PERTEMUAN 13:

SISTEM PENYIMPANAN DISK

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai jenis-jenis keamanan sistem dan proteksi, Anda harus mampu:

1.1 Penyebab data hilang

1.2 Intruder

1.3 Membedakan keamanan sistem dan proteksi

B. URAIAN MATERI

Tujuan Pembelajaran 1.1:
Penyebab data hilang

Penjadwalan Disk

Penjadwalan disk merupakan salah satu hal yang sangat penting dalam mencapai efisiensi perangkat keras. Bagi disk drives, efisiensi dipengaruhi oleh kecepatan waktu akses dan besarnya disk bandwith. Waktu akses memiliki dua komponen utama yaitu waktu pencarian dan waktu rotasi disk(rotational latency). Waktu pencarian adalah waktu yang dibutuhkan disk arm untuk menggerakkan head ke bagian silinder disk yang mengandung sektor yang diinginkan. Waktu rotasi disk adalah waktu tambahan yang dibutuhkan untuk menunggu perputaran disk agar head dapat berada di atas sektor yang diinginkan. Disk bandwith adalah total jumlah bytes yang ditransfer dibagi dengan total waktu dari awal permintaan transfer sampai transfer selesai. Kita bisa meningkatkan waktu akses dan bandwidth dengan menjadwalkan permintaan dari I/O dalam urutan tertentu.

Apabila suatu proses membutuhkan pelayanan I/O dari atau menuju disk, maka proses tersebut akan melakukan system call ke sistem operasi. Permintaan tersebut membawa beberapa informasi, antara lain:

1. Apakah operasi input atau output.
2. Alamat disk untuk proses tersebut.
3. Alamat memori untuk proses tersebut

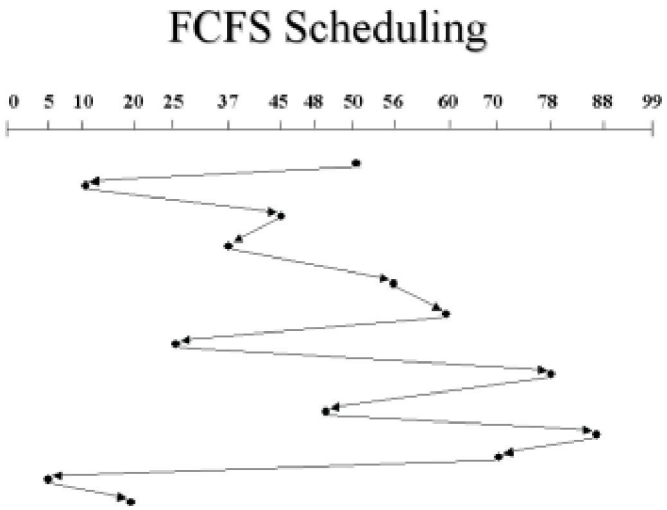
4. Jumlah bytes yang akan ditransfer

Pelayanan akan dilayani pada suatu proses apabila disk drive beserta pengendali tersedia untuk proses tersebut. Apabila disk drive dan pengendali sedang sibuk melayani proses lain, maka semua permintaan yang memerlukan pelayanan disk tersebut akan diletakkan pada suatu antrian permintaan untuk disk tersebut. Dengan demikian, jika suatu permintaan telah dilayani, maka sistem operasi melayani permintaan dari antrian berikutnya.

Penjadwalan FCFS

Penjadwalan disk FCFS melayani permintaan sesuai dengan antrian dari banyak proses yang meminta layanan. Secara umum algoritma FCFS ini sangat adil walaupun ada kelemahan dalam algoritma ini dalam hal kecepatannya yang lambat. Sebagai contoh, antrian permintaan pelayanan disk untuk proses I/O pada blok dalam silinder adalah sebagai berikut: 10, 45, 37, 56, 60, 25, 78, 48, 88, 70, 5, 20. Jika head pada awalnya berada pada 50, maka head akan bergerak dulu dari 50 ke 10, kemudian 45, 37, 56, 60, 25, 78, 48, 88, 70, 5 dan terakhir 20, dengan total pergerakan head sebesar 362 silinder.

Dari contoh diatas, kita dapat melihat permasalahan dengan menggunakan penjadwalan jenis ini yaitu pergerakan dari 78 ke 48 dan kembali lagi ke 88. Jika permintaan terhadap silinder 88 dapat dilayani setelah permintaan 78, setelah selesai baru melayani permintaan 48, maka pergerakan total head dapat dikurangi, sehingga dengan demikian pendayagunaan akan meningkat.



Gambar 13.1. Penjadwalan FCFS

Gambar ini diadaptasi dari [Silberschatz2002, halaman 494].

Penjadwalan SSTF

Shortest-Seek-Time-First (SSTF) merupakan algoritma yang melayani permintaan berdasarkan waktu pencarian atau waktu pencarian paling kecil dari posisi head terakhir. Karena waktu pencarian meningkat seiring dengan jumlah silinder yang dilewati oleh head, maka SSTF memilih permintaan yang paling dekat posisinya di disk terhadap posisi head terakhir. Pergerakan dari contoh diatas yaitu 50 ke 48, lalu ke 45, 37, 25, 20, 10, 5, 56, 60, 70, 78, 88.

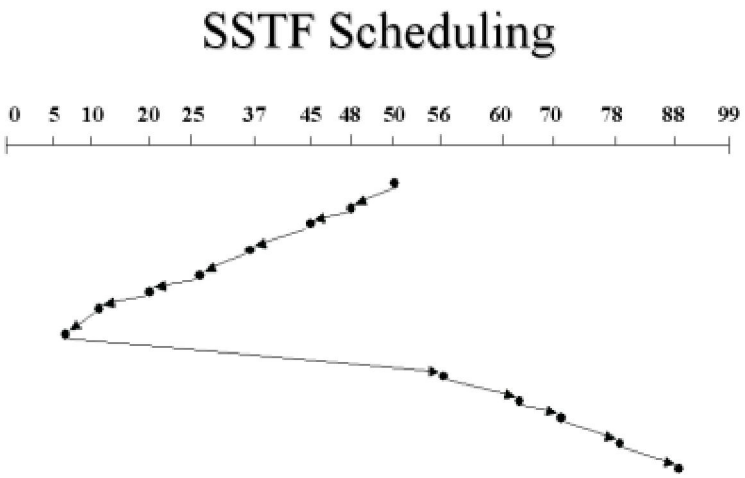
Perhatikan contoh antrian permintaan yang kita sajikan pada penjadwalan FCFS, permintaan paling dekat dengan posisi head saat itu (50) adalah silinder 48. Jika kita penuhi permintaan 48, maka yang terdekat berikutnya adalah silinder 45. Dari 45, silinder 37 letaknya lebih dekat ke 45 dibandingkan silinder 56,jadi 37 dilayani duluan. Selanjutnya, dilanjutkan ke silinder 25, 20, 10, 5, 56, 60, 70, 78 dan terakhir adalah 88.

