau menghapus suatu isi file.

Hak ini berguna dalam mengumpulkan data dari sejumlah sumber dengan sumber-sumber tidak dapat memodifikasi file selain menambahkan data.

> Updating.

Pemakai dapat memodifikasi, menghapus dan menambah data pada file.

Update biasanya termasuk menulis file, menulis ulang secara penuh atau sebagian, dan memindahkan semua atau sebagian data. Beberapa sistem membedakan menjadi derajat-derajat update secara berbeda.

> Changing protection.

Pemakai dapat mengubah hak-hak akses yang diberikan ke pemakai-pemakai lain. Biasanya hak ini hanya dipegang pemilik file. Pada beberapa sistem, pemilik file dapat melimpahkan hak ini pemakai lain. Untuk mencegah penyalahgunaan, pemilik file dapat menspesifikasikan hak-hak mana yang dapat diubah oleh penerima wewenang.

> Deletion.

Pemakai dapat menghapus file dari sistem file.

d. Kehandalan sistem file.

Kerusakan data lebih mahal dibanding kerusakan perangkat keras karena merupakan kehilangan yang tak dapat diganti bila tidak memiliki salinannya.

Manajemen blok buruk

Disk biasanya mempunyai blok-blok buruk, yaitu mempunyai cacat sehingga tak sempurna dalam menyimpan data. Kebanyakan produsen harddisk memberi daftar blok buruk yang ditemukan selama pengujian.

Terdapat dua solusi terhadap blok-blok buruk, yaitu :

> Secara perangkat keras.

Solusi dengan menyediakan track pengganti. Secara perangkat keras didedikasikan sektor untuk mencatat blok-blok buruk. Daftar blok buruk menyatakan blok pengganti pada track yang disediakan untuk pengganti.

Semua permintaan ke blok buruk akan diarahkan menggunakan blok pengganti.

> Secara perangkat lunak.

Sistem manajemen file membuat catatan semua blok buruk, menyingkirkan dari daftar blok bebas. Blok-blok ini tak pernah dipakai untuk menyimpan data. Selama pencatatan blok-blok buruk tidak terusik maka tak akan muncul masalah.

Pemulihan dari kegagalan disk

Penanggulangan kerusakan disk saat operasi dapat dilakukan dengan :

a. Backup.

Teknik yang paling biasa dilakukan adalah backup data secara periodik.

Backup adalah membuat kopian file secara teratur dan meletakkan kopian ditempat aman. Cara backup bergantung kapasitas penyimpan yang dibackup, yaitu :

> Untuk floppy disk.

Kopikan isi floppy disk secara keseluruhan. Cara iini dilakukan karena kapasitas masih kecil sehingga pengkopian dapat dilakukan secara cepat, aman, dan menyeluruh.

> Untuk small harddisk.

Dump isi harddisk.

> Untuk big harddisk.

Backup bersilangan, yaitu :

>> Partisi pertama disk A adalah data sedang partisi kedua adalah backup untuk data disk B.

>> Partisi pertama disk B adalah data sedang partisi kedua adalah backup untuk data disk A.

Agar tidak terjadi pengkopian berulang file-file yang tidak dimodifikasi pada disk berukuran besar maka dilakukan incremented dump. Incremented dump adalah hanya dump file-file yang berubah sejak terakhir kali di dump. Terdapat daftar file yang harus dibackup di disk.

Kelemahan :

> Sistem perlu shutdown selama operasi backup.

> Backup lengkap memakan waktu lama.

> Ketika terjadi kegagalan sistem, pemulihan dari backup terakhir biasanya memakai banyak waktu.

Ketika pemulihan selesai, data di sistem adalah data backup terakhir.

Seluruh taransaksi yang dilakukan sejak backup terakhir sampai terjadi kegagalan sistem hilang musnah. Semua teknik berbasis backup mempunyai keterbatasan tidak dapat memulihkan data/informasi diantara backup terakhir dan kejadian kegagalan sistem. Pendekatan untuk memperkecil data/informasi yang hilang adalah dengan transaction log.

b. Transaction log.

Setiap transaksi segera dicatat (log), menghasilkan transaction log.

Setiap transaksi dibackup. Teknik ini lebih mudah dilakukan di sistem interaktif karena aktifitas manusia relatif lebih lambat dibanding pencatatan transaksi sehingga tidak mempengaruhi waktu tanggap.

Bila terjadi kegagalan maka pemulihan memanfaatkan backup terakhir dan transaction log.Pemulihan dilakukan dengan cara sistem menjalankan setiap transaksi di transaction log terhadap backup terakhir, seterusnya sampai transaksi terakhir yang tercatat. Kehilangan data dikurangi, yaitu hanya satu transaksi terakhir yang tidak tercatat yang hilang musnah.

Konsistensi sistem manajemen file.

Masalah penting lain adalah konsistensi atau integritas. Terdapat beberapa teknik untuk mengatasi masalah konsistensi sistem manajemen file, antara lain :

> Atomic update.

Update rekord, blok dan file terjadi lengkap atau tidak sama sekali (meninggalkan sistem pada keadaan semula).

> Stable storage.

Menulis di drive 1, dilakukan verifikasi :

>> Jika baik maka ditulis ke drive 2 dan diversifikasi. Verifikasi untuk menangani bad sector.

>> Jika baik maka diulangi penulisannya.

Cara ini tidak pernah menjadikan sistem dalam keadaan ambigu.

> Multiversion files.

Dibuat file baru pada saat pengaksesan.

Kendali kongkurensi

Teknik untuk menangani keadaan simultan secara serial disebut serializability. Teknik untuk mendapatkan properti ini disebut kendali
kongkurensi, antara lain :

> Penguncian (locking).

Adalah teknik kendali kongkurensi yang biada dipakai. Ketika file dikunci, semua usaha menggunakan atau mengunci fle oleh klien-klien lain akan ditolak. Masalahnya yaitu jika klien mengunci file, kemudian crash.

Solusi menerapkan timer begitu memulai lock. Bila proses telah melewati suatu batas waktu maka klien dianggap telah crash dan dilepaskan penguncian yang dilakukannya. Cara ini menimbulkan masalah baru, yaitu jika ternyata sebenarnya klien masih berjalan baik, hanya lamban maka karena pengunciannya dilepaskan menyebabkan inkonsistensi dapat terjadi bila klien lain memakai file itu.

> Transaksi (transaction).

Pemakai diberi wewenang mendefinisikan transaksi yaitu seluruh aksi di transaksi harus berjalan sukses seluruhnya atau bila terdapat aksi yang gagal maka dianggap tidak terdapat aksi sama sekali. Transaksi yang gagal akan meninggalkan sistem file tanpa perubahan, tidak dalam keadaan ditengah-tengah yang tidak tentu.

