

Bab 1

SISTEM KOMPUTER

Empat komponen dalam sistem komputer, yaitu :

1.1. Pemroses

- Berfungsi untuk mengendalikan operasi komputer dan melakukan fungsi pemrosesan data.
- Pemroses melakukan operasi logika dan mengelola aliran data dengan membaca instruksi dari memori dan mengeksekusinya.
- Langkah kerja pemroses :
 - a. Mengambil instruksi biner dari memori
 - b. Mendekode instruksi menjadi aksi sederhana
 - c. Melakukan aksi

3 tipe operasi komputer :

- a. Operasi aritmatika (ADD, SUBSTRACT, MULTIPLY, DIVIDE)
- b. Operasi logika (OR, AND, XOR, INVERTION)
- c. Operasi pengendalian (LOOP, JUMP)

Pemroses terdiri :

- a. ALU (Aritmatic Logic Unit)

Berfungsi untuk melakukan operasi aritmatika dan logika.
- b. CU (Control Unit)

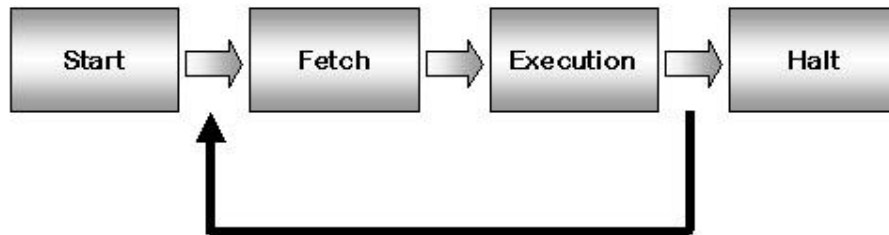
Berfungsi untuk mengendalikan operasi yang dilaksanakan sistem komputer.
- c. Register-register
Berfungsi untuk :
 - Membantu pelaksanaan operasi yang dilakukan pemroses
 - Sebagai memori yang bekerja secara cepat, biasanya untuk tempat operand-operand dari operasi yang akan dilakukan.

- ☑ Terbagi menjadi register data dan register alamat.
- ☑ Register data terdiri dari general dan special purpose register.
- ☑ Register alamat berisi :
 - a. Alamat data di memori utama
 - b. Alamat instruksi
 - c. Alamat untuk perhitungan alamat lengkapContoh : register indeks, register penunjuk segmen, register penunjuk stack, register penanda (flag)

Pemroses melakukan tugasnya dengan mengeksekusi instruksi-instruksi di program dengan mekanisme instruksi sebagai berikut :

- a. Pemroses membaca instruksi dari memori (fetch)
- b. Pemroses mengeksekusi instruksi (execute)

Eksekusi program berisi pengulangan fetch dan execute. Pemrosesan satu instruksi disebut satu siklus instruksi (instruction cycle).



Gambar 1-1 : Siklus eksekusi instruksi

1.2. Memori

- ☑ Berfungsi untuk menyimpan data dan program
- ☑ Biasanya volatile, tidak dapat mempertahankan data dan program yang disimpan bila sumber daya energi (listrik) dihentikan.

- ☑ Konsep program tersimpan (stored program concept), yaitu program (kumpulan instruksi) yang disimpan di suatu tempat (memori) dimana kemudian instruksi tersebut dieksekusi.
- ☑ Setiap kali pemroses melakukan eksekusi, pemroses harus membaca instruksi dari memori utama. Agar eksekusi dilakukan secara cepat maka harus diusahakan instruksi tersedia di memori pada lapisan berkecepatan akses lebih tinggi. Kecepatan eksekusi ini akan meningkatkan kinerja sistem.

Hirarki memori berdasarkan kecepatan akses :

- ☑ Register (tercepat)
- ☑ Chace memory
Memori berkapasitas terbatas, berkecepatan tinggi yang lebih mahal dibanding memori utama. Chace memory adalah diantara memori utama dan register, sehingga pemroses tidak langsung mengacu memori utama tetapi di cache memory yang kecepatan aksesnya lebih tinggi.
- ☑ Main memory
- ☑ Disk chace (buffering)
Bagian memori utama untuk menampung data yang akan ditransfer dari/ke perangkat masukan/keluaran dan penyimpan sekunder. Buffering dapat mengurangi frekuensi pengaksesan dari/ke perangkat masukan/keluaran dan penyimpan sekunder sehingga meningkatkan kinerja sistem.
 - Magnetic disk
 - Magnetic tape, optical disk (terlambat)

1.3. Perangkat masukan dan keluaran (I/O)

- ☑ Adalah perangkat nyata yang dikendalikan chip controller di board sistem atau card.
- ☑ Controller dihubungkan dengan pemroses dan komponen lainnya melalui bus.
- ☑ Controller mempunyai register-register untuk pengendaliannya yang berisi status kendali.

- ☑ Tiap controller dibuat agar dapat dialamati secara individu oleh pemroses sehingga perangkat lunak device driver dapat menulis ke register-registernya sehingga dapat mengendalikannya.
- ☑ Sistem operasi lebih berkepentingan dengan pengendali dibanding dengan perangkat fisik mekanis.
- ☑ Perangkat I/O juga memindahkan data antara komputer dan lingkungan eksternal.
- ☑ Lingkungan eksternal dapat diantarmuka (interface) dengan beragam perangkat, seperti :
 - a. Perangkat penyimpan sekunder
 - b. Perangkat komunikasi
 - c. Terminal

1.4. Interkoneksi antar komponen

- ☑ Adalah struktur dan mekanisme untuk menghubungkan antar komponen dalam sistem komputer yang disebut bus.
- ☑ Bus terdiri dari tiga macam, yaitu :
 - a. Bus alamat (address bus)

Berisi 16, 20, 24 jalur sinyal paralel atau lebih. CPU mengirim alamat lokasi memori atau port yang ingin ditulis atau dibaca di bus ini. Jumlah lokasi memori yang dapat dialamati ditentukan jumlah jalur alamat. Jika CPU mempunyai N jalur alamat maka dapat mengalami 2^N lokasi memori dan/atau port secara langsung.
 - b. Bus data (data bus)

Berisi 8, 16, 32 jalur sinyal paralel atau lebih. Jalur-jalur data adalah dua arah (bidirectional). CPU dapat membaca dan mengirim data dari/ke memori atau port. Banyak perangkat pada sistem yang dihubungkan ke bus data tetapi hanya satu perangkat pada satu saat yang dapat memakainya.
 - c. Bus kendali (control bus)

Berisi 4-10 jalur sinyal paralel. CPU mengirim sinyal-sinyal pada bus kendali untuk memerintahkan memori atau port.

Sinyal bus kendali antara lain :

 - Memory read

Untuk memerintahkan melakukan pembacaan dari memori.

- Memory write

Untuk memerintahkan melakukan penulisan ke memori.

- I/O read

Untuk memerintahkan melakukan pembacaan dari port I/O.

- I/O write

Untuk memerintahkan melakukan penulisan ke port I/O.

- Mekanisme pembacaan

Untuk membaca data suatu lokasi memori, CPU mengirim alamat memori yang dikehendaki melalui bus alamat kemudian mengirim sinyal memory read pada bus kendali. Sinyal tersebut memerintahkan ke perangkat memori untuk mengeluarkan data pada lokasi tersebut ke bus data agar dibaca CPU.

- Interkoneksi antar komponen ini membentuk satu sistem sendiri, seperti ISA (Industry Standard Architecture), EISA (Extended ISA) dan PCI (Peripheral Component Interconnect).

- Secara fisik interkoneksi antar komponen berupa "perkawatan".

- Interkoneksi memerlukan tata cara atau aturan komunikasi agar tidak kacau (chaos) sehingga mencapai tujuan yang diharapkan.

Bab 2

SISTEM OPERASI

2.1. Pendahuluan

Tanpa perangkat lunak (software) sebuah komputer sebenarnya sekumpulan besi yang tidak berguna. Dengan software sebuah komputer dapat digunakan untuk menyimpan, mengolah data (memproses) dan mendapatkan kembali informasi yang telah disimpan, menemukan kesalahan dalam program, memainkan games dan menggunakan banyak aktivitas lainnya yang bernilai. Pada umumnya program komputer dapat dibagi menjadi dua kategori, yaitu :

a. Program sistem (system program)

Yang berfungsi untuk mengatur operasi dari komputer itu sendiri.

b. Program aplikasi (application program)

Yang berfungsi untuk membantu menyelesaikan masalah untuk pemakainya.

Dasar yang sangat penting dari semua program sistem adalah operating system yang mengontrol semua sumber daya komputer dan menyediakan landasan sehingga sebuah program aplikasi dapat ditulis atau dijalankan.

Sebuah sistem komputer modern berisi satu atau lebih prosesor, banyak memori utama (sering disebut sebagai "core memory", walaupun magnetic cores yang sudah tidak digunakan dalam memory selama lebih satu dekade), clocks, terminal, disk, antarmuka (interface) jaringan, dan peralatan masukan/keluaran lainnya. Semua dalam semua, sebuah sistem yang kompleks, yaitu untuk menulis program yang disimpan dalam track sehingga dapat digunakan secara benar, apalagi dengan pengoptimalannya adalah sebuah pekerjaan atau tugas yang sangat sulit.