Metode penjadwalan ini hanya menghasilkan total pergerakan head sebesar 128 silinder -- kira-kira sepertiga dari yang dihasilkan penjadwalan FCFS. Algoritma SSTF ini memberikan peningkatan yang cukup signifikan dalam hal pendayagunaan atau kinerja sistem.

Penjadwalan SSTF merupakan salah satu bentuk dari penjadwalan shortest-job-first (SJF), dan karena itu maka penjadwalan SSTF juga dapat mengakibatkan starvation pada suatu saat tertentu. Hal ini dapat terjadi bila ada permintaan untuk mengakses bagian yang berada di silinder terdalam. Jika kemudian berdatangan lagi permintaan-permintaan yang letaknya lebih dekat dengan permintaan terakhir yang dilayani maka permintaan dari silinder terluar akan menunggu lama dan sebaliknya. Walaupun algoritma SSTF jauh lebih cepat dibandingkan dengan FCFS, namun untuk keadilan layanan SSTF lebih buruk dari penjadwalan FCFS.

Gambar 7-8. Penjadwalan SSTF

Gambar 7-8. Penjadwalan SSTF



Gambar 13.2. Penjadwalan SSTF

Gambar ini diadaptasi dari [Silberschatz2002, halaman 494]

Penjadwalan SCAN

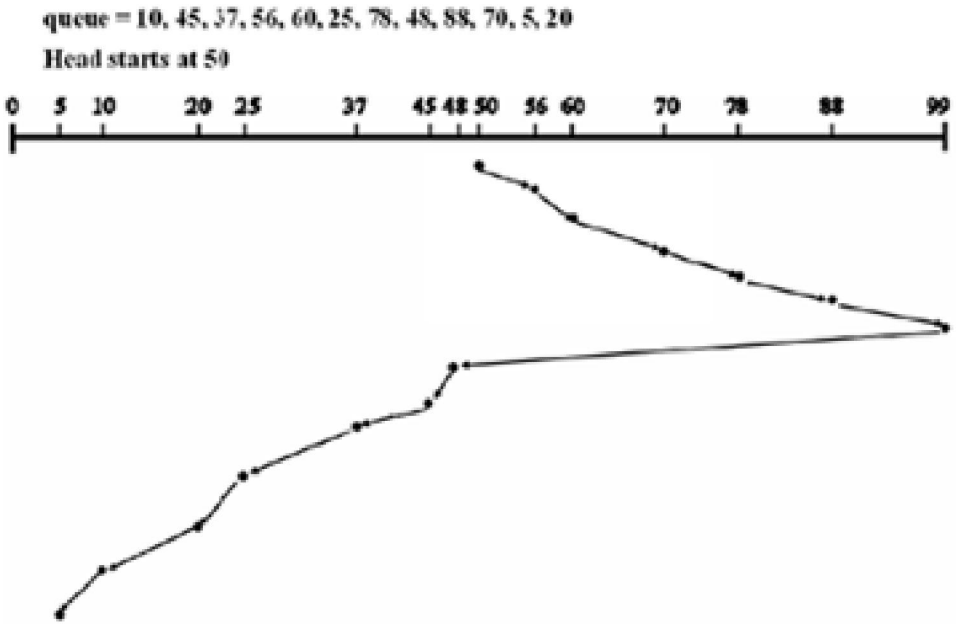
Pada algoritma ini disk arm bergerak menuju ke silinder paling ujung dari disk, kemudian setelah sampai di silinder paling ujung, disk arm akan berbalik arah geraknya menuju ke silinder paling ujung lainnya. Algoritma SCAN disebut juga Algoritma lift/ elevator karena algoritma ini cara kerjanya sama seperti algoritma yang umum dipakai oleh lift untuk melayani penggunanya, yaitu lift akan melayani orang-orang yang akan naik ke atas dulu, setelah sampai di lantai tertinggi, baru lift akan berbalik arah geraknya untuk melayani orang-orang yang akan turun. Dalam pergerakannya yang seperti lift itu, disk arm hanya bisa melayani permintaan-permintaan yang berada di depan arah geraknya terlebih dahulu. Bila ada permintaan yang berada di belakang arah geraknya, permintaan tersebut harus menunggu sampai disk arm mencapai salah satu silinder paling ujung dari disk, kemudian berbalik arah geraknya untuk melayani permintaan tersebut.

Contoh : (lihat gambar 7-7) Jika disk head sedang berada di silinder 50, dan sedang bergerak menuju silinder 99, maka permintaan yang bisa dilayani berikutnya adalah yang terdekat dengan silinder 50, tetapi masih berada di depan arah geraknya, yaitu: silinder 56. Begitu seterusnya disk arm melayani permintaan yang berada di depannya sampai disk arm mencapai silinder 99 dan berbalik arah gerak menuju ke silinder 0. Maka setelah disk arm berbalik arah gerak, permintaan di silinder 45 baru bisa dilayani.

Keunggulan dari algoritma SCAN adalah total pergerakan disk arm memiliki batas atas, yaitu 2 kali dari jumlah total silinder pada disk. Tetapi di samping itu masih ada beberapa kelemahan yang dimiliki oleh algoritma ini.

Dari contoh di gambar 7-7 terlihat salah satu kelemahan algoritma SCAN: permintaan di silinder 88 sebenarnya sudah merupakan permintaan yang paling ujung, tetapi disk arm harus bergerak sampai silinder 99 dulu, baru kemudian bisa berbalik arah geraknya. Bukankah hal seperti itu sangat tidak efisien? Mengapa disk arm tidak langsung berbalik arah geraknya sesudah sampai di silinder 88? Kelemahan ini akan dijawab oleh algoritma LOOK yang akan dibahas pada sub-bab berikutnya.

