

**MATA KULIAH SISTEM OPERASI
KODE MATA KULIAH CCS210**

**DISUSUN OLEH
NIZIRWAN ANWAR & TEAM**

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL
JAKARTA
20181**

MATERI
STRUKTUR SISTEM OPERASI

1. Pendahuluan

Secara umum, Sistem Operasi adalah software pada lapisan pertama yang ditempatkan pada memori komputer pada 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.



Gambar 1 Hirarki Sistem Operasi

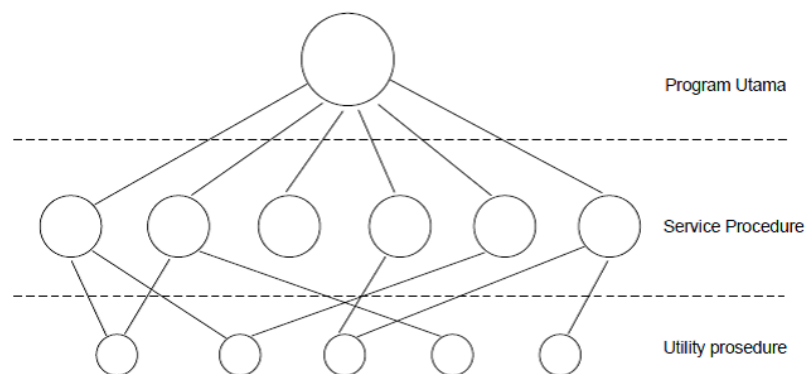
2. Model Struktur Sistem Operasi

Suatu sistem operasi modern merupakan suatu sistem yang besar dan kompleks sehingga strukturnya harus dirancang dengan hati-hati dan saksama supaya dapat berfungsi seperti yang diinginkan serta dapat dimodifikasi dengan mudah. Struktur sistem operasi merupakan komponen-komponen sistem operasi yang dihubungkan dan dibentuk di dalam kernel ¹. Ada beberapa struktur sistem operasi dan pernah dicoba, diantaranya sebagai berikut:

- (1) Sistem Monolithic
- (2) Sistem Lapisan
- (3) Mesin Virtual
- (4) Model Client Server

beberapa struktur system operasi diatas dapat diuraikan sebagai berikut ;

2.1 Sistem monolithic (*Monolithic System*)



Gambar 2 Sistem Monolithic

Sistem Monolitik ini merupakan sistem yang paling umum di dalam struktur sistem operasi. Pendekatan sistem monolitik ini biasa juga disebut dengan "The Big Mess". Sistem ini pada dasarnya tidak memiliki struktur. Sistem operasi ditulis sebagai sekumpulan prosedur, yang mana dapat dipanggil oleh yang lain pada saat prosedur itu diperlukan. Pada saat teknik ini digunakan, tiap prosedur dalam sistem mempunyai penjelasan tersendiri dari tiap parameter dan hasil yang ada. Masing-masing dapat secara bebas

¹ Iwan Binanto (2005). *Sistem Operasi*. Yogyakarta: Andi. hlm. 47-54

memanggil satu sama lain, apabila terdapat beberapa komputasi yang berguna dari kebutuhan yang sebelumnya.

Di dalam Sistem Monolitik, terdapat struktur yang kecil, yaitu Systemcalls. Systemcalls ini disediakan oleh sistem operasi yang mana permintaannya disediakan dengan meletakkan parameter-parameter yang tersedia, seperti register atau pada stack dan kemudian mengeksekusi setiap instruksi yang dikenal sebagai kernel call atau supervisor call.

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.

Struktur Dasar dari Sistem Monolitik

Didalam sistem monolitik terdapat struktur dasar untuk sistem operasi, yaitu :

- (a) Program utama membuat permintaan dari service procedure
- (b) Sekumpulan service procedure membawa keluar system call
- (c) Sekumpulan utilitas procedure membantu service procedure

Pada model ini, untuk setiap system call hanya terdapat satu service procedure yang ada. Prosedur utilitas sendiri berfungsi untuk memenuhi kebutuhan dari beberapa service procedure, seperti mengambil data dari program user. Divisi ini dibagi menjadi 3 layer, yaitu :

Kelemahan dari Sistem Monolitik

Ada beberapa kelemahan dari sistem monolitik, yaitu :

- (a) Disappearance dan pengujian dari eror sangatlah sulit, karena sistem tidak dapat dipisahkan dan dilokalisasi.
- (b) Sistem sulit untuk menyediakan fasilitas pengamanan (security).
- (c) Merupakan pemborosan yang sangat besar apabila setiap komputer harus menjalankan kernel monolitik. Sementara sistem tersebut sebenarnya tidak memerlukan seluruh layanan yang disediakan oleh kernel. (Tidak Fleksibel)
- (d) Apabila adanya kesalahan pemrograman pada suatu bagian dari kernel, maka menyebabkan matinya seluruh sistem.