Jika setiap programmer memfokuskan tentang bagaimana disk drive bekerja dan apa saja yang dapat dijalankan dengan benar ketika membaca sebuah block disk, mustahil bahwa banyak program dapat ditulis secara baik.

Beberapa tahun yang lalu telah menjadi kejelasan bahwa banyak cara atau metode menyediakan pelindung (sheild) atau sesuatu yang dapat menyembunyikan programmer dari kekomplekan perangkat keras. Cara yang telah berangsur-angsur berkembang adalah dengan menentukan atau membuat sebuah lapis (layer) perangkat lunak yang mengendalikan komponen pokok perangkat keras, mengatur semua bagian sistem, dan menyediakan pemakai dengan antarmuak (interface) atau virtual machine yang memudahkan memahami dan memprogram. Lapisan dari perangkat lunak adalah operating system dan hal tersebut menjadi pokok bahasan dalam buku ini.

Diagram atau lapis (layer) dapat dilihat sebagai berikut :

| | | | |
|-------------------------|----------------------------|----------------------------|-----------------------------|
| Banking System | Airline Reservation | Adventure Games | Application Programs |
| Compilers | Editors | Command Interpreter | |
| Operating System | | | System Programs |
| Machine Language | | | |
| Microprogramming | | | Hardware |
| Physical Devices | | | |

Gambar 2.1 : Sebuah sistem komputer yang terdiri dari hardware, system programs dan application program

Pada bagian bawah dari gambar diatas adalah perangkat keras yang didalamnya banyak terdapat alasan yang membentuk dua atau lebih lapis. Lapis yang paling bawah (physical devices) berisi perangkat fisik, yang terdiri dari integrasi circuit

chips, kabel (wires), power supplies, cathode ray tubes (CRT) dan perangkat lainnya. Bagaimana perangkat tersebut dibuat dan bagaimana sistem kerjanya adalah bagian dari insinyur elektronik.

Lapis berikutnya dari software primitive yang secara langsung mengontrol peralatan tersebut (physical devices) dan menyediakan interface (a cleaner interface) untuk layer berikut/diatasnya. Software ini disebut microprogram yang biasanya ditempatkan dalam read-only memory (ROM). Sebenarnya microprogram berfungsi sebagai penerjemah, mengambil (fetching) intruksi bahasa mesin, seperti ADD, MOVE dan JUMP serta melaksanakan proses-proses kecil yang saling terkait. Contohnya apabila melaksanakan instruksi ADD, microprogram harus menentukan dimana angka-angka yang ditambahkan ditempatkan (disimpan sementara), mengambilnya, menambahkannya dan menyimpan hasilnya disuatu tempat tertentu.

Sekumpulan instruksi yang diterjemahkan microprogram disebut machine language (bahasa mesin), yang kenyataannya bukan bagian dari perangkat mesin semuanya, tetapi pembuat komputer selalu menjelaskannya dalam manualnya (sejenis buku panduan perangkat kerasnya), sehingga banyak orang mengira sebagai mesin sesungguhnya. Dalam banyak mesin microprogram diimplementasikan dalam hardware dan sebenarnya tidak sebuah lapis yang jelas.

Bahasa mesin secara khusus memiliki antara 50 sampai 300 instruksi, yang sebagian besar untuk memindahkan data disekeliling mesin, mengerjakan aritmatika dan membandingkan nilai. Dalam lapis ini, perangkat masukan/keluaran mengontrol pemuatan nilai ke dalam register khusus. Untuk contoh, sebuah disk dapat diperintahkan membaca untuk memuat nilai dari alamat disk, alamat memori utama, besarnya byte dan memerintahkan (READ atau WRITE) ke dalam registernya. Dalam prakteknya, banyak parameter yang dibutuhkan, dan status drive sesudah operasi adalah masalah yang memiliki kekomplekan masalah yang tinggi. Terlebih banyak perangkat I/O, waktu akses adalah sebuah aturan yang penting dalam pemograman.

Sebuah fungsi utama dari Operating System adalah menyembunyikan semua kekomplekan atau kerumitan dan memberikan programmer sebuah kenyamanan,

kemudahan, kepraktisan dari instruksi yang digunakan untuk bekerja dengannya, sebagai contoh, READ BLOCK FROM FILE adalah konsep atau gagasan penyederhanaan daripada harus berpikir tentang bagaimana memindahkan head disk secara rinci, menunggu untuk menjadi tenang (posisi istirahat).

Diatas lapis Operating System (OS) adalah untuk menempatkan software system. Dari sini ditemukan penerjemah perintah (command interpreter/shield), compiler, editor, dan application program independent. Adalah penting untuk menyadari/merealisasikan bahwa program-program ini dengan pasti bukan bagian dari SO, walaupun secara khas disediakan oleh pembuat komputer. Ini adalah penting sekali, tetapi sebagai titik kecerdikan. OS adalah bagian dari software yang dijalankan dalam kernel mode atau supervisor mode.

SO memproteksi dari pemakai yang mengganggu pemakain hardware (diabaikan untuk microprocessors yang sebelumnya yang tidak memiliki proteksi hardware). Compiler dan editors dijalankan dalam user mode. Jika seorang tidak suka sebuah compiler khusus, pemakai bebas menulis miliknya jika dapat memilih, dan tentunya tidak bebas untuk menangani interrupt disk yang merupakan bagian dari OS dan normalnya akan memproteksi hardware terhadap usaha pemakai untuk memodifikasinya.

Akhirnya, diatas layer system program berikutnya adalah application program. Program ini ditulis oleh user untuk memecahkan masalah khusus, seperti proses pengolahan data, perhitungan rekayasa atau memainkan sebuah permainan.

2.2. Fungsi dan sasaran Sistem Operasi

Secara garis besar mempunyai 2 tugas utama, yaitu sebagai :

a. Pengelola seluruh sumber daya pada sistem komputer (resource manager)

Yang dimaksud dengan sumber daya pada sistem komputer adalah semua komponen yang memberikan fungsi (manfaat) atau dengan pengertian lain adalah semua yang terdapat atau terhubung ke sistem komputer yang dapat

untuk memindahkan, menyimpan, dan memproses data, serta untuk mengendalikan fungsi-fungsi tersebut.

Sumber daya pada sistem komputer, antara lain :

a.1 Sumber daya fisik, misalnya :

Keyboard, bar-code reader, mouse, joystick, light-pen, track-ball, touch-screen, pointing devices, floppy disk drive, harddisk, tape drive, optical disk, CD ROM drive, CRT, LCD, printer, modem, ethernet card, PCMCIA, RAM, cache memory, register, kamera, sound card, radio, digitizer, scanner, plotter, dan sebagainya.

Salah satu sasaran yang harus dicapai sistem operasi adalah dapat memanfaatkan seluruh sumber daya agar dapat digunakan secara efektif dan efisien mungkin.

a.2 Sumber daya abstrak, terdiri dari :

a.2.1 Data, misalnya :

Semaphore untuk pengendalian sinkronisasi proses-proses, PCB (Process Control Block) untuk mencatat dan mengendalikan proses, tabel segmen, tabel page, i-node, FAT, file dan sebagainya.

a.2.2 Program

Adalah kumpulan instruksi yang dapat dijalankan oleh sistem komputer, yang dapat berupa utilitas dan program aplikasi pengolahan data tertentu.

b. Penyedia layanan (extended/virtual machine)

Secara spesifik berfungsi :

b.1 Memberi abstraksi mesin tingkat tinggi yang lebih sederhana dan menyembunyikan kerumitan perangkat keras.

Sistem operasi menyediakan system call (API=Application Programming Interface) yang berfungsi menghindarkan kompleksitas pemrograman dengan memberi sekumpulan instruksi yang mudah digunakan.

b.2 Basis untuk program lain.

Program aplikasi dijalankan di atas sistem operasi yang bertujuan untuk memanfaatkan dan mengendalikan sumber daya sistem komputer secara benar, efisien, dan mudah dengan meminta layanan sistem operasi.

Untuk memberikan fungsi-fungsi tersebut, sistem operasi mempunyai beberapa subsistem, antara lain :

- a. Manajemen proses
- b. Manajemen memori
- c. Manajemen berkas
- d. Manajemen perangkat masukan/keluaran
- e. Pengamanan sistem
- f. Sistem komunikasi
- g. Dan sebagainya

2.3. Sejarah Perkembangan Sistem Operasi

Perkembangan sistem komputer dibagi menjadi empat generasi termasuk perangkat keras dan perangkat lunaknya (sistem operasi).

Generasi Pertama (1945-1955)

Pada generasi ini belum ada sistem operasi, sistem komputer diberi instruksi yang harus dikerjakan secara langsung.

Generasi Kedua (1955-1965)

Job dikumpulkan dalam satu rangkaian kemudian dieksekusi secara berurutan. Sistem komputer belum dilengkapi sistem operasi, tapi beberapa fungsi dasar sistem operasi telah ada, misalnya FMS (Fortran Monitoring System) dan IBSYS, keduanya merupakan bagian yang fungsinya merupakan komponen sistem operasi.