Kelemahan lain dari algoritma SCAN yaitu bisa menyebabkan terjadinya starvation. Begitu disk arm berbalik arah geraknya dari silinder 99, maka silinder yang berada dekat di depan arah gerak disk arm baru saja dilayani, sedangkan silinder-silinder yang dekat dengan silinder 0 sudah lama menunggu untuk dilayani. Bila kemudian bermunculan permintaan-permintaan baru yang dekat dengan silinder 99 lagi, maka permintaan-permintaan baru itulah yang akan dilayani, sehingga permintaan-permintaan yang dekat dengan silinder 0 akan semakin "lapar". Karena kelemahan yang kedua inilah muncul modifikasi dari algoritma SCAN, yaitu C-SCAN yang akan kita bahas berikutnya.



Gambar 13.3. Penjadwalan SCAN

Gambar ini diadaptasi dari [Silberschatz2002, halaman 495]

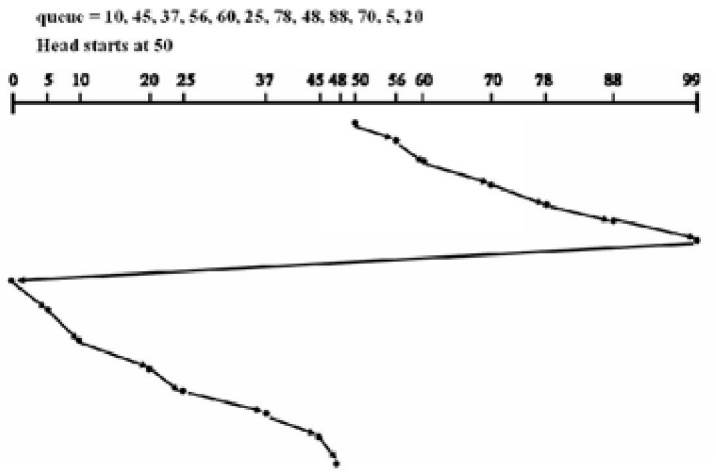
Penjadwalan C-SCAN

Algoritma Circular SCAN (C-SCAN) merupakan hasil modifikasi algoritma SCAN untuk mengurangi kemungkinan starvation yang bisa terjadi pada SCAN. Perbedaan C-SCAN dengan SCAN hanya pada bagaimana pergerakan disk arm setelah sampai ke salah satu silinder paling ujung. Pada algoritma SCAN, disk arm akan berbalik arah menuju ke silinder paling ujung yang lain sambil tetap melayani permintaan yang berada di depan arah pergerakan disk arm, sedangkan pada algoritma C-SCAN sesudah mencapai silinder paling ujung, maka disk arm akan bergerak cepat ke silinder paling ujung lainnya tanpa melayani permintaan.

Contoh: (lihat gambar 7-8) Setelah sampai di silinder 99, disk arm akan bergerak dengan cepat ke silinder 0 tanpa melayani permintaan selama dalam perjalanannya. Kemudian setelah sampai di silinder 0, baru disk arm akan bergerak ke arah silinder 99 lagi sambil melayani permintaan.

Dengan pergerakan yang seperti demikian, seolah-olah disk arm hanya bergerak 1 arah dalam melayani permintaan. Tetapi dalam algoritma C-SCAN masih terkandung kelemahan yang juga dimiliki oleh algoritma SCAN, yaitu disk arm harus sampai di silinder 99 atau silinder 0 terlebih dahulu sebelum bisa berbalik arah. Untuk itulah dibuat algoritma LOOK yang akan kita bahas berikutnya.

Gambar 7-10. Penjadwalan C-SCAN



Gambar ini diadaptasi dari [Silberschatz2002, halaman 496].

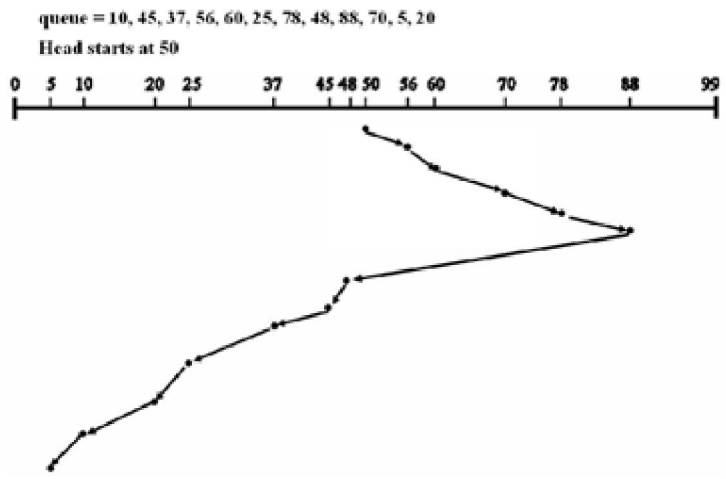
Penjadwalan LOOK

Sesuai dengan namanya, algoritma ini seolah-olah seperti bisa "melihat". Algoritma ini memperbaiki kelemahan SCAN dan C-SCAN dengan cara melihat apakah di depan arah pergerakannya masih ada permintaan lagi atau tidak. Bila tidak ada lagi permintaan di depannya, disk arm bisa langsung berbalik arah geraknya. Penjadwalan LOOK seperti SCAN yang lebih "pintar".

Contoh: (lihat gambar 7-9) Ketika disk head sudah selesai melayani permintaan di silinder 88, algoritma ini akan "melihat" bahwa ternyata di depan arah pegerakannya sudah tidak ada lagi permintaan yang harus dilayani. Oleh karena itu disk arm bisa langsung berbalik arah geraknya sehingga permintaan yang menunggu untuk dilayani bisa mendapatkan pelayanan lebih cepat.

Kelemahan algoritma ini sama seperti kelemahan algoritma SCAN bahwa bisa terjadi starvation untuk situasi yang sama pula dengan yang menyebabkan terjadinya starvationpada algoritma SCAN. Oleh karena itulah dibuat lagi suatu algoritma yang lebih baik untuk memperbaiki algoritma ini, yaitu: C-LOOK.

Gambar 7-11. Penjadwalan LOOK



Gambar ini diadaptasi dari [Silberschatz2002, halaman 497].