Sedangkan untuk keunggulan dari system monolitik adalah layanan sistem monolitik dapat dilakukan sangat cepat karena terdapat pada satu address space.

Dengan demikian, itu adalah pembahasan mengenai Sistem Monolitik (*Monolithic System*). Sistem Monolitik merupakan salah satu sistem yang terdapat pada struktur sistem operasi. Sistem ini biasanya merupakan sistem yang paling umum yang menggunakan address space di dalam penggunaannya.

2.2 Sistem lapisan

Sistem pertama yang dibangun dengan sistem lapisan adalah THE yang dibuat di **Technische Hogeschool Eindhoven di Belanda pada tahun 1968 oleh E.W. Dijkstra** dan para mahasiswanya. Sistem ini mempunyai 6 (enam) lapis ²

Contoh lain adalah sistem Venus yang mempunyai tujuh lapisan. Lapisan bawah (0 sampai 4) digunakan oleh penjadwalan CPU dan manajemen memori yang kemudian diletakkan dalam suatu microcode. Pengaturan ini memberikan keuntungan, seperti eksekusi yang lebih cepat dan interface yang didefinisikan menjadi lebih jelas antara lapisan microcode dengan lapisan yang lebih tinggi.

<u>Lapis-5</u>	:	<u>user program</u>
<u>Lapis-4</u>	:	<u>buffering untuk I/O device</u>
<u>Lapis-3</u>	:	<u>operator-console device driver</u>
<u>Lapis-2</u>	:	<u>menejemen memori</u>
<u>Lapis-1</u>	:	<u>penjadwalan CPU</u>
<u>Lapis-0</u>	:	<u>hardware</u>

Gambar 3 Struktur THE

² Abas Ali Pangera (2010). *Sistem Operasi*. Yogyakarta: Andi. hlm. 106-118.

<u>Lapis-6</u>	:	<u>user program</u>
<u>Lapis-5</u>	:	<u>device driver dan sceduler</u>
<u>Lapis-4</u>	:	<u>virtual memory</u>
<u>Lapis-3</u>	:	<u>I/O channel</u>
<u>Lapis-2</u>	:	<u>penjadwalan CPU</u>
<u>Lapis-1</u>	:	<u>instruksi interpreter</u>
<u>Lapis-0</u>	:	<u>hardware</u>

Gambar 4 Struktur Venus

Kelebihan Sistem Berlapis (*layered system*)

- (a) Memiliki rancangan modular, yaitu sistem dibagi menjadi beberapa modul & tiap modul dirancang secara independen.
- (b) 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 Dijkstra dan mahasiswa-mahasiswanya, serta sistem operasi MULTICS.

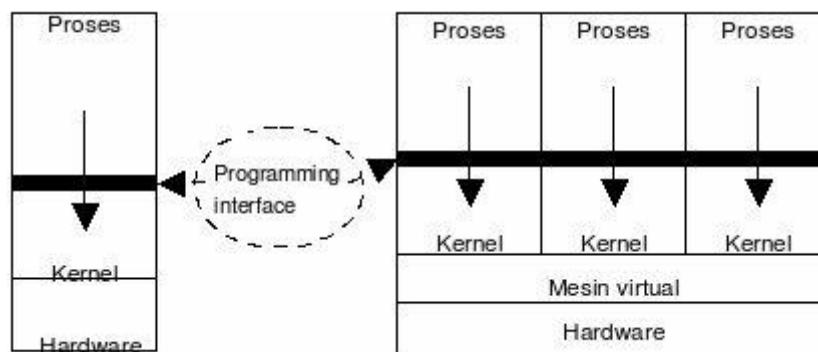
2.3 Mesin Virtual

Secara konsep, sistem computer dibuat berdasarkan lapisan, hardware atau perangkat lunak merupakan tingkatan terbawah dari keseluruhan sistem.³

³ Bambang Hariyanto (2009). *Sistem Operasi*. Bandung: Informatika. hlm. 41-45.

Kernel yang berjalan ditingkatkan berikutnya menggunakan instruksi-intruksi perangkat keras untuk membuat kumpulan sistem call yang digunakan oleh lapisan luarnya.

