ARSITEKTUR SISTEM OPERASI

Abas Ali Pangera, Dony Ariyus, *Jurusan Teknik Informatika, STMIK AMIKOM Yogyakarta,*
Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta - Indonesia

Abstract

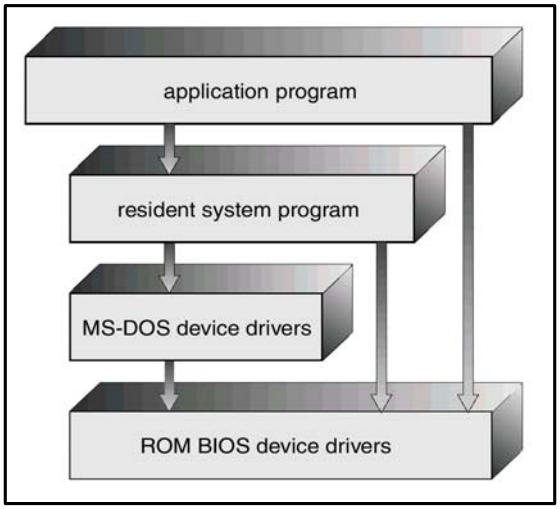
Arsitektur perangkat lunak merupakan struktur-struktur yang menjadikan landasan untuk menentukan keberadaan komponen-komponen perangkat lunak, cara komponen-komponen untuk saling berinteraksi dan organisasi komponen-komponen dalam membentuk perangkat lunak. Arsitektur system operasi merupakan arsitektur perangkat lunak yang digunakan dalam membangun perangkat lunak system operasi arsitktur system operasi modern yang semakin komplek dan rumit memerlukan sistem operasi yang dirancang dengan sangat hati-hati agar dapat berfungsi secara optimum dan mudah untuk dimodifikasi

Arsitektur system operasi moderen diantaranya adalah :

- System Monolitik
- System Berlapis
- System Client/server
- System Virtual mesin
- System Berorientasi objek

Ada sejumlah sistem komersial yang tidak memiliki struktur yang cukup baik. Sistem operasi tersebut sangat kecil, sederhana dan memiliki banyak keterbatasan. Salah satu contoh sistem tersebut adalah MS-DOS. MS-DOS dirancang oleh orang-orang yang tidak memikirkan akan kepopuleran *software* tersebut.

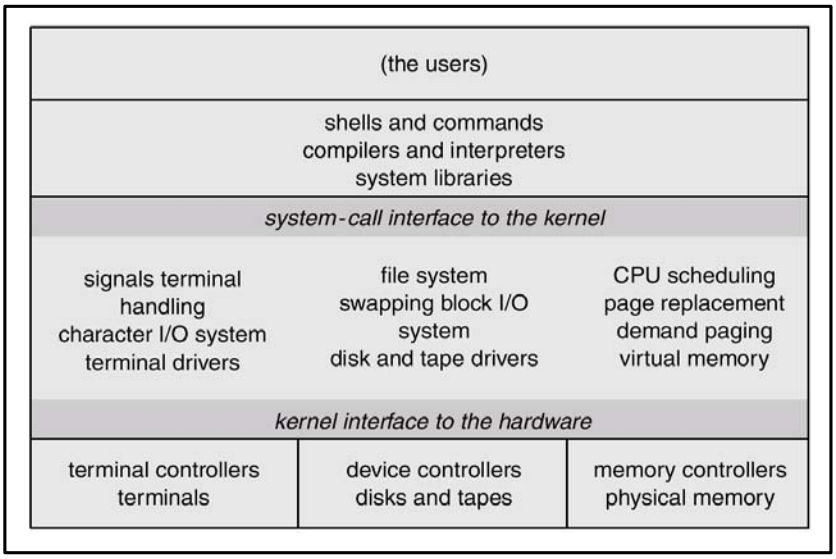
Sistem operasi tersebut terbatas pada perangkat keras sehingga tidak terbagi menjadi modul-modul. Meskipun MS-DOS mempunyai beberapa struktur, antar muka dan tingkatan fungsionalitas tidak terpisah secara baik seperti pada Gambar di bawah ini. Karena Intel 8088 tidak menggunakan dual-mode sehingga tidak ada proteksi hardware. Oleh karena itu orang mulai enggan menggunakannya.