Generasi Ketiga (1965-1980)

Dikembangkan untuk melayani banyak pemakai secara online, sehingga menuntut sistem komputer dapat digunakan secara :

- a. Multiuser
Berarti komputer yang memiliki resource yang dapat digunakan oleh banyak orang sekaligus.
- b. Multiprogramming
Berarti komputer melayani banyak proses/job sekaligus pada waktu bersamaan, yaitu dengan membagi (mempartisi) memori menjadi beberapa bagian dengan

satu bagian memori adalah satu job berbeda. Saat satu job menunggu operasi masukan/keluaran selesai, job lain dapat menggunakan proses. Teknik ini meningkatkan efisiensi pemroses yang juga memerlukan perangkat keras khusus untuk mencegah satu job mengganggu job lain. Karena pemakai-pemakai berinteraksi dengan komputer, komputer harus menanggapi permintaan-permintaan secara cepat, atau akan menyebabkan produktivitas pemakai menurun drastis. Untuk kebutuhan itu dikembangkan timesharing.

Timesharing

Varian dari multiprogramming, dimana tiap pemakai mempunyai satu terminal online dengan pemroses hanya memberi layanan pada pemakai yang aktif secara bergantian secara cepat. Para pemakai akan merasa dilayani terus-menerus, padahal sebenarnya digilir persatuan waktu yang singkat.

Karena sumber daya yang digunakan secara bersamaan akan sering menimbulkan bottleneck, maka dikembangkan spooling.

Spooling

Membuat peripheral seolah-olah dapat digunakan bersama-sama sekaligus, dapat diakses secara simultan, yaitu dengan cara menyediakan beberapa partisi memori. Saat terdapat permintaan layanan peripheral, langsung diterima dan data disimpan lebih dulu di memori yang disediakan (berupa antrian), kemudian dijadwalkan agar secara nyata dilayani oleh peripheral.

Generasi Keempat (1980-199x)

Sistem operasi yang dapat melayani banyak mode, yaitu mendukung batch processing, timesharing dan (soft) real time applications. Perkembangan dengan meningkatnya kemampuan komputer dekstop (PC) dan teknologi jaringan (TCP/IP).

2.4. Sistem komputer dalam pandangan perancang sistem operasi

Sistem operasi bertugas :

1. Menutupi atau mendandani perangkat keras komputer agar tampil indah, mudah dan nyaman bagi pemakai.

2. Menghindarkan rincian operasi perangkat keras dan menyediakan antarmuka untuk pemogram dalam menggunakan sistem.
3. Bertindak sebagai mediator, mempermudah pemogram dan program aplikasi mengakses dan menggunakan fasilitas dan sumber daya sistem komputer.

Perancang sistem operasi yang bertugas mendandani perangkat keras, dituntut mengetahui secara mendalam dan menyeluruh perangkat yang akan didandannya agar sistem operasi yang dirancang benar-benar berfungsi secara benar dan efisien.

2.5. Konsep dasar sistem operasi

Antarmuka (interface) antara sistem operasi dengan program aplikasi (user programs) dikenal sebagai extended instruction (perluasan instruksi).

Extended instruction dapat juga disebut sebagai panggilan sistem (system call).

a. Process

- Merupakan konsep utama dalam semua sistem operasi
- Pada dasarnya adalah sebuah program yang dieksekusi
- Berisi executable program, program's data dan stack, program counter, stack pointer dan register lainnya.
- Tabel proses (process table) adalah tabel yang berisi informasi tentang semua proses yang dijalankan.
- Sebuah proses dipanggil oleh command interpreter atau shell yang membaca perintah dari terminal.
- Child process dapat membuat satu atau lebih proses lainnya.

b. Files

- Sistem operasi mendukung konsep sebuah direktori.
- Hirarki proses dan file diorganisasi sebagai trees.
- Setiap proses memiliki direktori kerja.

c. System Call

Layanan langsung sistem operasi ke pemrogram adalah system call atau API (Application Programming Interface). System call adalah tata cara pemanggilan di program aplikasi untuk memperoleh layanan sistem operasi.

System call berupa rutin sistem operasi untuk keperluan tertentu yang spesifik. Bentuk system call beragam, terbanyak berupa rutin prosedur atau fungsi.

- User programs berkomunikasi dengan sistem operasi dan meminta layanan darinya dengan membuat system calls.
- Contoh system call READ dari bahasa C : `count=read(file, buffer, nbytes)`

d. Shell

- Sistem operasi adalah kode yang dilaksanakan system calls.
- Dalam UNIX command interpreter disebut shell.
- Dalam prompt UNIX, sebuah karakter tanda dollar (\$), memberitahu pemakai bahwa shell menunggu permintaan perintah.
- Jika user menuliskan : `$date`, memiliki arti bahwa shell membuat a child process dan menjalankan date program sebagai child. Selama child process dijalankan, shell menunggunya untuk dihentikan. Ketika child selesai, shell akan menampilkan prompt kembali dan mencoba membaca masukan perintah berikutnya.

2.6. Struktur dasar sistem operasi

a. Sistem monolitik (monolithic system)

Sistem operasi sebagai kumpulan prosedur dimana prosedur dapat saling dipanggil oleh prosedur lain di sistem bila diperlukan. Kernel berisi semua layanan yang disediakan sistem operasi untuk pemakai. Sistem operasi ditulis sebagai sekumpulan prosedur (a collection of procedures), yang dapat dipanggil setiap saat oleh pemakai saat dibutuhkan.

Kelemahan :

- 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 pemrograman satu bagian dari kernel menyebabkan matinya seluruh sistem.

Keunggulan :

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

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).

b. Sistem lapis (layered system)

Sistem operasi dibentuk secara hirarki berdasar lapisan-lapisan, dimana lapisan-lapisan bawa memberi layanan lapisan lebih atas.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.

Lapis-lapis dalam sistem operasi ada 6 lapis, 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-operatot 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,multiprogramming dan pengaturan prosessor.

Lapisan n memberi layanan untuk lapisan n+1. Proses-proses di lapisan n dapat meminta layanan lapisan n-1 untuk membangun 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.

Keunggulan :

- Memiliki semua keunggulan rancangan modular, yaitu sistem dibagi menjadi beberapa modul dan tiap modul dirancang secara independen. Tiap lapisan dapat dirancang, dikode dan diuji secara independen.
- Pendekatan berlapis menyederhanakan rancangan, spesifikasi dan implementasi sistem operasi.

Kelemahan :

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

c. Virtual machines (mesin maya)

- Multiprogramming
- Time sharing systems

Awalnya struktur ini membuat seolah-olah pemakai mempunyai seluruh komputer dengan simulasi atas pemroses yang digunakan. Sistem operasi melakukan simulasi mesin nyata. Mesin hasil simulasi digunakan pemakai, mesin maya merupakan tiruan seratus persen atas mesin nyata. Semua pemakai diberi ilusi mempunyai satu mesin yang sama-sama canggih.

Pendekatan ini memberikan fleksibilitas tinggi sampai memungkinkan sistem operasi-sistem operasi berbeda dapat dijalankan di mesin-mesin maya berbeda. Implementasi yang efisien merupakan masalah sulit karena sistem menjadi besar dan kompleks.

Teknik ini mulanya digunakan pada IBM S/370. VM/370 menyediakan mesin maya untuk tiap pemakai. Bila pemakai log (masuk) sistem, VM/370 menciptakan satu mesin maya baru untuk pemakai itu.

Teknik ini berkembang menjadi operating system emulator sehingga sistem operasi dapat menjalankan aplikasi-aplikasi untuk sistem operasi lain.

- Sistem operasi MS-Windows NT dapat menjalankan aplikasi untuk MS-DOS, OS/2 mode teks dan aplikasi Win16. Aplikasi tersebut dijalankan sebagai masukan bagi subsistem di MS-Windows NT yang mengemulasikan system calls yang dipanggil aplikasi dengan Win32 API (system calls di MS-Windows NT).
- IBM mengembangkan WABI yang mengemulasikan Win32 API sehingga diharapkan sistem operasi yang menjalankan WABI dapat menjalankan aplikasi-aplikasi untuk MS-Windows.
- Para sukarelawan pengembang Linux telah membuat DOSEMU agar aplikasi-aplikasi untuk MS-DOS dapat dijalankan di Linux, WINE agar aplikasi untuk MS-Windows dapat dijalankan di Linux, iBCS agar aplikasi-aplikasi untuk SCO-UNIX dapat dijalankan di Linux, dan sebagainya.

d. Client-server model

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 pemakai.
Kernel hanya mengatur komunikasi antara client dan server. Kernel yang ini populer dengan sebutan mikrokernel.
- Permintaan pelayanan, seperti membaca sebuah blok file, sebuah user process (disebut client process) mengirimkan permintaan kepada sebuah server process, yang kemudian bekerja dan memberikan jawaban balik.
- Keuntungan : kemampuan diadaptasi untuk digunakan dalam distributed system.

Masalah :

- Tidak semua tugas dapat dijalankan di tingkat pemakai (sebagai proses pemakai).