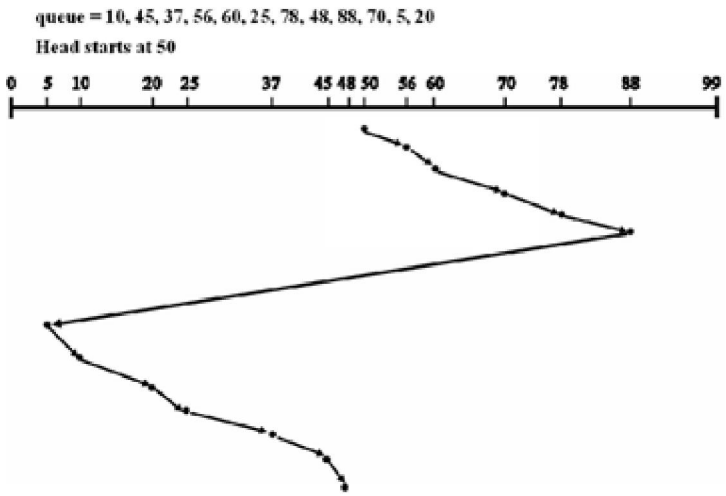
Penjadwalan C-LOOK

Algoritma ini berhasil memperbaiki kelemahan-kelemahan algoritma SCAN, C-SCAN, dan LOOK. Algoritma C-LOOK memperbaiki kelemahan LOOK sama seperti algoritma C-SCAN memperbaiki kelemahan SCAN, yaitu pada cara pergerakan disk arm setelah mencapai silinder yang paling ujung.

Contoh: (lihat gambar 7-10) dengan memiliki kemampuan "melihat" algoritma LOOK, setelah melayani permintaan di silinder 88, disk arm akan bergerak dengan cepat ke silinder 5, yaitu permintaan di silinder yang terletak paling dekat dengan silinder 0.

Dengan cara pergerakan disk arm yang mengadaptasi keunggulan dari C-SCAN dan LOOK, algoritma ini bisa mengurangi terjadinya starvation, dengan tetap menjaga efektifitas pergerakan disk arm.

Gambar 7-12. Penjadwalan C-LOOK



Gambar ini diadaptasi dari [Silberschatz2002, halaman 497].

Pemilihan Algoritma Penjadwalan Disk

Dari seluruh algoritma yang sudah kita bahas di atas, tidak ada algoritma yang terbaik untuk semua keadaan yang terjadi. SSTF lebih umum dan memiliki prilaku yang lazim kita temui. SCAN dan C-SCAN memperlihatkan kemampuan yang lebih baik bagi sistem yang menempatkan beban pekerjaan yang berat kepada disk, karena algoritma tersebut memiliki

masalah starvation yang paling sedikit. SSTF dan LOOK sering dipakai sebagai algoritma dasar pada sistem operasi.

Dengan algoritma penjadwalan yang mana pun, kinerja sistem sangat tergantung pada jumlah dan tipe permintaan. Sebagai contoh, misalnya kita hanya memiliki satu permintaan, maka semua algoritma penjadwalan akan dipaksa bertindak sama. Sedangkan permintaan sangat dipengaruhi oleh metode penempatan berkas. Karena kerumitan inilah, maka algoritma penjadwalan disk harus ditulis dalam modul terpisah dari sistem operasi, jadi dapat saling mengganti dengan algoritma lain jika diperlukan.

Namun perlu diingat bahwa algoritma-algoritma di atas hanya mempertimbangkan jarak pencarian, sedangkan untuk disk modern, rotational latency dari disk sangat menentukan. Tetapi sangatlah sulit jika sistem operasi harus memperhitungkan algoritma untuk mengurangi rotational latency karena disk modern tidak memperlihatkan lokasi fisik dari blok-blok logikanya. Oleh karena itu para produsen disk telah mengurangi masalah ini dengan mengimplementasikan algoritma penjadwalan disk di dalam pengendali perangkat keras, sehingga kalau hanya kinerja I/O yang diperhatikan, maka sistem operasi bisa menyerahkan algoritma penjadwalan disk pada perangkat keras itu sendiri.

2. MANAJEMEN DISK

Struktur Disk

Struktur disk merupakan suatu hal yang penting bagi penyimpanan informasi. Sistem komputer modern menggunakan Disk sebagai media penyimpanan sekunder. Dulu pita magnetik digunakan sebelum penggunaan disk sebagai media penyimpanan,sekunder yang memiliki waktu akses yang lebih lambat dari disk. Sejak digunakan disk, tape digunakan untuk backup, untuk menyimpan informasi yang tidak sering digunakan, sebagai media untuk memindahkan informasi dari satu sistem ke sistem lain, dan untuk menyimpan data yang cukup besar bagi sistem disk.

Bentuk penulisan Disk drive modern adalah array blok logika satu dimensi yang besar. Blok logika merupakan satuan unit terkecil dari transfer. Ukuran blok logika umumnya sebesar 512 bytes walaupun disk dapat diformat di level rendah (low level formatted) sehingga ukuran blok logika bisa ditentukan, misalnya 1024 bytes.

Array adalah blok logika satu dimensi yang dipetakan ke sektor dari disk secara sekuensial. Sektor 0 merupakan sektor pertama dari track pertama yang terletak di silinder paling luar (outermost cylinder). Proses pemetaan dilakukan secara berurut dari Sektor 0, lalu ke seluruh track dari silinder tersebut, lalu ke seluruh silinder mulai dari silinder terluar sampai silinder terdalam.

Kita dapat mengubah sebuah nomor blok logika dengan pemetaan menjadi sebuah alamat disk yang terdiri atas nomor silinder, nomor track di silinder tersebut, dan nomor sektor dari track tersebut. Dalam prakteknya, sulit untuk menerapkan pengubahan tersebut karena ada dua alasan. Pertama, kebanyakan disk memiliki sejumlah sektor yang rusak, tetapi pemetaan menyembunyikan hal ini dengan mensubstitusikan dengan sektor lain yang diambil dari suatu tempat di disk. Kedua, jumlah dari sektor tidak track tidak konstan. Pada media yang menggunakan ketentuan CLV (Constant Linear Velocity) kepadatan bit tiap track sama, jadi semakin jauh sebuah track dari tengah disk, semakin besar panjangnya, dan juga semakin banyak sektor yang dimilikinya. Trek di zona terluar memiliki 40% sektor lebih banyak dibandingkan dengan track di zona terdalam. Untuk menjamin aliran data yang sama, sebuah drive menaikan kecepatan putarannya ketika disk head bergerak dari zona luar ke zona dalam. Metode ini digunakan dalam CD-ROM dan DVD-ROM. Metode lain yang digunakan agar rotasi tetap konstan dan aliran data juga konstan dikenal dengan metode CAV (Constant Angular Velocity). CAV memungkinkan aliran data yang konstan karena kepadatan bit dari zona terdalam ke zona terluar semakin berkurang, sehingga dengan kecepatan rotasi yang konstan diperoleh aliran data yang konstan.

