Sistem Operasi

Oleh :

(Digunakan di lingkungan sendiri, sebagai buku ajar

mata kuliah Sistem Operasi)

Fakultas Teknik dan Ilmu Komputer

Program Studi Sistem Informasi

Universitas Komputer Indonesia

1.1. Silabus

Minggu Ke 1 : Pengantar Perkuliahan

Sistem perkuliahan, tujuan dan cakupan materi perkuliahan, silabus, perberitahuan daftar pustaka.

Minggu Ke 2 : Pengenalan Umum Sistem Operasi dan Struktur Sistem Komputer

Minggu Ke 3 : Struktur Sistem Operasi

Minggu Ke 4 : Manajemen Proses

Minggu Ke 5 : Penjadwalan Proses

Minggu Ke 6 : Sinkronisasi dan Deadlock

Minggu Ke 7 : Manajemen Memory

Minggu Ke 8 : UTS

Minggu Ke 9 : Virtual Memori

Minggu Ke 10 : Manajemen Sistem File

Minggu Ke 11 : Manajemen Sistem Input/Output

Minggu Ke 12 dan 13: Sistem Proteksi dan Sekuriti Komputer

Minggu Ke 14 dan 15 : Review semua Materi dan Penjelasan Umum materi Sistem Terdistribusi

Minggu ke 16 : UAS

1.2. Materi Perkuliahan

1. Pertemuan Pertama

a. Perkuliahan diselenggarakan 14 kali pertemuan (2 SKS)
b. Wajib kehadiran Mahasiswa 80% (-3 kali tidak masuk)
c. Materi perkuliahan akan diberikan salinannya kepada Mahasiswa
d. Mahasiswa dianjurkan membawa flashdisk
e. Batas keterlambatan 15 menit setelah perkuliahan dimulai
f. Mahasiswa diperbolehkan berkonsultasi dengan dosen; mengenai materi perkuliahan secara personal atau kelompok di luar jam perkuliahan (tatap muka; via email; kuliah online)
g. Mengikuti tata tertib Lab
h. Tidak diperbolehkan menggunakan perangkat komunikasi selama perkuliahan (setting silent/vibrate)
i. Bersikap sopan dan tidak mengganggu keberlangsungan perkuliahan
j. Tersedia waktu Shalat bagi yang beragama Islam.

2. Pertemuan kedua

1. Sistem Operasi

Sistem operasi merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Sebelum ada sistem operasi, orang hanya mengunakan komputer dengan menggunakan sinyal analog dan sinyal digital. Seiring dengan berkembangnya pengetahuan dan teknologi, pada saat ini terdapat berbagai sistem operasi dengan keunggulan masing-masing. Untuk lebih memahami sistem operasi maka sebaiknya perlu diketahui terlebih dahulu beberapa konsep dasar mengenai sistem operasi itu sendiri.

Pengertian sistem operasi secara umum ialah pengelola seluruh sumber-daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (system calls) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumber-daya sistem komputer.

- **Fungsi Dasar**

Sistem komputer pada dasarnya terdiri dari empat komponen utama, yaitu perangkat-keras, program aplikasi, sistem-operasi, dan para pengguna. Sistem operasi berfungsi untuk mengatur dan mengawasi penggunaan perangkat keras oleh berbagai program aplikasi serta para pengguna.

Sistem operasi berfungsi ibarat pemerintah dalam suatu negara, dalam arti membuat kondisi komputer agar dapat menjalankan program secara benar. Untuk menghindari konflik yang terjadi pada saat pengguna menggunakan sumber-daya yang sama, sistem operasi mengatur pengguna mana yang dapat mengakses suatu sumber-daya. Sistem operasi juga sering disebut resource allocator. Satu lagi fungsi penting sistem operasi ialah sebagai program pengendali yang bertujuan untuk menghindari kekeliruan (error) dan penggunaan komputer yang tidak perlu.

- **Tujuan Mempelajari Sistem Operasi**

Tujuan mempelajari sistem operasi agar dapat merancang sendiri serta dapat memodifikasi sistem yang telah ada sesuai dengan kebutuhan kita, agar dapat memilih alternatif sistem operasi, memaksimalkan penggunaan sistem operasi dan agar konsep dan teknik sistem operasi dapat diterapkan pada aplikasi-aplikasi lain.

- **Sasaran Sistem Operasi**

Sistem operasi mempunyai tiga sasaran utama yaitu kenyamanan --membuat penggunaan komputer menjadi lebih nyaman, efisien --penggunaan sumber-daya sistem komputer secara efisien, serta mampu berevolusi --sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru.

- **Sejarah Sistem Operasi**

Menurut Tanenbaum, sistem operasi mengalami perkembangan yang sangat pesat, yang dapat dibagi kedalam empat generasi:

• Generasi Pertama (1945-1955)
Generasi pertama merupakan awal perkembangan sistem komputasi elektronik sebagai pengganti sistem komputasi mekanik, hal itu disebabkan kecepatan manusia untuk menghitung terbatas dan manusia sangat mudah untuk membuat kecerobohan, kekeliruan bahkan kesalahan. Pada generasi ini belum ada sistem operasi, maka sistem komputer diberi instruksi yang harus dikerjakan secara langsung.

• Generasi Kedua (1955-1965)
Generasi kedua memperkenalkan Batch Processing System, yaitu Job yang dikerjakan dalam satu rangkaian, lalu dieksekusi secara berurutan.Pada generasi ini sistem komputer belum dilengkapi sistem operasi, tetapi beberapa fungsi sistem operasi telah ada, contohnya fungsi sistem operasi ialah FMS dan IBSYS.

• Generasi Ketiga (1965-1980)
Pada generasi ini perkembangan sistem operasi dikembangkan untuk melayani banyak pemakai sekaligus, dimana para pemakai interaktif berkomunikasi lewat terminal secara on-line ke komputer, maka sistem operasi menjadi multi-user (di gunakan banyak pengguna sekali gus) dan multi-programming (melayani banyak program sekali gus).

• Generasi Keempat (Pasca 1980an)
Dewasa ini, sistem operasi dipergunakan untuk jaringan komputer dimana pemakai menyadari keberadaan komputer-komputer yang saling terhubung satu sama lainnya. Pada masa ini para pengguna juga telah dinyamankan dengan Graphical User Interface yaitu antar-muka komputer yang berbasis grafis yang sangat nyaman, pada masa ini juga dimulai era komputasi tersebar dimana komputasi-komputasi tidak lagi berpusat di satu titik, tetapi dipecah dibanyak komputer sehingga tercapai kinerja yang lebih baik.

- **Layanan Sistem Operasi**

Sebuah sistem operasi yang baik menurut Tanenbaum harus memiliki layanan sebagai berikut: pembuatan program, eksekusi program, pengaksesan I/O Device, pengaksesan terkendali terhadap berkas pengaksesan sistem, deteksi dan pemberian tanggapan pada kesalahan, serta akunting.

Pembuatan program yaitu sistem operasi menyediakan fasilitas dan layanan untuk membantu para pemrogram untuk menulis program; Eksekusi Program yang berarti Instruksi-instruksi dan data-data harus dimuat ke memori utama, perangkat-parangkat masukan/ keluaran dan berkas harus di-inisialisasi, serta sumber-daya yang ada harus disiapkan, semua itu harus di tangani oleh sistem operasi; Pengaksesan I/O Device, artinya Sistem Operasi harus mengambil alih sejumlah instruksi yang rumit dan sinyal kendali menjengkelkan agar pemrogram dapat berfikir sederhana dan perangkat pun dapat beroperasi; Pengaksesan terkendali terhadap berkas yang artinya disediakannya mekanisme proteksi terhadap berkas untuk mengendalikan pengaksesan terhadap berkas; Pengaksesan sistem artinya pada pengaksesan digunakan bersama (shared system); Fungsi pengaksesan harus menyediakan proteksi terhadap sejumlah sumber-daya dan data dari pemakai tak terdistorsi serta menyelesaikan konflik-konflik dalam perebutan sumber-daya; Deteksi dan Pemberian tanggapan pada kesalahan, yaitu jika muncul
permasalahan muncul pada sistem komputer maka sistem operasi harus memberikan tanggapan yang menjelaskan kesalahan yang terjadi serta dampaknya terhadap aplikasi yang sedang berjalan; dan Akunting yang artinya Sistem Operasi yang bagus mengumpulkan data statistik penggunaan beragam sumber-daya dan memonitor parameter kinerja.

3. Pertemuan ketiga

STRUKTUR SISTEM OPERASI

Secara umum, Sistem Operasi adalah software pada lapisan pertama yang ditempatkan pada memori komputerpada saat komputer dinyalakan. Sedangkan software-software lainnya dijalankan setelah Sistem Operasi berjalan, dan Sistem Operasi akan melakukan layanan inti umum untuk software-software itu. Layanan inti umum tersebut seperti akses ke disk, manajemen memori, skeduling task, dan antar-muka user. Sehingga masing-masing software tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan dilakukan oleh Sistem Operasi. Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan "kernel" suatu Sistem Operasi. Kalau sistem komputer terbagi dalam lapisan-lapisan, maka Sistem Operasi adalah penghubung antara lapisan hardware dan lapisan software. Lebih jauh daripada itu, Sistem Operasi melakukan semua tugas-tugas penting dalam komputer, dan menjamin aplikasi-aplikasi yang berbeda dapat berjalan secara bersamaan dengan lancar. Sistem Operasi menjamin aplikasi software lainnya dapat menggunakan memori, melakukan input dan output terhadap peralatan lain dan memiliki akses kepada sistem file. Apabila beberapa aplikasi berjalan secara bersamaan, maka Sistem Operasi mengatur skedule yang tepat, sehingga sedapat mungkin semua proses yang berjalan mendapatkan waktu yang cukup untuk menggunakan prosesor (CPU) serta tidak saling mengganggu.

1. STRUKTUR SISTEM OPERASI

Sebuah sistem yang besar dan kompleks seperti sistem operasi modern harus diatur dengan cara membagi task kedalam komponen-komponen kecil agar dapat berfungsi dengan baik dan mudah.
Brikut ini adalah Struktur Sistem Operasi;

- *Struktur Sederhana*
- Sistem Berlapis (layered system)
- Kernel Mikro
- *Modular (Modules)*
- Mesin Maya *(Virtual Machine)*
- Client-Server Model
- Sistem Berorientasi Objek

1. Struktur Sederhana

Sistem operasi sebagai kumpulan prosedur dimana prosedur dapat saling dipanggil oleh prosedur lain di sistem bila diperlukan. Banyak sistem operasi komersial yang tidak terstruktur dengan baik. Kemudian sistem operasi dimulai dari yang terkecil, sederhana dan terbatas lalu berkembang dengan ruang lingkup originalnya. Contoh dari sistem operasi ini adalah MS-DOS dan UNIX. MS-DOS merupakan sistem operasi yang menyediakan fungsional dalam ruang yang sedikit sehingga tidak dibagi menjadi beberapa modul, sedangkan UNIX menggunakan struktur *monolitik* dimana prosedur dapat saling dipanggil oleh prosedur lain di sistem bila diperlukan dan kernel berisi semua layanan yang disediakan sistem operasi untuk pengguna. Inisialisasi-nya terbatas pada fungsional perangkat keras yang terbagi menjadi dua bagian yaitu kernel dan sistem program. Kernel terbagi menjadi serangkaian interface dan device driver dan menyediakan sistem file, penjadwalan CPU, manajemen memori, dan fungsi-fungsi sistem operasi lainnya melalui system calls.

Kelebihan Struktur Sederhana:

- Layanan dapat dilakukan sangat cepat karena terdapat di satu ruang alamat.

Kekurangan Struktur Sederhana:

- Pengujian dan penghilangan kesalahan sulit karena tidak dapat dipisahkan dan dilokalisasi.
- Sulit dalam menyediakan fasilitas pengamanan.
- Merupakan pemborosan bila setiap komputer harus menjalankan kernel monolitik sangat besar sementara sebenarnya tidak memerlukan seluruh layanan yang disediakan kernel.
- Tidak fleksibel.
- Kesalahan pemograman satu bagian dari kernel menyebabkan matinya seluruh sistem.

Evolusi :

Kebanyakan UNIX sampai saat ini berstruktur monolitik. Meskipun monolitik, yaitu seluruh komponen/subsistem sistem operasi terdapat di satu ruang alamat tetapi secara rancangan adalah berlapis. Rancangan adalah berlapis yaitu secara logik satu komponen/subsistem merupakan lapisan lebih bawah dibanding lainnya dan menyediakan layanan-layanan untuk lapisan-lapisan lebih atas. Komponen-komponen tersebut kemudia dikompilasi dan dikaitkan (di-link) menjadi satu ruang alamat. Untuk mempermudah dalam pengembangan terutama pengujian dan fleksibilitas, kebanyakan UNIX saat ini menggunakan konsep kernel loadable modules,yaitu:

- Bagian-bagian kernel terpenting berada di memori utama secara tetap.
- Bagian-bagian esensi lain berupa modul yang dapat ditambahkan ke kernel saat diperlukan dan dicabut begitu tidak digunakan lagi di waktu jalan (run time).

Contoh : UNIX berstruktur monolitik, MS-DOS

2. Sistem Berlapis (layered system)

Sistem operasi dibentuk secara hirarki berdasar lapisan-lapisan, dimana lapisan-lapisan bawa memberi layanan lapisan lebih atas. Lapisan yang paling bawah adalah perangkat keras, dan yang paling tinggi adalah user-interface. Sebuah lapisan adalah implementasi dari obyek abstrak yang merupakan enkapsulasi dari data dan operasi yang bisa memanipulasi data tersebut. Struktur berlapis dimaksudkan untuk mengurangi kompleksitas rancangan dan implementasi sistem operasi. Tiap lapisan mempunyai fungsional dan antarmuka masukan-keluaran antara dua lapisan bersebelahan yang terdefinisi bagus.

Sedangkan menurut Tanenbaum dan Woodhull, sistem terlapis terdiri dari enam lapisan, yaitu:

Lapis 5 – The operator

Berfungsi untuk pemakai operator.

Lapis 4 – User programs

Berfungsi untuk aplikasi program pemakai.

Lapis 3 – I/O management

Berfungsi untuk menyederhanakan akses I/O pada level atas.

Lapis 2 -Operator-operator communication

Berfungsi untuk mengatur komunikasi antar proses.

Lapis 1 -Memory and drum management

Berfungsi untuk mengatur alokasi ruang memori atau drum magnetic.

Lapis 0 -Processor allocation and multiprogramming

Berfungsi untuk mengatur alokasi pemroses dan switching, multi programming dan pengaturan prosessor.

Menurut Stallings, model tingkatan sistem operasi yang mengaplikasikan prinsip ini dapat dilihat pada tabel berikut, yang terdiri dari level-level dibawah ini:

- Level 1

Terdiri dari sirkuit elektronik dimana obyek yang ditangani adalah register memory cell, dan gerbang logika. Operasi pada obyek ini seperti membersihkan register atau membaca lokasi memori.

- Level 2

Pada level ini adalah set instruksi pada prosesor. Operasinya adalah instruksi bahasa-mesin, seperti menambah, mengurangi, load dan store.

- Level 3

Tambahan konsep prosedur atau subrutin ditambah operasi call atau return.

- Level 4

Mengenalkan interupsi yang menyebabkan prosesor harus menyimpan perintah yang baru dijalankan dan memanggil rutin penanganan interupsi. Empat level pertama bukan bagian sistem operasi tetapi bagian perangkat keras. Meski pun demikian beberapa elemen sistem operasi mulai tampil pada level-level ini, seperti rutin penanganan interupsi. Pada level 5, kita mulai masuk kebagian sistem operasi dan konsepnya berhubungan dengan multi-programming.

- Level 5

Level ini mengenalkan ide proses dalam mengeksekusi program. Kebutuhan-kebutuhan dasar pada sistem operasi untuk mendukung proses ganda termasuk kemampuan men-suspend dan me-resume proses. Hal ini membutuhkan register perangkat keras untuk menyimpan agar eksekusi bisa ditukar antara satu proses ke proses lainnya.

- Level 6

Mengatasi penyimpanan sekunder dari komputer. Level ini untuk menjadualkan operasi dan menanggapi permintaan proses dalam melengkapi suatu proses.

- Level 7

Membuat alamat logik untuk proses. Level ini mengatur alamat virtual ke dalam blok yang bisa dipindahkan antara memori utama dan memori tambahan. Cara-cara yang sering dipakai adalah menggunakan ukuran halaman yang tetap, menggunakan segmen sepanjang variabelnya, dan menggunakan cara keduanya. Ketika blok yang dibutuhkan tidak ada dimemori utama, alamat logis pada level ini meminta transfer dari level 6. Sampai point ini, sistem operasi mengatasi sumber daya dari prosesor tunggal. Mulai level 8, sistem operasi mengatasi obyek eksternal seperti peranti bagian luar, jaringan, dan sisipan komputer kepada jaringan.

Ø Level 8

Mengatasi komunikasi informasi dan pesan-pesan antar proses. Dimana pada level 5 disediakan mekanisme penanda yang kuno yang memungkinkan untuk sinkronisasi proses, pada level ini mengatasi pembagian informasi yang lebih banyak. Salah satu peranti yang paling sesuai adalah pipe (pipa) yang menerima output suatu proses dan memberi input ke proses lain.

- Level 9

Mendukung penyimpanan jangka panjang yang disebut dengan berkas. Pada level ini, data dari penyimpanan sekunder ditampilkan pada tingkat abstrak, panjang variabel yang terpisah. Hal nini bertentangan tampilan yang berorientasikan perangkat keras dari penyimpanan sekunder.

- Level 10

Menyediakan akses ke peranti eksternal menggunakan antarmuka standar.

- Level 11

Bertanggung-jawab mempertahankan hubungan antara internal dan eksternal identifier dari sumber daya dan obyek sistem. Eksternal identifier adalah nama yang bisa dimanfaatkan oleh aplikasi atau pengguna. Internal identifier adalah alamat atau indikasi lain yang bisa digunakan oleh level yang lebih rendah untuk meletakkan dan mengontrol obyek.

- Level 12

Menyediakan suatu fasilitator yang penuh tampilan untuk mendukung proses. Hal ini merupakan lanjutan dari yang telah disediakan pada level 5. Pada level 12, semua info yang dibutuhkan untuk managemen proses dengan berurutan disediakan, termasuk alamat virtual di proses, daftar obyek dan proses yang berinteraksi dengan proses tersebut serta batasan interaksi tersebut, parameter yang harus dipenuhi proses saat pembentukan, dan karakteristik lain yang mungkin digunakan sistem operasi untuk mengontrol proses.

- Level 13

Menyediakan antarmuka dari sistem operasi dengan pengguna yang dianggap sebagai shell atau dinding karena memisahkan pengguna dengan sistem operasi dan menampilkan sistem operasi dengan sederhana sebagai kumpulan servis atau pelayanan.

Dari ketiga sumber diatas dapat kita simpulkan bahwa lapisan sistem operasi secara umum terdiri atas 4 bagian, yaitu:

1. Perangkat keras

Lebih berhubungan kepada perancang sistem. Lapisan ini mencakup lapisan 0 dan 1 menurut Tanenbaum, dan level 1 sampai dengan level 4 menurut Stallings.