Kesulitan ini diatasi dengan :

- Proses server kritis tetap di kernel, yaitu proses yang biasanya berhubungan dengan perangkat keras.
- Mekanisme ke kernel seminimal mungkin, sehingga pengaksesan ruang pemakai dapat dilakukan dengan cepat.

Untuk sistem-sistem besar dengan banyak server dikehendaki supaya client transparan dalam meminta layanan sehingga tidak menyulitkan pemrogram.

Keunggulan :

- 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.

Kelemahan :

- Layanan dilakukan lambat karena harus melalui pertukaran pesan.
- Pertukaran pesan dapat menjadi bottleneck.

e. Sistem berorientasi Objek

Sistem operasi merealisasikan layanan sebagai kumpulan proses disebut sistem operasi bermodel proses. Pendekatan lain implementasi layanan adalah sebagai objek-objek. Sistem operasi 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. Contoh sistem operasi yang berorientasi objek, antara lain : eden, choices, x-kernel, medusa, clouds, amoeba, muse, dan sebagainya.

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

2.7. Ringkasan

Dalam bab ini telah membahas sistem operasi dari dua pandangan : resource managers dan extended machines. Dalam pandangan resource managers, tugas sistem operasi adalah mengatur bagian-bagian yang berbeda secara efisien dalam sistem. Dalam pandangan extended machines, tugas sistem operasi adalah menyediakan pemakai dengan virtual machine virtual machine yang sesuai digunakan daripada actual machine.

Pembahasan juga menjelaskan perkembangan komputer dan sistem operasi, dan mengetahui saat ini termasuk generasi berapa. Empat dekade telah ditinjau, dari vacuum tubes ke personal computer (PC). Dijelaskan pula dua konsep sistem

operasi, yaitu proses dan file. Penjabaran system calls dan memberikan contoh sederhana (READ).

Terakhir dijelaskan perbedaan struktur sistem operasi ; sebagai monolithic system, a hierarchy of layers, a virtual machine, dan client-server model.

Bab 3

PROSES

3.1. Pengertian proses (task/job)

Proses berisi instruksi, data, program counter, register pemroses, stack data, alamat pengiriman dan variabel pendukung lainnya.

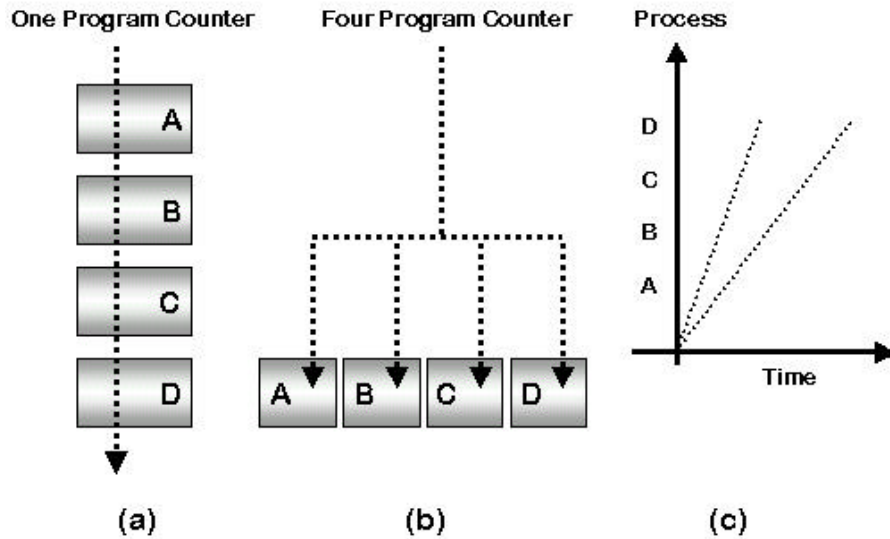
Terdapat beberapa definisi mengenai proses, antara lain :

- a. Merupakan konsep pokok dalam sistem operasi, sehingga masalah manajemen proses adalah masalah utama dalam perancangan sistem operasi.
- b. Proses adalah program yang sedang dieksekusi.
- c. Proses adalah unit kerja terkecil yang secara individu memiliki sumber daya dan dijadwalkan oleh sistem operasi.

Peran sistem operasi dalam kegiatan proses adalah mengelola semua proses di sistem dan mengalokasikan sumber daya ke proses tersebut. Banyak proses yang dijalankan bersamaan, dimana setiap proses mendapat bagian memori dan kendali sendiri-sendiri (peran SO), sehingga setiap proses (program) memiliki prinsip :

- a. Independent, artinya program-program tersebut berdiri sendiri, terpisah dan saling tidak bergantung.
- b. One program at any instant, artinya hanya terdapat satu proses yang dilayani pemroses pada satu saat.

Dalam multiprogramming, teknik penanganan proses adalah dengan mengeksekusi satu proses dan secara cepat beralih ke proses lainnya (bergiliran), sehingga menimbulkan efek paralel semu (*pseudoparallelism*).



Gambar 3.1 : (a). Multiprogramming dengan 4 proses (program/job)
 (b) : Model konseptual dari 4 proses independent, sequential proses
 (c) : Hanya 1 proses yang aktif pada suatu saat

3.2. Pengendalian proses

Dalam pengendalian antar proses, sistem operasi menggunakan metode :

a. Saling melanjutkan (interleave)

Sistem operasi harus dapat kembali melanjutkan proses setelah melayani proses lain.

b. Kebijakan tertentu

Sistem operasi harus mengalokasikan sumber daya ke proses berdasar prioritasnya.

c. Komunikasi antar proses dan penciptaan proses

Sistem operasi harus mendukung komunikasi dan penciptaan antar proses (menstrukturkan aplikasi).

Pada sistem dengan banyak proses aktif, proses-proses pada satu saat berada dalam beragam tahap eksekusinya. Proses mengalami beragam state (ready, running, blocked) selama siklus hidupnya sebelum berakhir dan keluar dari sistem. Sistem operasi harus dapat mengetahui state masing-masing proses dan merekam semua perubahan yang terjadi secara dinamis. Informasi tersebut digunakan untuk kegiatan penjadwalan dan memutuskan alokasi sumber daya.

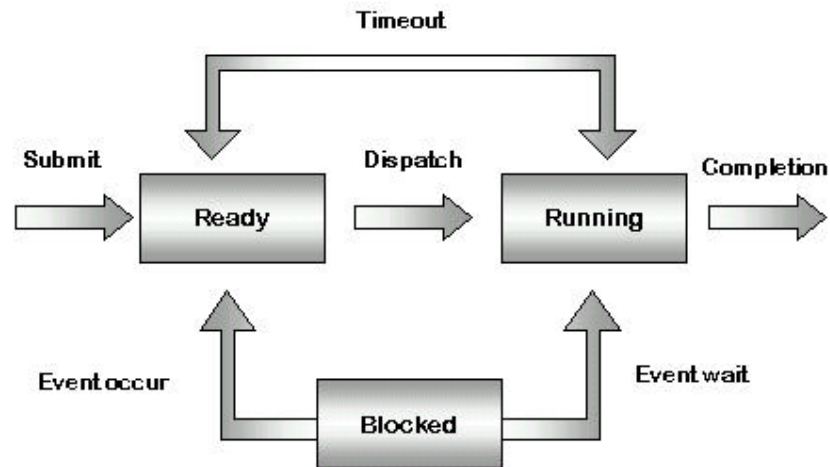
3.3 Status (state) proses

Sebuah proses akan mengalami serangkaian state diskrit. Beragam kejadian dapat menyebabkan perubahan state proses. Tiga state tersebut adalah sebagai berikut :

Tabel 3.1 : Tiga state dasar proses

| No | STATUS (STATE) | DESKRIPSI |
|----|----------------|--|
| 1 | Running | Proses sedang mengeksekusi instruksi proses |
| 2 | Ready | Proses siap dieksekusi, tetapi proses tidak tersedia untuk eksekusi proses ini. |
| 3 | Blocked | Proses menunggu kejadian untuk melengkapinya tugasnya Contoh : a. Selesinya operasi perangkat masukan/keluaran b. Tersedianya memory c. Tibanya pesan jawaban d. Dan sebagainya |

Hubungan ketiga state dasar digambarkan dalam diagram berikut :



Gambar 3.2 Diagram tiga state dasar proses

Penjelasan diagram :

1. Proses yang baru diciptakan akan mempunyai state ready.
2. Proses berstate running menjadi blocked, karena sumber daya yang diminta belum tersedia atau meminta layanan perangkat masukan/keluaran, sehingga menunggu kejadian muncul. Proses menunggu kejadian alokasi sumber daya atau selesainya layanan perangkat masukan/keluaran (event wait).
3. Proses berstate running menjadi ready, karena penjadwal memutuskan eksekusi proses lain karena jatah waktu untuk proses tersebut telah habis (time out).
4. Proses berstate blocked menjadi ready saat sumber daya yang diminta/ diperlukan telah tersedia atau layanan perangkat masukan/keluaran selesai (event occurs).
5. Proses berstate ready menjadi running, karena penjadwal memutuskan penggunaan pemroses untuk proses itu karena proses yang saat itu running berubah statenya (menjadi ready atau blocked) atau telah menyelesaikan sehingga disingkirkan dari sistem. Proses menjadi mendapatkan jatah pemroses.