Program di atas kernel dapat menggunakan sistem call atau instruksi-instruksi perangkat keras. Dalam beberapa hal, program sistem tidak membedakan kedua lapisan tersebut. Program sistem memperlakukan sistem call dan perangkat keras di lapisan yang sama, meskipun program beberapa sistem pada tingkat yang lebih tinggi dari rutin-rutin program aplikasi yang lain, program-program aplikasi akan melihat semua yang berada di bawahnya. Pendekatan sistem lapisan ini menjadi konsep dari mesin virtual. Dengan menggunakan penjadwalan CPU dan teknik virtual memori, sebuah sistem operasi dapat membuat bayangan proses dalam jumlah banyak, yang masing-masing dieksekusi oleh prosesornya sendiri dengan memori (virtual) sendiri. Pada mesin virtual tidak tersedia fungsi-fungsi tambahan tetapi interface yang identik dengan perangkat keras yang mendasarinya. Setiap proses dilengkapi dengan salinan dari computer yang mendasarinya.⁴



Gambar 5 Mesin Virtual

Kelebihan Mesin Maya (*Virtual Machine*)

- (a) 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.
- (b) 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.

⁴ "[Struktur sistem operasi](#)". *Ashared*. Diakses tanggal 15 Mei 2014.

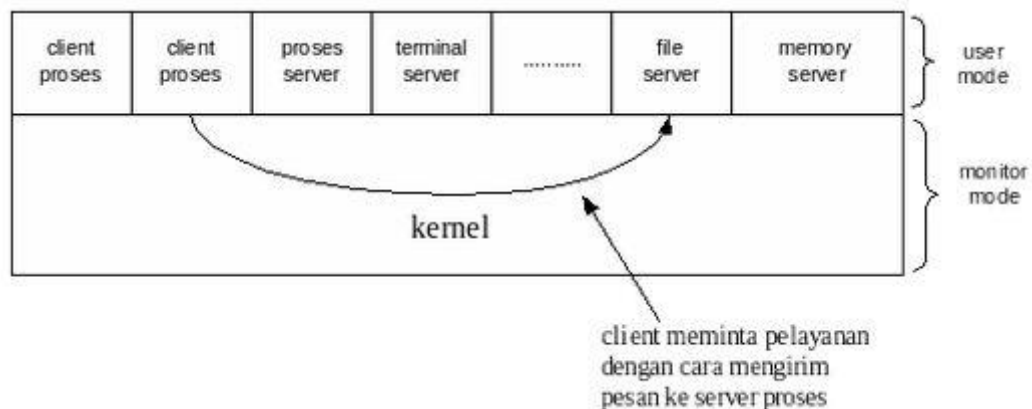
Kekurangan Mesin Maya (*Virtual Machine*)

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

2.4 Model Client Server

Dalam model ini, semua kernel bekerja menangani komunikasi antara server dan client. Dengan membagi system operasi menjadi banyak bagian yang masing-masing hanya menangani bagian tertentu dari sistem. Seperti layanan file, layanan proses, layanan terminal atau layanan memori, maka setiap bagian menjadi lebih kecil dan lebih mudah diatur. Kemudian semua server berjalan dalam proses user mode dan tidak dalam kernel mode, maka bagian-bagian tidak mempunyai akses langsung ke perangkat keras.⁵

Keuntungannya, bila ada kesalahan di file server, maka layanan file mungkin akan crash, tetapi tidak akan mempengaruhi keseluruhan sistem, yang akhirnya dapat mengakibatkan keseluruhan sistem tidak berfungsi.[5] Keuntungan lain dari sebuah model client-server adalah dapat diadaptasi untuk digunakan dalam sistem terdistribusi. Jika client berkomunikasi dengan server dengan mengirimkan pesan, klien tidak perlu tahu pesan tersebut ditangani secara local dalam mesinnya sendiri atau pesan tersebut dikirimkan melalui jaringan ke server di mesin yang lain.



Gambar 6 Model Client Server

⁵ Elmasri DKK. *Operating System a Spiral Approach. Texas: Higher Education. hlm. 33-42.*

Kelebihan Client-Server Model

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

Kekurangan Client-Server Model

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

3. 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 menandai 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. 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. Komponen-komponen Sistem

Pada kenyataannya tidak semua sistem operasi mempunyai struktur yang sama. Namun menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

- (a) Manajemen Proses.
- (b) Manajemen Memori Utama.
- (c) Manajemen Secondary-Storage.
- (d) Manajemen Sistem I/O.
- (e) Manajemen Berkas.
- (f) Sistem Proteksi.
- (g) Jaringan.

- (h) Command-Interpreter system.