1. Sistem operasi

Lebih berhubungan kepada programer. Lapisan ini mencakup lapisan 2 menurut Tanenbaum, dan level 5 sampai dengan level 7 menurut Stallings.

1. Kelengkapan

Lebih berhubungan kepada programer. Lapisan ini mencakup lapisan 3 menurut Tanenbaum, dan level 8 sampai dengan level 11 menurut Stallings.

1. Program aplikasi

Lebih berhubungan kepada pengguna aplikasi komputer. Lapisan ini mencakup lapisan 4 dan lapisan 5 menurut Tanebaum, dan level 12 dan level 13 menurut Stallings.

Lapisan n memberi layanan untuk lapisan n+1. Proses-proses di lapisan n dapat meminta layanan lapisan n-1 untuk membangunan layanan bagi lapisan n+1. Lapisan n dapat meminta layanan lapisan n-1. Kebalikan tidak dapat, lapisan n tidak dapat meminta layanan n+1. Masing-masing berjalan di ruang alamat-nya sendiri. Kelanjutan sistem berlapis adalah sistem berstruktur cincin seperti sistem MULTICS. Sistem MULTICS terdiri 64 lapisan cincin dimana satu lapisan berkewenangan berbeda. Lapisan n-1 mempunyai kewenangan lebih dibanding lapisan n. Untuk meminta layanan lapisan n-1, lapisan n melakukan trap. Kemudian, lapisan n-1 mengambil kendali sepenuhnya untuk melayani lapisan n.

Kelebihan Sistem Berlapis (layered system):

- Memiliki rancangan modular, yaitu sistem dibagi menjadi beberapa modul & tiap modul dirancang secara independen.
- Pendekatan berlapis menyederhanakan rancangan, spesifikasi dan implementasi sistem operasi.

Kekurangan Sistem Berlapis (layered system):

- Fungsi-fungsi sistem operasi diberikan ke tiap lapisan secara hati-hati.

Contoh: Sistem operasi yang menggunakan pendekatan berlapis adalah THE yang dibuat oleh Djikstra dan mahasiswa-mahasiswanya, serta sistem operasi MULTICS.

3. Kernel Mikro

Metode struktur ini adalah menghilangkan komponen-komponen yang tidak diperlukan dari kernel dan mengimplementasikannya sebagai sistem dan program-program level user. Hal ini akan menghasilkan

kernel yang kecil. Fungsi utama dari jenis ini adalah menyediakan fasilitas komunikasi antara program client dan bermacam pelayanan yang berjalan pada ruang user.

Kelebihan Kernel Mikro:

- kemudahan dalam memperluas sistem operasi
- mudah untuk diubah ke bentuk arsitektur baru
- kode yang kecil dan lebih aman

Kekurangan Kernel Mikro:

- kinerja akan berkurang selagi bertambahnya fungsi-fungsi yang digunakan.

Contoh: sistem operasi yang menggunakan metode ini adalah TRU64 UNIX, MacOSX dan QNX.

4. Modular (Modules)

Kernel mempunyai kumpulan komponen-komponen inti dan secara dinamis terhubung pada penambahan layanan selama waktu boot atau waktu berjalan. Sehingga strateginya menggunakan pemanggilan modul secara dinamis (*Loadable Kernel Modules*). Umumnya sudah diimplementasikan oleh sistem operasi modern seperti Solaris, Linux dan MacOSX.

Sistem Operasi Apple Macintosh Mac OS X menggunakan **struktur** ***hybrid*****.** Strukturnya menggunakan teknik berlapis dan satu lapisan diantaranya menggunakan Mach microkernel.

5. **Mesin Maya** ***(Virtual Machine)***

Mesin maya mempunyai sistem timesharing yang berfungsi untuk ,menyediakan kemampuan untuk multiprogramming dan perluasan mesin dengan antarmuka yang lebih mudah.

Struktur Mesin maya (CP/CMS, VM/370) terdiri atas komponen dasar utama :

- Control Program, yaitu virtual machine monitor yang mengatur fungsi ari prosessor, memori dan piranti I/O. Komponen ini berhubungan langsung dengan perangkat keras.
- Conventional Monitor System, yaitu sistem operasi sederhanayang mengatur fungsi dari proses, pengelolaan informasi dan pengelolaan piranti.

Kelebihan Mesin Maya ***(Virtual Machine)*****:**

- Konsep mesin virtual menyediakan proteksi yang lengkap untuk sumber daya system sehingga masing-masing mesin virtual dipisahkan mesin virtual yang lain. Isolasi ini tidak memperbolehkan pembagian sumber daya secara langsung.
- Sistem mesin virtual adalah mesin yang sempurna untuk riset dan pengembangan system operasi. Pengembangan system dikerjakan pada mesin virtual, termasuk di dalamnya mesin fisik dan tidak mengganggu operasi system yang normal.

Kekurangan Mesin Maya ***(Virtual Machine)*****:**

- Konsep mesin virtual sangat sulit untuk mengimplementasikan kebutuhan dan duplikasi yang tepat pada mesin yang sebenarnya.

Contoh:

- Sistem operasi MS-Windows NT dapat menjalankan aplikasi untuk MS-DOS, OS/2 mode teks dan aplikasi WIN16.
- IBM mengembangkan WABI untuk meng-emulasikan Win32 API sehingga sistem operasi yang menjalankan WABI dapat menjalankan aplikasi-aplikasi untuk MS-Windows.
- Para pengembang Linux membuat DOSEMU untuk menjalankan aplikas-aplikasi DOS pada sistem operasi Linux, WINE untuk menjalankan aplikasi-aplikasi MS-Windows.
- VMWare merupakan aplikasi komersial yang meng-abstraksikan perangkat keras intel 80×86 menjadi virtual mesin dan dapat menjalan beberapa sistem operasi lain (*guest operating system*) di

dalam sistem operasi MS-Windos atau Linux (*host operating system*). **VirtualBox** merupakan salah satu aplikasi sejenis yang opensource.

6. **Client-Server Model**

Mengimplementasikan sebagian besar fungsi sistem operasi pada mode pengguna (user mode). Sistem operasi merupakan kumpulan proses dengan proses-proses dikategorikan sebagai server dan client, yaitu :

Server, adalah proses yang menyediakan layanan.

Client, adalah proses yang memerlukan/meminta layanan.

Proses client yang memerlukan layanan mengirim pesan ke server dan menanti pesan jawaban. Proses server setelah melakukan tugas yang diminta, mengirim hasil dalam bentuk pesan jawaban ke proses client. Server hanya menanggapi permintaan client dan tidak memulai dengan percakapan client. Kode dapat diangkat ke level tinggi, sehingga kernel dibuat sekecil mungkin dan semua tugas diangkat ke bagian proses pemaka. Kernel hanya mengatur komunikasi antara client dan server. Kernel yang ini popular dengan sebutan mikrokernel.

Kelebihan Client-Server Model:

- Pengembangan dapat dilakukan secara modular.
- Kesalahan (bugs) di satu subsistem (diimplementasikan sebagai satu proses) tidak merusak subsistem-subsistem lain, sehingga tidak mengakibatkan satu sistem mati secara keseluruhan.
- Mudah diadaptasi untuk sistem tersebar.

Kekurangan Client-Server Model:

- Layanan dilakukan lambat karena harus melalui pertukaran pesan.
- Pertukaran pesan dapat menjadi bottleneck.
- Tidak semua tugas dapat dijalankan di tingkat pemakai (sebagai proses pemakai).

7. Sistem Berorientasi Objek

Sisten operasi merealisasikan layanan sebagai kumpulan proses disebut sistem operasi bermodel proses. Pendekatan lain implementasi layanan adalah sebagai objek-objek. Sistem operasu yang distrukturkan menggunakan objek disebut sistem operasi berorientasi objek. Pendekatan ini dimaksudkan untuk mengadopsi keunggulan teknologi berorientasi objek. Pada sistem yang berorientasi objek, layanan diimplementasikan sebagai kumpulan objek. Objek mengkapsulkan struktur data dan sekumpulan operasi pada struktur data itu. Tiap objek diberi tipe yang menandadi properti objek seperti proses, direktori, berkas, dan sebagainya. Dengan memanggil operasi yang didefinisikan di objek, data yang dikapsulkan dapat diakses dan dimodifikasi. Model ini sungguh terstruktur dan memisahkan antara layanan yang disediakan dan implementasinya. Sistem operasi MS Windows NT telah mengadopsi beberapa teknologi berorientasi objek tetapi belum keseluruhan.

Kelebihan Sistem Berorientasi Objek:

- Terstruktur dan memisahkan antara layanan yang disediakan dan implementasinya.

Kekurangan Sistem Berorientasi Objek:

- Sistem operasi MS Windows NT telah mengadopsi beberapa teknologi berorientasi objek tetapi belum keseluruhan.

Contoh sistem operasi yang berorientasi objek, antara lain : eden, choices, x-kernel, medusa, clouds, amoeba, muse, dan sebagainya.

4. Pertemuan keempat

MANAJEMEN PROSES PADA SISTEM OPERASI

Manajemen proses merupakan konsep pokok dalam sistem operasi, sehingga masalah manajemen proses adalah masalah utama dalam perancangan sistem operasi. Proses adalah program yang sedang dieksekusi. Proses dapat juga didefinisikan sebagai unit kerja terkecil yang secara individu memiliki sumber daya dan dijadwalkan oleh sistem operasi. Proses berisi instruksi, data, program counter, register pemroses, stack data, alamat pengiriman dan variabel pendukung lainnya.

Sebagaimana proses bekerja, maka proses tersebut merubah state (keadaan statis/ asal). Status dari sebuah proses didefinisikan dalam bagian oleh aktivitas yang ada dari proses tersebut. Tiap proses mungkin adalah satu dari keadaan berikut ini:

- New: Proses sedang dikerjakan/ dibuat.
- Running: Instruksi sedang dikerjakan.
- Waiting: Proses sedang menunggu sejumlah kejadian untuk terjadi (seperti sebuah penyelesaian I/O
- Ready: Proses sedang menunggu untuk ditugaskan pada sebuah prosesor.
- Terminated: Proses telah selsesai melaksanakan tugasnya/ mengeksekusi.

PROCESS CONTROL BLOCK

Tiap proses digambarkan dalam sistem operasi oleh sebuah process control block (PCB) – juga disebut sebuah control block. Sebuah PCB ditunjukkan dalam Gambar 2. PCB berisikan banyak bagian dari informasi yang berhubungan dengan sebuah proses yang spesifik, termasuk ini:

- Keadaan proses: Keadaan mungkin, new, ready, running, waiting, halted, dan juga banyak lagi.
- Program counter: Counter mengindikasikan address dari perintah selanjutnya untuk dijalankan untuk proses ini.
- CPU register: Register bervariasi dalam jumlah dan jenis, tergantung pada rancangan komputer.
-

PENJADWALAN PROSES

Tujuan dari multiprogramming adalah untuk memiliki sejumlah proses yang berjalan pada sepanjang waktu, untuk memaksimalkan penggunaan CPU.

Tujuan dari pembagian waktu adalah untuk mengganti CPU diantara proses-proses yang begitu sering sehingga pengguna dapat berinteraksi dengan setiap program sambil CPU bekerja. Untuk sistem uniprosesor, tidak akan ada lebih dari satu proses berjalan. Jika ada proses yang lebih dari itu, yang lainnya akan harus menunggu sampai CPU bebas dan dapat dijadualkan kembali.

Terdapat 3 konsep dasar Penjadwalan proses yaitu :

Penjadualan Antrian (Scheduling Queue)

Ketika proses memasuki sistem, mereka diletakkan dalam antrian job. Antrian ini terdiri dari seluruh proses dalam sistem. Proses yang hidup pada memori utama dan siap dan menunggu/ wait untuk mengeksekusi disimpan pada sebuah daftar bernama ready queue. Antrian ini biasanya disimpan sebagai daftar penghubung. Sebuah header ready queue berisikan penunjuk kepada PCB-PCB awal dan akhir. Setiap PCB memiliki pointer field yang menunjukkan proses selanjutnya dalam ready queue.

Penjadual / Scheduler

Sebuah proses berpindah antara berbagai penjadualan antrian selama umur hidupnya. Sistem operasi harus memilih, untuk keperluan penjadualan, memproses antrian-antrian ini dalam cara tertentu. Pemilihan proses dilaksanakan oleh penjadual yang tepat/ cocok. Dalam sistem batch, sering ada lebih banyak proses yang diserahkan daripada yang dapat dilaksanakan segera. Proses ini dipitakan/ disimpan

pada suatu alat penyimpan masal (biasanya disket), dimana proses tersebut disimpan untuk eksekusi dilain waktu. Penjadualan long term, atau penjadual job, memilih proses dari pool ini dan mengisinya kedalam memori eksekusi.

Alih Konteks / Switch Context

Mengganti CPU ke proses lain memerlukan penyimpanan suatu keadaan proses lama (state of old process) dan kemudian beralih ke proses yang baru. Tugas tersebut diketahui sebagai alih konteks (context switch). Alih konteks sebuah proses digambarkan dalam PCB suatu proses; termasuk nilai dari CPU register, status proses (lihat Gambar 7). dan informasi managemen memori. Ketika alih konteks terjadi, kernel menyimpan konteks dari proses lama kedalam PCB nya dan mengisi konteks yang telah disimpan dari process baru yang telah terjadual untuk berjalan. Pergantian waktu konteks adalah murni overhead, karena sistem melakukan pekerjaan yang tidak perlu. Kecepatannya bervariasi dari mesin ke mesin, bergantung pada kecepatan memori, jumlah register yang harus di copy, dan keberadaan instruksi khusus (seperti instruksi tunggal untuk mengisi atau menyimpan seluruh register). Tingkat kecepatan umumnya berkisar antara 1 sampai 1000 mikro detik

OPERASI PADA PROSES

Proses dalam sistem dapat dieksekusi secara bersama-sama, proses tersebut harus dibuat dan dihapus secara dinamis. Maka, sistem operasi harus menyediakan suatu mekanisme untuk pembuatan proses dan erminasi proses. Sistem operasi dalam mengelola proses dapat melakukan operasi-operasi terhadap proses. Operasi tersebut adalah :

a. Penciptaan proses
b. Penghancuran/terminasi proses
c. Penundaan proses
d. Pelanjutan kembali proses
e. Pengubahan prioritas proses
f. Memblok proses
g. Membangunkan proses
h. Menjadwalkan proses
i. Memungkinkan proses berkomunikasi dengan proses lain

Pembuatan Proses

Melibatkan banyak aktivitas, yaitu :

a. Memberi identitas proses
b. Menyisipkan proses pada senarai atau tabel proses
c. Menentukan prioritas awal proses
d. Menciptakan PCB
e. Mengalokasikan sumber daya awal bagi proses

Ketika proses baru ditambahkan, sistem operasi membangun struktur data untuk mengelola dan mengalokasikan ruang alamat proses.

Kejadian yang dapat menyebabkan penciptaan proses :

a. Pada lingkungan batch, sebagai tanggapan atas pemberian satu kerja (job). Sistem operasi dengan kendali batch job, setelah menciptakan proses baru, kemudian melanjutkan membaca job berikutnya.
b. Pada lingkungan interaktif, ketika pemakai baru berusaha logon.
c. Sebagai tanggapan suatu aplikasi, seperti permintaan pencetakan file, sistem operasi dapat menciptakan proses yang akan mengelola pencetakan itu. Sistem operasi menciptakan proses untuk memenuhi satu fungsi pada program pemakai, tanpa mengharuskan pemakai menunggu.
d. Proses penciptaan proses lain (proses anak). Untuk mencapai modularitas atau mengeksploitasi kongkurensi, program pemakai memerintahkan pembuatan sejumlah proses.

Tahap-tahap penciptaan proses

Penciptaan proses dapat disebabkan beragam sebab. Penciptaan proses meliputi beberapa tahap :

1. Beri satu identifier unik ke proses baru. Isian baru ditambahkan ke tabel proses utama yang berisi satu isian perproses.
2. Alokasikan ruang untuk proses.
3. PCB harus diinisialisasi.
4. Kaitan-kaitan antar tabel dan senarai yang cocok dibuat.
5. Bila diperlukan struktur data lain maka segera dibuat struktur data itu.

Penghancuran / Terminasi Proses

Penghancuran proses melibatkan pembebasan proses dari sistem, yaitu :

a. Sumber daya-sumber daya yang dipakai dikembalikan.
b. Proses dihancurkan dari senarai atau tabel sistem.
c.PCB dihapus (ruang memori PCB dikembalikan ke pool memori bebas).
ci.

Penghancuran lebih rumit bila proses telah menciptakan proses-proses lain. Terdapat dua pendekatan, yaitu :

a. Pada beberapa sistem, proses-proses turunan dihancurkan saat proses induk dihancurkan secara otomatis.
b. Beberapa sistem lain menganggap proses anak independen terhadap proses induk, sehingga proses anak tidak secara otomatis dihancurkan saat proses induk dihancurkan.

Alasan penghancuran proses :

1 Selesainya proses secara manual. Proses mengeksekusi panggilan layanan sistem operasi untuk menandakan bawah proses telah berjalan secara lengkap.
2 Batas waktu telah terlewati. Proses telah berjalan melebihi batas waktu total yang dispesifikasikan.
3 Memori tidak tersedia. Proses memerlukan memori lebih banyak daripada yang dapat disediakan sistem.
4 Pelanggaran terhadap batas memori Proses mencoba mengakses lokasi memori yang tidak diijinkan diakses.
5 Terjadi kesalahan karena pelanggaran proteksi
6 Terjadi kesalahan aritmatika
7 Waktu telah kedaluwarsa
8 Terjadi kegagalan Masukan/keluaran
9 Instruksi yang tidak benar
10 Terjadi usaha memakai nstruksi yang tidak Diijinkan
11 Kesalahan Penggunaan data Bagian data adalah tipe yang salah atau tidak diinisialisasi.
12 Diintervensi oleh sistem operasi
13 Berakhirnya proses induk
14 Atas permintaan dari proses induk

Proses yang Kooperatif

Proses yang bersifat simultan (concurrent) dijalankan pada sistem operasi dapat dibedakaan menjadi yaitu proses independent dan proses kooperatif. Suatu proses dikatakan independen apabila proses tersebut tidak dapat terpengaruh atau dipengaruhi oleh proses lain yang sedang dijalankan pada sistem.

Komunikasi Proses Dalam Sistem

Cara lain untuk meningkatkan efek yang sama adalah untuk sistem operasi yaitu untuk menyediakan alat-alat proses kooperatif untuk berkomunikasi dengan yang lain lewat sebuah komunikasi dalam proses (IPC = Inter-Process Communication). IPC menyediakan sebuah mekanisme untuk mengizinkan

proses- proses untuk berkomunikasi dan menyelaraskan aksi-aksi mereka tanpa berbagi ruang alamat yang sama. IPC adalah khusus digunakan dalam sebuah lingkungan yang terdistribusi dimana proses komunikasi tersebut mungkin saja tetap ada dalam komputer-komputer yang berbeda yang tersambung dalam sebuah jaringan. IPC adalah penyedia layanan terbaik dengan menggnakan sebuah sistem penyampaian pesan, dan sistem- sistem pesan dapat diberikan dalam banyak cara.