3.4 Diagram state lanjut

Penundaan (suspend) adalah operasi penting dan telah diterapkan dengan beragam cara. Penundaan biasanya berlangsung singkat. Penundaan sering dilakukan sistem untuk memindahkan proses-proses tertentu guna mereduksi beban sistem selama beban puncak.

Proses yang ditunda (suspended blocked) tidak berlanjut sampai proses lain meresume. Untuk jangka panjang, sumber daya-sumber daya proses dibebaskan (dilucuti). Keputusan membebaskan sumber daya-sumber daya bergantung sifat masing-masing sumber daya. Memori utama seharusnya segera dibebaskan begitu proses tertunda agar dapat dimanfaatkan proses lain. Resuming (pengaktifan kembali) proses, yaitu menjalankan proses dari titik (instruksi) dimana proses ditunda.

Operasi suspend dan resume penting, sebab :

- a. Jika sistem berfungsi secara buruk dan mungkin gagal maka proses-proses dapat disuspend agar diresume setelah masalah diselesaikan.

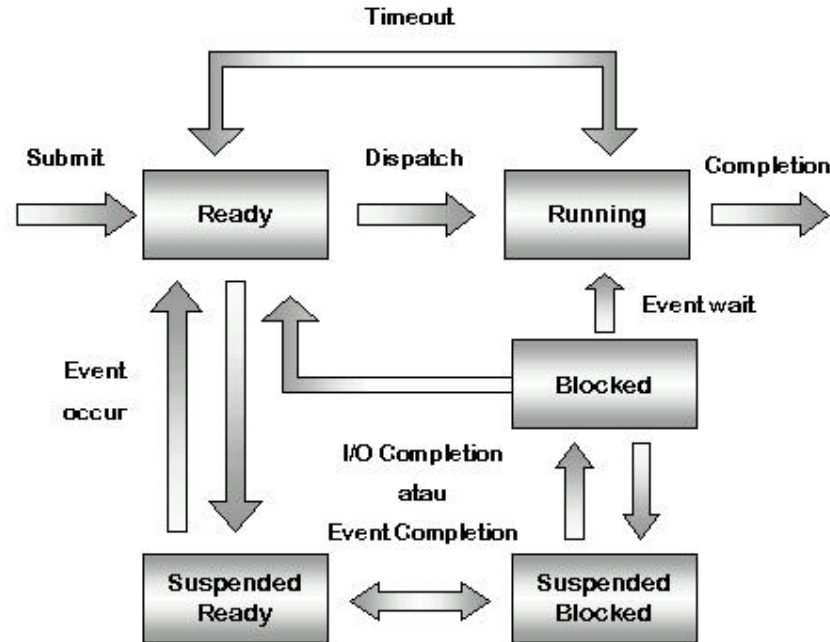
Contoh :

Pada proses pencetakan, bila tiba-tiba kertas habis maka proses disuspend.

Setelah kertas dimasukkan kembali, proses pun dapat diresume.

- b. Pemakai yang ragu/khawatir mengenai hasil proses dapat mensuspend proses (bukan membuang (abort) proses). Saat pemakai yakin proses akan berfungsi secara benar maka dapat me-resume (melanjutkan kembali di instruksi saat disuspend) proses yang disuspend.
- c. Sebagai tanggapan terhadap fluktuasi jangka pendek beban sistem, beberapa proses dapat disuspend dan diresume saat beban kembali ke tingkat normal.

Gambar berikut menunjukkan modifikasi diagram state dengan memasukkan kejadian suspend dan resume.



Gambar 3.3 : Diagram lima state proses

Dua state baru dimasukkan sehingga membentuk diagram 5 state, yaitu :

1. Suspended ready
2. Suspended blocked

Penundaan dapat diinisialisasi oleh proses itu sendiri atau proses lain.

- a. Pada sistem monoprocessor, proses running dapat mensuspend dirinya sendiri karena tak ada proses lain yang juga running yang dapat memerintahkan suspend.
- b. Pada sistem multiprocessor, proses running dapat disuspend proses running lain pada pemroses berbeda. Proses ready hanya dapat di suspend oleh proses lain.

Pada proses blocked terdapat transisi menjadi suspended blocked. Pilihan ini dirasa aneh. Apakah tidak cukup menunggu selesainya operasi masukan/keluaran atau kejadian yang membuat proses ready atau suspended ready ?. Bukankah state blocked, ready, blocked, suspended, blocked sama-sama tidak mendapat jatah waktu pemroses ?. Kenapa dibedakan ?.

Alasannya, karena penyelesaian operasi masukan/keluaran bagi proses blocked mungkin tak pernah terjadi atau dalam waktu tak terdefiniskan sehingga lebih baik disuspend agar sumber daya-sumber daya yang dialokasikan untuk proses tersebut dapat digunakan proses-proses lain. Untuk kondisi ini, lebih baik sumber daya-sumber daya yang dipegang proses yang berkondisi seperti ini dipakai proses-proses lain. Proses blocked disuspend sistem atau secara manual menjadi suspended blocked.

Bila akhirnya operasi masukan/keluaran berakhir maka segera proses suspended blocked mengalami transisi. Karena resume dan suspend mempunyai prioritas tinggi maka transisi segera dilakukan. Suspend dan resume dapat digunakan untuk menyeimbangkan beban sistem saat mengalami lonjakan di atas normal.

3.5. Program Control Block (PCB)

Struktur data PCB menyimpan informasi lengkap mengenai proses sehingga dapat terjadi siklus hidup proses. Sistem operasi memerlukan banyak informasi mengenai proses guna pengelolaan proses. Informasi ini berada di PCB. Sistem berbeda akan mengorganisasikan secara berbeda.

Informasi dalam PCB :

1. Informasi identifikasi proses

Informasi ini berkaitan dengan identitas proses yang berkaitan dengan tabel lainnya.

Informasi tersebut meliputi :

- a. Identifier proses
- b. Identifier proses yang menciptakan
- c. Identifier pemakai

2. Informasi status pemroses

Informasi tentang isi register-register pemroses. Saat proses berstatus running, informasi tersebut berada di register-register. Ketika proses diinterupsi, semua informasi register harus disimpan agar dapat dikembalikan saat proses

dieksekusi kembali. Jumlah dan jenis register yang terlibat tergantung arsitektur komputer.

Informasi status terdiri dari :

a. Register-register yang terlihat pemakai

Adalah register-register yang dapat ditunjuk instruksi bahasa assembly untuk diproses pemroses.

b. Register-register kendali dan status

Adalah register-register yang digunakan untuk mengendalikan operasi pemroses.

c. Pointer stack

Tiap proses mempunyai satu atau lebih stack, yang digunakan untuk parameter atau alamat prosedur pemanggil dan system call. Pointer stack menunjukkan posisi paling atas dari stack.

3. Informasi kendali proses

Adalah informasi lain yang diperlukan sistem operasi untuk mengendalikan dan koordinasi beragam proses aktif.

Informasi kendali terdiri dari :

a. Informasi penjadwalan dan status

Informasi-informasi yang digunakan untuk menjalankan fungsi penjadwalan, antara lain :

a.1 Status proses

Mendefinisikan keadaan/status proses (running, ready, blocked)

a.2 Prioritas

Menjelaskan prioritas proses.

a.3 Informasi berkaitan dengan penjadwalan

Berkaitan dengan informasi penjadwalan, seperti lama menunggu, lama proses terakhir dieksekusi.

a.4 Kejadian

Identitas kejadian yang ditunggu proses.

b. Penstrukturan data

Satu proses dapat dikaitkan dengan proses lain dalam satu antrian atau ring, atau struktur lainnya. PCB harus memiliki pointer untuk mendukung struktur ini.

c. Komunikasi antar proses

Beragam flag, sinyal dan pesan dapat diasosiasikan dengan komunikasi antara dua proses yang terpisah.

d. Manajemen memori

Bagian yang berisi pointer ke tabel segmen atau page yang menyatakan memori maya (virtual memori) proses.

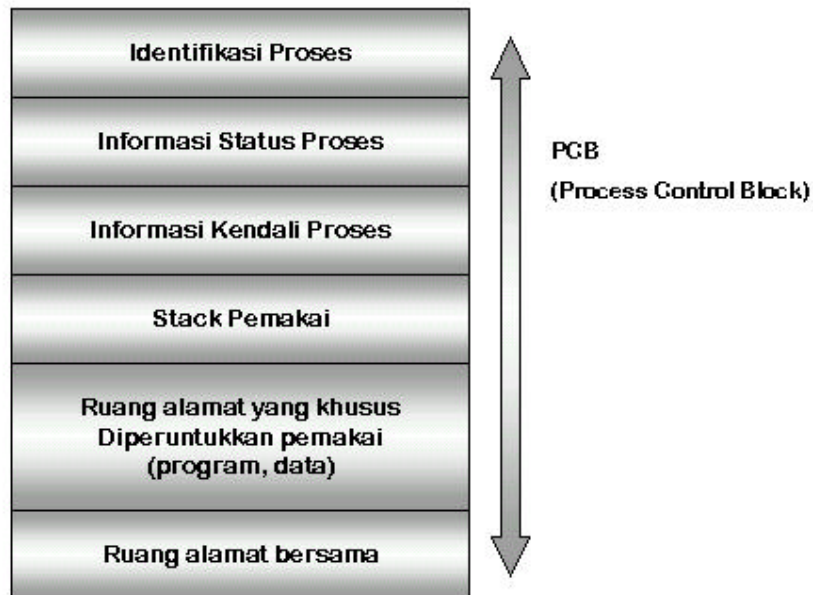
e. Kepemilikan dan utilisasi sumber daya

Sumber daya yang dikendalikan proses harus diberi tanda, misalnya :

- e.1 Berkas yang dibuka
- e.2 Pemakaian pemroses
- e.3 Pemakaian sumber daya lainnya

Informasi ini diperlukan oleh penjadwal.