> Replikasi file (file replication).

Sistem tidak hanya menyimpan satu kopian tapi menyimpan N kopian.

Jika salah satu kopian rusak, data tidak hilang. Ketika satu kopian dimodifikasi, beragam cara untuk menangani replikasi agar sistem file tetap konsisten.

Terdapat dua strategi, yaitu :

>> Menempatkan duplikasi-duplikasi pada banyak direktori dan mengirim blok-blok yang telah diubah ke tiap duplikasi. Duplikasi-duplikasi akan up-to-minute.

>> Tinggalkan duplikasi-duplikasi yang telah kedaluwarsa, buat kopian-kopian baru file yang dimodifikasi dan masukan kopian-kopian itu ke direktori.

e. Kinerja sistem file.

Sasaran utama peningkatan kinerja sistem manajemen file adalah mereduksi jumlah akses ke disk. Cara-cara yang dapat digunakan, adalah :

> Buffer cache.

Mengakses data dari/ke disk dibanding mengakses dari/ke memori utama (RAM) lebih lamban 100.000 kali. Karena itu harus diusahakan mereduksi jumlah pengaksesan ke disk. Teknik untuk mereduksi adlah block chace atau buffer cache atau chace. Chace adalah kumpulan blok yang secara logik dipunyai disk tetapi tersimpan di memori utama. Cara kerja chace adalah sebagai berikut :

>> Selalu memeriksa semua permintaan baca untuk menentukan apakah blok yang diperlukan telah berada di chace.

>> Jika blok telah berada di chace, maka permintaan baca dapat dipenuhi dari cache tanpa pengaksesan disk.

>> Jika blok data tidak berada di chace, maka dilakukan pembacaan dari disk sebanyak satu blok dan kopikan lebih dulu ke chace. Setelah itu kopikan ke proses yang meminta. Permintaan berikutnya untuk blok yang sama dapat dipenuhi dari chace tanpa perlu pengaksesan disk.

>> Jika chace telah penuh, suatu blok di chace dipindahkan dan jika blok tersebut telah dimodifikasi maka harus dituliskan ke disk.

> Penempatan data.

Penempatan data diusahakan sehingga memperkecil jumlah seek times dan rotasi. Interleave digunakan untuk memperkecil rotasi. Pada sistem dengan i-node terdapat bottleneck disebabkan dua pengaksesan, yaitu :

>> Pengaksesan i-node.

>> Pengaksesan blok-blok data.

8.10. Sistem akses file

Rekord dan blocking

Pada sistem akses, maka rekord adalah unit terkecil penyimpanan data di level logik atau file. Panjang rekord dapat tetap atau bervariasi.

Tiga metode untuk penandaan awal dan akhir rekord berukuran variasi, yaitu :

> End of record mark.

> Indikator panjang.

> Tabel posisi rekord.

Rekord-rekord harus ditempatkan di blok. Satu blok dapat terdiri satu rekord atau lebih. Penempatan rekord-rekord ke blok disebut blocking.

Blocking factor (Bfr) adalah parameter yang menunjukkan jumlah rekord yang diharapkan (maksimum) ditampung di 1 blok.

Penempatan rekord-rekord pada block

Kombinasi penempatan rekord-rekord pada blok dapat berupa :

> Fixed blocking.

Rekord berukuran tetap. Blok berisi jumlah rekord yang tetap. Rekord hanya menempati di satu blok, tidak boleh di pecah di beberapa blok.

Rekord tidak boleh melebihi ukuran blok.

Keunggulan :

Memudahkan implementasi.

Kelemahan :

Memboroskan ruang penyimpan karena fragmentasi internal.

> Variable length spanned blocking.

Rekord dapat berukuran bervariasi ditempatkan memenuhi blok dan dapat dipecah untuk menempati blok-blok berbeda. Satu rekord dapat ditempatkan di lebih dari satu blok. Keterhubungan rekord yang terpecah pada blok-blok berbeda diwujudkan dengan pointer blok menunjuk alamat blok-blok berikutnya tempat bagian rekord itu.

Keuntungan :

>> Fleksibel bagi pemakai.

>> Ukuran rekord tidak dibatasi ukuran blok.

>> Mengurangi kesiaan ruang penyimpan karena fragmentasi internal sungguh berkurang.

Kelemahan :

>> Sulit diimplementasikan.

>> Mahal dalam pencariannya.

>> Sulit dalam perbaruan (update).

> Variable length unspanned blocking.

Rekord-rekord walaupun bervariabel panjangnya harus secara utuh ditempatkan pada satu blok, tidak boleh dipecah ke blok-blok lain.

Kelemahan :

>> Terjadi pemborosan tempat karena rekord yang akan ditempatkan terlalu panjang untuk sisa blok akan ditempatkan di blok berikutnya.

>> Panjang rekord tidak boleh lebih panjang daripada ukuran blok.

Operasi-operasi di sistem akses file

Sistem akses harus mampu menyediakan operasi-operasi berikut terhadap organisasi akses yang dipilih, yaitu :

- Pencarian suatu rekord tertentu.
- Bergerak ke rekord berikutnya.
- Memperbarui rekord berupa penghapusan rekord atau modifikasi suatu rekord.
- Pembacaan kumpulan rekord dengan kriteria tertentu.
- Pembacaan seluruh rekord di file.
- Reorganisasi.

Tiap organisasi akses mempunyai keunggulan dan kelemahan tersendiri sehingga tidak mungkin menerapkan satu organisasi akses untuk seluruh kebutuhan aplikasi sistem komputer.

Bab 9

KEAMANAN SISTEM

Saat ini sistem komputer yang terpasang makin mudah diakses, sistem time-sharing dana akses jarak jauh menyebabkan kelemahan komuniksai data menjadi pokok masalah keamanan. Terlebih dengan meningkatnya perkembangan jaringan komputer. Kecenderungan lain saat ini adalah memberi tanggungjawab pengelolaan aktivitas pribadi dan bisnis ke komputer, seperti :

- Sistem transfer dana elektronis (electronic fund transfer system) melewatkan uang sebagai aliran bit.
- Sistem kendali lalu-lintas udara (air trafic control system) melakukan banyak kerja yang sebelumnya ditangani pengendali manusia.
- Unit rawat intensif di rumah sakit sudah sangat terkomputerisasi.
- Dan sebagainya.