Sistem Penyampaian Pesan

Fungsi dari sebuah sistem pesan adalah untuk memperbolehkan komunikasi satu dengan yang lain tanpa perlu menggunakan pembagian data. Sebuah fasilitas IPC menyediakan paling sedikit dua operasi yaitu kirim (pesan) dan terima (pesan). Pesan dikirim dengan sebuah proses yang dapat dilakukan pada ukuran pasti atau variabel. Jika hanya pesan dengan ukuran pasti dapat dikirimkan, level sistem implementasi adalah sistem yang sederhana. Pesan berukuran variabel menyediakan sistem implementasi level yang lebih kompleks.

Berikut ini ada beberapa metode untuk mengimplementasikan sebuah jaringan dan operasi pengiriman/penerimaan secara logika:

Komunikasi langsung atau tidak langsung.

Komunikasi secara simetris/ asimetris.

Buffer otomatis atau eksplisit.

engiriman berdasarkan salinan atau referensi.

Pesan berukuran pasti dan variabel.

THREAD

Model proses yang didiskusikan sejauh ini telah menunjukkan bahwa suatu proses adalah sebuah program yang menjalankan eksekusi thread tunggal. Sebagai contoh, jika sebuah proses menjalankan sebuah program Word Processor, ada sebuah thread tunggal dari instruksi-instruksi yang sedang dilaksanakan.

Konsep Dasar

Secara informal, proses adalah program yang sedang dieksekusi. Ada dua jenis proses, proses berat (heavyweight) atau biasa dikenal dengan proses tradisional, dan proses ringan atau kadang disebut thread.

Thread saling berbagi bagian program, bagian data dan sumber daya sistem operasi dengan thread lain yang mengacu pada proses yang sama. Thread terdiri atas ID thread, program counter, himpunan register, dan stack. Dengan banyak kontrol thread proses dapat melakukan lebih dari satu pekerjaan pada waktu yang sama.

User Threads

User thread didukung oleh kernel dan diimplementasikan oleh thread library ditingkat pengguna. Library mendukung untuk pembentukan thread, penjadualan, dan managemen yang tidak didukung oleh kernel.

Kernel Threads

Kernel thread didukung secara langsung oleh sistem operasi: pembentukan thread, penjadualan, dan managemen dilakukan oleh kernel dalam ruang kernel. Karena managemen thread telah dilakukan oleh sistem operasi, kernel thread biasanya lebih lambat untuk membuat dan mengelola daripada pengguna thread.

Model Multithreading

Dalam sub bab sebelumnya telah dibahas pengertian dari thread, keuntungannya, tingkatan atau levelnya seperti pengguna dan kernel. Sistem-sistem yang ada sekarang sudah banyak yang bisa mendukung untuk kedua pengguna dan kernel thread, sehingga model-model multithreading-nya pun

menjadi beragam. Implementasi multithreading yang umum akan kita bahas ada tiga, yaitu model many-to-one, one-to-one, dan many-to-many.

Model Many to One

Model many-to-one ini memetakan beberapa tingkatan pengguna thread hanya ke satu buah kernel thread. Managemen proses thread dilakukan oleh (di ruang) pengguna, sehingga menjadi efisien, tetapi apabila sebuah thread melakukan sebuah pemblokingan terhadap sistem pemanggilan, maka seluruh proses akan berhenti (blocked). Kelemahan dari model ini adalah multihreads tidak dapat berjalan atau bekerja secara paralel di dalam multiprosesor dikarenakan hanya satu thread saja yang bisa mengakses kernel dalam suatu waktu.

Model One to One

Model one-to-one memetakan setiap thread pengguna ke dalam satu kernel thread. Hal ini membuat model one-to-one lebih sinkron daripada model many-to-one dengan mengizinkan thread lain untuk berjalan ketika suatu thread membuat pemblokingan terhadap sistem pemanggilan; hal ini juga mengizinkan multiple thread untuk berjalan secara parallel dalam multiprosesor. Kelemahan model ini adalah dalam pembuatan thread pengguna dibutuhkan pembuatan korespondensi thread pengguna. Karena dalam proses pembuatan kernel thread dapat mempengaruhi kinerja dari aplikasi maka kebanyakan dari implementasi model ini membatasi jumlah thread yang didukung oleh sistem. Model one-to-one diimplementasikan oleh Windows NT dan OS/2.

Model Many to Many

Beberapa tingkatan thread pengguna dapat menggunakan jumlah kernel thread yang lebih kecil atau sama dengan jumlah thread pengguna. Jumlah dari kernel thread dapat dispesifikasikan untuk beberapa aplikasi dan beberapa mesin (suatu aplikasi dapat dialokasikan lebih dari beberapakernel thread dalam multiprosesor daripada dalam uniprosesor) dimana model many-to-one mengizinkan pengembang untuk membuat thread pengguna sebanyak mungkin, konkurensi tidak dapat tercapai karena hanya satu thread yang dapat dijadualkan oleh kernel dalam satu waktu. Model one-to-one mempunyai konkurensi yang lebih tinggi, tetapi pengembang harus hati-hati untuk tidak membuat terlalu banyak thread tanpa aplikasi dan dalam kasus tertentu mungkin jumlah thread yang dapat dibuat dibatasi.

5. Pertemuan kelima

Penjadwalan Proses

Deskripsi Penjadwalan Proses
Kumpulan kebijaksanaan dan mekanisme di sistem operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer.

Penjadwalan bertugas memutuskan hal-hal berikut :

- Proses yang harus berjalan
- Kapan dan selama berapa lama proses berjalan

Sasaran utama penjadwalan proses adalah Optimasi kinerja sistem komputer menurut kriteria tertentu. Kriteria untuk mengukur dan optimasi kinerja penjadwalan adalah sbb:

1. Adil (fairness)
2. Efisiensi
3. Waktu Tanggap (response time)
4. Turn arround Time
5. Troughput

Adil (fairness)
Proses-proses diperlakukan sama yaitu mendapat jatah waktu layanan pemroses yang sama dan tidak ada proses yang tidak kebagian layanan pemroses sehingga mengalami startvation.
Startvation adalah kondisi bahwa proses tidak pernah berjalan karena tidak dijadwalkan untuk berjalan. Sasaran penjadwalan seharusnya menjamin setiap proses mendapat pelayanan dari pemroses secara adil.

Efisiensi
Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses dengan total waktu operasi sistem komputer secara keseluruhan.
Sasaran penjadwalan adalah menjaga agar pemroses tetap dalam keadaan sibuk sehingga efisiensi sistem komputer mencapai nilai maksimum. Keadaan sibuk berarti pemroses tidak menganggur. Layanan pemroses termasuk waktu yang dihabiskan untuk mengeksekusi program pemakai dan layanan sistem operasi secara efektif, bukan untuk melakukan penjadwalan itu sendiri.

Waktu Tanggap (response time)
Waktu tanggap berbeda untuk :

- Sistem interaktif

Waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan oleh program atau transaksi sampai hasil pertama muncul di jperangkat masukan keluaran seperti layar (terminal).Waktu tanggap untuk sistem interaktif biasa disebut terminal responce time.

- Sistem waktu nyata (real time)

Pada sistem waktu nyata, waktu tanggap didefinisikan sebagai waktu dari saat kemunculan suatu kejadian (internal/eksternal) sampai instruksi pertama rutin layanan terhadap kejadian dieksekusi. Waktu untuk sistem waktu nyata biasa disebut event response time.
Sasaran penjadwalan adalah meminimalkan waktu tanggap sehingga menghasilkan sistem yang responsif.

Turn arround Time
Waktu yang dihabiskan dari saat proses atau job mulai masuk ke sistem sampai proses itu

diselesaikan sistem.Waktu yang dimaksud adalah waktu yang dihabiskan proses berada di sistem, diekspresikan sebagai penjumlahan waktu eksekusi (waktu layanan proses/job) dan waktu menunggu dari proses itu, yaitu :

Turn arround time = waktu eksekusi + waktu menunggu.

Sasaran penjadwalan adalah meminimalkan turn arround time.

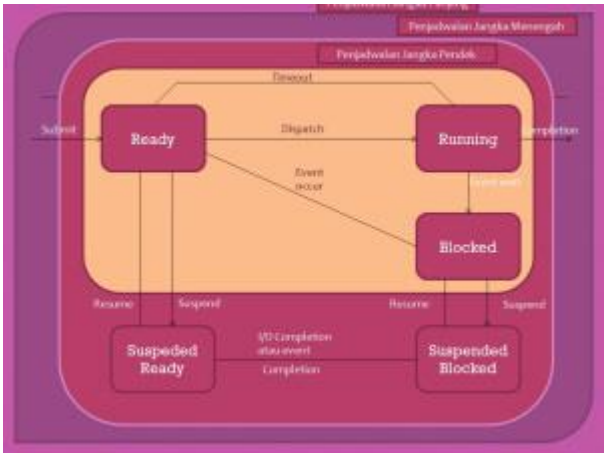
Troughput
Troughput adalah jumlah kerja yang dapat diselsesaikan selama satu selang/ unit waktu. Cara untuk mengekspresikan throughput adalah dengan jumlah proses/jobpemakai yang dapat dieksekusi dalam satu unit/ interval waktu tertentu.
Sasaran penjadwalan adalah memaksimalkan jumlah job/ proses yang dilayani per satu interval waktu. Lebih tinggi angka througput maka lebih banya kerja yang dilakukan sistem.
Kriteria tsb saling bergantung dan dapat saling bertentangan sehingga tidak dimungkinkan optimasi semua kriteria secara simultan.

Tipe-Tipe Penjadwalan
Dapat terdapat 3 tipe penjadwal berada secara bersama-sama pada sistem operasi yang kompleks, yaitu :

1. Penjadwal jangka pendek (short-term scheduller). Penjadwalan jangka pendek bertugas menjadwalkan alokasi pemroses di antara proses-proses Ready yang berada di memori utama. sasaran utama penjadwal jangka pendek adalah memaksimumkan kinerja sistem untuk memenuhi satu kumpulan kriteria yang diharapkan. Penjadwal ini dijalankan setiap terjadi pengalihan proses untuk memilih proses berikutnya yang harus dijalankan.

Gambar Posisi Tipe-Tipe Penjadwalan yang dapat terdapat di satu sistem operasi

2. Penjadwal jangka menengah (medium-term scheduller). Setelah eksekusi selama suatu waktu, proses mungkin ditunda karena permintaan layanan masukan/keluaran atau memanggil suatu system call. Proses-proses yang tertunda tidak dapat membuat suatu kemajuan untuk menuju selesai sampai ondisi yang menyebabkannya hilang. Agar ruang memori dapat bermanfaat maka proses dipindah dari memori utama ke memori sekunder sehingga tersedia ruang yang lebih besar untuk proses yang lain. Kapasitas memori utama terbatas untuk sejumlah proses yang aktif. Aktivitas pemindahan proses yang tertunda dari memori utama ke memori sekunder disebut *swapping*. Penjadwal jangka menengah bertugas menangani proses swapping . Proses yang mempunyai kepentingan kecil saat itu adalah proses yang tertunda. Tetapi begitu kondii yang membuat proses tertunda hilang dan proses dimasukkan kembali ke memori utama dan Ready. Penjadwal jangka menengah mengendalikan transisi dari suspended ke ready (dari state suspend ke Ready dari proses yang mengalami swapping).



1. Penjadwal jangka panjang (long-term scheduller). Penjadwal jangka panjang bekerja terhadap antrian batch dan memilih batch berikutnya yang harus dieksekusi sistem. Batch biasanya berupa proses-proses dengan penggunaan sumber daya yang intensif (yaitu waktu pemroses, memori, perangkat masukan/keluaran), program ini mempunyai prioritas yang rendah, dan biasa digunakan sebagai pengisi (agar pemroses sibu) selama periode aktivitas proses-proses interaktif rendah. Sasaran utama penjadwal jangka panjang adalah memberi keseimbangan proses-proses campuran. Tipe-tipe penjadwal dapat dikaitkan dengan state proses. Kaitan antara tipe-tipe penjadwalan dengan state proses digambarkan pada gambar berikut :

Strategi Penjadwalan
Terdapat 2 strategi penjadwalan, yaitu :

1. Penjadwalan nonpreemptive (run-to-completion). Begitu proses diberi jatah layanan pemroses aka pemroses tidak dapat diambil alih oleh proses lain sampai proses itu selesai. Non-preemptive juga disebut run-to-completion karena proses yang telah dijadwalkan akan dijalankan sampai selesainya atau proses tersebut meminta layanan masukan/keluaran.
2. Penjadwalan preemptive. Saat proses diberi jatah layanan pemroses maka pemroses dapat diambil alih proses lain yang mempunyai prioritas lebih tinggi berdasarkan kriteria sistem itu. Pada penjadwalan preemptive, proses dapat disela oleh proses lain sebelumnya selesainya dan harus dilanjutkan menunggu jatah waktu layanan pemroses tiba kembali pada proses itu. Proses yang disela berubah menjadi state Ready.

Penjadwalan preemptive berguna pada sistem yakni proses-proses yang perlu mendapat perhatian/ tanggapan pemroses secara cepat. Misalnya :

- Pada sistem-sistem waktu nyata, kehilangan interupsi (yaitu interupsi tidak segera dilayani) dapat berakibat fatal
- Pada sistem-sistem interatif timesharing, penjadwalan preemptive penting agar dapat menjamin waktu tanggap yang memadai.

Peralihan proses (yaitu layanan pemroses dari satu proses beralih ke proses lain) memerlukan overhead (karena banya tabel yang dikelola). Agar penjadwalan preemptive menjadi efektif, banyak proses harus berada di memori utama sehingga proses-proses tersebut dapat segera Running begitu diperlukan. Menyimpan banyak proses yang tidak Running di memori utama merupakan suatu overhead tersendiri.

Algoritma-Algoritma Penjadwalan Proses
Terdapat banyak algoritma penjadwalan, baik algoritma penjadwalan nonpreemptive maupun penjadwalan preemptive.
Algoritma-algoritma yang menerapkan strategi nonpreemptive diantaranya :

1. FIFO (First-In, First-Out) atau FCFS (First-Come, First-Serve)
2. SJF (Shortest Job First)

Algoritma-algoritma yang menerapkan strategi preemptive diantaranya :

1. RR (Round-Robin)
2. MFQ (Multiple Feedback Queues)
3. SRF (Shortest-Remaining-First)
4. HRN (Highest-Remaining-Next)
5. PS (Priority Schedulling)
6. GS (Guaranteed Schedulling)

Klasifikasi lain selain berdasarkan dapat/tidaknya suatu proses diambil alih secara paksa adalah klasifikasi yang berdasarkan adanya prioritas diproses-proses, yaitu :

1. Algoritma penjadwalan tanpa berprioritas
2. Algoritma penjadwalan berprioritas, terdiri dari :
    - Algoritma penjadwalan berprioritas statis
    - Algoritma penjadwalan berprioritas dinamis

Algoritma-Algoritma Penjadwalan Proses

1. Penjadwalan Round-Robin (RR)
2. Penjadwalan FIFO (FIFO)
3. Penjadwalan Berprioritas (PS)
4. Penjadwalan yang Terpendek yang Lebih Dahulu (SJF)
5. Penjadwalan dengan Banyak Antrian (MFQ)
6. Penjadwalan dengan Sisa Waktu Terpendek, Lebih Dahulu (SRF)
7. Penjadwalan Rasio Tanggapan Tertinggi, Lebih Dahulu(HRN)
8. Penjadwalan Terjamin (GS)

Penjadwalan Round Robin
Penjadwalan Round Robin merupakan

- Penjadwalan Preemptive, namun proses tidak di-preempt secara langsung oleh proses lain, namun oleh penjadwal berdasarkan lama waktu berjalannya suatu proses. Maka penjadwalan ini disebut preempt-by-time
- Penjadwalan tanpa prioritas

Semua proses dianggap penting dan diberi jumlah waktu pemroses yang disebut kwanta (quantum) atau time-slice tempat proses tsb berjalan. Proses berjalan selama 1 kwanta, kemudian penjadwal akan mengalihkan kepada proses berikutnya juga untuk berjalan satu kwanta, begitu seterusnya sampai kembali pada proses pertama dan berulang.

Ketentuan algoritma round robin adalah sbb :

1. Jika kwanta habis dan proses belum selesai maka proses Runing menjadi Ready dan pemroses dialihkan ke proses lain
2. Jika kwanta belum habis dan proses menunggu suatu kejadian (misalnya menunggu selesainya suatu operasi I/O), maka proses Running menjadi Blocked dan pemroses dialihkan ke proses lain.
3. Jika kwanta belum habis tapi proses telah selesai maka proses Running itu diakhiri dan pemroses dialihkan ke proses lain

Algoritma penjadwalan ini dapat diimplementasi sbb :

- Sistem mengelola senarai proses Ready sesuai urutan kedatangannya
- Sistem mengambil proses yang berada di ujung depan antrian Ready menjadi Running
- Bila kwanta belum habis dan proses selesai maka sistem mengambil proses di ujung depan antrian proses Ready
- Jika kwanta habis dan proses belum selesai maka ditempatkan proses Running ke ekor antrian proses Ready dan sistem mengambil proses di ujung depan antrian proses Ready

Masalah penjadwalan ini adalah dalam hal menentukan besar kwanta, yaitu :

- Kwanta terlalu besar menyebabkan waktu tanggap besar dan turn arround time rendah
- Kwanta terlalu kecil mengakibatkan peralihan proses terlalu banyak sehingga menurunkan efisiensi pemroses

Harus diterapkan besar kwanta waktu yang optimal berdasarkan kebutuhan sistem, terutama dari hasil percobaan atau data historis dari sistem. Besar kwanta waktu beragam yang bergantung beban sistem. Berdasarkan kriteria penilaian penjadwalan.

- Fairness, penjadwalan RR adil bila dipandang dari persamaan pelayanan oleh pemroses
- Efisiensi, penjadwalan ini cenderung efisien pada sistem interatif
- Respons Time(waktu tanggap), penjadwalan ini memuaskan untuk sistem interaktif, tidak memadai untuk sistem waktu nyata.Turn arround Time, penjadwalan RR cukup bagus
- Throughput, penjadwlan RR cukup bagus

Penjadwalan FIFO
Penjadwalan FIFO merupakan :

- Penjadwalan non preemptive (run-to-completion)
- Penjadwalan tidak berprioritas

Penjadwal FIFO adalah penjadwalan dengan ketentuan-ketentuan paling sederhana, yaitu :

- Proses-proses diberi jatah waktu pemroses diurutkan berdasarkan waktu kedatangan proses-proses itu ke sistem.
- Begitu proses mendapat jatah waktu pemroses, proses dijalankan sampai selesai

Penjadwalan ini dikatakan adil dalam arti resmi, tapi dikatakan tidak adil karena proses yang memerlukan waktu lama membuat proses pendek menunggu. Proses tidak penting dapat membuat proses penting menjadi menunggu.
FIFO jarang digunakan secara mandiri tapi dikombinasikan dengan skema lain, misalnya keputusan berdasarkan prioritas proses, sedangkan untuk proses berprioritas sama diputuskan berdasarkan FIFO.