Struktur citra proses digambarkan berurutan di satu ruang alamat. Implementasi penempatan citra proses yang sesungguhnya bergantung skema manajemen memori yang digunakan dan organisasi struktur kendali sistem operasi.



Gambar 3.4 : Proses pemakai

3.6. Operasi-operasi pada 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

3.7 Penciptaan 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.

Proses dapat menciptakan proses baru yaitu anak proses (child process), sedangkan proses yang menciptakannya disebut proses induk (parent process).

Proses anakpun kembali dapat menciptakan proses-proses anak lainnya.

Proses-proses dapat membentuk pohon hirarki proses.

3.8 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.

3.9. Penghancuran 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).

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.

Tabel 3.2 : Alasan-alasan penghancuran proses

| No | PENYEBAB TERMINASI | KETERANGAN |
|----|---|--|
| 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. Terdapat banyak kemungkinan untuk tipe waktu yang diukur, termasuk waktu total yang dijalani (wait clock time), jumlah waktu yang dipakai untuk eksekusi, dan jumlah waktu sejak pemakai terakhir kali memberi masukan (pada proses interaktif). |
| 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 | Proses berusaha menggunakan sumber data atau file yang tidak diijinkan dipakainya, atau proses mencoba menggunakannya tidak untuk peruntukannya, seperti menulis file read only. |
| 6 | Terjadi kesalahan aritmatika | Proses mencoba perhitungan terlarang, seperti pembagian dengan nol, atau mencoba menyimpan angka yang lebih besar daripada yang dapat diakomodasi oleh perangkat keras |
| 7 | Waktu telah kedaluwarsa | Proses telah menunggu lebih lama dari pada maksimum yang ditentukan untuk terjadinya suatu kejadian spesifik. |

| | | |
|----|--|---|
| 8 | Terjadi kegagalan masukan/keluaran | Kesalahan muncul pada masukan/keluaran, seperti ketidakmampuan menemukan file, kegagalan membaca atau menulis setelah sejumlah maksimum percobaan yang ditentukan (area rusak didapatkan pada tape atau operasi tak valid seperti membaca dari line printer). |
| 9 | Instruksi yang tidak benar | Proses berusaha mengeksekusi intruksi yang tak ada (akibat percabangan ke daerah data dan dan berusaha mengeksekusi data itu) |
| 10 | Terjadi usaha memakai instruksi yang tidak diijinkan | Proses berusaha menggunakan instruksi yang disimpan untuk sistem operasi. |
| 11 | Kesalahan penggunaan data | Bagian data adalah tipe yang salah atau tidak diinisialisasi. |
| 12 | Diintervensi oleh sistem operasi atau operator | Untuk suatu alasan, operator atau sistem operasi mengakhiri proses (terjadi deadlock) |
| 13 | Berakhirnya proses induk | Ketika parent berakhir, sistem operasi mungkin dirancang secara otomatis mengakhiri semua anak proses dari parent itu. |
| 14 | Atas permintaan dari proses induk | Parent process biasanya mempunyai otoritas mengakhiri suatu anak proses. |

3.10. Pengalihan proses

Kelihatannya pengalihan proses (process switching) adalah sepele. Pada suatu saat, proses running diinterupsi dan sistem operasi memberi proses lain state running dan menggilir kendali ke proses itu.

Dalam hal ini muncul beberapa masalah, yaitu :

1. Kejadian-kejadian apa yang memicu alih proses ?

2. Masalah lain adalah terdapatnya perbedaan antara alih proses (process switching) dan alih konteks (context switching).
3. Apa yang harus dilakukan sistem operasi terhadap beragam struktur data yang dibawah kendalinya dalam alih proses ?

3.11. Kejadian-kejadian penyebab pengalihan proses

Kejadian-kejadian yang menyebabkan terjadinya alih proses adalah :

1. Interupsi sistem

Disebabkan kejadian eksternal dan tak bergantung proses yang saat itu sedang running. Contoh : selesainya operasi masukan/keluaran.

Pada kejadian interupsi, kendali lebih dulu ditransfer ke interrupt handler yang melakukan penyimpanan data-data dan kemudian beralih ke rutin sistem operasi yang berkaitan dengan tipe interupsi itu.

Tipe-tipe interupsi antara lain :

a. Interupsi clock (clock interrupt)

Sistem operasi (penjadwal) menentukan apakah proses yang sedang running telah mengeksekusi selama jatah waktunya. Jika telah mencapai jatahnya maka proses dialihkan ke state ready dan proses lain dijadwalkan running.

b. Interupsi masukan/keluaran

Kejadian dimana peralatan masukan/keluaran melakukan interupsi meminta layanan sistem operasi. Sistem operasi segera menentukan aksi-aksi masukan/keluaran yang harus dilakukan.

c. Page/memory fault

Pemroses menemui pengacuan alamat memori maya yang tidak terdapat di memori utama (fisik). Sistem operasi segera memerintahkan untuk mengambil page yang terdapat alamat yang dimaksud untuk dipindah ke memori utama.

2. Trap

Adalah interupsi karena terjadinya kesalahan atau kondisi kekecualian (exception conditions) yang dihasilkan proses yang running, seperti usaha illegal dalam mengakses file.

Dengan trap, sistem operasi menentukan apakah kesalahan yang dibuat merupakan kesalahan fatal ?

- a. Jika fatal, proses yang saat itu running disingkirkan dan terjadi alih proses.
- b. Jika kesalahan tidak fatal maka bergantung sifat kesalahan dan rancangan sistem operasi.

Kemungkinan yang dilakukan adalah menjalankan prosedur pemulihan atau memperingatkan ke pemakai. Saat terjadi trap, mungkin terjadi pengalihan proses mungkin pula resume proses.

3. Supervisor call

Yaitu panggilan meminta atau mengaktifkan bagian sistem operasi.

Contoh :

Proses pemakai running meminta layanan masukan/keluaran seperti membuka file. Panggilan ini menghasilkan transfer ke rutin bagian sistem operasi.

Biasanya, penggunaan system call membuat proses pemakai blocked karena diaktifkan proses kernel (sistem operasi).

3.12. Pengalihan konteks

Pengalihan konteks dapat terjadi tanpa pengalihan state process yang sedang running, sedang pengalihan proses pasti melibatkan juga pengalihan konteks.

Siklus penanganan interupsi adalah :

1. Pemroses menyimpan konteks program saat itu yang sedang dieksekusi ke stack.
2. Pemroses menset register PC dengan alamat awal program untuk interuppt handler.

Setelah kedua aktivitas itu, pemroses melanjutkan menjalankan instruksi-instruksi berikutnya di interuppt handler yang melayani interrupt.

Pelaksanaan interupsi ini belum tentu mengakibatkan pengalihan ke proses lain (yaitu pengalihan PCB proses dari senarai running ke senarai lain (blocked, ready),

dan sebaliknya. Kita menyebut pengalihan konteks adalah untuk pengalihan sementara yang singkat, misalnya untuk mengeksekusi program interrupt handler.

Setelah penanganan interupsi selesai maka konteks yang terdapat pada stack dikembalikan sehingga kembali ke konteks proses semula tanpa terjadi pengalihan ke proses lain. Pengalihan proses terjadi jika proses yang running beralih menjadi state lain (ready, blocked), kemudian sistem operasi harus membuat perubahan-perubahan berarti terhadap lingkungannya. Rincian-rincian dalam pelaksanaan pengalihan proses dibahas setelah ini.

3.13. Pengalihan proses

Pengalihan proses terjadi jika proses yang running beralih menjadi state lain (ready, blocked) kemudian sistem operasi membuat perubahan-perubahan berarti terhadap lingkungan.

Langkah-langkah yang terlibat dalam pengalihan proses sebagai berikut :

1. Simpan konteks pemroses, termasuk register PC dan register-register lain.
2. Perbarui PCB proses yang running. Pelaksanaan termasuk mengubah state proses menjadi salah satu state (ready, blocked, suspendedready).
3. Field-field yang relevan juga diperbarui misalnya alasan meninggalkan state running dan informasi akunting.
4. Pindahkan PCB proses ke senarai yang cocok (ready, blocked).
5. Pilih satu proses lain untuk dieksekusi sesuai dengan teknik penjadwalan.
6. Perbarui PCB proses yang dipilih termasuk perubahan state menjadi running.
7. Perbarui struktur-struktur data manajemen memori. Pekerjaan ini sesuai dengan pengelolaan translasi alamat.
8. Kembalikan konteks pemroses dengan konteks simpanan yang memberitahu konteks proses terakhir saat dialihkan dari state running. Pengembalian konteks ini dilakukan dengan memuatkan nilai-nilai register PC dan register-register lain dengan nilai konteks yang tersimpan.