Managemen Proses

Proses adalah keadaan ketika sebuah program sedang di eksekusi. Sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, dan perangkat-perangkat I/O. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan managemen proses seperti:

- (a) Pembuatan dan penghapusan proses pengguna dan sistem proses.
- (b) Menunda atau melanjutkan proses.
- (c) Menyediakan mekanisme untuk proses sinkronisasi.
- (d) Menyediakan mekanisme untuk proses komunikasi.
- (e) Menyediakan mekanisme untuk penanganan deadlock.

Managemen Memori Utama

Memori utama atau lebih dikenal sebagai memori adalah sebuah array yang besar dari word atau byte, yang ukurannya mencapai ratusan, ribuan, atau bahkan jutaan. Setiap word atau byte mempunyai alamat tersendiri. Memori Utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama termasuk tempat penyimpanan data yang sementara (volatile), artinya data dapat hilang begitu sistem dimatikan. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan managemen memori seperti:

- (a) Menjaga track dari memori yang sedang digunakan dan siapa yang menggunakannya.
- (b) Memilih program yang akan di-load ke memori.
- (c) Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.

Managemen Secondary-Storage

Data yang disimpan dalam memori utama bersifat sementara dan jumlahnya sangat kecil. Oleh karena itu, untuk meyimpan keseluruhan data dan program komputer dibutuhkan secondary-storage yang bersifat permanen dan mampu menampung banyak data. Contoh dari secondary-storage adalah harddisk, disket, dll. Sistem operasi bertanggung-jawab atas aktivitas-aktivitas yang berkaitan dengan disk-management seperti: free-space management, alokasi penyimpanan, penjadualan disk.

Managemen Sistem I/O

Sering disebut device manager. Menyediakan "device driver" yang umum sehingga operasi I/O dapat seragam (membuka, membaca, menulis, menutup). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada hard-disk, CD-ROM dan floppy disk. Komponen Sistem Operasi untuk sistem I/O:

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

Managemen Berkas

Berkas adalah kumpulan informasi yang berhubungan sesuai dengan tujuan pembuat berkas tersebut. Berkas dapat mempunyai struktur yang bersifat hirarkis (direktori, volume, dll.). Sistem operasi bertanggung-jawab:

- (a) Pembuatan dan penghapusan berkas.
- (b) Pembuatan dan penghapusan direktori.
- (c) Mendukung manipulasi berkas dan direktori.
- (d) Memetakan berkas ke secondary storage.
- (e) Mem-backup berkas ke media penyimpanan yang permanen (non-volatile).

Sistem Proteksi

Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program, prosesor, atau pengguna ke sistem sumber daya. Mekanisme proteksi harus:

- (a) membedakan antara penggunaan yang sudah diberi izin dan yang belum.
- (b) specify the controls to be imposed.
- (c) provide a means of enforcement.

Jaringan

Sistem terdistribusi adalah sekumpulan prosesor yang tidak berbagi memori atau clock. Tiap prosesor mempunyai memori sendiri. Prosesor-prosesor tersebut terhubung melalui jaringan komunikasi. Sistem terdistribusi menyediakan akses pengguna ke bermacam sumber-daya sistem. Akses tersebut menyebabkan:

- (a) Computation speed-up.
- (b) Increased data availability.
- (c) Enhanced reliability.

Command-Interpreter System

Sistem Operasi menunggu instruksi dari pengguna (command driven). Program yang membaca instruksi dan mengartikan control statements umumnya disebut: control-card interpreter, command-line interpreter, dan UNIX shell. Command-Interpreter System sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan dan teknologi I/O devices yang ada. Contohnya: CLI, Windows, Pen-based (touch), dan lain-lain.

Layanan Sistem Operasi

Eksekusi program adalah kemampuan sistem untuk "load" program ke memori dan menjalankan program. Operasi I/O: pengguna tidak dapat secara langsung mengakses sumber daya perangkat keras, sistem operasi harus menyediakan mekanisme untuk melakukan operasi I/O atas nama pengguna.

Sistem manipulasi berkas adalah kemampuan program untuk operasi pada berkas (membaca, menulis, membuat, and menghapus berkas). Komunikasi adalah pertukaran data/ informasi antar dua atau lebih proses yang berada pada satu komputer (atau lebih). Deteksi error adalah menjaga kestabilan sistem dengan mendeteksi "error", perangkat keras mau pun operasi.

Efisiensi penggunaan sistem

- (a) Resource allocator adalah mengalokasikan sumber-daya ke beberapa pengguna atau job yang jalan pada saat yang bersamaan.
- (b) Proteksi menjamin akses ke sistem sumber daya dikendalikan (pengguna dikontrol aksesnya ke sistem).
- (c) Accounting adalah merekam kegiatan pengguna, jatah pemakaian sumber daya (keadilan atau kebijaksanaan).