Berdasarkan kriteria penilaian penjadwalan :

- Fairness, penjadwalan FIFO adil dalam arti resmi
- Efisiensi, FIFO sangat efisien dalam penggunaan pemroses
- Waktu tanggap, penjadwalan sangat tidak memuaskan karena proses dapat menunggu lama. Tidak cocok untuk sistem interaktifTurn arround time, penjadwalan FIFO tidak bagus
- Throughput, penjadwalan FIFO tidak bagus.

Penjadwalan Berprioritas
Gagasan penjadwalan adalah masing-masing proses diberi prioritas dan proses berprioritas tertinggi menjadi Running (yaitu mendapat jatah waktu pemroses).
Prioritas dapat diberikan secara

- Prioritas statis (static priorities), prioritas tak berubah.

keunggulan : Mudah diimplementasikan dan mempunyai overhead relatif kecil
kelemahan : penjadwalan prioritas statis tidak tanggap perubahan lingkungan yang mungkin menghendaki penyesuaian prioritas

- Prioritas dinamis (dynamic priorities), mekanisme menanggapi perubahan lingkungan sistem saat beroperasi di lingkungan nyata. Prioritas awal yang diberikan ke proses mungkin hanya berumur pendek. Dalam hal ini sistem dapat menyesuaikan nilai prioritasnya ke nilai yang lebih tepat sesuai lingkungan.

keunggulan : waktu tanggap sistem yang bagus
kelemahan : implementsi mekanisme prioritas dinamis lebih kompleks dan mempunyai overhead yang lebih besar dibanding mekanisme prioritas statik.

Contoh penjadwalan berprioritas
Proses-proses yang sangat banyak operasi masukan/keluaran dan menghabiskan kebanyakan waktu proses untuk menunggu selesainya operasi masukan/ keluaran. Proses demikian disebut I/O bound process. Proses-proses ini dapat diberi prioritas sangat tinggi sehingga begitu proses-proses memerlukan pemroses, segera saja diberikan dan proses akan segera memulai permintaaan masukan/keluaran berikutnya sehingga menyebabkan proses Blocked menunggu selesainya operasi masukan/keluaran. Dengan demikian pemroses segera dialihkan, dapat dipergunakan oleh proses lain tanpa mengganggu proses I/O bound. Proses I/O bound berjalan paralel bersama proses lain yang benar-benar memerlukan pemroses.

Proses-proses yang sangat banyak operasi masukan/keluaran jika harus menunggu lama untuk memakai pemroses(karena diberi prioritas rendah) hanya akan membebani memori, karena sistem harus menyimpan tanpa perlu proses-proses itu di memori karena tidak selesai-selesai menunggu operasi masukan/keluaran dan menunggu jatah pemroses.

Algoritma Prioritas Dinamis
Algoritma ini dituntun oleh keputusan untuk memenuhi kebijaksanaan tertentu yang menjadi tujuan sistem komputer.
Algoritma sederhana yang memberi layanan yang baik adalah dengan menge-set proses dengan prioritas berdasarkan rumus nilai 1/f bahwa f adalah rasio kwanta terakhir yang digunakan proses.

- Proses yang menggunakan 2 milidetik, kwanta 100 ms maka prioritasnya 50
- Proses yang berjalan selama 50 milidetik sebelum Blocked berprioritas 2
- Proses yang menggunakan seluruh kwanta berprioritas 1

Kebijaksanaan yang diterapkan adalah jaminan proses-proses mendapat layanan yang adil dari pemroses dalam arti jumlah waktu pemroses yang sama untuk masing-masing pemroses pada satu waktu.
Biasanya memenuhi kebijaksanaan yang ingin mencapai level maksimal berdasarkan suatu kriteria tertentu di sistem.

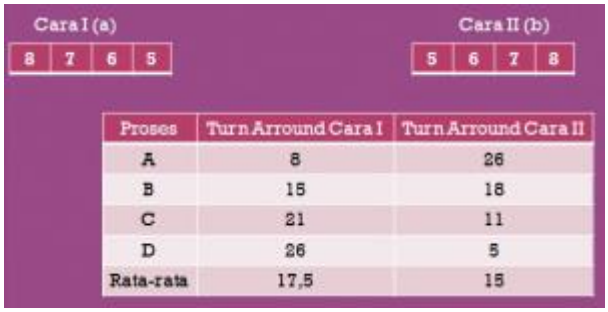
Algoritma penjadwalan berprioritas dapat dikombinasikan yaitu dengan mengelompokkan proses-proses menjadi kelas-kelas prioritas. Penjadwalan berprioritas diterapkan antar kelas- kelas proses itu. Penjadwlan round-robin atau penjadwalan FIFO diterapkan pada proses-proses di dalam satu kelas.

Penjadwalan yang Terpendek yang Lebih Dahulu (SJF)
Penjadwalan SJF ini merupakan

- Penjadwalan non preemptive
- Penjadwalan dapat dikatakan sebagai berprioritas. Di SJF, prioritas diasosiasikan dengan masing-masing proses dan pemroses dialokasikan ke proses dengan prioritas tertinggi. Proses-proses dengan prioritas yang sama akan dijadwalkan secara FIFO.

Penjadwalan ini mengasumsikan waktu jalan proses (sampai selesai) atau waktu lamanya proses diketahui sebelumnya. Mekanisme penjadwlan SJF adalah lebih dulu menjadwalkan proses dengan waktu jalan terpendek sampai selesai. Setelah proses itu selesai, maka proses dengan waktu jalan terpendek berikutnya dijadwalkan. Demikian seterusnya.
Keunggulan : penjadwalan SJF mempunyai efisiensi tinggi dan turn arround time rendah.

Contoh : Terdapat 4 proses A,B,C,D dengan waktu jalan selama 8,7,6,5 kwanta.
Gambar (a) menunjukkan penjadwalan cara I, dengan proses-proses dijadwalkan berurutan sebagai A,B,C,D. Gambar (b) menujukkan bila proses dijadwalkan secara SJF yaitu berurutan D,C,B,A.



Proses	Turn Arround Cara I	Turn Arround Cara II
A	8	26
B	15	18
C	21	11
D	26	5
Rata-rata	17,5	15

Kedua cara menghasilkan turn arround time yang ditunjukkan pada gambar (c). Cara I turn arround time rata-rata adalah 17,5 kwanta, sedangkan cara II adalah 15 kwanta
Walaupun mempunyai turn arround yang bagus, SJF mempunyai masalah, yaitu

- Tidak dapat mengetahui ukuran proses saat proses masuk
- Proses tidak datang bersamaan sehingga penetapannya harus dinamis

Untuk mengetahui ukuran lama proses agar dapat ditetapkan yang terpendek, biasanya dilakukan dengan cara pendekatan. Pendekatan yang biasa dilakukan adalah dengan membuat estimasi berdasarkan perilaku historis sistem. Merupakan kajian teoritis untuk pembandingan dalam pembandingan turn arround time.

Penjadwal dengan Banyak Antrian (MFQ)
Penjadwalan MFQ ini merupakan

- Penjadwalan preemptive
- Penjadwalan berprioritas dinamis

Sasaran penjadwalan ini adalah untuk mencegah banyaknya aktivitas swapping. Cara yang dilakukan adalah dengan

1. Proses-proses yang sangat banyak menggunakan pemroses (karena menyelesaikan tugasnya memakan waktu yang lama) diberi jatah waktu (jumlah kwanta) lebih banyak dalam satu waktu.
2. Penjadwalan ini menghendaki kelas prioritas bagi proses-proses yang ada. Kelas tertinggi berjalan selama satu kwanta, kelas berikutnya berjalan selama dua kwanta, kelas berikutnya lagi berjalan empat kwanta, kelas berikutnya-berikutnya lagi berjalan delapan kwanta dan seterusnya.

Ketentuan yang berlaku adalah sebagai berikut :

- Jalankan proses-proses yang berada pada kelas prioritas tertinggi
- Jika proses telah menggunakan seluruh kwanta yang dialokasikan maka proses itu diturunkan kelas prioritasnya
- Proses yang masuk untuk pertama kali ke sistem langsung diberi kelas tertinggi

Penjadwalan dengan Sisa Waktu Terpendek, Lebih Dahulu (SRF)
Penjadwalan ini merupakan

- Penjadwalan preemptive
- Penjadwalan berprioritas dinamis

Penjadwalan SRF merupakan perbaikan dari SJF, SJF merupakan penjadwalannonpreemptive sedang SRF adalah preemptive yang dapat digunakan untuk sistem timesharing.
Pada SRF, proses dengan sisa waktu jalan diestimasi terendah dijalankan, termasuk proses-proses yang baru tiba.

Perbedaan SRF dengan SJF

- Pada SJF, begitu proses dieksekusi, proses dijalankan sampai selesai
- Pada SRF proses yang sedang berjalan (Running) dapat diambil alih oleh proses baru dengan sisa waktu jalan yang diestimasi lebih rendah

SRF mempunyai overhead yang lebih besar dibanding SJF. SRF memerlukan penyimpanan waktu layanan yang telah dihabiskan proses dan kadang-kadang harus menangani peralihan.

- Tibanya proses-proses kecil akan segera dijalankan
- Proses-proses lebih lama berarti dengan lama dan variasi waktu tunggu lebih lama dibanding dengan SJF

Secara teoretis, SRF memberi waktu tunggu minimum tapi karena adanya overhead peralihan, maka pada situasi tertentu SJF bisa memberi kinerja yang lebih baik dibanding SRF.

Penjadwalan Rasio Tanggapan Tertinggi, Lebih Dahulu (HRN)
Penjadwalan HRN ini merupakan :

- Penjadwalan non preemptive
- Penjadwalan berprioritas dinamis

Penjadwalan ini juga untuk mengkoreksi kelemahan SJF. HRN adalah strategi penjadwalan non preemptive dengan prioritas proses tidak hanya merupakan fungsi dari waktu layanan, tapi juga jumlah waktu tunggu proses. Prioritas dinamis HRN dihitung berdasarkan rumus berikut :

Prioritas = (waktu tunggu + waktu layanan) / waktu layanan

Karena waktu layanan muncul sebagai pembagi maka proses yang lebih pendek mempunyai prioritas yang lebih baik. Karena waktu tunggu sebagai pembilang maka proses yang telah menunggu lebih lama juga mempunyai kesempatan lebih bagus untuk memperoleh layanan pemroses.
Disebut HRN (High respons next) karena waktu tanggap adalah (waktu tunggu + waktu layanan). Ketentuan HRN berarti agar memperoleh waktu tanggap tertinggi yang harus dilayani.

Penjadwalan Terjamin (GS)
Penjadwal GS ini adalah

- Penjadwalan preemptive
- Penjadwalan berprioritas dinamis

Penjadwalan ini berupaya memberi masing-masing pemakai daya pemroses yang sama. Jika terdapat N pemakai maka tiap pemakai diupayakan mendapat 1/N daya pemroses. Sistem merekam banyak waktu pemroses yang telah digunakan proses sejak login dan jumlah waktu proses yang digunakan seluruh proses.

Karena jumlah waktu pemroses tiap pemakai dapat diketahui, maka dapat dihitung rasio antara waktu pemroses yang sesungguhnya harus diperoleh yaitu 1/N waktu pemroses seluruhnya dan waktu pemroses telah diperuntukkan proses itu.
Penjadwal akan menjalankan proses dengan rasio terendah sampai rasio proses diatas pesaing terdekatnya.

Evaluasi Algoritma
Bagaimana memilih algoritma penjadwalan untuk sistem tertentu? Masing-masing algoritma mempunyai parameter-parameter tersendiri. Pemilihan algoritma penjadwalan merupakan hal yang sulit. Persoalan pertama adalah mendefinisikan kriteria untuk pemilihan algoritma.
Kriteria-kriteria yang sering digunakan adalah fairness (keadilan), efisiensi, waktu tanggap, turn arround time dan throughput. Kriteria kemudian dapat menjadi :

- Memaksimumkan utilisasi pemroses dengan konstrain waktu tanggap maksimum adalah 500 milidetik, atau
- Memaksimumkan throughput bahwa turn arround time adalah berbanding linier dengan waktu eksekusi total.

Begitu kriteria pemilihan telah didefinisikan, kita dapat mengeveluasi beragam algoritma. Terdapat sejumlah metode evaluasi untuk melakukan hal ini, yaitu :

- Pemodelan deterministis, merupakan evaluasi analitis. Evaluasi analitis menggunakan algoritma dan beban kerja sistem untuk menghasilkan satu rumus atau angka yang menunjukkan kinerja algoritma untuk beban kerja itu. Pemodelan deterministik menggunakan suatu beban kerja tertentu yang telah ditentukan dan mendefinisikan kinerja algoritma untuk beban kerja itu.
- Pemodelan antrian, sistem komputer dipandang sebagai satu jaringan pelayanan (server). Masing-masing pelayan mempunyai satu antrian dari proses-proses yang menunggu layanan. Pemroses adalah satu pelayan dengan satu antrian proses yang siap menerima layanan, begitu juga perangkat I/O adalah antrian perangkat. Dengan mengetahui rate kedatangan dan rate layanan, maka kita dapat mengkomputasi utilisasi, panjang antrian rata-rata, waktu tunggu rata-rata dsb. Bidang studi ini adalah analisis jaringan antrian (queueing network analys).
- Simulasi, simulasi dapat memberikan evaluasi algoritma penjadwalan dengan lebih akurat. Simulasi melibatkan pemrograman model sistem komputer. Dengan simulasi akan diperoleh statistik yang menyatakan kinerja algoritma.
- Implementasi, simulasi pun hanya memberikan akurasi yang terbatas. Satu-satunya cara paling akurat dalam mengevaluasi algoritma penjadwalan adalah mengimplementasikannya, menjalankannya pada sistem nyata dan melihatnya bekerja. Pendekatan ini adalah menjalankan algoritma nyata di sistem nyata untuk keperluan evaluasi pada beban atau kondisi operasi yang nyata.

Masing-masing cara evaluasi algoritma penjadwalan mempunyai kelebihan dan kelemahan.

6. Pertemuan 6

Sinkronisasi dan deadlock merupakan permasalahan sistem operasi. Dimana jika sinkronisasi dibutuhkan untuk menghindari terjadinya ketidak-konsistenan data akibat adanya akses data secara konkuren. Proses-proses disebut konkuren jika proses-proses itu ada dan berjalan pada waktu yang sama, proses-proses konkuren ini bisa bersifat independen atau bisa juga saling berinteraksi. Prosesproses konkuren yang saling berinteraksi memerlukan sinkronisasi agar terkendali dan juga menghasilkan output yang benar. Dan sedangkan deadlock suatu kondisi dimana proses tidak berjalan lagi atau tidak ada komunikasi lagi antar proses. Deadlock disebabkan karena proses yang satu menunggu sumber daya yang sedang dipegang oleh proses lain, proses lain itu pun sedang menunggu sumber daya yang dipegang olehnya. Dengan kata lain setiap proses dalam set menunggu untuk sumber yang hanya dapat dikerjakan oleh proses lain dalam set sedang menunggu.

1. Pengertian Sinkronisasi

Sinkronisasi merupakan suatu proses pengaturan jalannya beberapa proses pada waktu yang bersamaan untuk menyamakan waktu dan data supaya tidak terjadi inconsitensi (ketidak konsistenan) data akibat adanya akses data secara konkuren agar hasilnya bagus dan sesuai dengan apa yang diharapkan.

Sinkronisasi diperlukan untuk menghindari terjadinya ketidak-konsistenan data akibat adanya akses data secara konkuren. Proses-proses disebut konkuren jika proses-proses itu ada dan berjalan pada waktu yang sama, proses-proses konkuren ini bisa bersifat independen atau bisa juga saling berinteraksi. Proses-proses konkuren yang saling berinteraksi memerlukan sinkronisasi agar terkendali dan juga menghasilkan output yang benar.

1. Race Condition

Race condition adalah suatu kondisi dimana dua atau lebih proses mengakses shared memory/sumber daya pada saat yang bersamaan dan hasil akhir dari data tersebut tergantung dari proses mana yang terakhir selesai dieksekusi sehingga hasil akhirnya terkadang tidak sesuai dengan yang dikehendaki.

Kunci untuk mencegah masalah ini dan di situasi yang lain yang melibatkan memori bersama, berkas bersama, dan sumber daya lain yang digunakan secara bersama-sama adalah menemukan beberapa jalan untuk mencegah lebih dari satu proses melakukan proses tulis dan baca kepada data yang sama pada saat yang sama. Dengan kata lain, kita membutuhkan mutual exclusion, sebuah jalan yang menjamin jika sebuah proses sedang menggunakan variabel atau berkas yang digunakan bersama-sama, proses lain akan dikeluarkan dari pekerjaan yang sama. Cara untuk menghindari race condition adalah kita harus dapat menjamin bahwa jika suatu proses sedang menjalankan critical section, maka proses lain tidak boleh masuk ke dalam critical section tersebut.

2. Critical Section

Critical section adalah segmen kode yang mengakses data yang digunakan proses secara bersamasama yang dapat membawa proses itu ke bahaya race condition. Biasanya sebuah proses sibuk melakukan perhitungan internal dan hal-hal lainnya tanpa ada bahaya yang menuju ke race condition pada sebagian besar waktu. Akan tetapi, biasanya setiap proses memiliki segmen kode dimana proses itu dapat mengubah variabel, meng-update suatu tabel, menulis ke suatu file, dan lain-lainnya, yang dapat membawa proses itu ke bahaya race condition.

3. Prasyarat Solusi Critical Section

1. Mutual Exclusion. Mutual Exclusion merupakan sebuah jalan yang menjamin jika sebuah proses sedang menggunakan variabel atau berkas yang digunakan bersama-sama, proses lain akan dikeluarkan dari pekerjaan yang sama. Misal proses Pi sedang menjalankan critical section (dari proses Pi), maka tidak ada proses-proses lain yang dapat menjalankan critical section dari prosesproses tersebut. Dengan kata lain, tidak ada dua proses yang berada di critical section pada saat yang bersamaan.

a. do {

   i. entry section
   ii. critical section
   iii. exit section
   iv. remainder section

b. } while (1);

Setiap proses harus meminta izin untuk memasuki critical sectionnya. Bagian dari kode yang mengimplementasikan izin ini disebut entry section. Akhir dari critical section itu disebut exit section. Bagian kode selanjutnya disebut remainder section. Dari kode di atas, dapat kita lihat bahwa untuk bisa memasuki critical section sebuah proses harus melalui entry section.

2. Terjadi kemajuan (progress). Jika tidak ada proses yang sedang menjalankan critical section-nya dan jika terdapat lebih dari satu proses lain yang ingin masuk ke critical section, maka hanya prosesproses yang tidak sedang menjalankan remainder section-nya yang dapat berpartisipasi dalam memutuskan

siapa yang berikutnya yang akan masuk ke critical section, dan pemilihan siapa yang berhak masuk ke critical section ini tidak dapat ditunda secara tak terbatas (sehingga tidak terjadi deadlock).
Ada batas waktu tunggu (bounded waiting). Jika seandainya ada proses yang sedang menjalankan critical section, maka terdapat batasan waktu berapa lama suatu proses lain harus menunggu giliran untuk mengakses critical section. Dengan adanya batas waktu tunggu akan menjamin proses dapat mengakses ke critical section (tidak mengalami starvation: proses seolah-olah berhenti, menunggu request akses ke critical section diperbolehkan).