Pengalihan proses melibatkan pengalihan konteks dan perubahan state, memerlukan usaha lebih besar daripada pengalihan konteks.

3.14. Tabel-tabel proses

Tiap proses mempunyai state yang perlu diperhatikan sistem operasi yang dicatat dalam beragam tabel atau senarai yang saling berhubungan, yaitu :

a. Tabel informasi manajemen memori

Untuk menjaga keutuhan memori utama dan memori sekunder yang menyimpan informasi tentang :

- a.1 Alokasi memori utama yang dipakai proses.
- a.2 Alokasi memori sekunder yang dipakai proses (menggunakan manajemen memori dengan swapping).
- a.3 Atribut segmen memori utama dan sekunder.
- a.4 Informasi-informasi lain yang digunakan untuk pengelolaan memori.

b. Tabel informasi manajemen masukan/keluaran

Untuk mengelola perangkat masukan/keluaran, dimana perangkat tersebut digunakan proses tertentu, sehingga perlu dijaga agar proses lain tidak memakainya. Sistem operasi perlu mengetahui status operasi masukan/keluaran dan lokasi memori utama yang digunakan untuk transfer data.

c. Tabel informasi sistem file

Berisi informasi mengenai ekstensi file, lokasi pada memori sekunder, status saat itu dan menyimpan atribut-atribut file lainnya.

d. Tabel proses

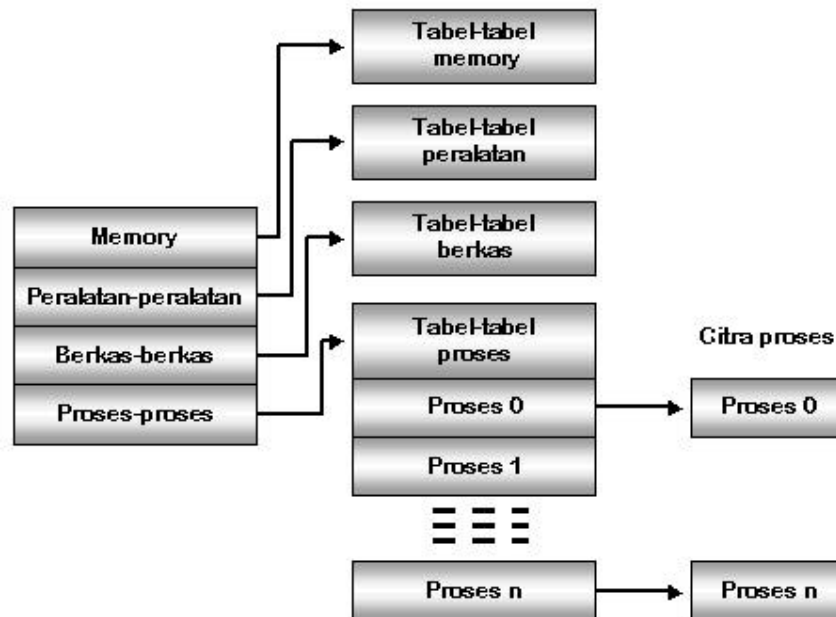
Untuk mengelola informasi proses di sistem operasi, lokasinya di memori, status dan atribut proses lainnya.

Proses ditempatkan di memori utama di lokasi tertentu, proses mempunyai satu ruang alamat tersendiri. Ruang alamat yang digunakan proses disebut citra proses (process image), karena selain seluruh kode biner program, proses ditambah atribut-atribut lain yang berkaitan penempatannya pada suatu lokasi memori dan status eksekusi pada saat itu.

Tabel 3.3 : Elemen-elemen citra proses

| No | ELEMEN CITRA PROSES | KETERANGAN |
|----|-----------------------------|--|
| 1 | Data pemakai | Bagian yang dapat memodifikasi berupa data program, daerah stack pemakai. |
| 2 | Program pemakai | Program biner yang dieksekusi. |
| 3 | Stack sistem | Digunakan untuk menyimpan parameter dan alamat pemanggilan untuk prosedur dan system calls |
| 4 | PCB (Program Control Block) | Berisi informasi yang diperlukan oleh sistem operasi dalam mengendalikan proses |

Struktur umum tabel-tabel kendali ditunjukkan pada gambar berikut :

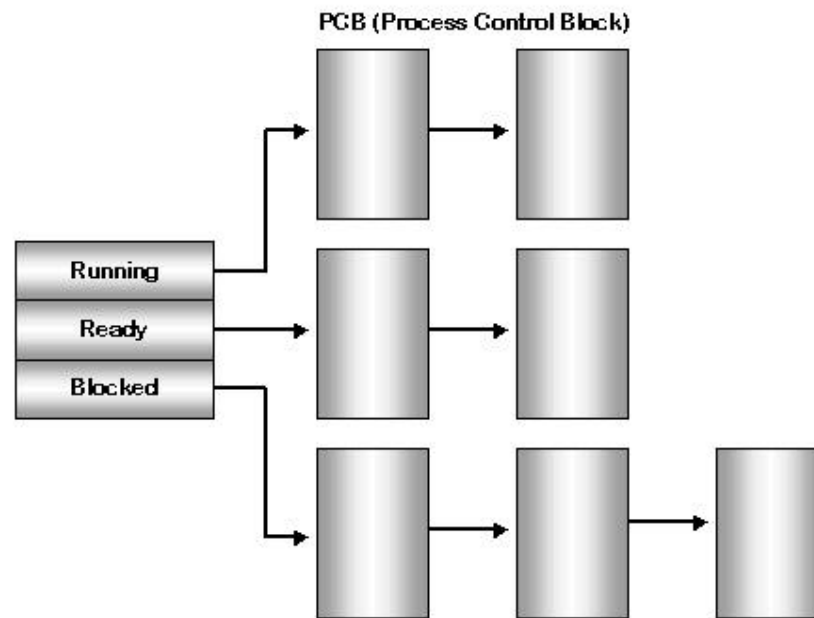


Gambar 3.5 : Struktur tabel-tabel kendali pada sistem operasi

3.15. PCB dan senarai proses

PCB berisi informasi mengenai proses yang diperlukan sistem operasi. PCB dibaca dan /atau dimodifikasi rutin sistem operasi seperti penjadwalan, alokasi sumber daya, pemrosesan interupsi, monitoring dan analisis kinerja. Kumpulan PCB mendefinisikan state sistem operasi.

Untuk menyatakan senarai proses di sistem operasi dibuat senarai PCB.



Gambar 3.6: Senarai PCB

Diagram memperlihatkan hanya satu PCB berada di senarai running. PCB ini menyatakan proses yang saat itu sedang dieksekusi pemroses sehingga hanya satu proses yang running. Tentu saja ini tidak berlaku untuk multiprocessing yang dapat mengeksekusi lebih dari satu proses sekaligus.

Prose-proses ready digambarkan dengan PCB proses-proses di senarai ready. Proses-proses menunggu dijadwalkan untuk dieksekusi pemroses. Proses yang dijadwalkan dieksekusi (yaitu mengalami transisi dari state ready menjadi running) maka PCBnya dipindah dari senarai ready ke senarai running.

Proses running (PCB-nya berada di senarai running) dipindah sesuai state yang dialami proses itu, sebagai berikut :

- a. Bila proses berakhir (selesai) maka dijalankan operasi terminasi sehingga PCB-nya tak ada lagi.
- b. Bila proses diblocked karena menunggu alokasi sumber daya maka PCBnya dipindah ke senarai blocked.
- c. Bila proses dijadwalkan habis jatah waktu eksekusinya maka PCBnya dipindahkan ke senarai ready.

Proses yang sedang blocked berpindah menjadi ready bila sumber daya yang ditunggu telah teralokasi untuknya. Untuk itu PCBnya dipindahkan ke senarai ready.

3.16. Pengaksesan informasi di PCB

Rutin-rutin sistem operasi perlu mengakses informasi di PCB. Tiap proses dilengkapi ID unik yang digunakan sebagai indeks (penunjuk) ke tabel untuk mengambil PCB.

Kesulitan bukan pada mekanisme pengaksesan, tetapi masalah proteksi terhadap PCB. Dua masalah utama proteksi terhadap PCB, yaitu :

1. Bug (kesalahan pemrograman) pada rutin tunggal, misalnya interrupt handler dapat merusak PCB sehingga dapat berakibat menghancurkan kemampuan sistem mengelola proses-proses yang diasosiasikan dengan PCB.
2. Perubahan rancangan struktur dan semantiks PCB dapat berdampak ke sejumlah modul sistem operasi yang memakai PCB.

Kedua masalah tersebut memberi gagasan agar semua rutin sistem operasi melewati satu rutin khusus, yaitu rutin penanganan PCB dalam mengakses PCB.

Tugas rutin adalah memproteksi PCB dan menjadi perantara pembacaan dan penulisan PCB.