3. Penanganan Swap-Space

Penanganan (management) swap-space (tempat pertukaran; tetapi karena istilah swap-space sudah umum dipakai, maka untuk seterusnya kita tetap memakai istilah swap-space) adalah salah satu dari low-level task pada sebuah sistem operasi. Memori Virtual menggunakan disk space sebagai perpanjangan (atau space tambahan) dari memori utama. Karena kecepatan akses disk lebih lambat daripada kecepatan akses memori, menggunakan swap-space akan mengurangi performa sistem secara signifikan. Tujuan utama dari perancangan dan implementasi swap-space adalah untuk menghasilkan kinerja memori virtual yang optimal. Dalam sub-bab ini, kita akan membicarakan bagaimana swap-space digunakan, dimana letak swap-space pada disk, dan bagaimana penanganan swap-space.

Manajemen Swap-Space

- Management swap-space merupakan salah satu dari lowlevel task pada sebuah sistem operasi.
- Swap-space merupakan Memori Virtual dengan cara menggunakan disk space sebagai space tambahan darimemori utama
    - kecepatan akses disk lebih lambat daripada kecepatan akses memori à swap-space akan mengurangi performa sistem secara signifikan.
- Tujuan utama dari perancangan dan implementasi swapspace adalah untuk menghasilkan kinerja memori virtual yang optimal.
- Dalam sub-bab ini, kita akan membicarakan bagaimana swap-space digunakan, dimana letak swap-space pada disk, dan bagaimana penanganan swap-space.

Penggunaan Swap-Space

- Penggunaan swap-space pada berbagai macam sistem operasi berbeda-beda, tergantung pada algoritma memory management yang diimplementasikan.
- Besar swap-space yang dibutuhkan sistem bermacammacam, tergantung dari banyaknya physical memory (RAM, seperti EDO DRAM, SDRAM, RD RAM), memori virtual yang disimpan di swap-space, dan caranya memori virtual digunakan. Besarnya bervariasi, antara beberapa megabytes sampai ratusan megabytes atau lebih.
- Beberapa sistem operasi, seperti UNIX, menggunakanswap-space sebanyak yang diperlukan.
- Swap-space ini biasanya disimpan dalam beberapa disk yang terpisah, jadi beban yang diterima oleh sistem I/O dari paging dan swapping bisa didistribusikan ke berbagai I/O device pada sistem.
- Menyediakan swap-space yang berlebih lebih aman daripada kekurangan swap-space, karena bila kekurangan maka ada kemungkinan sistem terpaksa menghentikan sebuah atau lebih proses atau bahkan membuat sistem menjadi crash.
- Swap-space yang berlebih memang membuang disk space yang sebenarnya bisa digunakan untuk menyimpan file, tapi tidak menimbulkan resiko yang lain.

Lokasi Swap-Space

- Ada dua tempat dimana swap-space bisa berada:

- ♥ swap-space diletakkan pada partisi yang sama dengan sistem operasi.
- ♥ swap-space diletakkan pada partisi yang berbeda dengan sistem operasi

Ø Apabila swap-space yang dipakai hanya berupa sebuah berkas yang besar di dalam sistem berkas, maka sistem berkas yang dipakai bisa digunakan untuk membuat, menamakan, dan mengalokasikan tempat swap-space.

- ♥ mudah diimplementasikan.
- ♥ tidak efisien.

Pengelolaan Swap-Space

Ø Dalam 4.3BSD, swap-space dialokasikan untuk proses ketika sebuah proses dimulai. Tempat yang cukup disediakan untuk menampung program, dikenal sebagai segmen teks dan segmen data proses tersebut.

Ø Kernel mempergunakan swap maps untuk melacak penggunaan swap-space.

Ø Pada Solaris 1 (SunOS 4), para pembuatnya membuat perubahan pada metode standar UNIX untuk meningkatkan efisiensi dan untuk mencermin kan perubahan teknologi.

- ♥ Ketika sebuah proses berjalan, halaman-halaman (pages) dari segmen teks dibawa kembali dari sistem berkas, diakses di memori utama, dan dibuang bila diputuskan untuk di-pageout.
- ♥ Akan lebih efisien untuk membaca ulang sebuah halaman (page) dari sistem berkas daripada menaruhnya di swap-space dan membacanya ulang dari sana.

Ø Pada Solaris 2 terjadi perubahan besar. Pengalokasian Swap-space hanya dilakukan ketika sebuah halaman (page) dipaksa keluar dari memori (tidak dilakukan ketika halaman (page) dari memori virtual pertama kali dibuat).

Ø Perubahan ini memberikan performa lebih baik pada komputer modern, yang sudah mempunyai memori lebih banyak daripada komputer-komputer dengan sistem yang sudah lama, dan lebih jarang melakukan paging.

5. Kehandalan Disk

- Ø Disk memiliki resiko untuk mengalami kerusakan yang dapat berakibat turunnya performa atau pun hilangnya data, sehingga reliabilitas dari suatu disk harus dapat terus ditingkatkan.
- Ø Berikut ini adalah beberapa macam penyebab terjadinya hilangnya data:

1. Ketidaksengajaan dalam menghapus.
    - ♥ Bisa saja pengguna secara tidak sengaja menghapus suatu berkas, hal ini dapat dicegah seminimal mungkin dengan cara melakukan backup data secara reguler.
2. Hilangnya tenaga listrik
    - ♥ Hilangnya tenaga listrik dapat mengakibatkan adanya corrupt data.
3. Blok rusak pada disk.
    - ♥ Rusaknya blok pada disk dapat saja disebabkan dari umur disk tersebut. Seiring dengan waktu, banyaknya blok pada disk yang rusak dapat terus terakumulasi. Blok yang rusak pada disk, tidak akan dapat dibaca.
4. Rusaknya Disk.
    - ♥ Bisa saja karena suatu kejadian disk rusak total. Sebagai contoh, dapat saja disk jatuh atau pun ditendang ketika sedang dibawa.
5. System Corrupt.
    - ♥ Ketika komputer sedang dijalankan, bisa saja terjadi OS error, program error, dan lain sebagainya.

- Ø Untuk meningkatkan kinerja disk, dilibatkan banyak disk sebagai satu unit penyimpanan.
- Ø Tiap-tiap blok data dipecah ke dalam beberapa sub-blok, dan dibagi-bagi ke dalam disk-disk tersebut.
- Ø Ketika mengirim data disk-disk tersebut bekerja secara pararel dan dilakukan sinkronisasi pada rotasi masingmasing disk, maka kinerja dari disk dapat ditingkatkan. Cara ini dikenal sebagai RAID (Redundant Array of Independent Disks).
- Ø Selain masalah kinerja, RAID dapat meningkatkan reabilitas disk dengan jalan melakukan redundansi data. Sistem Operasi Lanjut 5
- Ø Salah satu cara yang digunakan pada RAID adalah dengan mirroring atau shadowing, yaitu dengan membuat duplikasi dari tiap-tiap disk.