Implementasi pengamanan sangat penting untuk menjamin sistem tidak diinterupsi dan diganggu. Proteksi dan pengamanan terhadap perangkat keras dan sistem operasi sama pentingnya. Sistem operasi hanya satu bagian kecil dari seluruh perangkat lunak di suatu sistem.

Tetapi karena sistem operasi mengendalikan pengaksesan ke sumber daya, dimana perangkat lunak lain meminta pengaksesan sumber daya lewat sistem operasi maka sistem operasi menempati posisi yang penting dalam pengamanan sistem. Pengamanan perangkat lunak cenderung memfokuskan pada pengamanan sistem operasi, karena perangkat lunak aplikasi juga memberi resiko keamanan.

Keamanan sistem operasi merupakan bagian masalah keamanan sistem komputer secara total. Pengamanan sistem operasi berarti kecil jika setiap orang dapat melenggang di ruang sistem komputer. Pengamanan secara fisik dengan membatasi pengaksesan fisik secara langsung dengan fasilitas sistem komputer harus dilakukan juga.

9.1. Keamanan

Keamanan sistem komputer adalah untuk menjamin sumber daya tidak digunakan atau dimodifikasi orang tak terotorisasi. Pengamanan termasuk masalah teknis, manajerial, legalitas dan politis.

Keamanan sistem terbagi menjadi tiga, yaitu :

1. Keamanan eksternal (external security).
   Berkaitan dengan pengamanan fasilitas komputer dari penyusup (hacker) dan bencana seperti kebakaran dan kebanjiran.
2. Keamanan interface pemakai (user interface security).
   Berkaitan dengan identifikasi pemakai sebelum pemakai diijinkan mengakses program dan data yang disimpan.
3. Keamanan internal (internal security).
   Berkaitan dengan pengamanan beragam kendali yang dibangun pada perangkat keras dan sistem operasi yang menjamin operasi yang handal dan tak terkorupsi untuk menjaga integritas program dan data. Istilah keamanan (security) dan proteksi (protection) sering digunakan secara bergantian. Untuk menghindari kesalahpahaman, istilah keamanan mengacu ke seluruh masalah keamanan dan istilah mekanisme proteksi mengacu ke mekanisme sistem yang digunakan untuk memproteksi/melindungi informasi pada sistem komputer.

9.2. Masalah-masalah keamanan

Terdapat dua masalah penting, yaitu :

a. Kehilangan data (data loss).
   Dapat disebabkan karena :
   a.1. Bencana.
   - o Kebakaran.
   - o Banjir.
   - o Gempa bumi.
   - o Perang.
   - o Kerusuhan.
   - o Binatang.

a.2. Kesalahan perangkat keras dan perangkat lunak.

- o Ketidak berfungsian pemroses.
- o Disk atau tape yang tidak terbaca.
- o Kesalahan telekomunikasi.
- o Kesalahan program (bugs).

a.3. Kesalahan/kelalaian manusia.

- o Kesalahan pemasukan data.
- o Memasang tape atau disk yang salah.
- o Eksekusi program yang salah.
- o Kehilangan disk atau tape.

Kehilangan data dapat diatasi dengan mengelola beberapa backup dan backup ditempatkan jauh dari data yang online.

b. Penyusup (hacker).

Terdiri dari :

b.1. Penyusup pasif, yaitu yang membaca data yang tak diotorisasi.

b.2 Penyusup aktif, yaitu yang mengubah data yang tak diotorisasi.

Kateogri penyusupan :

- o Lirikan mata pemakai non teknis. Pada sistem time-sharing, kerja pemakai dapat diamati orang sekelilingnya. Bila dengan lirikan itu dapat mengetahui apa yang diketik saat pengisian password, maka pemakai non teknis dapat mengakses fasilitas yang bukan haknya.
- o Penyadapan oleh orang dalam.
- o Usaha hacker dalam mencari uang.
- o Spionase militer atau bisnis.

9.3. Ancaman-ancaman keamanan

Sasaran pengamanan adalah menghindari, mencegah dan mengatasi ancaman terhadap sistem. Kebutuhan keamanan sistem komputer dikategorikan tiga aspek, yaitu :

1. Kerahasiaan (secrecy).

   Adalah keterjaminan bahwa informasi disistem komputer hanya dapat diakses oleh pihak-pihak yang diotorisasi dan modifikasi tetap menjaga konsistensi dan keutuhan data di sistem.

2. Integritas (integrity).

   Adalah keterjaminan bahwa sumber daya sistem komputer hanya dapat dimodifikasi oleh pihak-pihak yang diotorisasi.

3. Ketersediaan (availability).

   Adalah keterjaminan bahwa susmber daya sistem komputer tersedia bagi pihak-pihak yang diotorisasi saat diperlukan.

Tipe-tipe ancaman terhadap keamanan sistem dapat dimodelkan dengan memandang fungsi sistem komputer sebagai penyedia informasi. Berdasarkan fungsi ini, ancaman terhadap sistem komputer dapat dikategorikan menjadi empat ancaman, yaitu :

1. Interupsi (interuption).

   Sumber daya sistem komputer dihancurkan atau menjadi tak tersedia atau tak berguna. Interupsi merupakan ancaman terhadap ketersediaan.

   Contoh : penghancuran bagian perangkat keras, seperti harddisk, pemotongan kabel komunikasi.

2. Intersepsi (interception).

   Pihak tak diotorisasi dapat mengakses sumber daya. Interupsi merupakan ancaman terhadap kerahasiaan. Pihak tak diotorisasi dapat berupa orang atau program komputer.

   Contoh : penyadapan untuk mengambil data rahasia, mengetahui file tanpa diotorisasi.

3. Modifikasi (modification).

   Pihak tak diotorisasi tidak hanya mengakses tapi juga merusak sumber daya. Modifikasi merupakan ancaman terhadap integritas.