- a. Masalah pertama dapat dicegah karena rutin penanganan PCB akan selalu menjaga agar PCB tidak rusak.
- b. Masalah kedua jelas langsung teratasi karena antarmuka terhadap rutin-rutin lain masih tetap dipertahankan walau rincian-rincian PCB diubah.

Rutin-rutin sistem operasi yang memakai antarmuka tidak perlu diubah. Teknik ini menghendaki didefinisikan antarmuka rutin penanganan PCB dan rutin-rutin lain dengan baik. Kelemahan teknik ini adalah adanya overhead kinerja karena harus memanggil rutin penanganan PCB. Pengaksesan langsung terhadap PCB tentu lebih cepat daripada harus memanggil rutin penanganan PCB.

3.17. Kedudukan sistem operasi

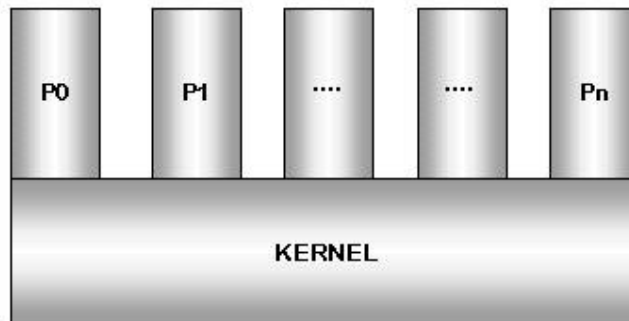
Sistem operasi pada dasarnya adalah seperti perangkat lunak lain, yaitu program yang perlu dieksekusi pemroses.

Kedudukan sistem operasi dibanding proses-proses lain, adalah :

- a. Sistem operasi sebagai kernel tersendiri yang berbeda dengan proses-proses lain (kernel sebagai non-proses).
- b. Fungsi-fungsi sistem operasi dieksekusi dalam proses pemakai.
- c. Sistem operasi juga sebagai kumpulan proses (process based operating systems).

3.18. Kernel sebagai non proses

Kernel sistem operasi adalah di luar proses, digambarkan pada gambar berikut :



Gambar 3.7 : Eksekusi kernel sebagai non-proses

Ketika proses running diinterupsi atau memanggil system call, maka konteks pemroses proses ini disimpan dan kendali dilewatkan ke kernel. Sistem operasi mempunyai daerah memori dan stack sendiri untuk pemanggilan prosedur.

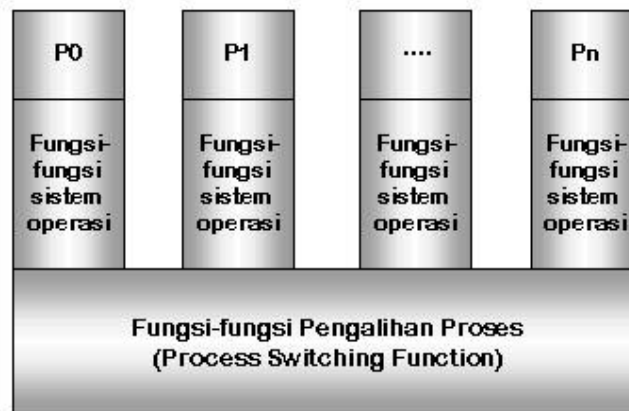
Sistem operasi melakukan fungsi yang diinginkan dan mengembalikan konteks proses yang diinterupsi. Eksekusi proses pemakai yang diinterupsi dilanjutkan. Alternatif lain, sistem operasi menyimpan lingkungan proses, melakukan penjadwalan dan menjadwalkan proses lain.

Konsep proses hanya diterapkan untuk program-program pemakai. Kode sistem operasi dieksekusi sebagai satu entitas terpisah, beroperasi pada mode kernel. Proses adalah non-kernel, sedang sistem operasi adalah kernel yang bukan proses.

3.19. Dieksekusi dalam proses pemakai

Alternatif lain dieksekusi sistem operasi adalah mengeksekusi sistem operasi di konteks proses pemakai. Pendekatan ini didasarkan terutama pada pandangan bahwa sistem operasi sebagai kumpulan rutin yang dipanggil pemakai untuk melakukan beragam fungsi dan dieksekusi dalam lingkungan proses pemakai.

Pendekatan ini digambarkan pada gambar berikut :



Gambar 3.8 : Eksekusi dalam proses pemakai

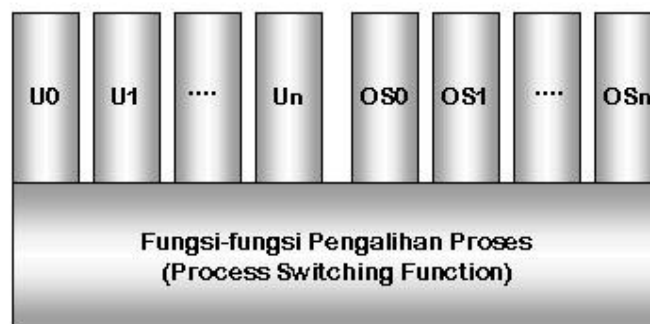
Pada seluruh waktu, sistem operasi mengelola N citra proses. Tiap citra tidak hanya mempunyai daerah untuk proses tapi juga daerah program, data dan stack untuk kernel. Terdapat juga ruang alamat yang dipakai bersama semua proses. Ketika diinterupsi, trap atau supervisor call terjadi, pemroses ditempatkan ke mode kernel dan kendali dilewatkan ke sistem operasi. Konteks pemroses disimpan dan alih konteks ke rutin sistem operasi.

Eksekusi dilanjutkan dalam proses pemakai saat itu, tidak dilakukan alih proses, hanya alih konteks di proses yang sama. Jika sistem operasi telah menyelesaikan tugas, menentukan apakah proses berlanjut, maka alih konteks meresume program yang diinterupsi dalam proses itu juga. Keunggulan pendekatan ini adalah program pemakai yang diinterupsi untuk memperoleh rutin sistem operasi dan diresume tidak mengalami overhead peralihan dua proses.

Jika sistem operasi menentukan bahwa alih proses terjadi bukan kembali ke proses semula yang dieksekusi, maka kendali dilewatkan ke rutin alih proses. Rutin ini boleh dijalankan pada proses boleh juga tidak, bergantung rancangan sistem. Pada keadaan ini, proses saat itu menjadi state non-running dan proses lain menjadi running.

3.20. Sistem operasi sebagai kumpulan proses

Pendekatan ini mengimplementasikan sistem operasi sebagai kumpulan proses. Pendekatan ini digambarkan pada gambar berikut :



Gambar 3.9: Eksekusi sistem operasi sebagai proses

Variannya adalah perangkat lunak bagian kernel dieksekusi dalam mode kernel. Fungsi-fungsi kernel utama diorganisasi sebagai proses-proses terpisah. Terdapat kode kecil pengalihan proses yang dieksekusi di luar proses.

Pendekatan ini mempunyai beberapa keunggulan, yaitu :

- a. Perancangan modular dapat diterapkan ke perancangan sistem operasi. Sistem operasi menjadi modul-modul dengan interface yang bersih dan minimal antar modul-modul itu.
- b. Fungsi-fungsi sistem operasi tak kritis secara bagus dapat diimplementasikan sebagai proses-proses terpisah.
- c. Mudah diterapkan pada lingkungan multiprocessor atau multicomputer dimana beberapa layanan sistem operasi ditaruh di pemroses-pemroses tertentu. Teknik ini akan meningkatkan kinerja.

3.21. Mikrokernel

Saat ini, mikrokernel mendapat banyak perhatian. Mikrokernel adalah inti sistem operasi yang menyebabkan landasan perluasan sistem operasi. Pendekatan mikrokernel dipopularkan sistem operasi MACH. Secara teoritis, pendekatan mikrokernel menyediakan derajat fleksibilitas dan modularitas tinggi. Sistem operasi yang memakai pendekatan mikrokernel adalah MS Windows NT. Landasan pendekatan mikrokernel adalah hanya fungsi-fungsi sistem operasi inti yang secara mutlak esensi yang harus berada di kernel.

Layanan-layanan dan aplikasi-aplikasi yang kurang esensi dibangun di atas mikrokernel itu. Meskipun pembagian antara yang perlu dan tidak perlu ada di mikrokernel beragam. Terdapat ciri yang sama yaitu banyak layanan yang secara tradisional merupakan bagian sistem operasi menjadi subsistem eksternal. Subsistem ini berinteraksi dengan kernel dan subsistem-subsistem lain.

Layanan-layanan itu antara lain sistem file, sistem windowing dan layanan-layanan keamanan. Komponen-komponen sistem operasi di luar mikrokernel saling

berinteraksi melalui pesan yang dilewatkan melalui mikrokernel. Fungsi mikrokernel adalah sebagai mediator pertukaran pesan.

Mikrokernel memvalidasi pesan, melewatkan pesan antara komponen-komponen dan memberi hak pengaksesan perangkat keras. Struktur ini ideal untuk lingkungan pemrosesan terdistribusi karena mikrokernel dapat melewatkan pesan baik secara lokal atau jarak jauh tanpa perubahan komponen-komponen sistem operasi yang lain.