2. Deadlock

Deadlock dalam arti sebenarnya adalah kebuntuan. Kebuntuan yang dimaksud dalam sistem operasi adalah kebuntuan proses. Jadi deadlock ialah suatu kondisi dimana proses tidak berjalan lagi atau tidak ada komunikasi lagi antar proses. Deadlock disebabkan karena proses yang satu menunggu sumber daya yang sedang dipegang oleh proses lain, proses lain itu pun sedang menunggu sumber daya yang dipegang olehnya. Dengan kata lain setiap proses dalam set menunggu untuk sumber yang hanya dapat dikerjakan oleh proses lain dalam set sedang menunggu. Kejadian deadlock selalu tidak lepas dari sumber daya, bahwa hampir seluruhnya merupakan masalah sumber daya yang digunakan bersama-sama. Oleh karena itu, kita juga perlu tahu tentang jenis sumber daya, yaitu: sumber daya dapat digunakan lagi berulang-ulang dan sumber daya yang dapat digunakan dan habis dipakai atau dapat dikatakan sumber daya sekali pakai.

1. Faktor Penyebab Terjadinya Deadlock

Ada empat kondisi yang dapat menyebabkan terjadinya deadlock. Keempat kondisi tersebut tidak dapat berdiri sendiri, namun saling mendukung.

- Mutual exclusion. Hanya ada satu proses yang boleh memakai sumber daya, dan proses lain yang ingin memakai sumber daya tersebut harus menunggu hingga sumber daya tadi dilepaskan atau tidak ada proses yang memakai sumber daya tersebut.
- Hold and wait. Proses yang sedang memakai sumber daya boleh meminta sumber daya lagi maksudnya menunggu hingga benar-benar sumber daya yang diminta tidak dipakai oleh proses lain, hal ini dapat menyebabkan kelaparan sumber daya sebab dapat saja sebuah proses tidak mendapat sumber daya dalam waktu yang lama.
- No preemption. Sumber daya yang ada pada sebuah proses tidak boleh diambil begitu saja oleh proses lainnya. Untuk mendapatkan sumber daya tersebut, maka harus dilepaskan terlebih dahulu oleh proses yang memegangnya, selain itu seluruh proses menunggu dan mempersilahkan hanya proses yang memiliki sumber daya yang boleh berjalan.
- Circular wait. Kondisi seperti rantai, yaitu sebuah proses membutuhkan sumber daya yang dipegang proses berikutnya.

Ketiga kondisi pertama merupakan syarat perlu (necessary conditions) bagi terjadinya deadlock. Keberadaan deadlock selalu berarti terpenuhi kondisi-kondisi diatas, tak mungkin terjadi deadlock bila tidak ada ketiga kondisi itu. Deadlock terjadi berarti terdapat ketiga kondisi itu, tetapi adanya ketiga kondisi itu belum berarti terjadi deadlock. Deadlock baru benar-benar terjadi bila kondisi keempat terpenuhi. Kondisi keempat merupakan keharusan bagi terjadinya peristiwa deadlock. Bila salah satu saja dari kondisi tidak terpenuhi maka deadlock tidak terjadi.

2. Metode Menangani Deadlock

Terdapat tiga metode untuk menangani permasalahan deadlock yaitu :

1. Menggunakan protocol untuk menjamin bahwa sistem tidak pernah memasuki status deadlock.
2. Mengijinkan sistem memasuki status deadlock dan kemudian memperbaikinya.
3. Mengabaikan permasalahan dan seakan-akan deadlock tidak pernah terjadi pada sistem. Model ini yang banyak digunakan pada sistem operasi termasuk UNIX.

- **Strategi Burung Onta**

Strategi ini mengasumsikan kejadian deadlock jarang terjadi, sehingga mengabaikan kemungkinan deadlock. Jika terjadi deadlock, maka reboot sistem. Strategi ini berarti sama sekali tidak berusaha mengatasi deadlock.

- **Mencegah Deadlock (Deadlock Prevention)**

Metode ini berkaitan dengan pengkondisian sistem agar menghilangkan kemungkinan terjadinya deadlock. Pencegahan merupakan solusi yang bersih dipandang dari sudut tercegahnya deadlock. Metode ini sering menghasilkan utilisasi sumber daya yang buruk. Pencegahan deadlock merupakan

metode yang banyak dipakai. Untuk mencegah deadlock dilakukan dengan meniadakan salah satu dari syarat perlu sebagai berikut :

a. Mencegah Mutual Exclusion

Mutual exclusion benar-benar tak dapat dihindari. Hal ini dikarenakan tidak ada sumber daya yang dapat digunakan bersama-sama, jadi sistem harus membawa sumber daya yang tidak dapat digunakan bersamasama.

b. Mencegah Hold and Wait

Untuk mencegah hold and wait, sistem harus menjamin bila suatu proses meminta sumber daya, maka proses tersebut tidak sedang memegang sumber daya yang lain. Proses harus meminta dan dialokasikan semua sumber daya yang diperlukan sebelum proses memulai eksekusi atau mengijinkan proses meminta sumber daya hanya jika proses tidak membawa sumber daya lain. Model ini mempunyai utilitas sumber daya yang rendah dan kemungkinan terjadi starvation jika proses membutuhkan sumber daya yang popular sehingga terjadi keadaan menunggu yang tidak terbatas karena setidaknya satu dari sumber daya yang dibutuhkannya dialokasikan untuk proses yang lain.

c. Mencegah Non Preemption

Peniadaan non preemption mencegah proses-proses lain harus menunggu. Seluruh proses menjadi preemption, sehingga tidak ada tunggu menunggu. Cara mencegah kondisi non preemption :

- Jika suatu proses yang membawa beberapa sumber daya meminta sumber daya lain yang tidak dapat segera dipenuhi untuk dialokasikan pada proses tersebut, maka semua sumber daya yang sedang dibawa proses tersebut harus dibebaskan.
- Proses yang sedang dalam keadaan menunggu, sumber daya yang dibawanya ditunda dan ditambahkan pada daftar sumber daya.
- Proses akan di-restart hanya jika dapat memperoleh sumber daya yang lama dan sumber daya baru yang diminta.

d. Mencegah Kondisi Menunggu Sirkular

Sistem mempunyai total permintaan global untuk semua tipe sumber daya. Proses dapat meminta proses kapanpun menginginkan, tapi permintaan harus dibuat terurut secara numerik. Setiap proses yang membutuhkan sumber daya dan memintanya maka nomor urut akan dinaikkan. Cara ini tidak akan menimbulkan siklus. Masalah yang timbul adalah tidak ada cara pengurutan nomor sumber daya yang memuaskan semua pihak.

2.2.3Menghindari Deadlock (Deadlock Avoidance)

Metode alternatif untuk menghindari deadlock adalah digunakan informasi tambahan tentang bagaimana sumber daya diminta. Misalnya pada sistem dengan satu tape drive dan satu printer, proses

P pertama meminta tape drive dan kemudian printer sebelum melepaskan kedua sumber daya tersebut. Sebaliknya proses Q pertama meminta printer kemudian tape drive. Dengan mengetahui urutan permintaan dan pelepasan sumber daya untuk setiap proses, dapat diputuskan bahwa untuk setiap permintaan apakah proses harus menunggu atau tidak. Setiap permintaan ke sistem harus dipertimbangkan apakah sumber daya tersedia, sumber daya sedang dialokasikan untuk proses dan permintaan kemudian serta pelepasan oleh proses untuk menentukan apakah permintaan dapat dipenuhi atau harus menunggu untuk menghindari deadlock. Model yang sederhana dan sangat penting dibutuhkan adalah setiap proses menentukan jumlah maksimum sumber daya dari setiap tipe yang mungkin diperlukan. Algoritma deadlock avoidance secara dinamis memeriksa status sumber daya yang dialokasikan untuk menjamin tidak pernah terjadi kondisi menunggu sirkular. Status alokasi sumber daya ditentukan oleh jumlah sumber daya yang tersedia dan yang dialokasikan dan maksimum permintaan oleh prosesproses.

- **Mendeteksi Deadlock**

Jika sistem tidak menyediakan algoritma mencegah deadlock dan menghindari deadlock, maka terjadi deadlock. Pada lingkungan ini sistem harus menyediakan :

- Algoritma yang menguji state sistem untuk menentukan apakah deadlock telah terjadi.
- Algoritma untuk memperbaiki dari deadlock.

- **Pemulihan Deadlock**

Terdapat dua pilihan untuk membebaskan deadlock. Satu solusi sederhana adalah dengan menghentikan satu atau beberapa proses untuk membebaskan kondisi menunggu sirkular. Pilihan kedua adalah menunda beberapa sumber daya dari satu atau lebih proses yang deadlock.

7. **Pertemuan ketujuh**

Manajemen Memori

- **Konsep Dasar Memori**

Memori adalah pusat dari operasi pada sistem komputer modern, berfungsi sebagai tempat penyimpanan informasi yang harus diatur dan dijaga sebaik-baiknya. Memori adalah array besar dari word atau byte, yang disebut alamat. CPU mengambil instruksi dari memory berdasarkan nilai dari program counter.

Sedangkan manajemen memori adalah suatu kegiatan untuk mengelola memori komputer. Proses ini menyediakan cara mengalokasikan memori untuk proses atas permintaan mereka, membebaskan untuk digunakan kembali ketika tidak lagi diperlukan serta menjaga alokasi ruang memori bagi proses. Pengelolaan memori utama sangat penting untuk sistem komputer, penting untuk memproses dan fasilitas masukan/keluaran secara efisien, sehingga memori dapat menampung sebanyak mungkin proses dan sebagai upaya agar pemogram atau proses tidak dibatasi kapasitas memori fisik di sistem komputer (Eko, 2009).

Memory manager merupakan salah satu bagian sistem operasi yang mempengaruhi dalam menentukan proses mana yang diletakkan pada antrian.

Jenis Memori

a) Memori Kerja
b) ROM/PROM/EPROM/EEPROM
c) RAM
d) Cache memory
e) Memori Dukung
f) Floppy, harddisk, CD, dll.

Alamat Memori

a) Alamat memori mutlak (alamat fisik)
b) Alamat memori relatif (alamat logika)
c) Hubungan antara alamat multak dan alamat relatif
d) Jenis memori dan alamat memori

Isi Memori

a) Sistem bahasa penataolahan
b) Sistem Utilitas
c) Inti Sistem Operasi
d) Sistem Operasi
e) Pengendali alat (device drivers)

f) File pemakai

Fungsi Manajemen Memori

a) Mengelola informasi yang dipakai dan tidak dipakai.

b) Mengalokasikan memori ke proses yang memerlukan.

c) Mendealokasikan memori dari proses telah selesai.

d) Mengelola swapping atau paging antara memori utama dan disk.

Sistem operasi memberikan tanggapan terhadap manajemen memori utama untuk aktivitas-aktivitas sebagai berikut:

a) Menjaga dan memelihara bagian-bagian memori yang sedang digunakan dan dari yang menggunakan.

b) Memutuskan proses-proses mana saja yang harus dipanggil kememori jika masih ada ruang di memori.

c) Mengalokasikan dan mendelokasikan ruang memori jika diperlukan

Jenis-Jenis Manajemen Memory

Manajemen Memory Untuk Monoprogramming

Bila program komputer yang dijalankan hanya satu jenis selama proses berlangsung maka dikatakan mode kerja komputer itu adalah monoprogramming. Selama komputer itu bekerja maka memory RAM seluruhnya di kuasai oleh program tersebut. Jadi RAM tidak dapat di masuki oleh program lain. Mode serupa ini di temui pada komputer berbasis DOS.

Penempatan program di memory diatur sedemikain rupa sehingga (Eko, 2009) :

a) BIOS selalu di ROM (BIOS)

b) Sistem Operasi di RAM bawah (alamat rendah)

c) Program Aplikasi di RAM tengah (alamat sesudah OS terakhir)

d) Data Sementara di RAM atas (alamat sesudah Aplikasi terakhi).

Bila sistem operasi telah selasai dimuat maka tampillah prompt di layar monitor, dan itu adalah tanda bahwa komputer siap menerima program aplikasi. Letakkan disk yang berisi program aplikasi pada diskdrive yang aktif lalu eksekusi , sehingga program itu termuat seluruhnya ke RAM. Dengan demikian program aplikasi siap digunakan menurut semestinya.Kita lihat ketika komputer mula-mula dinyalakan maka proses yang dibaca pertama kali adalah apa yang tertulis di dalam ROM. Setelah semua perintah di adalam ROM BIOS selesai dibaca maka komputer meminta kita memasukkan DOS ke dalam RAM-nya.Ketika DOS dibaca maka diletakkan sebagian dari program DOS yang terpenting saja ke dalam RAM, seperti : COMMAND.COM dan INTERNAL COMMAND. Sedangkan program

DOS yang lain masih tetap di dalam disk dan apabila kita perlukan dapat di eksekusi. Hal itu berguna untuk mrnjaga agar RAM tidak penuh oleh Sistem Operasi saja.

Ketika kita bekerja dengan program aplikasi tasdi maka kita akan menghasilkan data. Data itu akan di simpan sementara di RAM yang masih tersisa. Data yang disimpan di RAM bersifat voletile, artinya data hanya bisa bertahan selama catudaya komputer masih ON. Untuk berjaga-jaga biasakan menyimpan data ke disk dalam jangka waktu yang tidak terlalu lama, misalnya setiap 5 menit sekali. Selain menjaga data agar tidak amblas menyimpan ke disk bertujuan juga untuk mengosongkan RAM agar tidak cepat penuh.

Didalam sistem juga dapat kita lihat bahwa sistem operasi terletak berdekatan dengan program lain di RAM sehingga kemungkinan sistem operasi ter ganggu atau terubah oleh proses yang sedang berjalan sangat besar .Hal itu tidak boleh terjadi.Untuk mencegah terganggu sitem operasi tersebut maka alamat tertinggi dari sistem operasi dletakkan pada register batas dalam CPU. Jika ada proses yang mengacu ke alamat itu atau yang lebih rendah dari itu maka proses di hentikan dan program akan menampilkan pesan kesalahan.

Manajemen Memory Untuk Multiprogramming

Untuk sistem komputer yang berukuran besar (bukan small computers), membutuhkan pengaturan memori, karena dalam multiprogramming akan melibatkan banyak pemakai secara simultan sehingga di memori akan terdapat lebih dari satu proses bersamaan. Oleh karena itu dibutuhkan sistem operasi yang mampu mendukung dua kebutuhan tersebut, meskipun hal tersebut saling bertentangan, yaitu (Ama, 2003) :

a) Pemisahan ruang-ruang alamat.

b) Pemakaian bersama memori.

Manajer memori harus memaksakan isolasi ruang-ruang alamat tiap proses agar mencegah proses aktif atau proses yang ingin berlaku jahat mengakses dan merusak ruang alamat proses lain. Manajer memori di lingkungan multiprogramming sekalipun melakukan dua hal, yaitu :

a) Proteksi memori dengan isolasi ruang-ruang alamat secara dis-joint.

b) Pemakaian bersama memori.

Memungkinkan proses-proses bekerja sama mengakses daerah memori bersama. Ketika konsep multiprogramming digunakan, pemakaian CPU dapat ditingkatkan. Sebuah model untuk mengamati pemakaian CPU secara probabilistic :

CPU utilization = 1 – p n

Dengan :

a) N menunjukkan banyaknya proses pada suatu saat, sehingga kemungkinan bahwa semua n proses akan menunggu menggunakan I/O (masalah CPU menganggur) adalah sebesar pn. Fungsi dari n disebut sebagai degree of multiprogramming.

b) P menunjukkan besarnya waktu yang digunakan sebuah proses

B. Strategi Manajemen Memori

Strategi yang dikenal untuk mengatasi hal tersebut adalah memori maya. Memori maya menyebabkan sistem seolah-olah memiliki banyak memori dibandingkan dengan keadaan memori fisik yang sebenarnya. Memori maya tidak saja memberikan peningkatan komputasi, akan tetapi memori maya juga memiliki bberapa keuntungan seperti :

Large Address Space

Membuat sistem operasi seakan-akan memiliki jumlah memori melebihi kapasitas memori fisik yang ada. Dalam hal ini memori maya memiliki ukuran yang lebih besar daripada ukuran memori fisik.

Proteksi.

Setiap proses di dalam sistem memiliki virtual address space. Virtual address space tiap proses berbeda dengan proses yang lainnya lagi, sehingga apapun yang terjadi pada sebuah proses tidak akan berpengaruh secara langsung pada proses lainnya

Memory Mapping

Memory mapping digunakan untuk melakukan pemetaan image dan file-file data ke dalam alamat proses. Pada pemetaan memori, isi dari file akan di link secara langsung ke dalam virtual address space dari proses.

Fair Physical Memory Allocation

Digunakan oleh Manajemen Memori untuk membagi penggunaan memori fisik secara “adil” ke setiap proses yang berjalan pada sistem.

Shared Virtual Memory.

Meskipun tiap proses menggunakan address space yang berbeda dari memori maya, ada kalanya sebuah proses dihadapkan untuk saling berbagi penggunaan memori.

C. Ruang Alamat Logika dan Fisik

Alamat Logika adalah alamat yg dibentuk di CPU, disebut juga alamat virtual. Alamat fisik adalah alamat yang terlihat oleh memori. Untuk mengubah dari alamat logika ke alamat fisik diperlukan

suatu perangkat keras yang bernama MMU (Memory Management Unit). Pengubahan dari alamat logika ke alamat fisik adalah pusat dari manajemen memori. Alamat yang dibangkitkan oleh CPU disebut alamat logika (logical address) dimana alamat terlihat sebagai uni memory yang disebut alamat fisik (physical address). Tujuan utama manajemen memori adalah konsep meletakkan ruang alamat logika ke ruang alamat fisik (Ama, 2003).

Hasil skema waktu kompilasi dan waktu pengikatan alamat pada alamat logika dan alamat memori adalah sama. Tetapi hasil skema waktu pengikatan alamat waktu eksekusi berbeda. dalam hal ini, alamat logika disebut dengan alamat maya (virtual address). Himpunan dari semua alamat logika yang dibangkitkan oleh program disebut dengan ruang alamat logika (logical address space); himpunan dari semua alamat fisik yang berhubungan dengan alamat logika disebut dengan ruang alamat fisik (physical address space).

Memory Manajement Unit (MMU) adalah perangkat keras yang memetakan alamat virtual ke alamat fisik. Pada skema MMU, nilai register relokasi ditambahkan ke setiap alamat yang dibangkitkan oleh proses user pada waktu dikirim ke memori.

Register basis disebut register relokasi. Nilai dari register relokasi ditambahkan ke setiap alamat yang dibangkitkan oleh proses user pada waktu dikirim ke memori, sebagai contoh, apabila basis 14000, maka user mencoba menempatkan ke alamat lokasi 0 dan secara dinamis direlokasi ke lokasi 14000. Pengaksesan ke lokasi logika 346, maka akan dipetakan ke lokasi 14346. Sistem operasi MS-DOS yang masih keluarga intel 80X86 menggunakan empat register relokasi ketika proses loading dan running.