System Calls

System call menyediakan interface antara program (program pengguna yang berjalan) dan bagian OS. System call menjadi jembatan antara proses dan sistem operasi. System call ditulis dalam bahasa assembly atau bahasa tingkat tinggi yang dapat mengendalikan mesin (C). Contoh: UNIX menyediakan system call: read, write => operasi I/O untuk berkas. Sering pengguna program harus memberikan data (parameter) ke OS yang akan dipanggil. Contoh pada UNIX: read(buffer, max_size, file_id); Tiga cara memberikan parameter dari program ke sistem operasi:

- (a) Melalui registers (sumber daya di CPU).
- (b) Menyimpan parameter pada data struktur (table) di memori, dan alamat table tsb ditunjuk oleh pointer
- (c) yang disimpan di register.
- (d) Push (store) melalui "stack" pada memori dan OS mengambilnya melalui pop pada stack tsb.

Mesin Virtual

Sebuah mesin virtual (Virtual Machine) menggunakan misalkan terdapat sistem program => control program yang mengatur pemakaian sumber daya perangkat keras. Control program = trap System call + akses ke perangkat keras. Control program memberikan fasilitas ke proses pengguna. Mendapatkan jatah CPU dan memori. Menyediakan interface "identik" dengan apa yang disediakan oleh perangkat keras => sharing devices untuk berbagai proses.

Mesin Virtual (MV) (MV) => control program yang minimal MV memberikan ilusi multitasking: seolah-olah terdapat prosesor dan memori eksklusif digunakan MV. MV memilah fungsi multitasking dan implementasi extended machine (tergantung proses pengguna) => flexible dan lebih mudah untuk pengaturan. Jika setiap pengguna diberikan satu MV => bebas untuk menjalankan OS (kernel) yang diinginkan pada MV tersebut. Potensi lebih dari satu OS dalam satu komputer. Contoh: IBM VM370: menyediakan MV untuk berbagai OS: CMS (interaktif), MVS, CICS, dll. Masalah: Sharing disk => OS mempunyai sistem berkas yang mungkin berbeda. IBM: virtual disk (minidisk) yang dialokasikan untuk pengguna melalui MV. Konsep MV menyediakan proteksi yang lengkap untuk sumberdaya sistem, dikarenakan tiap MV terpisah dari MV yang lain. Namun, hal tersebut menyebabkan tidak adanya sharing sumberdaya secara langsung. MV merupakan alat yang tepat untuk penelitian dan pengembangan sistem operasi. Konsep MV susah untuk diimplementasi sehubungan dengan usaha yang diperlukan untuk menyediakan duplikasi dari mesin utama.

Perancangan Sistem dan Implementasi

Target untuk pengguna: sistem operasi harus nyaman digunakan, mudah dipelajari, dapat diandalkan, aman dan cepat. Target untuk sistem: sistem operasi harus gampang dirancang, diimplementasi, dan dipelihara, sebagaimana fleksibel, error, dan efisien. Mekanisme dan Kebijakan:

- (a) Mekanisme menjelaskan bagaimana melakukan sesuatu kebijakan memutuskan apa yang akan dilakukan. Pemisahan kebijakan dari mekanisme merupakan hal yang sangat penting; ini mengizinkan fleksibilitas yang tinggi bila kebijakan akan diubah nanti.
- (b) Kebijakan memutuskan apa yang akan dilakukan.

Pemisahan kebijaksanaan dari mekanisme merupakan hal yang sangat penting; ini mengizinkan fleksibilitas yang tinggi bila kebijaksanaan akan diubah nanti. Implementasi Sistem biasanya menggunakan bahasa assembly, sistem operasi sekarang dapat ditulis dengan menggunakan bahasa tingkat tinggi. Kode yang ditulis dalam bahasa tingkat tinggi: dapat dibuat dengan cepat, lebih ringkas, lebih mudah dimengerti dan didebug. Sistem operasi lebih mudah dipindahkan ke perangkat keras yang lain bila ditulis dengan bahasa tingkat tinggi.

System Generation (SYSGEN)

Sistem operasi dirancang untuk dapat dijalankan di berbagai jenis mesin; sistemnya harus di konfigurasi untuk tiap komputer. Program SYSGEN mendapatkan informasi mengenai konfigurasi khusus dari sistem perangkat keras.

- (a) Booting: memulai komputer dengan me-load kernel.
- (b) Bootstrap program: kode yang disimpan di code ROM yang dapat menempatkan kernel, memasukkannya kedalam memori, dan memulai eksekusinya.