- Pada cara ini, berarti diperlukan media penyimpanan yang dua kali lebih besar daripada ukuran data sebenarnya.
- Akan tetapi, dengan cara ini pengaksesan disk yang dilakukan untuk membaca dapat ditingkatkan dua kali lipat.
- Cara lain yang digunakan pada RAID adalah block interleaved parity. Pada cara ini, digunakan sebagian kecil dari disk untuk penyimpanan parity block.
    - Sebagai contoh, dimisalkan terdapat 10 disk pada array. Karenanya setiap 9 data block yang disimpan pada array, 1 parity block juga akan disimpan. Bila terjadi kerusakan pada salah satu block pada disk maka dengan adanya informasi pada parity block ini, ditambah dengan data block lainnya, diharapkan kerusakan pada disk tersebut dapat ditanggulangi, sehingga tidak ada data yang hilang.
- Penggunaan parity block ini juga akan menurunkan kinerja sama seperti halnya pada mirroring. Pada parity block ini, tiap kali subblock data ditulis, akan terjadi perhitungan dan penulisan ulang pada parity block.

6. Implementasi Stable-Storage

- Berdasarkan definisi, informasi yang berada di dalamstable storage tidak akan pernah hilang.
- Untuk mengimplementasikan storage seperti itu, kita perlu mereplikasi informasi yang dibutuhkan ke banyak peralatan storage (biasanya disk-disk) dengan failure modes yang independen.
- Kita perlu mengkoordinasikan penulisan update-update dalam sebuah cara yang menjamin bila terjadi kegagalan selagi meng-update tidak akan membuat semua kopi yang ada menjadi rusak, dan bila sedang recover dari sebuah kegagalan, kita bisa memaksa semua kopi yang ada ke dalam keadaan yang bernilai benar dan konsisten, bahkan bila ada kegagalan lain yang terjadi ketika sedang recovery.
- Sebuah disk write menyebabkan satu dari tiga kemungkinan:

1. Successful completion.
    - Data disimpan dengan benar di dalam disk.
2. Partial failure.

- ♥ Kegagalan terjadi di tengah-tengah transfer, menyebabkan hanya bebe rapa sektor yang diisi dengan data yang baru, dan sektor yang diisi ketika terjadi kegagalan menjadi rusak.

3. Total failure.

- ♥ Kegagalan terjadi sebelum disk write dimulai, jadi data yang sebelumnya ada pada disk masih tetap ada.

Ø Sebuah operasi output dieksekusi seperti berikut:

1. Tulis informasinya ke blok physical yang pertama.
2. Ketika penulisan pertama berhasil, tulis informasi yang sama ke blok physical yang kedua.
3. Operasi dikatakan berhasil hanya jika penulisan kedua berhasil.

7. Tertiary-Storage Structure

Ø Ciri-ciri Tertiary-Storage Structure:

- ♥ Biaya produksi lebih murah.
- ♥ Menggunakan removable media.
- ♥ Data yang disimpan bersifat permanen.
- ♥

Macam-macam Tertiary-Strorage Structure

1. Floppy Disk

Floopy disk adalah fleksible disk yang tipis, dilapisi material yang bersifat magnet, dan ditutupi oleh plastik.

Ciri-ciri:

• Umumnya mempunyai kapasitas antara 1-2 MB.
• Kemampuan akses hampir seperti hardisk.

2. Magneto-optic disk

Magneto-optic Disk adalah Piringan optic yang keras dilapisi oleh material yang bersifat magnet, kemudian dilapisi pelindung dari plastik atau kaca yang berfungsi untuk menahan head yang hancur.

3. Optical Disk

Disk ini tidak menggunakan sifat magnet, tetapi menggunakan bahan khusus yang dimodifikasi menggunakan sinar laser. Setelah dimodifikasi dengan dengan sinar laser pada disk akan terdapat spot yang gelap atau terang. Spot ini menyimpan satu bit.
Optical-disk teknologi terbagi atas:

1. Phase-change disk, dilapisi oleh material yang dapat membeku menjadi crystalline atau amorphous state.
2. Dye-polimer disk, merekam data dengan membuat bump.
4. WORM Disk (Write Once, Read Many Times)
WORM adalah Aluminium film yang tipis dilapisi oleh piringan plastik atau kaca pada bagian atas dan bawahnya. Untuk menulis bit, drive tersebut menggunakan sinar laser untuk membakar hole yang kecil pada aluminium. Hole ini tidak dapat diubah seperti sebelumnya. Oleh karena itu, disk hanya dapat ditulis sekali.
Ciri-ciri:
• Data hanya dapat ditulis sekali.
• Data lebih tahan lama dan dapat dipercaya.
• Read Only disk, seperti CD-ROM dan DVD yang berasal dari
pabrik sudah berisi data.
5. Tapes
• Dibandingkan dengan disk, tape lebih murah dan lebih kapasitasnya lebih besar, tetapi random access tape lebih lambat daripada disk karena tape menggunakan operasi forward dan rewind.
• Tape adalah media yang ekonomis apabila media yang ingin digunakan tidak membutuhkan kemampuan random access, contoh: backup data dari data disk, menampung data yang besar.
• Pemasangan tape yang besar menggunakan robotic tape changers. Robotic tape changers memindahkan beberapa tape antara beberapa tape drive dan beberapa slot penyimpanan yang berada di dalam tape library.
• Library yang menyimpan beberapa tape disebut tape stacker.
• Library yang menyimpan ribuan tape disebut tape silo.
• Robotic tape library mengurangi biaya penyimpanan data. File yang ada di disk dapat dipindahkan ke tape dengan tujuan mengurangi biaya penyimpanan. Jika file ingin digunakan, maka komputer akan memindahkan file ke disk.

8. kesimpulan

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. Adapun yang bisa menyebabkan data hilang, seperti

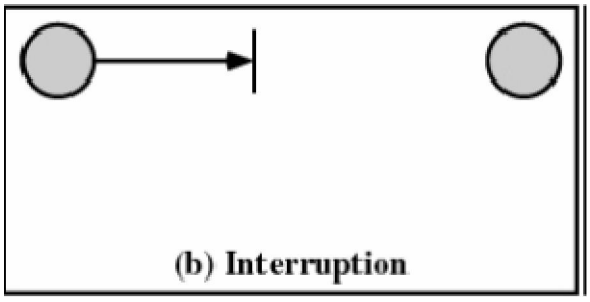
- Bencana alam dan perang
- Kesalahan Hardware atau software, Contohnya CPU malfunction, bad disk, program bugs
- Kesalahan manusia, Contohnya data entry, wrong tape mounted.