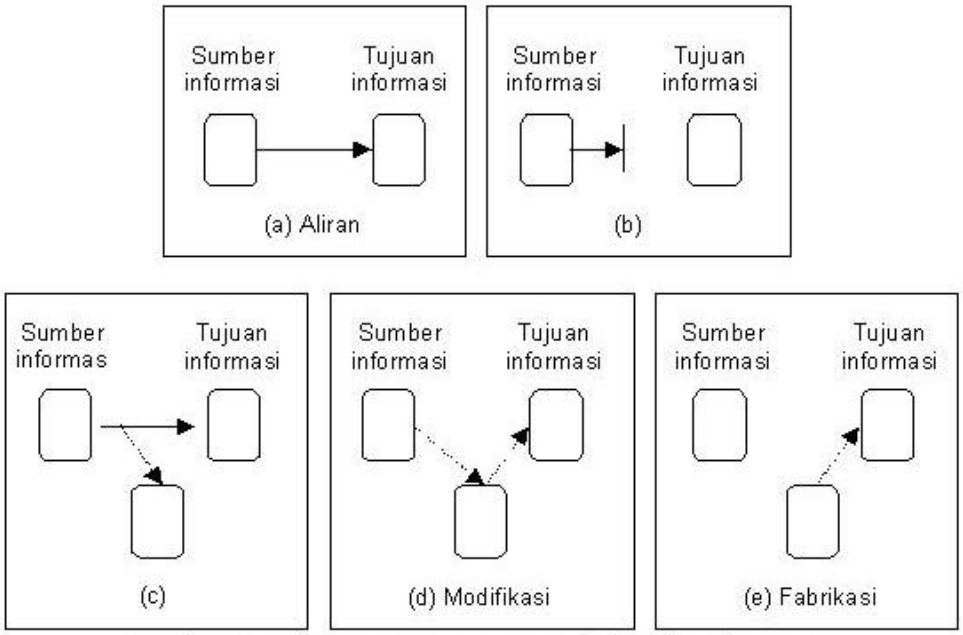
   Contoh : mengubah nilai-nilai file data, mengubah program sehingga bertindak secara berbeda, memodifikasi pesan-pesan yang ditransmisikan pada jaringan.

4. Fabrikasi (fabrication).

   Pihak tak diotorisasi menyisipkan/memasukkan objek-objek palsu ke sistem.

   Fabrikasi merupakan ancaman terhadap integritas.

   Contoh : memasukkan pesan-pesan palsu ke jaringan, penambahan record ke file.



Gambar 9.1 : Skema ancaman-ancaman terhadap sistem komputer.

9.4. Petunjuk pengamanan sistem

Terdapat beberapa prinsip pengamanan sistem komputer, yaitu :

1. Rancangan sistem seharusnya publik.

   Keamanan sistem seharusnya tidak bergantung pada kerahasiaan rancangan mekanisme pengamanan. Mengasumsikan penyusup tidak akan mengetahui cara kerja sistem pengamanan hanya menipu/memperdaya perancang sehingga tidak membuat mekanisme proteksi yang bagus.

2. Dapat diterima.

   Skema yang dipilih harus dapat diterima secara psikologis. Mekanisme proteksi seharusnya tidak menganggu kerja pemakai dan memenuhi kebutuhan otorisasi

pengaksesan. Jika mekanisme tidak mudah digunakan maka tidak akan digunakan atau digunakan secara tak benar.

3. Pemeriksaan otoritas saat itu.

   Sistem tidak seharusnya memeriksa ijin dan menyatakan pengaksesan diijinkan, serta kemudian menetapkan terus informasi ini untuk penggunaan selanjutnya. Banyak sistem memeriksa ijin ketika file dibuka dan setelah itu (operasi-operasi lain) tidak diperiksa. Pemakai yang membuka file dan lupa menutup gile akan terus dapat walaupun pemilik file telah mengubah atribut proteksi file.

4. Kewenangan serendah mungkin.

   Program atau pemakai sistem seharusnya beroperasi dengan kumpulan wewenang serendah mungkin yang diperlukan untuk menyelesaikan tugasnya.

   Default sistem yang digunakan harus tak ada akses sama sekali.

5. Mekanisme yang ekonomis.

   Mekanisme proteksi seharusnya sekecil, sesederhana mungkin dan seragam sehingga memudahkan verifikasi. Proteksi seharusnya dibangun dilapisan terbawah. Proteksi merupakan bagian integral rancangan sistem, bukan mekanisme yang ditambahkan pada rancangan yang telah ada.

9.5. Otentifikasi pemakai

Kebanyakan proteksi didasarkan asumsi sistem mengetahui identitas pemakai. Masalah identifikasi pemakai ketika login disebut otentifikasi pemakai (user authentication). Kebanyakan metode otentifikasi didasarkan pada tiga cara, yaitu :

1. Sesuatu yang diketahui pemakai, misalnya :
   - o Password.
   - o Kombinasi kunci.
   - o Nama kecil ibu mertua.
   - o Dan sebagainya.
2. Sesuatu yang dimiliki pemakai, misalnya :
   - o Badge.
   - o Kartu identitas.
   - o Kunci.
   - o Dan sebagainya.
3. Sesuatu mengenai (ciri) pemakai, misalnya :

- o Sidik jari.
- o Sidik suara.
- o Foto.
- o Tanda tangan.

Password

Pemakai memilih satu kata kode, mengingatnya dan mengetikkan saat akan mengakses sistem komputer. Saat diketikkan, komputer tidak menampilkan dilayar. Teknik ini mempunyai kelemahan yang sangat banyak dan mudah ditembus. Pemakai cenderung memilih password yang mudah diingiat.

Seseorang yang kenal dengan pemakai dapat mencoba login dengan sesuatu yang diketahuinya mengenai pemakai.

Proteksi password dapat ditembus dengan mudah, antara lain :

- o Terdapat file berisi nama depan, nama belakang, nama jalan, nama kota dari kamus ukuran sedang, disertai dengan pengejaan dibalik), nomor plat mobil yang valid, dan string-string pendek karakter acak.
- o Isian di file dicocokkan dengan file password.

Upaya untuk lebih mengamankan proteksi password, antara lain :

1. Salting.

   Menambahkan string pendek ke string password yang diberikan pemakai sehingga mencapai panjang password tertentu.
2. One time password.

   Pemakai harus mengganti password secara teratur. Upaya ini membatasi peluang password telah diketahui atau dicoba-coba pemakai lain.

   Bentuk ekstrim pendekatan ini adalah one time password, yaitu pemakai mendapat satu buku berisi daftar password. Setiap kali pemakai login, pemakai menggunakan password berikutnya yang terdapat di daftar password.

   Dengan one time password, pemakai direpotkan keharusan menjaga agar buku passwordnya jangan sampai dicuri.
3. Satu daftar panjang pertanyaan dan jawaban.

   Variasi terhadap password adalah mengharuskan pemakai memberi satu daftar pertanyaan panjang dan jawabannya. Pertanyaan-pertanyaan dan jawabannya

dipilih pemakai sehingga pemakai mudah mengingatnya dan tak perlu menuliskan di kertas.

Pertanyaan berikut dapat dipakai, misalnya :

o Siapa mertua abang ipar Badru ?

o Apa yang diajarkan Pak Harun waktu SD ?

o Di jalan apa pertama kali ditemukan simanis ?