Stuktur Sistem MS – DOS

Sistem operasi UNIX (Original UNIX) juga terbatas pada fungsi perangkat keras dan struktur yang terbatas. UNIX hanya terdiri atas 2 bagian, yaitu Kernel dan program sistem. Kernel berada di bawah

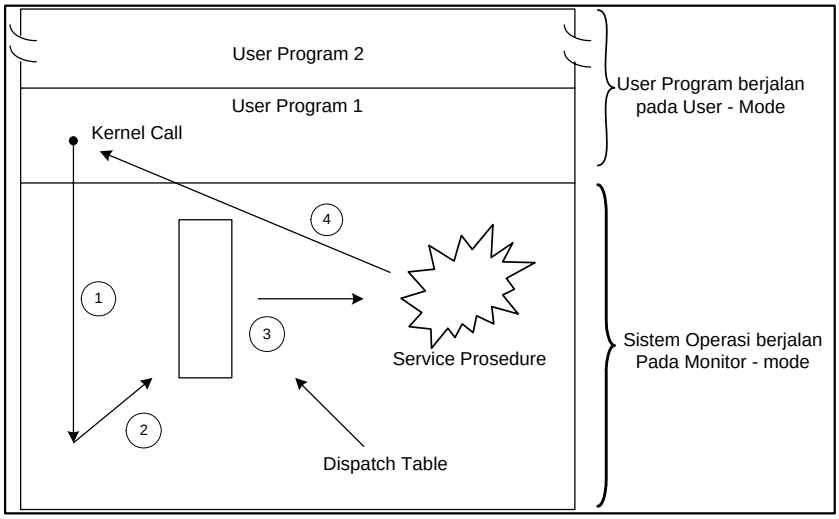
tingkat antarmuka *system call* dan di atas perangkat lunak secara fisik. Kernel ini berisi sistem file, penjadwalan CPU, menejemen memori, dan fungsi sistem operasi lainnya yang ada pada sistem call berupa sejumlah fungsi yang besar pada satu level. Program sistem meminta bantuan kernel untuk memanggil fungsi-fungsi dalam kompilasi dan manipulasi file. Struktur system UNIX dapat dilihat pada Gambar di bawah ini.



Stuktur Sistem UNIX

Sistem Monolitik

Merupakan struktur sederhana yang melengkapi dengan operasi "dual" pelayanan {sistem call} yang diberikan oleh sistem operasi model yang dilakukan dengan cara mengambil sejumlah parameter pada tempat yang telah ditentukan sebelumnya, seperti register atau stack dan kemudian mengeksekusi suatu intruksi trap tertentu pada monitor mode.



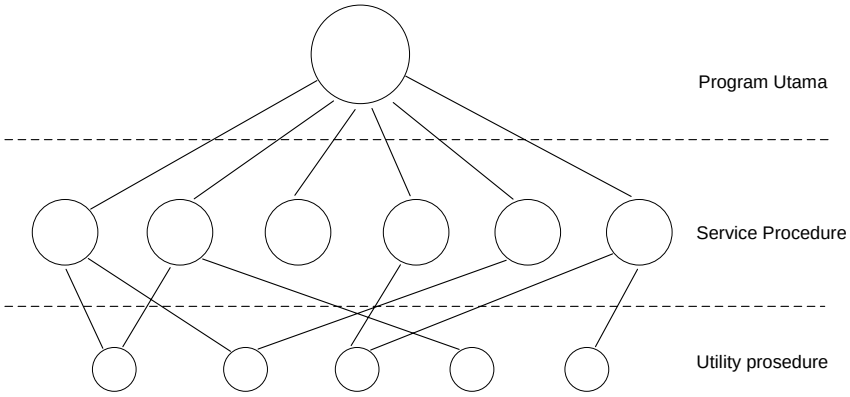
Bagaimana Sistem Call Dibuat

- User program melakukan "trap" pada karnel
- Intruksi berpindah dari user mode ke monitor modedan mentransfer control ke sistem operasi