User program tidak pernah melihat alamat fisik secara real. Program dapat membuat sebuah penunjuk ke lokasi 346, mengirimkan ke memory, memanipulasinya, membandingkan dengan alamat lain, semua menggunakan alamat 346. Hanya ketika digunakan sebagai alamat memory akan direlokasi secara relatif ke register basis.

D. Swapping

Sebuah proses, sebagaimana telah diterangkan di atas, harus berada di memori sebelum dieksekusi. Proses swapping menukarkan sebuah proses keluar dari memori untuk sementara waktu ke sebuah penyimpanan sementara dengan sebuah proses lain yang sedang membutuhkan sejumlah alokasi memori untuk dieksekusi. Tempat penyimpanan sementara ini biasanya berupa sebuah fast disk dengan kapasitas yang dapat menampung semua salinan dari semua gambaran memori serta menyediakan akses langsung ke gambaran tersebut. Jika eksekusi proses yang dikeluarkan tadi akan dilanjutkan beberapa saat kemudian, maka ia akan dibawa kembali ke memori dari tempat penyimpanan sementara tadi. Bagaimana sistem mengetahui proses mana saja yang akan dieksekusi? Hal ini dapat dilakukan dengan

ready queue. Ready queue berisikan semua proses yang terletak baik di penyimpanan sementara maupun memori yang siap untuk dieksekusi. Ketika penjadwal CPU akan mengeksekusi sebuah proses, ia lalu memeriksa apakah proses bersangkutan sudah ada di memori ataukah masih berada dalam penyimpanan sementara. Jika proses tersebut belum berada di memori maka proses swapping akan dilakukan seperti yang telah dijelaskan di atas.

Sebuah contoh untuk menggambarkan teknik swapping ini adalah sebagai berikut: Algoritma Round-Robin yang digunakan pada multiprogramming environment menggunakan waktu kuantum (satuan waktu CPU) dalam pengeksekusian proses-prosesnya. Ketika waktu kuantum berakhir, memory manager akan mengeluarkan (swap out) proses yang telah selesai menjalani waktu kuantumnya pada suatu saat serta memasukkan (swap in) proses lain ke dalam memori yang telah bebas tersebut. Pada saat yang bersamaan penjadwal CPU akan mengalokasikan waktu untuk proses lain dalam memori. Hal yang menjadi perhatian adalah, waktu kuantum harus cukup lama sehingga waktu penggunaan CPU dapat lebih optimal jika dibandingkan dengan proses penukaran yang terjadi antara memori dan disk.

Teknik swapping roll out, roll in menggunakan algoritma berbasis prioritas dimana ketika proses dengan prioritas lebih tinggi tiba maka memory manager akan mengeluarkan proses dengan prioritas yang lebih rendah serta me-load proses dengan prioritas yang lebih tinggi tersebut. Saat proses dengan prioritas yang lebih tinggi telah selesai dieksekusi maka proses yang memiliki prioritas lebih rendah dapat dimasukkan kembali ke dalam memori dan kembali dieksekusi.

Sebagian besar waktu swapping adalah waktu transfer. Sebagai contoh kita lihat ilustrasi berikut ini: sebuah proses pengguna memiliki ukuran 5 MB, sedangkan tempat penyimpanan sementara yang berupa harddisk memiliki kecepatan transfer data sebesar 20 MB per detiknya. Maka waktu yang dibutuhkan untuk mentransfer proses sebesar 5 MB tersebut dari atau ke dalam memori adalah sebesar 5000 KB / 20000 KBps = 250 ms.

Perhitungan di atas belum termasuk waktu latensi, sehingga jika kita asumsikan waktu latensi sebesar 2 ms maka waktu swap adalah sebesar 252 ms. Oleh karena terdapat dua kejadian dimana satu adalah proses pengeluaran sebuah proses dan satu lagi adalah proses pemasukan proses ke dalam memori, maka total waktu swap menjadi 252 + 252 = 504 ms.

Agar teknik swapping dapat lebih efisien, sebaiknya proses-proses yang di- swap hanyalah proses-proses yang benar-benar dibutuhkan sehingga dapat mengurangi waktu swap. Oleh karena itulah, sistem harus selalu mengetahui perubahan apapun yang terjadi pada pemenuhan kebutuhan terhadap memori. Disinilah sebuah proses memerlukan fungsi system call, yaitu untuk memberitahukan sistem operasi kapan ia meminta memori dan kapan membebaskan ruang memori tersebut.

Jika kita hendak melakukan swap, ada beberapa hal yang harus diperhatikan. Kita harus menghindari menukar proses dengan M/K yang ditunda (asumsinya operasi M/K tersebut juga sedang mengantri di antrian karena peralatan M/Knya sedang sibuk). Contohnya seperti ini, jika proses P1dikeluarkan dari memori dan kita hendak memasukkan proses P2, maka operasi M/K yang juga berada di antrian akan mengambil jatah ruang memori yang dibebaskan P1 tersebut. Masalah ini dapat diatasi jika kita tidak melakukan swap dengan operasi M/K yang ditunda. Selain itu, pengeksekusian operasi M/K hendaknya dilakukan pada buffer sistem operasi.

Tiap sistem operasi memiliki versi masing-masing pada teknik swapping yang digunakannya. Sebagai contoh pada UNIX, swapping pada dasarnya tidak diaktifkan, namun akan dimulai jika banyak proses yang membutuhkan alokasi memori yang banyak. Swapping akan dinonaktifkan kembali jika jumlah proses yang dimasukkan berkurang. Pada sistem operasi Microsoft Windows 3.1, jika sebuah proses baru dimasukkan dan ternyata tidak ada cukup ruang di memori untuk menampungnya, proses yang lebih dulu ada di memori akan dipindahkan ke disk. Sistem operasi ini pada dasarnya tidak menerapkan teknik swapping secara penuh, hal ini disebabkan pengguna lebih berperan dalam menentukan proses mana yang akan ditukar daripada penjadwal CPU. Dengan ketentuan seperti ini proses-proses yang telah dikeluarkan tidak akan kembali lagi ke memori hingga pengguna memilih proses tersebut untuk dijalankan.

Manajemen Memori Berdasarkan Keberadaan Swapping Atau Paging

a) Manajemen tanpa swapping atau paging

b) Manajemen dengan swapping atau paging

Memori Tanpa Swapping Or Paging

Merupakan manajemen memori tanpa pemindahan citra proses antara memori utama dan disk selama eksekusi. Manajemen ini terdiri dari :

Monoprogramming

Ciri-ciri :

a) Hanya satu proses pada satu saat

b) Hanya satu proses menggunakan semua memori

c) Pemakai memuatkan program ke seluruh memori dari disk atau tape

d) Program mengambil kendali seluruh mesin

Multiprogramming Dengan Pemartisian Statis

a) Pemartisian menjadi partisi-partisi berukuran sama, yaitu ukuran semua partisi memori adalah sama

b) Pemartisian menjadi partisi-partisi berukuran berbeda, yaitu ukuran semua partisi memori adalah berbeda.

Strategi Penempatan Program Ke Partisi

Satu Antrian Tunggal Untuk Semua Partisi Keuntungan : Lebih fleksibel serta implementasi dan operasi lebih minimal karena hanya mengelola satu antrian. Kelemahan : Proses dapat ditempatkan di partisi yang banyak diboroskan, yaitu proses kecil ditempatkan di partisi sangat besar.

Tetap dengan Satu Antrian Satu Antrian Untuk Tiap Partisi (banyak antrian Untuk Seluruh Partisi). Keuntungan : Meminimalkan pemborosan memori. Kelemahan : Dapat terjadi antrian panjang di suatu partisi sementara antrian partisi – partisi lain kosong.

Multiprogramming Dengan Swapping

Merupakan manajemen memori dengan pemindahan citra proses antara memori utama dan disk selama eksekusi, atau dengan kata lain merupakan manajemen pemindahan proses dari memori utama ke disk dan kembali lagi (swapping). Manajemen ini terdiri dari :

Multiprogramming Dengan Pemartisisan Dinamis

Jumlah, lokasi dan ukuran proses di memori dapat beragam sepanjang waktu secara dinamis. Kelemahan: a) Dapat terjadi lubang-lubang kecil memori di antara partisi-partisi yang dipakai; b) Merumitkan alokasi dan dealokasi memori.

Solusi:

Lubang-lubang kecil di antara blok-blok memori yang digunakan dapat diatasi dengan pemadatan memori yaitu menggabungkan semua lubang kecil menjadi satu lubang besar dengan memindahkan semua proses agar saling berdekatan.

Strategi Alokasi Memori

a) First fit algorithm : memory manager men-scan list untuk menemukan hole yg cukup untuk menampung proses yg baru. Proses akan menempati hole pertama yg ditemuinya yg cukup untuk dirinya.

b) Next fit algorithm : sama dengan first fit, tetapi pencarian hole dimulai dari hole ditemuinya dari scan sebelumnya.

c) Best fit algorithm : dicari hole yang akan menghasilkan sisa paling sedikit setelah dimasuki proses.

d) Worst fit algorithm : kebalikan dari best fit.

e) Quick fit algorithm : mengelompokkan hole-hole dan membuat listnya sendiri. Misalnya, ada list untuk hole 4K, satu list untuk 8K, dst.

f) Sistem Buddy : Memori di susun dalm senari blok-blok bebas berukuran 1,2,4,8,16 byte dst, sampai kapasitas memori.

Dari berbagai cara alokasi tsb. Di atas, sebuah hole yg ditempati proses akan terbagi menjadi bagian yang dipakai proses dan memori yang tidak terpakai (fragmen). Timbulnya memori yang tidak terpakai disebut fragmentasi. Ada dua macam fragmen :

a) Internal : sisa hole yang tidak terpakai setelah terisi proses.

b) Eksternal : hole yang secara utuh terlalu kecil untuk dipakai oleh proses manapun.

Alokasi Ruang Swap pada Disk (Penempatan proses pada disk setelah di-swap-out dari memori)

a) Ruang disk tempat swap dialokasikan begitu diperlukan

b) Ruang disk tempat swap dialokasikan lebih dahulu.

Algoritma untuk pengaturan ruang swap pada disk sama dengan untuk memori utama. Perbedaannya adalah ruang pada disk harus dialokasikan sebagai kelipatan bilangan bulat dari disk block.

8. Pertemuan kedelapan

UTS

9. Pertemuan 9

VIRTUAL MEMORY

Selama bertahun-tahun, pelaksanaan manajemen memori pada intinya adalah dengan menempatkan semua bagian proses yang akan dijalankan ke dalam memori sebelum proses dapat mulai dieksekusi. Dengan demikian semua bagian proses tersebut harus memiliki alokasi sendiri di dalam memori fisik.
Pada kenyataannya tidak semua bagian dari program tersebut akan diproses, misalnya:

- Ada pernyataan-pernyataan atau pilihan yang hanya akan dieksekusi jika kondisi tertentu dipenuhi
- Terdapat fungsi-fungsi yang jarang digunakan
- Pengalokasian memori yang lebih besar dari yang sebenarnya dibutuhkan.

Pada memori berkapasitas besar, hal-hal ini tidak akan menjadi masalah. Namun pada memori dengan kapasitas yang sangat terbatas, hal ini akan menurunkan optimalisasi utilitas dari ruang memori fisik (memori utama).
Setiap program yang dijalankan harus berada di memori. Memori merupakan suatu tempat penyimpanan utama (*primary storage*) yang bersifat sementara (*volatile*). Ukuran memori yang terbatas dapat menimbulkan masalah bagaimana menempatkan program yang berukuran yang lebih besar dari ukuran memori fisik (memori utama) dan masalah penerapan *multiprogramming* yang membutuhkan tempat yang lebih besar di memori.
Memori virtual adalah suatu teknik yang memisahkan antara memori logis dan memori fisiknya. Memori logis merupakan kumpulan keseluruhan halaman dari suatu program. Tanpa memori virtual, memori logis akan langsung dibawa ke memori fisik (memori utama). Disinilah memori virtual melakukan pemisahan dengan menaruh memori logis ke *secondary storage* (disk sekunder) dan hanya membawa halaman yang diperlukan ke memori utama (memori fisik). Teknik ini menempatkan keseluruhan program di disk sekunder dan membawa halaman-halaman yang diperlukan ke memori fisik sehingga memori utama hanya akan menyimpan sebagian alamat proses yang sering digunakan dan sebagian lainnya akan disimpan dalam disk sekunder dan dapat diambil sesuai dengan kebutuhan. Jadi jika proses yang sedang berjalan membutuhkan instruksi atau data yang terdapat pada suatu halaman tertentu maka halaman tersebut akan dicari di memori utama. Jika halaman yang diinginkan tidak ada maka akan dicari ke disk sekunder.

Pada gambar diatas ditunjukkan ruang sebuah memori virtual yang dibagi menjadi bagian-bagian yang sama dan diidentifikasikan dengan nomor *virtual pages*. Memori fisik dibagi menjadi *page frames* yang berukuran sama dan diidentifikasikan dengan nomor *page frames*. Bingkai (*frame*) menyimpan data dari halaman. Atau memori virtual memetakan nomor *virtual pages* ke nomor *page frames*. *Mapping* (pemetaan) menyebabkan halaman virtual hanya dapat mempunyai satu lokasi alamat fisik.
Dalam sistem *paging*, jika sebuah ruang diperlukan untuk proses dan halaman yang bersangkutan tidak sedang digunakan, maka halaman dari proses akan mengalami *paged out* (disimpan ke dalam disk) atau *swap out*, memori akan kosong untuk halaman aktif yang lain. Halaman yang dipindah dari disk ke memori ketika diperlukan dinamakan *paged in* (dikembalikan ke memori) atau *swap in*. Ketika sebuah item dapat mengalami *paging*, maka item tersebut termasuk dalam item yang menempati ruang virtual, yang diakses dengan alamat virtual dan ruangan yang ada dialokasikan untuk informasi pemetaan. Sistem operasi mengalokasikan alamat dari item tersebut hanya ketika item tersebut mengalami *paging in*.
Keuntungan yang diperoleh dari penyimpanan hanya sebagian program saja pada memori fisik adalah:

- Berkurangnya proses M/K yang dibutuhkan (lalu lintas M/K menjadi rendah)
- Ruang menjadi lebih leluasa karena berkurangnya memori fisik yang digunakan
- Meningkatnya respon karena menurunnya beban M/K dan memori
- Bertambahnya jumlah pengguna yang dapat dilayani. Ruang memori yang masih tersedia luas memungkinkan komputer untuk menerima lebih banyak permintaan dari pengguna.

Teknik memori virtual akan memudahkan pekerjaan seorang programmer ketika besar data dan programnya melampaui kapasitas memori utama. Sebuah *multiprogramming* dapat mengimplementasikan teknik memori virtual sehingga sistem *multiprogramming* menjadi lebih efisien. Contohnya: 10 program dengan ukuran 2 MB dapat berjalan di memori berkapasitas 4 MB. Tiap program dialokasikan 256 Kbyte dan bagian – bagian proses (*swap in*) masuk ke dalam memori fisik begitu diperlukan dan akan keluar (*swap out*) jika sedang tidak diperlukan.
Prinsip dari memori virtual adalah bahwa "Kecepatan maksimum ekseskusi proses di memori virtual dapat sama, tetapi tidak akan pernah melampaui kecepatan eksekusi proses yang sama di sistem yang tidak menggunakan memori virtual".
Memori virtual dapat diimplementasikan dengan dua cara:

1. *Demand Paging* yaitu dengan menerapkan konsep pemberian halaman pada proses
2. *Demand segmentation*, lebih kompleks diterapkan ukuran segmen yang bervariasi.

- **Virtual Memory Di Windows**

Pada komputer kita, jumlah memory yang tersedia adalah jumlah antara memory fisik/RAM dengan virtual memory. Virtual memory adalah sebuah porsi pada hard disk yang di-set menyerupai RAM oleh system. Virtual memory merupakan ruang penyimpanan sementara yang digunakan untuk menjalankan program yang membutuhkan memory yang lebih besar dari memory fisik.
Virtual memory berupa file yang bernama pagefile.sys yang di-set hidden oleh Windows. File ini disebut paging file, yang digunakan untuk menampung program dan data yang tidak cukup di memory fisik. Virtual memory lebih lambat daripada memory fisik, dan penggunaan yang terlalu banyak dapat menurunkan kinerja sistem. Sehubungan dengan itu, windows memindahkan proses yang tidak terlalu sering ke virtual memory, dan membiarkan proses yang sering digunakan di memory fisik. Jadi ini sangat efisien.

Ukuran dari virtual memory dapat kita rubah, Windows merekomendasikan ukuran minimal dari vitual memory adalah 1.5 kali dari memory fisik kita. Jika anda memiliki beberapa harddisk, misal hardisk pertama adalah C: dan harddisk kedua adalah D: dan anda jarang menggunakan drive D:, anda dapat memindahkan virtual memory ke drive D:. Memindahkan virtual memory ke harddisk yang jarang digunakan akan sedikit meningkatkan performa. Alasannya adalah, pada harddisk pertama biasanya head dari harddisk sangat sibuk untuk membuka program, dokumen, menyimpan file dan masih banyak lagi. Tetapi ingat, cara ini tidak akan berguna bila drivenya terletak pada harddisk yang sama atau dengan kata lain sebuah partisi.

Cara Kerja

Virtual Memory digunakan dengan membuat suatu file khusus yang disebut swapfile atau paging file. Virtual memory digunakan pada saat operating system kehabisan memory, dimana o.s. akan memindahkan data yang paling terakhir diakses ke dalam swapfile di hardisk. Hal ini mengosongkan/ membebaskan beberapa ruang kosong pada memory untuk aplikasi yang akan digunakan selanjutnya. Operating system akan melakukan hal ini secara terus menerus ketika data baru diisi pada ram. Kemudian, pada saat data yang tersimpan di swapfile diperlukan, data tersebut ditukar (swap) dengan data yang paling terakhir dipakai di dalam memory (ram). Hal ini mengakibatkan swapfile bersifat

seperti ram, walaupun program tidak dapat secara langsung dijalankan dari swapfile. Satu hal yang perlu dicatat bahwa karena operating system tidak dapat secara langsung menjalankan program dari swapfile, beberapa program mungkin tidak akan berjalan walau dengan swapfile yang besar jika kita hanya memiliki ram yang kecil.

- **Virtual Memory Di Linux**

Managemen Memori di Linux

1. Managemen Memori Fisik

Memori managemen merupakan salah satu bagian terpenting dalam sistem operasi. Karena adanya keterbatasan memori, diperlukan suatu strategi dalam menangani masalah ini. Jalan keluarnya adalah dengan menggunakan memori virtual. Dengan memori virtual, memori tampak lebih besar daripada ukuran yang sebenarnya.
Dengan memori virtual kita dapat:
1. Ruang alamat yang besar
Sistem operasi membuat memori terlihat lebih besar daripada ukuran memori sebenarnya. Memori virtual bisa beberapa kali lebih besar daripada memori fisiknya.
2. Pembagian memori fisik yang adil
Managemen memori membuat pembagian yang adil dalam pengalokasian memori antara proses-proses.
3. Perlindungan
Memori managemen menjamin setiap proses dalam sistem terlindung dari proses-proses lainnya. Dengan demikian, program yang crash tidak akan mempengaruhi proses lain dalam sistem tersebut.
4. Penggunaan memori virtual bersama
Memori virtual mengizinkan dua buah proses berbagi memori diantara keduanya, contohnya dalam shared library. Kode library dapat berada di satu tempat, dan tidak dikopi pada dua program yang berbeda.