Pada saat login, komputer memilih salah satu dari pertanyaan-pertanyaan secara acak, menanyakan ke pemakai dan memeriksa jawaban yang diberikan.

4. Tantangan tanggapan (chalenge response).

Pemakai diberi kebebasan memilih suatu algoritma, misalnya x3.

Ketika pemakai login, komputer menuliskan di layar angka 3. Dalam kasus ini pemakai mengetik angka 27. Algoritma dapat berbeda di pagi, sore, dan hari berbeda, dari terminal berbeda, dan seterusnya.

Identifikasi fisik

Pendekatan lain adalah memberikan yang dimiliki pemakai, seperti :

Kartu berpita magnetik

Kartu pengenal dengan selarik pita magnetik. Kartu ini disisipkan ke suatu perangkat pembaca kartu magnetik jika akan mengakses komputer.

Teknik ini biasanya dikombinasikan dengan password, sehingga pemakai dapat login sistem komputer bila memenuhi dua syarat berikut :

o Mempunyai kartu.

o Mengetahui password yang spesifik kartu itu.

ATM merupakan mesin yang bekerja dengan cara ini.

Sidik jari

Pendekatan lain adalah mengukur ciri fisik yang sulit ditiru, seperti :

o Sidik jari dan sidik suara.

o Analisis panjang jari.

o Pengenalan visual dengan menggunakan kamera diterapkan.

o Dan sebagainya.

Analisis tanda tangan

Disediakan papan dan pena khusus dimana pemakai menulis tanda tangan.

Pada teknik ini, bukan membandingkan bentuk tanda tangan tapi gerakan (arah) dan tekanan pena saat menulis. Seorang dapat meniru bentuk tanda tangan tapi sulit meniru persis cara (gerakan dinamis dan irama tekanan) saat pembuatan tanda tangan.

Analisis suatu yang dipunyai pemakai

Pendekatan lain adalah meniru perilaku kucing dan anjing dalam menandai batas wilayah, yaitu urine. Disediakan alat urunalysis. Bila pemakai ingin login, maka pemakai harus membawa sampel urine-nya. Sampel urine dimasukkan ke tabung dan segera dilakukan analisis dan ditentukan apakah termasuk salah satu pemakai sistem. Urinalysis harus dapat dilakukan sesaat.

Pendekatan pengamanan yang bagus, tapi tidak diterima secara psikologis.

Analisis darah

Disediakan satu jarum dimanan pemakai dapat mencobloskan jari sampai menetes darahnya. Darah itu kemudian dianalisis dengan spektografi (blood spectographic analysis). Dari analisis dapat ditentukan mengenai pemilik darah. Pendekatan ini relatif aman tapi tidak diterima secara psikologis.

9.6. Pembatasan

Pembatasan-pembatasan dapat dilakukan sehingga memperkecil peluang penembusan oleh pemakai yang tak diotorisasi, misalnya :

o Pembatasan login.

Login hanya diperbolehkan :

> Pada terminal tertentu.

> Hanya ada waktu dan hari tertentu.

> Pembatasan dengan call-back.

Login dapat dilakukan siapapun. Bila telah sukses login, sistem segera memutuskan koneksi dan memanggil nomor telepon yang telah disepakati.

Penyusup tidak dapat menghubungi lewat sembarang saluran telepon, tapi hanya pada saluran telepon tertentu.

> Pembatasan jumlah usaha login.

Login dibatasi sampai tiga kali dan segera dikunci dan diberitahu ke administrator.

Semua login direkam dan sistem operasi melaporkan informasi-informasi berikut :

>> Waktu, yaitu waktu pemakai login.

>> Terminal, yaitu terminal dimana pemakai login.

Mekanisme proteksi sistem komputer

Pada sistem komputer banyak objek yang perlu diproteksi, yaitu :

1. Objek perangkat keras.

   Objek yang perlu diproteksi, antara lain :

   - o Pemroses.
   - o Segment memori.
   - o Terminal.
   - o Disk drive.
   - o Printer.
   - o Dan sebagainya.

2. Objek perangkat lunak.

   Objek yang perlu diproteksi, antara lain :

   - o Proses.
   - o File.
   - o Basis data.
   - o Semaphore.
   - o Dan sebagainya.

Matriks pengaksesan objek

Masalah proteksi adalah mengenai cara mencegah proses-proses mengakses objek-objek yang tidak diotorisasi. Mekanisme ini juga harus memungkinkan membatasi proses-proses ke suatu subset operasi-operasi legal yang diperlukan. Misalnya proses A dapat membaca file F, tapi tidak menulisinya.

Agar dapat menyediakan mekanisme proteksi berbeda dikembangkan berdasar konsep domain. Domain adalah himpunan pasangan (hak, objek). Tiap pasangan

menspesifikasikan objek dan suatu subset operasi yang dapat dilakukan terhadapnya. Hak dalam konteks ini berarti ijin melakukan suatu operasi.

Proses berjalan pada suatu domain proteksi, yaitu proses merupakan anggota suatu domain atau beberapa domain. Terdapat kumpulan objek yang dapat diakses proses. Untuk tiap objek, proses mempunyai suatu kumpulan hak terhadap objek itu. Proses-proses dapat juga beralih dari satu domain ke domain lain selama eksekusi. Aturan peralihan domain ini bergantung pada sistem.

Domain ditetapkan dengan mendaftarkan pemakai-pemakai yang termasuk domain itu. Proses-proses yang dijalankan pemakai adalah proses-proses pada domain itu dan mempunyai hak akses terhadap objek seperti ditentukan domainnya.

Cara penyimpanan informasi anggota domain

Secara konseptual adalah berupa satu matriks besar, dimana :

o Baris menunjukkan domain.

o Kolom menunjukkan objek.

Tiap elemen matriks mendaftar hak-hak yang dimiliki domain terhadap objek.

Dengan matriks ini, sistem dapat mengetahui hak pengaksesan terhadap objek.

Gambar berikut menunjukkan matriks pengaksesan objek.

	File 1	File 2	Printer 1	Plotter 1	Modem 1
Domain 1	Read	Read Write	Write		
Domain 2	Read			Write	Write
Domain 3		Read Write Execute	Write	Write	Write

Gambar 9.2 : Matriks pengaksesan objek

Untuk sistem-sistem yang mengijinkan peralihan domain dimodelkan dengan menganggap domain sebagai objek, yaitu :

o Jika terdapat operasi enter, berarti mempunyai hak berpindah domain.