Selain itu terdapat ancaman lain pada sistem keamanan komputer yang bisa dikategorikan dalam empat macam Penyusup / Intruder, sehingga sumber daya sistem komputer dihancurkan menjadi tak berguna. Seperti

- Iseng-iseng, biasanya pada yang bisa diakses semua user
- Snooping, seseorang masuk kedalam sistem jaringan dan berusaha menebus pengamanan.
- Berusaha mencari keuntungan dengan motivasi uang
- Spionase / militer

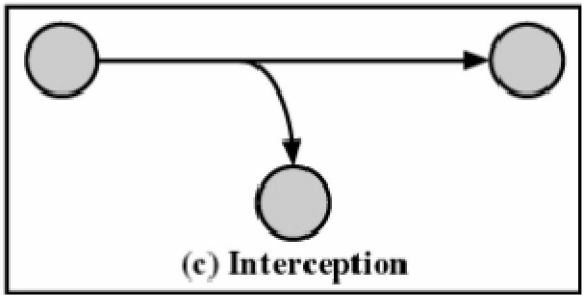
Penyusup / Intruder dikelompokkan menjadi empat bagian:

a. Interupsi, Orang yang tak diotorisasi dapat masuk / mengakses ke sumber daya sistem.
    a. Sumber daya sistem komputer dihancurkan atau menjadi tak tersedia
    b. Penghancuran harddisk
    c. Pemotongan kabel komunikasi
    d. Sistem file management menjadi tidak tersedia



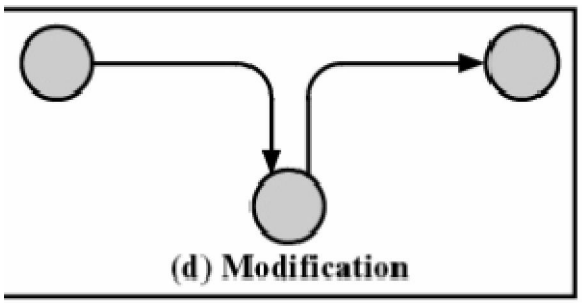
Gambar 14.1 Interupsi

b. Intersepsi
    a. Pihak tak diotorisasi dapat mengakses sumber daya
    b. Ancaman terhadap kerahasiaan data
    c. Penyadapan terhadap data dijaringan
    d. Mengkopi file tanpa diotorisasi



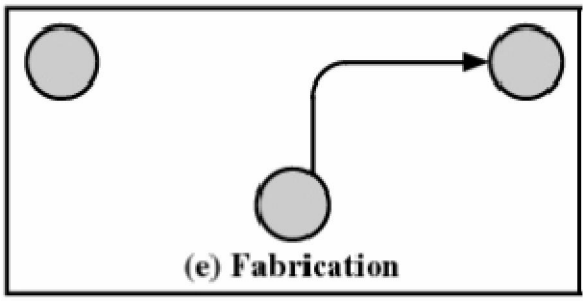
Gambar 14.2 Intersepsi

c. Modification, Orang yang tak diotorisasi tidak hanya dapat mengakses tapi juga mengubah,merusak sumber daya. Ciri – cirinya :
    a. Mengubah nilai-nilai file data
    b. Mengubah program sehingga bertindak secara beda
    c. Memodifikasi pesan-pesan yang ditransmisikan pada jaringan



Gambar 14.3 Modification

d. Fabrication, Orang yang tak diotorisasi menyisipkan objek palsu ke dalam sistem.
   Ciri-cirinya :
   a. Pihak tak diotorisasi menyisipkan objek palsu ke sistem
   b. Memasukkan pesan-pesan palsu ke jaringan
   c. Penambahan record ke file



Gambar 14.4 Fabrication

Keamanan sistem

Adapun aspek keamanan sistem pada sistem operasi

- Kerahasiaan (Secrecy)
- Integritas (Integrity)
- Ketersediaan (Availability)

Prinsip Pengamanan Sistem Komputer

a. Rancangan sistem seharusnya publik
b. Dapat diterima
c. Pemeriksaan otoritas saat itu
d. Kewenangan serendah mungkin
e. Mekanisme yang ekonomis

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

Autentikasi pemakai, Suatu yang diketahui pemakai, misalnya :

a. passsword
b. kombinasi kunci
c. nama kecil ibu, dsb

Sesuatu yang dimiliki pemakai, misalnya :

a. badge
b. kartu identitas
c. kunci, dsb

Sesuatu mengenai(merupakan ciri) pemakai, misalnya ::

a. sidik jari
b. sidik suara
c. foto
d. tanda tangan, dsb

Contoh Autentikasi

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 :

- 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.
- 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

LOGIN : ken	LOGIN : carol
PASSWORD : FooBar	INVALID LOGIN NAME
SUCCESSFUL LOGIN	LOGIN:
(a)	(b)

LOGIN : carol
PASSWORD : Idunno
INVALID LOGIN
LOGIN :
(c)

(a)Login berhasil
(b)Login ditolak setelah nama dimasukkan

(c)Login ditolak setelah nama dan password

Bobbie, 4238, e(Dog4238)
Tony, 2918, e(6%%TaeFF2918)
Laura, 6902, e(Shakespeare6902)
Mark, 1694, e(XaB@Bwcz1694)
Deborah, 1092, e(LordByron,1092)

Contoh Autentikasi Menggunakan Objek Fisik

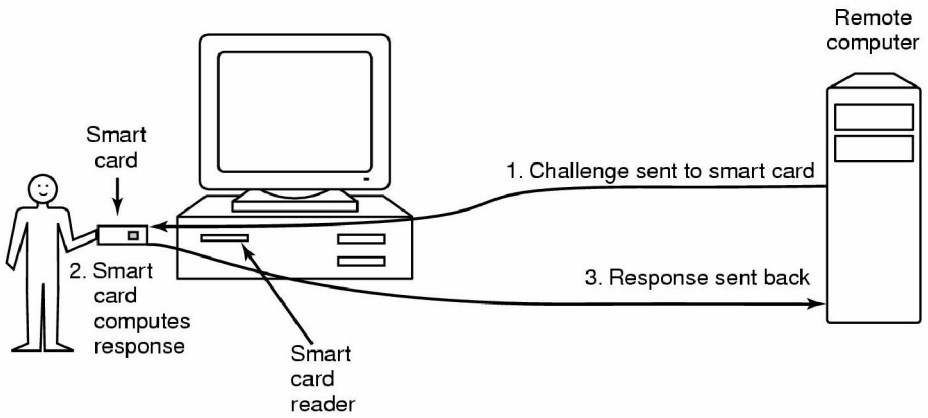
Identifikasi Fisik

Pendekatan lain adalah memberikan yang dimiliki pemakai, seperti :

a. Magnetic cards (Kartu berpita magnetic)

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 system komputer bila memenuhi dua syarat berikut : Mempunyai kartu dan mengetahui password yang spesifik kartu itu.