MemoriVirtual

Memori fisik dan memori virtual dibagi menjadi bagian-bagian yang disebut page. Page ini memiliki ukuran yang sama besar. Tiap page ini punya nomor yang unik, yaitu Page Frame Number (PFN). Untuk setiap instruksi dalam program, CPU melakukan mapping dari alamat virtual ke memori fisik yang sebenarnya.
Penerjemahan alamat di antara virtual dan memori fisik dilakukan oleh CPU menggunakan tabel page untuk proses x dan proses y. Ini menunjukkan virtial PFN 0 dari proses x dimap ke memori fisik PFN
1. Setiap anggota tabel page mengandung informasi berikut ini:
1. Virtual PFN
2. PFN fisik
3. informasi akses page dari page tersebut

Untuk menerjemahkan alamat virtual ke alamat fisik, pertama-tama CPU harus menangani alamat virtual PFN dan offsetnya di virtual page. CPU mencari tabel page proses dan mancari anggota yang sesuai degan virtual PFN. Ini memberikan PFN fisik yang dicari. CPU kemudian mengambil PFN fisik dan mengalikannya dengan besar page untuk mendapat alamat basis page tersebut di dalam memori fisik. Terakhir, CPU menambahkan offset ke instruksi atau data yang dibutuhkan. Dengan cara ini, memori virtual dapat dimap ke page fisik dengan urutan yang teracak.

Demand Paging

Cara untuk menghemat memori fisik adalah dengan hanya meload page virtual yang sedang digunakan oleh program yang sedang dieksekusi. Tehnik dimana hanya meload page virtual ke memori hanya ketika program dijalankan disebut demand paging.
Ketika proses mencoba mengakses alamat virtual yang tidak ada di dalam memori, CPU tidak dapat menemukan anggota tabel page. Contohnya, dalam gambar, tidak ada anggota tabel page untuk proses x untuk virtual PFN 2 dan jika proses x ingin membaca alamat dari virtual PFN 2, CPU tidak dapat menterjemahkan alamat ke alamat fisik. Saat ini CPU bergantung pada sistem operasi untuk menangani masalah ini. CPU menginformasikan kepada sistem operasi bahwa page fault telah terjadi, dan sistem operasi membuat proses menunggu selama sistem operasi menagani masalah ini.
CPU harus membawa page yang benar ke memori dari image di disk. Akses disk membutuhkan waktu yang sangat lama dan proses harus menunggu sampai page selesai diambil. Jika ada proses lain yang dapat dijalankan, maka sistem operai akan memilihnya untuk kemudian dijalankan. Page yang diambil kemudian dituliskan di dalam page fisik yang masih kosong dan anggota dari virtual PFN ditambahkan dalam tabel page proses. Proses kemudian dimulai lagi pada tempat dimana page fault terjadi. Saat ini terjadi pengaksesan memori virtual, CPU membuat penerjemahan dan kemudian proses dijalankan kembali.
Demand paging terjadi saat sistem sedang sibuk atau saat image pertama kali diload ke memori. Mekanisme ini berarti sebuah proses dapat mengeksekusi image dimana hanya sebagian dari image tersebut terdapat dalam memori fisik.

Swaping

Jika memori fisik tiba-tiba habis dan proses ingin memindahkan sebuah page ke memori, sistem operasi harus memutuskan apa yang harus dilakukan. Sistem operasi harus adil dalam mambagi page fisik dalam sistem diantara proses yang ada, bisa juga sistem operasi menghapus satu atau lebih page dari memori untuk membuat ruang untuk page baru yang dibawa ke memori. Cara page virtual dipilih dari memori fisik berpengaruh pada efisiensi sistem.

Linux menggunakan tehnik page aging agar adil dalam memilih page yang akan dihapus dari sistem. Ini berarti setiap page memiliki usia sesuai dengan berapa sering page itu diakses. Semakin sering sebuah page diakses, semakin muda page tersebut. Page yang tua adalah kandidat untuk diswap. Pengaksesan Memori Virtual Bersama Memori virtual mempermudah proses untuk berbagi memori saat semua akses ke memori menggunakan tabel page. Proses yang akan berbagi memori virtual yang sama, page fisik yang sama direference oleh banyak proses. Tabel page untuk setiap proses mengandung anggota page table yang mempunyai PFN fisik yang sama.

Efisiensi

Desainer dari CPU dan sistem operasi berusaha meningkatkan kinerja dari sistem. Disamping membuat prosesor, memori semakin cepat, jalan terbaik adalah manggunakan cache. Berikut ini adalah beberapa cache dalam managemen memori di linux:

1. *PageCache*
   Digunakan untuk meningkatkan akses ke image dan data dalam disk. Saat dibaca dari disk, page dicache di page cache. Jika page ini tidak dibutuhkan lagi pada suatu saat, tetapi dibutuhkan lagi pada saat yang lain, page ini dapat segera diambil dari page cache.
   2.BufferCache
   Page mungkin mengandung buffer data yang sedang digunakan oleh kernel, device driver dan lain-lain. Buffer cache tampak seperti daftar buffer. Contohnya, device driver membutuhkan buffer 256 bytes, adalah lebih cepat untuk mengambil buffer dari buffer cache daripada

mengalokasikan page fisik lalu kemudian memecahnya menjadi 256 bytes buffer-buffer.

3.SwapCache

Hanya page yang telah ditulis ditempatkan dalam swap file. Selama page ini tidak mengalami perubahan setelah ditulis ke dalam swap file, maka saat berikutnya page di swap out tidak perlu menuliskan kembali jika page telah ada di swap file. Di sistem yang sering mengalami swap, ini dapat menghemat akses disk yang tidak perlu.

Salah satu implementasi yang umum dari hardware cache adalah di CPU, cache dari anggota tabel page. Dalam hal ini, CPU tidak secara langsung membaca tabel page, tetap mencache terjemahan page yang dibutuhkan.

Load dan Eksekusi Program

1.Penempatan program dalam memori

Linux membuat tabel-tabel fungsi untuk loading program, memberikan kesempatan kepada setiap fungsi untuk meload file yang diberikan saat sistem call exec dijalankan. Pertama-tama file binari dari page ditempatkan pada memori virtual. Hanya pada saat program mencoba mengakses page yang telah diberikan terjadi page fault, maka page akan diload ke memori fisik.

2. Linking statis dan linking dinamis

a. Linking statis:

librari-librari yang digunakan oleh program ditaruh secara langsung dalam file binari yang dapat dieksekusi. Kerugian dari linking statis adalah setiap program harus mengandung kopi library sistem yang umum.

b. Linking dinamis:

hanya sekali meload librari sistem menuju memori. Linking dinamis lebih efisien dalam hal memori fisik dan ruang disk.

10. Pertemuan 10

Pengertian manajemen file dalam sistem operasi dan manfaatnya

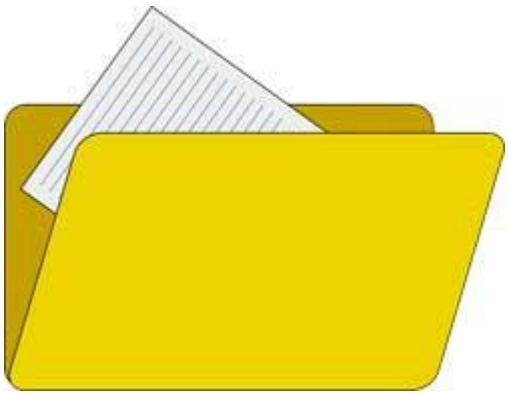
File system atau disebut juga dengan manajemen file adalah suatu metode dan struktur data yang dipakai oleh sistem operasi untuk mengatur serta menorganisir file yang terdapat pada disk atau partisi disk. Manajemen file *(File system)* ini dapat diartikan sebagai disk atau partisi yang dipakai untuk menyimpan file-file dalam cara tertentu.

A. Inilah manfaat manajemen file

Adapun manfaat dari manajemen file diantaranya yaitu, dapat mengurangi resiko kehilangan file misalnya seperti terhapusnya file secara tidak sengaja, file tersimpan dimana saja dan tidak teraturnya letak file serta dapat memudahkan kita dalam pencarian file, dapat menghemat kapasitas penyimpanan dengan cara melakukan penghapusan file yang tidak terpakai. Untuk mendapatkan manfaat dari manajemen file kamu harus dapat melakukan manajemen file dengan baik dan benar.

B. Sasaran sistem file

- Untuk memenuhi kebutuhan dari manajemen data bagi pemakai atau *user*.
- Untuk menjamin data yang terdapat pada file adalah *valid*.
- Untuk optimasi kinerja.
- Untuk menyediakan dukungan masukan *(input)* dan keluaran *(output)* bagi beragam tipe perangkat penyimpanan.
- Untuk meminimalkan atau mengeliminasi potensi kehilangan data.
- Untuk menyediakan sekumpulan rutin *interface* masukan *(input)* dan keluaran *(output)*.
- Dan untuk menyediakan dukungan masukan *(input)* dan keluaran *(output)* bagi banyak pemakai *(user)* di sistem *multiuser*.



apakah itu manajemen file?

C. Beberapa fungsi yang diharapkan dari pengelolaan file

- Mekanisme pemakaian file secara bersama.
- Penciptaan, modifikasi dan penghapusan file.
- Kemampuan men-*beckup* dan *recovery* untuk dapat mencegah kehilangan file dikarenakan kecelakaan atau adanya upaya penghancuran file.

- Pemakai bisa mengacu file dengan nama simbolik *(Symbolic name)* bukan menggunakan penamaan yang mengacu perangkat fisik.
- Supaya pada lingkungan sensitif, informasi dapat tersimpan dengan aman dan rahasia.
- Sistem file harus menyediakan *interface user-friendly*.

D. Arsitektur Pengelolaan File

Pengelolaan file, biasanya terdiri seperti di bawah ini:

- Yang pertama adalah sistem akses, yaitu berhubungan dengan bagaimana cara data yang disimpan pada file akses.
- Yang kedua adalah manajemen file, yaitu berhubungan dengan penyediaan mekanisme operasi pada file, misalnya seperti: penyimpanan, pengacuan, pemakaian bersama, dan juga pengamanan.
- Yang ketiga adalah, manajemen ruang penyimpanan yaitu berhubungan dengan alokasi ruang untuk file di perangkat penyimpan.
- Dan yang keempat adalah mekanisme Integritas file, yaitu berhubungan dengan jaminan informasi pada file yang tidak terkorupsi.

E. Tipe file yang terdapat pada sistem operasi

Terdapat tiga tipe file pada sistem operasi, diantaranya seperti di bawah ini:

- File regular yang berisi informasi, yang terdiri dari file teks dan biner. File teks ini berisi baris-baris teks (txt). Lalu file biner eksekusi (exe), dan juga biner hasil dari program aplikasi. Struktur internal file biner eksekusi hanya diketahui oleh sistem operasi, sedangkan struktur internal dari file biner hasil program aplikasi hanya diketahui oleh program aplikasi saja yang menggunakan file tersebut.
- File folder yaitu file yang dimiliki oleh sistem operasi, biasanya berisi informasi-informasi mengenai daftar file yang termasuk didalam folder tersebut.
- Dan file khusus merupakan nama logic dari perangkat *input* dan perangkat *output*.

11. Pertemuan 11

Pengertian Management Input Output

Dalam sistem komputer manajemen i/o sangat diperlukan karena i/o adalah sarana user untuk bisa berkomunikasi dengan komputer. Contoh perangkat i/o seperti keyboard, mice, audio controllers, video controllers, disk drives, networking ports, dll. Manajemen i/o pun diperlukan agar user dapat langsung menggunakan perangkat i/o tanpa harus menginialisasi terlebih dahulu. Oleh karena itu, dalam setiap system operasi selalu terdapat i/o manager.

Beberapa fungsi management input /output :

1. Mengirim perintah ke perangkat input / output agar menyediakan layanan.
2. Menangani interupsi perangkat input / output
3. Menangani kesalahan perangkat input /output.
4. Menyediakan interface ke pemakai.

Teknik Management Input Output

I/O Terprogram

Pada I/O terprogram, data saling dipertukarkan antara CPU dan modul I/O. CPU mengeksekusi program yang memberikan operasi I/O kepada CPU secara langsung, seperti pemindahan data, pengiriman perintah baca maupun tulis, dan monitoring perangkat.

Kelemahan teknik ini adalah CPU akan menunggu sampai operasi I/O selesai dilakukan modul I/O sehingga akan membuang waktu, apalagi CPU lebih cepat proses operasinya. Dalam teknik ini, modul I/O tidak dapat melakukan interupsi kepada CPU terhadap proses – proses yang diinteruksikan padanya. Seluruh proses merupakan tanggung jawab CPU sampai operasi lengkap dilaksanakan

I/O Interrupt

Teknik interrupt – driven I/O memungkinkan proses tidak membuang – buang waktu. Prosesnya adalah CPU mengeluarkan perintah I/O pada modul I/O, bersamaan perintah I/O dijalankan modul I/O maka CPU akan melakukan eksekusi perintah – perintah lainnya. Apabila modul I/O telah selesai menjalankan instruksi yang diberikan padanya akan melakukan interupsi pada CPU bahwa tugasnya telah selesai.

Dalam teknik ini kendali perintah masih menjadi tanggung jawab CPU, baik pengambilan perintah dari memori maupun pelaksanaan isi perintah tersebut. Terdapat selangkah kemajuan dari teknik sebelumnya, yaitu CPU melakukan *multitasking* beberapa perintah sekaligus sehingga tidak ada waktu tunggu bagi CPU. Teknik interrupt – driven I/O memungkinkan proses tidak membuang buang waktu. Prosesnya adalah CPU mengeluarkan perintah I/O pada modul I/O, bersamaan perintah I/O dijalankan modul I/O maka CPU akan melakukan eksekusi perintah – perintah lainnya. Apabila modul I/O telah selesai menjalankan instruksi yang diberikan padanya akan melakukan interupsi pada CPU bahwa tugasnya telah selesai.

Dalam teknik ini kendali perintah masih menjadi tanggung jawab CPU, baik pengambilan perintah dari memori maupun pelaksanaan isi perintah tersebut. Terdapat selangkah kemajuan dari teknik sebelumnya, yaitu CPU melakukan *multitasking* beberapa perintah sekaligus sehingga tidak ada waktu tunggu bagi CPU

Direct Memory Access (DMA)

Teknik yang dijelaskan sebelumnya yaitu I/O terprogram dan Interrupt-Driven I/O memiliki kelemahan, yaitu proses yang terjadi pada modul I/O masih melibatkan CPU secara langsung. Hal ini berimplikasi pada :

- Kelajuan transfer I/O yang tergantung pada kecepatan operasi CPU.
- Kerja CPU terganggu karena adanya interupsi secara langsung.

Bertolak dari kelemahan di atas, apalagi untuk menangani transfer data bervolume besar dikembangkan teknik yang lebih baik, dikenal dengan *Direct Memory Access* (DMA). Prinsip kerja DMA adalah CPU akan mendelegasikan kerja I/O kepada DMA, CPU hanya akan terlibat pada awal proses untuk memberikan instruksi lengkap pada DMA dan akhir proses saja. Dengan demikian CPU dapat menjalankan proses lainnya tanpa banyak terganggu dengan interupsi.

Komponen Management Input Output

a. Buffer : menampung sementara data dari/ke perangkat I/O.
b. Spooling : melakukan penjadwalan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
c. Menyediakan "driver" untuk dapat melakukan operasi "rinci" untuk perangkat keras I/O tertentu.

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

INPUT :

1. Keyboard : berisi tombol-tombol yang terdiri dari fungsi,angka,huruf,tanda baca dan tombol kontrol
2. Mouse : Perangkat keras untuk terhubung dengan layar komputer,berfungsi mengarahkan pointer dilayar monitor
3. Microphone : Merekam suara kedalam kompputer yang akan disimpan ke sound card
4. Scanner : alat yang digunakan untuk memasukan data berbentuk object kedalam komputer untuk diubah kedalam bentuk digital
5. CD-ROOM : alat komputer untuk membaca dan memutar compact disk (cd),salah satu media penyimpanan memori
6. DVD-ROOM : memutar dan membaca CD dan DVD

OUTPUT :

1. Monitor : perangkat yang akan menampilkan apa-apa saja yang kita kerjakan dikomputer baik itu teks atau grafik
2. Printer : Pernagkat keras yang mencetak hasil inputa berupa image atau teks pada kertas
3. Speaker : Perangkat yang mengeluarkan data berupa suara

Perangkat Input Output

a. Perangkat Keras

Piranti keras I/O atau device pada sistem komputer amatlah beragam. Masing-masing piranti I/O memiliki karakteristik yang khas. Karakteristik yang dapat digunakan sebagai pembeda antara piranti I/O yang satu dengan yang lain antara lain :

1. Modus Transfer Data.
2. Metode Akses.
3. Jadwal Transfer.
4. Sharing.
5. Kecepatan Akses.
6. Modus Operasi I/O

b. Perangkat Lunak

Tujuan perancangan perangkat lunak I/O adalah :

1. Device Independence
2. Uniform Naming
3. Error Handling
4. Transfer Sinkron Vs. Asinkron
5. Sharable Vs. Dedicated Device

12 & 13. Pertemuan keduabelas dan tigabelas

PROTEKSI DAN SECURITY SISTEM KOMPUTER

1. PROTEKSI

Proteksi adalah mekanisme sistem yang di gunakan untuk memproteksi atau melindungi informasi pada sistem komputer.Proteksi mengacu pada mekanisme untuk mengontrol akses yang di lakukan oleh program, prosesor atau pengguna ke sistem sumber daya.Dalam beberapa sistem, proteksi di lakukan oleh sebuah yang bernama reference monitor. Jika ada pengaksesan sumber daya PC yang di proteksi, sistem pertama kali akan bertanya kepada reference monitor tentang boleh atau tidak nya akses tersebut (keabsahan). Selanjutnya reference monitor akan menentukan keputusan apakah akses tersebut di perbolehkan atau di tolak. Secara sederhana, mekanisme proteksi ini dapat di gambarkan dengan konsep domain.

Domain merupakan himpunan yang berisi pasangan objek dan hak akses. Masing-masing pasangan domain berisi sebuah objek dan beberapa akses operasi contoh nya read, write, execute, operasi yang dapat di lakukan terhadap objek tersebut. Dalam setiap waktu, setiap proses berjalan dalam beberapa domain proteksi.Hal itu berarti terdapat beberapa objek yang dapat diakses oleh proses tersebut, dan operasi-operasi apa yang boleh dilakukan oleh proses terhadap objek tersebut. Proses juga bisa berpindah dari domain ke domain lain dalam eksekusi.

Implementasi pengamanan sangat penting untuk menjamin sistem tidak diinterupsi dan di ganggu. Proteksi dan pengamanan terhadap perangkat keras dan sistem operasi sama pentingnya.