	File 1	File 2	Printer 1	Plotter 1	Modem 1	Domain 1	Domain 1	Domain3
Domain 1	Read	Read Write	Write				Enter	
Domain 2	Read			Write	Write	Enter		
Domain 3		Read Write	Write	Write	Write			

Gambar 9.3 : Matriks pengaksesan objek dengan operasi peralihan domain

Gambar diatas menunjukkan matriks pengaksesan objek dengan operasi pengalihan domain. Proses-proses pada domain 1 dapat berpindah ke domain 2 dan proses pada domain 2 dapat berpindah ke domain 1.

ACL (Access Control List)

Matriks pengaksesan objek akan berbentuk matrik jarang (sparse matrix).
Matrik jarang memboroskan ruang penyimpanan dan lambat karena memerlukan ruang besar, Dua alternatif untuk memperbaikinya adalah :
o Menyimpan matriks sebagai perbaris.
o Menyimpan matriks sebagai perkolom.

Teknik yang digunakan adalah mengasosiasikan tiap objek dengan senarai terurut berisi semua domain yang boleh mengakses dan operasi-operasi yang dibolehkan (bagaimana). Temnik ini menghasilkan senarai disebut ACL.
Contoh :

File 1	: (Yani, *, rwx), (Soni, *, rw-)
File 2	: (Yani, system, rwx)
Printer 1	: (Yani, *, -w-), (Elsa, karyawan, -w-)
Plotter 1	: (Yusuf, *, -w-)
Modem 1	: (Yuniar, *, -w-)

Gambar 9.4 : ACL (Access Control List)

Tiap ACL yang disebutkan di kurung menyatakan komponen uid (user ID), gid (group ID) dan hak akses. Dengan ACL, dimungkinkan mencegah uid, gid spesifik mengakses objek sementara mengijinkan yang lain. Pemilik objek dapat mengubah ACL kapanpun.Cara ini untuk mempermudah pencegahan/pelarangan pengaksesan yang sebelumnya diperbolehkan. Penyimpanan dilakukan hanya untuk isian yang tak kosong.

Kapabilitas

Cara lain adalah memecah matrik perbaris. Diasosiasikan tiap proses satu daftar objek yang boleh diakses, bila terdapat tanda operasi yang diijinkan padanya atau domainnya.
Senarai ini disebut senarai kapabilitas (capabilities list).
Contoh :

	Tipe	Hak	Objek
0	Berkas	Rwx	Pointer ke file 2
1	Printer	-w-	Pointer ke printer 1
2	Plotter	-w-	Pointer ke plotter 1
3	Modem	-w-	Pointer ke modem 1

Gambar 9.5 : Senarai kapabilitas untuk domain 3 dari gambar 9.2

9.8. Program-program jahat

Ancaman-ancaman canggih terhadap sistem komputer adalah program yang mengeksploitasi kelemahan sistem operasi. Kita berurusan dengan program aplikasi begitu juga program utilitas seperti editor dan kompilator.
Terdapat taksonomi ancaman perangkat lunak atau klasifikasi program jahat (malicious program), yaitu :

1. Program-program yang memerlukan program inang (host program).
   Fragmen program tidak dapat mandiri secara independen dari suatu program aplikasi, program utilitas atau program sistem.
2. Program-program yang tidak memerlukan program inang.
   Program sendiri yang dapat dijadwalkan dan dijalankan oleh sistem operasi.

Pembagian atau taksonomi menghasilkan tipe-tipe program jahat sebagai berikut :

1. Bacteria.
   Bacteria adalah program yang mengkonsumsi sumber daya sistem dengan mereplikasi dirinya sendiri. Bacteria tidak secara eksplisit merusak file. Tujuan program ini hanya satu yaitu mereplikasi dirinya.
   Program bacteria yang sederhana bisa hanya mengeksekusi dua kopian dirinya secara simultan pada sistem multiprogramming atau menciptakan dua file baru, masing-masing adalah kopian file program bacteria.
   Kedua kopian in kemudian mengkopi dua kali, dan seterusnya.
2. Logic bomb.
   Logic bomb adalah logik yang ditempelkan pada program komputer agar memeriksa suatu kumpulan kondisi di sistem. Ketika kondisi-kondisi yang dimaksud ditemui, logik mengeksekusi suatu fungsi yang menghasilkan aksi-aksi tak diotorisasi.
   Logic bomb menempel pada suatu program resmi yang diset meledak ketika kondisi-kondisi tertentu dipenuhi. Contoh kondisi-kondisi untuk memicu logic bomb adalah ada atau tudak adanya file-file tertentu, hari tertentu baru minggu atau tanggal, atau pemakai menjalankan aplikasi tertentu.
   Begitu terpicu, bomb mengubah atau menghapus data atau seluruh file, menyebabkan mesin terhenti, atau mengerjakan perusakan lain.
3. Trapdoor.
   Trapdoor adalah titik masuk tak terdokumentasi rahasia di satu program untuk memberikan akses tanpa metode-metode otentifikasi normal.
   Trapdoor telah dipakai secara benar selama bertahun-tahun oleh pemogram untuk mencari kesalahan program. Debugging dan testing biasanya dilakukan pemogram saat mengembangkan aplikasi. Untuk program yang mempunyai prosedur otentifikasi atau setup lama atau memerlukan pemakai memasukkan nilai-nilai berbeda untuk menjalankan aplikasi maka debugging akan lama bila harus melewati prosedur-prosedur tersebut.

Untuk debug program jenis ini, pengembang membuat kewenangan khusus atau menghilangkan keperluan setup dan otentifikasi. Trapdoor adalah kode yang menerima suatu barisan masukan khusus atau dipicu dengan menjalankan ID pemakai tertentu atau barisan kejahatan tertentu. Trapdoor menjadi ancaman ketika digunakan pemrogram jahat untuk memperoleh pengkasesan tak diotorisasi. Pada kasus nyata, auditor (pemeriks) perangkat lunak dapat menemukan trapdoor pada produk perangkat lunak dimana nama pencipta perangkat lunak berlakuk sebagai password yang memintas proteksi perangkat lunak yang dibuatnya. Adalah sulit mengimplementasikan kendali-kendali perangkat lunak untuk trapdoor.

4. Trojan horse.

   Trojan horse adalah rutin tak terdokumentasi rahasia ditempelkan dalam satu program berguna. Program yang berguna mengandung kode tersembunyi yang ketika dijalankan melakukan suatu fungsi yang tak diinginkan.

   Eksekusi program menyebabkan eksekusi rutin rahasia ini. Program-program trojan horse digunakan untuk melakukan fungsi-fungsi secara tidak langsung dimana pemakai tak diotorisasi tidak dapat melakukannya secara langsung. Contoh, untuk dapat mengakses file-file pemakai lain pada sistem dipakai bersama, pemakai dapat menciptakan program trojan horse.