Contohnya ATM, merupakan mesin yang bekerja dengan cara ini.



Gambar 14.5 Objek Fisik

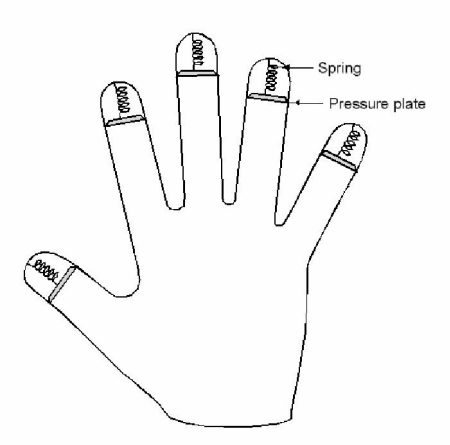
Magnetic cards

a. magnetic stripe cards
b. chip cards: stored value cards, smart cards

b. Autentikasi Menggunakan Biometric (sidik jari)

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

- Sidik jari dan sidik suara.
- Analisis panjang jari.
- Pengenalan visual dengan menggunakan kamera diterapkan.



Gambar 14.6 Biometric

Countermeasures (Tindakan Balasan)

a. Pembatasan waktu ketika seseorang login
b. Panggilan otomatis pada nomor yang disiapkan
c. Pembatasan upaya melakukan login
d. Ketersediaan database login
e. Penggunaan simple login sebagai perangkap

Sekuriti Sistem Operasi

Logic Bomb adalah Logik yang ditempelkan pada program komputer, dimana pada saat program menjalankan kondisi tertentu logik tersebut menjalankan fungsi yang merusak

Trap Door

Kode yang menerima suatu barisan masukan khusus atau dipicu dengan menjalankan ID pemakai tertentu

```
while (TRUE) {
    printf("login: ");
    get_string(name);
    disable_echoing();
    printf("password: ");
    get_string(password);
    enable_echoing();
    v = check_validity(name, password);
    if (v) break;
}
execute_shell(name);
```

(a)

```
while (TRUE) {
    printf("login: ");
    get_string(name);
    disable_echoing();
    printf("password: ");
    get_string(password);
    enable_echoing();
    v = check_validity(name, password);
    if (v || strcmp(name, "zzzzz") == 0) break;
}
execute_shell(name);
```

(b)

Serangan Pengamanan Umum

- Permintaan page memori
-Mencoba system calls
- Mencoba login dan langsung menekan DEL, RUBOUT atau BREAK
- Mencoba memodifikasi struktur sistem operasi
- Mencari informasi yang tidak boleh dilakukan pada manual book
- Menggunakan kelemahan sifat manusia.

Prinsip Dasar Sekuriti

-Sistem sebaiknya bersifat publik
-Nilai default tidak boleh diakses
-Pengecekan otoritas
-Memberikan setiap proses kemamapuan akses sesedikit mungkin
-Mekanisme proteksi sederhana, uniform dan buil in kelapis terbawah
-Skema pengamanan harus dapat diterima secara psikologis

Sekuriti Jaringan Komputer

-Ancaman Eksternal
-Kode ditransfer kemesin target
-Saat kode dieksekusi, kerusakan pun terjadi
-Tujuan virus ditulis dijaringan komputer
-Penyebarannya yang cepat
-Sulit terdeteksi
-Virus = program yang dapat memperbanyak diri sendiri

Skenario Pengrusakan oleh Virus

-Blackmail

-Denial of Service selama virus masih jalan

-Kerusakan permanen pada hardware

-Kompetitor komputer

-sabotase

Siklus Hidup Virus

-Fase Tidur (Dormant Phase)

Virus dalam keadaan menganngur sampai terjadi suatu kejadian tertentu

-Fase Propagasi

Virus menempatkan kopi dirinya keprogram lain didisk.

-Fase Pemicuan (Triggering Phase)

Virus diaktifkan untuk melakukan fungsi tertentu

-Fase Eksekusi

Virus menjalankan fungsinya.

Tipe-tipe Virus

-Parasitic Virus

Menggantung kefile .exe dan melakukan replikasi ketika file tersebut dieksekusi

- Memory Resident Virus

Menempatkan diri kemeori utama dan menginfeksi setiap program yang dieksekusi

-Boot Sector Virus

Menginfeksiboot record danmenyebarsaatsistemdiboot

-Stealth Virus

Bentuknyadirancangagar tidakterdeteksiolehantivirus

-Polymorphic Virus

Bermutasi setiap kali melakukan infeksi

Anti virus

Pendekatan Antivirus

-Deteksi

-Identifikasi

-Penghilangan dengan program antivirus (biasanya dibuat dengan bahasa assembler)

Generasi Antivirus

-G1 : Sekedar scanner biasa

-G2 : heuristic scanner

-G3 : activity trap

-G4 : full featue protection

C. SOAL LATIHAN/TUGAS

1. Buatlah contoh Login berhasil?
2. Buatlah contoh Login ditolak setelah nama dimasukkan?
3. Buatlah contoh Login ditolak setelah nama dan password ?

D. DAFTAR PUSTAKA

Buku

Bambang Hariyanto. 1997. Sistem Operasi, Bandung:Informatika Bandung.

Dali S. Naga. 1992. Teori dan Soal Sistem Operasi Komputer,Jakarta: Gunadarma.

Silberschatz Galvin. 1995. 4 Edition Operating System Concepts: Addison Wesley.

Sri Kusumadewi. 2000. Sistem Operasi. Yogyakarta: J&J Learning.

Tanenbaum, A.1992. Modern Operating Systems.New York: Prentice Hall

.

6.

Link and Sites:

http://www.ilmukomputer.com

http://vlsm.bebas.org

http://www.wikipedia.com