Pada sistem komputer banyak objek yang perlu diproteksi, yaitu :

1. Objek perangkat keras.

Objek yang perlu diproteksi antara lain :

- Pemroses.
- Segment memori.
- Terminal.
- Disk drive.
- Printer.

2. Objek perangkat lunak.

Objek yang perlu diproteksi antara lain :

- Proses.
- File.
- Basis data.
- Semaphore.[2]

2. TUJUAN PROTEKSI

Proteksi memiliki beberapa tujuan antara lain :

1. Untuk melindungi, memberikan ijin dan mengatur pemakaian sumber daya yang ada dalam sistem tersebut baik sumber daya fisik (memori, diks, prosesor, jaringan komputer) maupun data / informasi

2. Menjamin sistem tidak di interupsi dan di ganggu

3. Menghindari, mencegah dan mengatasi ancaman terhadap sistem.[3]

- DOMAIN PROTEKSI

Yang di maksud domain proteksi yaitu melindungi objek-objek pada sistem komputer agar tidak terjadi kerusakan. Setiap domain harus memiliki nama yang unik dan sekumpulan operasi yang dapat di lakukan terhadap domain.

Agar dapat menyediakan mekanisme proteksi berbeda, dikembangkan berdasarkan konsep domain.Domain : himpunan pasangan (objek, hak).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.

Sistem komputer merupakan gabungan dari banyak proses dan objek. Objek dalam hal ini kita artikan sebagai objek hardware (seperti CPU, segmen memori,printer, disket, dan drive), dan objek software (seperti berkas, program, dan semaphore). Tiap objek mempunyai nama yang khusus yang membedakan mereka dengan lainnya pada suatu sistem, dan tiap-tiap dari mereka dapat diakses hanya melalui operasi yang khusus pula. Secara esensial objek adalah tipe data abstrak.Operasi yang ada memungkinkan untuk bergantung pada objeknya. Contoh CPU hanya bisa dinyalakan. Segmen memori dapat membaca maupun menulis, dimanacard reader hanya bisa membaca saja. Drive dapat dibaca, ditulis, ataupun, di-rewound.Berkas data dapat dibuat, dibuka, dibaca, ditulis, ditutup, dihapus; berkas program dapat dibaca, ditulis, dijalankan, dan dihapus.Jelasnya, sebuah proses hanya boleh mengakses resource yang memang dibolehkan. Untuk lebih lanjut, kapan saja, hal ini diharuskan untuk hanya mengaksesresource yang memang dibutuhkan saat itu.

- HAK AKSES

Hak akses adalah hak yang diberikan kepada user untuk mengakses sistem. Mungkin hak akses adalah hal yang paling mandasar dalam bidang sekuriti. Dalam strategi sekuriti, setiap objek dalam sistem (user, administratir, software, sistem itu sendiri) harus diberikan hak akses yang berguna untuk menunjang fungsi kerja dari objek tersebut. Dengan kata lain, objek hanya memperoleh hak akses minimum.

Dengan demikian, kerja objek terhadap sistem dapat di batasi sehingga objek tidak akan melakukan hal-hal yang membahayakan sekuriti jaringan komputer. Hak akses minimum akan membuat para menyusup dari internet tidak dapat berbuat banyak saat berhasil menembus sebuah user account pada sistem jaringan komputer. Hak akses minimum juga bisa mengurangi bahaya yang mengancam sistem dari dalam. Itulah beberapa keuntungan yang dapat di peroleh dari strategi ini.

Subjek (pengguna) dapat memodifikasi atribut akses (read, write, run) setiap objek (sumber daya) yang dibuatnya dengan melakukan proses granting (mengijinkan akses) dan revoking (menolak akses).

- SISTEM BERDASARKAN KAPABILITAS

Sistem Berbasis Kapabilitas , kita akan mensurvey 2 macam sistem. Sistem ini memiliki banyak variasi pada tingkat kompleksitas yang berbeda-beda dan pada tingkat policyyang mengacu pada bagaimana kita mengimplementasikannya.

- Hydra

hidra adalah sistem proteksi berbasis kapabilitas yang menyediakan fleksibilitas yang baik. Sistem ini menyediakan sebuah set yang pasti dari access right yang mungkin dapat diketahui dan diinterpretasikan oleh sistem. Right ini termasuk bentuk dasar dari access sebagai right untuk membaca, menulis ataupun mengeksekusi pada sebuah segmen memory. Dalam kenyataannya, sistem ini menyediakan cara bagiuser untuk mendeklarasikan right tambahan. Interpretasi dari user-diviner

rightdilaksanakan hanya oleh program dari user, tetapi sistem menyediakan proteksi access dalam penggunaan right ini, seiring dengan penggunaan dari sistem-diviner right.

Operasi dari objek didefinisikan sesuai dengan prosedur. Prosedur yang mengimplementasikan operasi ini merupakan bentuk dari sebuah objek dan diaksessecara tidak langsung oleh kapabilitas. Nama dari user-divine procedure harusdiidentifikasi kepada sistem proteksi bila hal ini berhubungan dengan objek dari user-divide-type. Ketika definisi dari sebuah objek yang telah dibuat diperkenalkan kepadahydra, nama dari operasi yang ada pada tipe ini berubah dari auxiliary right. Auxiliaryright dapat dijabarkan dalam sebuah kapabilitas dari sebuah tipe.Konsep lainnya adalah right implification. Skema ini memungkinkan sebuah sertifikasi dari sebuah prosedur yang aman untuk dapat bergerak pada sebuah parameter formal dari sebuah tipe yang telah terspesifikasi. Amplifikasi sangat berguna dalam menjalankan implementasi dari access procedure ke variabel representasi daritipe data abstrak.Sebuah hydra sub system dibuat di atas kernel proteksinyadan mungkinmembutuhkan proteksi dari komponenya sendiri. Sebuah sub sistem berinteraksidengan kernel melalui panggilan pada sebuah set dari kernel-divine primitive yang mendefinisikan access right kepada resourse yang diimplementasikan selanjutnya oleh sub sistem.

- Cambridge CAP System

Sebuah pendekatan kepada proteksi berbasis kapabilitas telah dibentuk dalamsebuah desain dari Cambridge CAP System. Sistem kapabilitas dari CAP lebihsederhana dan tidak lebih baik daripada hydra. Hanya saja ditunjukkan bahwa sistemini dapat digunakan untuk menyediakan sebuah proteksi sekuritas dari user-divine-object. Pada CAP ada 2 kapabilitas. Yang satu dinamakan data kapabilitas, yang dapatdigunakan untuk menyediakan akses kepada objek tetapi right yang disediakan hanyaright standar seperti membaca, tulis ataupun mengeksekusi segmen penyimpananterpisah yang terasosiakan dengan objek.

Data kapabilitas diintepretasikan oleh microcode di dalam mesin CAP.Sebuah software kapabilitas dilindungi oleh CAP microcode tetapi tidak diinterpretasikan. Hal ini diinterpretasikan oleh sebuah prosedur yang terlindungi, yangmungkin bisa ditulis oleh sebuah progammer aplikasi sebagai sebuah bagian dari subsystem. Walaupun seorang programmer dapat mendefinisikan prosedur proteksinyasendiri, tetapi secara global sistem tidak dapat disatukan dengan sistem proteksi dasar tidak memperbolehkan prosedur-prosedur lain yang dibuat oleh user untuk mengakseskepada segmen penyimpanan yang bukan milik dari lingkungan yang ada. Designer dari system CAP telah menyadari bahwa penggunaan software kapabilitasmemungkinkan mereka untuk membuat sebuah formula dengan harga yang terjangkadan mengimplementasi policy dari proteksi yang sesuai dengan kebutuhan.[4]

3. SECURITY

- MASALAH SECURITY

Keamanan sistem komputer adalah untuk menjamin sumber daya tidak digunakan ataudimodifikasi oleh orang yang tak berhak. Pengamanan termasuk masalah teknis, manajeral, legalitas dan politis.

Keamanan sistem terbagi menjadi 3 yaitu :

1. Keamanan eksternal (external security)

Berkaitan dengan pengamanan fasilitas komputer dari penyusup dan bencana seperti kebakaran dan kebanjiran.

2. Keamanan interface pemakai (user interface security)

Berkaitan dengan identifikasi pemakai sebelum pemakai diijinkan mengkases program dan data yang disimpan.

3. Keamanan internal (internal security)

Berkaitan dengan keamanan 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 dan proteksi sering digunakan secara bergantian. Untuk menghindari kesalahpahaman, istilah kemanan mengacu ke seluruh masalah kemanan dan istilah mekanisme proteksi mengacu ke mekanisme sistem yang digunakan untuk memproteksi / melindungi informasi pada sistem komputer.

Masalah masalah keamanan :

a. Kehilangan data (data loss) dapat disebabkan karena :

· Bencana

· Kesalahan pernagkat keras dan perangkat lunak

· Kesalahan / kelalaian manusia.

b. Penyusup

· Penyusup pasif, yaitu yang membaca data yang tak diotoritaskan

· Penyusup aktif, yaitu yang mengubah data yang tak diotoritaskan[5]

Tujuan Security

1. Integritas Data

Misalnya pengguna yang tidak memiliki autorisasi untuk mengakses data tidak mungkin dapat mengubah atau memodifikasi data.

2. Kerahasiaan Data

Sistem dapat menjamin bahwa data yang telah ditentukan untuk tidak dapat dibaca oleh pengguna lain pada sistem, data tersebut benar-benar aman dan rahasia.

3. Ketersediaan Akses ke Sistem

Tidak ada seorangpun, sekalipun dengan akses ke sistem dapat menyebabkan sistem menjadi tidak dapat digunakan. Contoh nya dengan serangan denial of service melalui internet.[6]

- AUTENTIKASI

Proses pengenalan peralatan, sistem operasi , kegiatan, aplikasi dan identitas user yang terhubung dengan jaringan komputer. Autentikasi dimulai pada saat user login ke jaringan dengan cara memasukkan password.

Tahapan Autentikasi

1. Autentikasi untuk mengetahui lokasi dari peralatan pada suatu simpul jaringan (data link layer dan network layer)

2. Autentikasi untuk mengenal sistem operasi yang terhubung ke jaringan (transport layer)

3. Autentikasi untuk mengetahui fungsi/proses yang sedang terjadi di suatu simpul jaringan (session dan presentation layer)

4. Autentikasi untuk mengenali user dan aplikasi yang digunakan (application layer)

Autentifikasi pemakai

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

1. Sesuatu yang diketahui pemakai, misalanya :

· Password

· Kombinasi kunci

· Dan sebagainya

2. Sesuatu yang dimiliki pemakai, misalanya;

· Badge

· Kartu identitas

· Kunci

· Dan sebgainya

3. Sesutau mengenai (ciri) pemakai, misalnya;

· Sidik jari

· Sidik suara

· Foto

· Tanda tangan[7]

- **ANCAMAN PROGRAM**

A. Program-program jahat

Ancaman ancaman canggih terhadap sistem komputer adalah program yang mengekspolitasi 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 :

· 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.

· 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.

· Trapdoor

Trapdoor adalah titik masuk tak terdokumentasi rahasia di satu program untuk memberikan akses tanpa metode-metode otentifikasi normal.

· Trojan horse

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..

· Virus

Virus komputer adalah buatan manusia yang bertujuan merugikan orang lain. Ancaman yang paling serius dari sebuah virus komputer adalah sifatnya yang merusak. Tidak semua program virus dibuat untuk merusak. Ada yan g mungkin membuat virus dengan tujuan melindungi hasil karya sendiri.[8]

B. Tipe-tipe virus

· 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.

· Memory resident virus

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

· Boot sector virus

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

· Stealth virus

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

· 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

C. 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 :

- Deteksi. : Begitu infeksi telah terjadi, tentukan apakah infeksi memang telah terjadi dan cari lokasi virus.
- Identifikasi. : Begitu virus terdeteksi maka identifikasi virus yang menginfeksi program.
- 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 yangsederhana menjadi semakin canggih, begitu juga paket perangkat lunakantivirus. Saat ini program antivirus semakin kompleks dan canggih .

- **ANCAMAN SISTEM**

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.[9]

- **MONITORING ANCAMAN**

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 file 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.[10]

- **ENKRIPSI**

Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. atau bisa didefinisikan juga Enkripsi, merupakan proses untuk mengubah plainteks menjadi chiperteks.Planteks sendiri adalah data atau pesan asli yang ingin dikirim, sedangkan Chiperteks adalah data hasil enkripsi.Definisi lain tentang Enkripsi adalah proses mengacak data sehingga tidak dapat dibaca oleh pihak lain.

Enkripsi mempunyai kelebihan dan kekurangan yang diantaranya adalah:

Kelebihan dari Enkripsi :

· Kerahasiaan suatu informasi terjamin

· Menyediakan autentikasi dan perlindungan integritas pada algoritma checksum/hash

· Menanggulangi penyadapan telepon dan email

· Untuk digital signature

Kekurangan dari Enkripsi

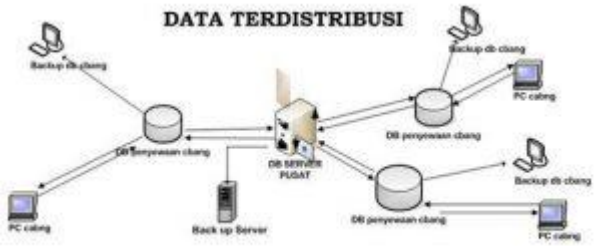
· Penyandian rencana teroris

· Penyembunyian record kriminal oleh seorang penjahat

· Pesan tidak bisa dibaca bila penerima pesan lupa atau kehilangan kunci

13 & 15. Pertemuan keempatbelas dan limabelas

Pengertian Sistem Terdistribusi dan Macam-Macam nya



Definisi Sistem Terdistribusi

Sistem Terdistribusi terdiri dari dua kata yaitu " Sistem" dan "Terdistribusi". Jadi Sistem terdistribusi adalah sekumpulan elemen yang saling berhubungan satu dengan yang lainnya dan juga membentuk satu kesatuan guna menyelesaikan satu tujuan yang spesifik atau menjalangkan sperangkat fungsi. Adapun terdistribusi yaitu berasal dari kata "distribusi" yang ialah lawan kata dari "sentralisasi" yang artinya adalah penyebaran, sirkulasi, penyerahan, pembagian menjadi bagian-bagian kecil.

Contoh Sistem Terdistribusi adalah sebagai berikut

1. **Intranet**

Itranet adalah Jaringan (proprietary) yang teradministrasi secara lokal dan bisa terhubung ke internet melalui firewall juga adanya layanan internal dan juga eksternal didalamnya.

2. Internet

Internet adalah jaringan global yang menghubungkan komputer satu sama lain dan bisa berkomunikasi dengan media IP sebagai protokol.

3. World Wide Web

World Wide Web (www) Arsiteketur client/server terbuka yang diterapkan di atas infrastuktur internet dan juga shared resources melalui URL.

4. .Mobile dan sistem komputasi ubiguitos

Sistem telepon Celullar (e.g. GSM) re. Resources yaang dishare : frekuensi radio, waktu transmisi dalam satu frekuensi, komputer laptop, bergerak, handheld devices, ubiquitos computing, PDA, etc

5. Sistem terdistribusi multimedia

System terdistribusi ini biasanya digunakan pada infrastruktur internet

-karakteristik

Sumber data yang heterogen dan juga memerlukan sinkronisasi secara real time

-video, audio, text Multicast

contohnya:

-Teletaching tools (mbone-based, etc.)

-Video-conferencing

-Video and audio on demand

6. Contoh sistem terdistribusi lainnya adalah :

Sistem telepon seperti ISDN, PSTN

Manajemen jaringan misalnya Administrasi sesumber jaringan

Network File System (NFS) seperti Arsitektur guna mengakses sistem file melalui jaringan.

Model Sistem Terdistribusi

Dalam sistem terdistribusi terdapat beberapa model, antaranya adalah :

Model Client Server

Sistem Client-server ini mempunyai satu atau lebih proses client dan satu atau lebih proses server, dan juga sebuah proses client biasanya mengirim query ke sembarang proses server. Client bertanggung jawab di antar muka untuk user, sedangkan server mengatur data dan juga mengeksekusi transaaksi. Sehingga suatu proses client berjalan di sebuah personal computer dan juga mengirim query ke sebuah server yang berjalan pada mainframe.

Aarsitektur jenis ini menjadi sangat popular untuk beberapa alasan. Pertama, ialah implementasi yang relatif sederhana karena pembagian fungsi yang baik dan karena server tersentralisasi. Yang Kedua, mesin server yang cukup mahal utilisasinya tidak terperngaruh pada interaksi pemakai. Meskipun mesin client tidak terlalu mahal. Ketiga adalah, pemakai dapat menjalankan antarmuka berbasis grafis maka dari itu pemakailebih mudah dibandingkan antar muka pada server yang tidak user-friendl, perlu diingat batasan antara client dan juga server dan untuk menjaga komunikasi antara keduanya yang berorientasi himpunan. Khususnya untuk membuka kursor dan mengambil tupel pada satu waktu membangkitkan beberapa pesan dan bisa diabaikan.

Client:

– Proses akses data

– Melakukan operasi pada komputer yang lain

Server:

– Proses mengatur data

– Proses mengatur resources

– dan Proses komputasi

Interaksi:

– Invocation atau result

Model Multiple Server

Service disediakan oleh beberapa server

Contohnya ialah:

– Sebuah situs yang jalankan dibeberapa server

Server menggunakan replikasi /database terdistribusi

Model Proxy Server

Proxy server menyediakan hasil copy (replikasi) dari resource yang di atur oleh server lain. Biasa nya proxy server di gunakan untuk menyimpan hasil copy web resources. Apabila client melakukan request ke server, hal yang pertama dilakukan ialah memeriksa proxy server apakah yang diminta oleh client terdapat pada proxy server. Proxy server bisa diletakkan pada setiap client atau bisa di pakai bersama oleh beberapa client. Tujuannya ialah meningkatkan performance dan availibity dengan mencegah frekwensi akses ke server.

Proxy server membuat duplikasi beberapa server yang diakses oleh client

Caching:

– Penyimpanan lokal untuk item yang sering diaksess

– Meningkatkan kinerja

– Mengurangi beban pada server

Contoh:

– Searching satu topik namun dilakukan dua kali maka searching terakhir mempunyai waktu yang lebih kecil

Model Peer To Peer

Bagian dari model sistem terdistribusi dimana sistem bisa sekaligus berfungsi sebagai client maupun server. Sebuah arsitektur di mana tidak terdapat mesin khusus yang melayani suatu pelayanan tertentu ataupun mengatur sumber daya dalam jaringan dan semua kewajiban dibagi rata ke seluruh mesin, yang dikenal dengan peer. Pola komunikasi yang digunakan berdasarkan aplikasi yang digunakan. Peer-to-peer adalah model yang paling general dan fleksible.

Model Mobile Code

Kode yang berpindah dan dijalankan pada pc yang berbeda

Contohnya: Applet

Model Mobile Agent

Sebuah program yang berpindah dari satu komputer ke komputer yang lainnya

Melakukan perkerjaan yang otomatis

Contoh:

– Untuk install dan juga pemeliharan software pada komputer sebuah organisasi