- Sistem operasi mengecek parameter-parameter dari pemanggilan tersebut, untuk menentukan sistem call mana yang memanggil
- Sistem operasi menunjuk ke suatu table yang berisi slol ke-k yang menunjuk sistem call K
- Kontrol akan dikembalikan kepada user program, jika sistem call telah selesai mengerjakan tugasnya

Tatanan ini memberikan suatu struktur dasar dari sistem operasi sebagai berikut :

a) Program utama meminta service procedure
b) Kumpulan service procedure yang dibaca oleh sistem call
c) Kumpulan utility procedure yang membantu service procedure



Model Struktur Monolitik

Pada model ini, tiap-tiap sistem call memiliki satu service procedure. Ulitity procedure mengerjakan segala sesuatu yang dibutuhkan oleh beberapa service procedure, seperti mengambil data dari user program, seperti terlihat pada gambar di atas.

Keunggulan dari system Monolitik ini adalah:

- Layanan terhadap job-job yang ada bisa dilakukan dengan cepat karena berada pada satu ruang alamat

Kelemahan dari system Monolitik adalah

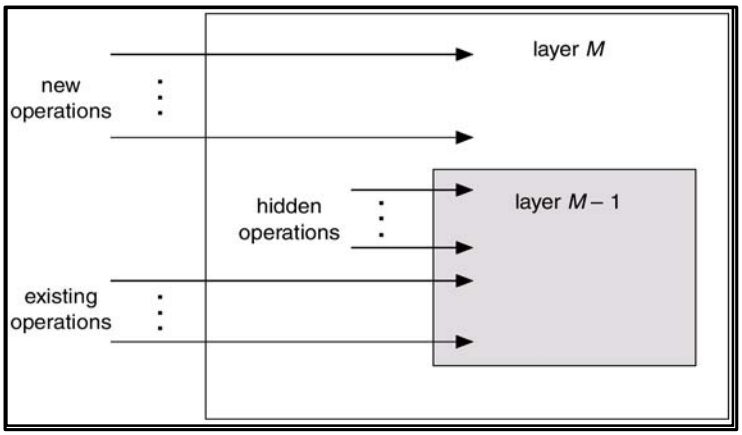
- Pengujian dan penghilangan kesalahan sulit dilakukan karena tidak dapat dipisahkan dan dilokasikan, namum secara prakteknya pemrograman yang berdisiplin bagus dapta mempermudah pengembangannya
- Sulit dalam menyediakan fasilitas pengamanan
- Merupakan pemborosan apabila setiap computer harus menjalankan kernel minilitik sangat besar sementara sebenarnya tidak memerlukan seluruh layanan yang disediakan kernel atau dengan kata lain bisa dibilang tidak fleksibel
- Kesalahan pemrograman di satu bagian kernel menyebakan matinya seluruh sistem

Sistem Berlapis

Teknik pendekatan terlapis pada dasarnya dibuat dengan menggunakan pendekatan *top-down*, semua fungsi ditentukan dan dibagi menjadi komponen komponen. Modularisasi sistem dilakukan dengan cara memecah sistem operasi menajdi beberapa lapis (tingkat).

Lapisan terendah (layer 0) adalah perangkat keras dan lapisan teratas (layer N) adalah *user interface*. Dengan system modularisasi, setiap lapisan mempunyai fungsi (operasi) tertentu dan melayani lapisan yang lebih rendah.

Table dan gambar di bawah ini menunjukkan system pendekatan terlapis tersebut. System operasi pertama kali yang memakai system berlapis adalah THE. System operasi THE yang dibuat oleh Dijkstra dan mahasiswa-mahasiswanya. Pada dasarnya system operasi berlapis dimaksudkan untuk mengurangi kompleknya rancangan dan implementasi dari suatu system operasi. Stuktur system operasi berlapis seperti gambar :