   Trojan horse ini ketika program dieksekusi akan mengubah ijin-ijin file sehinga file-file dapat dibaca oleh sembarang pemakai. Pencipta program dapat menyebarkan ke pemakai-pemakai dengan menempatkan program di direktori bersama dan menamai programnya sedemikian rupa sehingga disangka sebagai program utilitas yang berguna. Program trojan horse yang sulit dideteksi adalah kompilator yang dimodifikasi sehingga menyisipkan kode tambahan ke program-program tertentu begitu dikompilasi, seperti program login. Kode menciptakan trapdoor pada program login yang mengijinkan pencipta log ke sistem menggunakan password khusus.

   Trojan horse jenis ini tak pernah dapat ditemukan jika hanya membaca program sumber. Motivasi lain dari trojan horse adalah penghancuran data. Program muncul sebagai melakukan fungsi-fungsi berguna (seperti kalkulator), tapi juga secara diam-diam menghapus file-file pemakai.

   Trojan horse biasa ditempelkan pada program-program atau rutin-rutin yang diambil dari BBS, internet, dan sebagainya.

5. Virus.

   Virus adalah kode yang ditempelkan dalam satu program yang menyebabkan pengkopian dirinya disisipkan ke satu program lain atau lebih.

   Program menginfeksi program-program lain dengan memodifikasi program-program itu. Modifikasi itu termasuk memasukkan kopian program virus yang kemudian dapat menginfeksi program-program lain. Selain hanya progasi, virus biasanya melakuka fungsi yang tak diinginkan. Seperti virus biologis, pada virus komputer terdapat kode intruksi yang dapat membuat kpian sempurna dirinya. Ketika komputer yang terinfeksi berhubungan (kontak) dengan perangkat lunak yang belum terinfeksi, kopian virus memasuki program baru. Infeksi dapat menyebar dari komputer ke komputer melalui pemakai-pemakai yang menukarkan disk atau mengirim program melalui jaringan. Pada lingkungan jaringan, kemampuan mengakses aplikasi dan layanan-layanan komputer lain merupakan fasilitas sempurna penyebaran virus.

6. Worm.

   Adalah program yang dapat mereplikasi dirinya dan mengirim kopian-kopian dari komputer ke komputer lewat hubungan jaringan. Begitu tiba, worm diaktifkan untuk mereplikasi dan progasai kembali. Selain hanya propagasi, worm biasanya melakukan fungsi yang tak diinginkan. Network worm menggunakan hubungan jaringan untuk menyebar dari sistem ke sistem lain.

   Sekali aktif di suatu sistem, network worm dapat berlaku seperti virus atau bacteria, atau menempelkan program trojan horse atau melakukan sejumlah aksi menjengkelkan atau menghancurkan. Untuk mereplikasi dirinya, network worm menggunakan suatu layanan jaringan, seperti :

   - o Fasilitas surat elektronik (electronic mail facility), yaitu worm mengirimkan kopian dirinya ke sistem-sistem lain.
   - o Kemampuan eksekusi jarak jauh (remote execution capability), yaitu worm mengeksekusi kopian dirinya di sistem lain.
   - o Kemampuan login jarak jauh (remote login capability), yaitu worm log pada sistem jauh sebagai pemakai dan kemudian menggunakan perintah untuk mengkopi dirinya dari satu sistem ke sistem lain.

   Kopian program worm yang baru kemudian dijalankan di sistem jauh dan melakukan fungsi-fungsi lain yang dilakukan di sistem itu, worm terus menyebar

dengan cara yang sama. Network worm mempunyai ciri-ciri yang sama dengan virus komputer, yaitu mempunyai fase-fase sama, yaitu :

- Dormant phase.
- Propagation phase.
- Trigerring phase.
- Execution phase.

Network worm juga berusaha menentukan apakah sistem sebelumnya telah diinfeksi sebelum mengirim kopian dirinya ke sistem itu.

9.9. Virus dan antivirus

Virus adalah sama dengan program komputer lain. Perbedaan dengan program lain adalah virus dapat mencantolkan dirinya ke program lain dan mengeksekusi kodenya secara rahasia setiap kali program inang berjalan.

Masalah yang ditimbulkan virus adalah virus sering merusak sistem komputer seperti menghapus file, partisi disk, atau mengacaukan program. Virus mengalami siklus hidup empat fase (tahap), yaitu :

1. Fase tidur (dormant phase).

   Virus dalam keadaan menganggur. Virus akan tiba-tiba aktif oleh suatu kejadian seperti tibanya tanggal tertentu, kehadiran program atau file tertentu, atau kapasitas disk yang melewati batas. Tidak semua virus mempunyai tahap ini.

2. Fase propagasi (propagation phase).

   Virus menempatkan kopian dirinya ke program lain atau daerah sistem tertentu di disk. Program yang terinfeksi virus akan mempunyai kloning virus. Kloning virus itu dapat kembali memasuki fase propagasi.

3. Fase pemicuan (triggering phase).

   Virus diaktifkan untuk melakukan fungsi tertentu. Seperti pada fase tidur, fase pemicuan dapat disebabkan beragam kejadian sistem termasuk penghitungan jumlah kopian dirinya.

4. Fase eksekusi (execution phase).

   Virus menjalankan fungsinya, fungsinya mungkin sepele seperti sekedar menampilkan pesan dilayar atau merusak seperti merusak program dan file-file data, dan sebagainya.

Kebanyakan virus melakukan kerjanya untuk suatu sistem operasi tertentu, lebih spesifik lagi pada platform perangkat keras tertentu. Virus-virus dirancang memanfaatkan rincian-rincian dan kelemahan-kelemahan sistem tertentu.

Infeksi virus

Sekali virus telah memasuki sistem dengan menginfeksi satu program, virus berada dalam posisi menginfeksi beberapa atau semua file .exe lain di sistem itu saat program yang terinfeksi dieksekusi. Infeksi virus dapat sepenuhnya dihindari dengan mencegah virus masuk sistem. Pencegahan ini sangat luar biasa sulit karena virus dapat menjadi bagian program di luar sistem. Kebanyakan virus mengawali infeksinya pengkopian disk yang telah terinfeksi virus. Banyak disk berisi game atau utilitas di rumah dikopikan ke mesin kantor. Disk berisi virus pun dapat terdapt di disk yang dikirim produsen aplikasi. Hanya sejumlah kecil infeksi virus yang dimulai dari hubungan jaringan.

Tipe-tipe virus

Saat ini perkembangan virus masih berlanjut, terjadi perlombaan antara penulis virus dan pembuat antivirus. Begitu satu tipe dikembangkan antivirusnya, tipe virus yang lain muncul. Klasifikasi tipe virus adalah sebagai berikut :

o Parasitic virus.

Merupakan virus tradisional dan bentuk virus yang paling sering.

Tipe ini mencantolkan dirinya ke file .exe. Virus mereplikasi ketika program terinfeksi dieksekusi dengan mencari file-file .exe lain untuk diinfeksi.

o Memory resident virus.

Virus memuatkan diri ke memori utama sebagai bagian program yang menetap.

Virus menginfeksi setiap program yang dieksekusi.

o Boot sector virus.

Virus menginfeksi master boot record atau boot record dan menyebar saat sistem diboot dari disk yang berisi virus.

o Stealth virus.

Virus yang bentuknya telah dirancang agar dapat menyembunyikan diri dari deteksi perangkat lunak antivirus.

o Polymorphic virus.
Virus bermutasi setiap kali melakukan infeksi. Deteksi dengan penandaan virus tersebut tidak dimungkinkan.

Penulis virus dapat melengkapi dengan alat-alat bantu penciptaan virus baru (virus creation toolkit, yaitu rutin-rutin untuk menciptakan virus-virus baru). Dengan alat bantu ini penciptaan virus baru dapat dilakukan dengan cepat. Virus-virus yang diciptakan dengan alat bantu biasanya kurang canggih dibanding virus-virus yang dirancang dari awal.

Antivirus

Solusi ideal terhadap ancaman virus adalah pencegahan. Jaringan diijinkan virus masuk ke sistem. Sasaran ini, tak mungkin dilaksanakan sepenuhnya.
Pencegahan dapat mereduksi sejumlah serangan virus. Setelah pencegahan terhadap masuknya virus, maka pendekatan berikutnya yang dapat dilakukan adalah :
o Deteksi.
Begitu infeksi telah terjadi, tentukan apakah infeksi memang telah terjadi dan cari lokasi virus.
o Identifikasi.
Begitu virus terdeteksi maka identifikasi virus yang menginfeksi program.
o Penghilangan.
Begitu virus dapat diidentifikasi maka hilangkan semua jejak virus dari program yang terinfeksi dan program dikembalikan ke semua (sebelum terinfeksi).

Jika deteksi virus sukses dilakukan, tapi identifikasi atau penghilangan jejak tidak dapat dilakukan, maka alternatif yang dilakukan adalah menghapus program yang terinfeksi dan kopi kembali backup program yang masih bersih.
Sebagaimana virus berkembang dari yang sederhana menjadi semakin canggih, begitu juga paket perangkat lunak antivirus. Saat ini program antivirus semakin kompleks dan canggih.

Perkembangan program antivirus dapat diperiode menjadi empat generasi, yaitu :

1. Generasi pertama : sekedar scanner sederhana.

   Antivirus menscan program untuk menemukan penanda (signature) virus. Walaupun virus mungkin berisi karakter-karakter varian, tapi secara esensi mempunyai struktur dan pola bit yang sama di semua kopiannya. Teknis ini terbatas untuk deteksi virus-virus yang telah dikenal.

   Tipe lain antivirus generasi pertama adalah mengelola rekaman panjang (ukuran) program dan memeriksa perubahan panjang program.

2. Generasi kedua : scanner yang pintar (heuristic scanner).

   Antivirus menscan tidak bergantung pada penanda spesifik. Antivirus menggunakan aturan-aturan pintar (heuristic rules) untuk mencari kemungkinan infeksi virus.Teknik yang dipakai misalnya mencari fragmen-fragmen kode yang sering merupakan bagian virus. Contohnya, antivirus mencari awal loop enkripsi yang digunakan polymorphic virus dan menemukan kunci enkripsi. Begitu kunci ditemukan, antivirus dapat mendeskripsi virus untuk identifikasi dan kemudian menghilangkan infeksi virus.

   Teknik ini adalah pemeriksanaan integritas. Checksum dapat ditambahkan di tiap program. Jika virus menginfeksi program tanpa mengubah checksum, maka pemeriksaan integritas akan menemukan perubahan itu.

   Untuk menanggulangi virus canggih yang mampu mengubah checksum saat menginfeksi program, fungsi hash terenkripsi digunakan. Kunci enkripsi disimpan secara terpisah dari program sehingga program tidak dapat menghasilkan kode hash baru dan mengenkripsinya. Dengan menggunakan fungsi hash bukan checksum sederhana maka mencegah virus menyesuaikan program yang menghasilkan kode hash yang sama seperti sebelumnya.

3. Generasi ketiga : jebakan-jebakan aktivitas (activity trap).

   Program antivirus merupakan program yang menetap di memori (memory resident program). Program ini mengidentifikasi virus melalui aksi-aksinya bukan dari struktur program yang diinfeksi. Dengan antivirus semacam in tak perlu mengembangkan penanda-penanda dan aturan-aturan pintar untuk beragam virus yang sangat banyak. Dengan cara ini yang diperlukan adalah mengidentifikasi kumpulan instruksi yang berjumlah sedikit yang mengidentifikasi adanya usaha infeksi. Kalau muncul kejadian ini, program antivirus segera mengintervensi.

4. Generasi keempat : proteksi penuh (full featured protection).

   Antivirus generasi ini menggunakan beragam teknik antivirus secara bersamaan. Teknik-teknik ini meliputi scanning dan jebakan-jebakan aktivitas. Antivirus juga mempunyai senarai kapabilitas pengaksesan yang membatasi kemampuan virus memasuki sistem dan membatasi kemampuan virus memodifikasi file untuk menginfeksi file.

Pertempuran antara penulis virus dan pembuat antivirus masih berlanjut. Walau beragam strategi lebih lengkap telah dibuat untuk menanggulangi virus, penulis virus pun masih berlanjut menulis virus yang dapat melewati barikade-barikade yang dibuat penulis antivirus. Untuk pengaman sistem komputer, sebaiknya pengaksesandan pemakaian komputer diawasi dengan seksama sehingga tidak menjalankan program atau memakai disk yang belum terjamin kebersihannya dari infeksi virus. Pencegahan terbaik terhadap ancaman virus adalah mencegah virus memasuki sistem disaat yang pertama.

DAFTAR PUSTAKA

1. Hariyanto, Bambang, Ir., *Sistem Operasi*, Penerbit Informatika, Bandung, 1999
2. Tanenbaum, Andrew S., *Modern Operating Systems*, Prentice Hall Inc., 1992