Struktur Sistem Operasi Berlapis

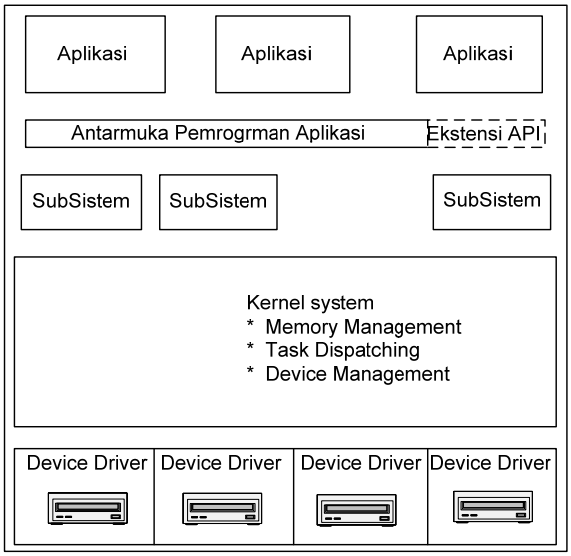
Contoh sistem operasi yang menggunakan sistem ini adalah: UNIX termodifikasi, THE, Venus dan OS/2 (Gambar 2-8). Lapisan pada struktur THE adalah:

Lapisan	Nama	Fungsi
Lapisan - 5	User Program	Untuk program Pemakai
Lapisan - 4	Buffering untuk I/O device	Penyederhana akses I/O pada level atas
Lapisan - 3	Operator-console device driver	Mengantur komunikasi antar proses
Lapisan - 2	Manajemen memori	Pengalokasian ruang memori
Lapisan - 1	Penjadwalan CPU	Mengatur alokasi CPU dan switching pengaturan prosessor
Lapisan - 0	Hardware	Untuk operator dan menjalan keseluruhan sistem

Sedangkan lapisan pada struktur Venus adalah:

Lapisan	Nama	Fungsi
Lapisan - 6	User Program	Untuk program Pemakai
Lapisan - 5	Device driver dan Scheduler	Mengantur dan penjadwalan komunikasi antar proses
Lapisan - 4	Virtual Memory	Penyimpan proses untuk sementara
Lapisan - 3	I/O Channel	Mengatur akses I/O
Lapisan - 2	Penjadwalan CPU	Mengatur alokasi CPU dan switching pengaturan prosessor
Lapisan - 1	Instruksi Interpreter	Fungsi
Lapisan - 0	Hardware	Untuk operator dan menjalan keseluruhan sistem

Struktur Sistem Operasi OS/2 adalah



Struktur OS/2

Keuntungan dari sistem berlapis adalah:

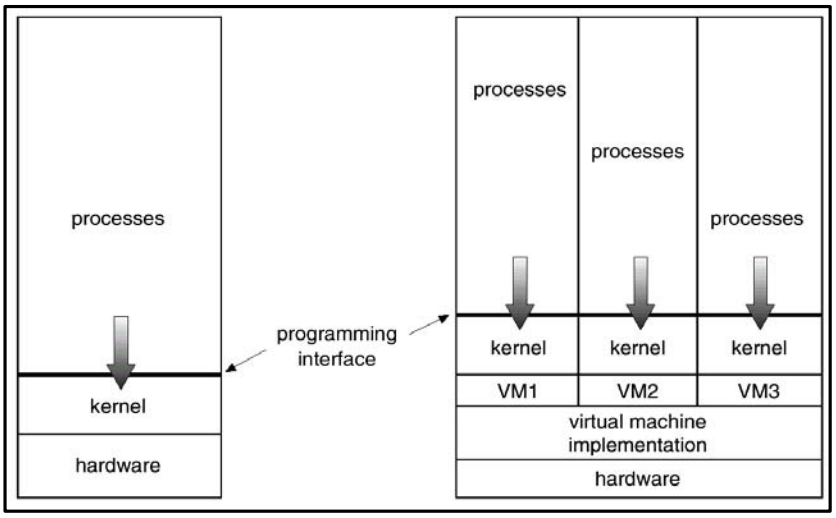
Sistem berlapis memiliki semua keunggulan rancangan modular merupakan dirancang secara terpisah atau beberapa modul, setiap modul dan lapisan bisa dirancang, di uji, secara independen sehingga jika terjadi suatu kesalahan mudah untuk menanganinya.

Kelemahannya :

Fungsi-fungsi dari sistem operasi harus terdapat di masing-masing lapisan, jika terjadi suatu kesalahan bisa jadi semua lapisan harus diprogram ulang.

Sistem Mesin Virtual (Virtual Machine)

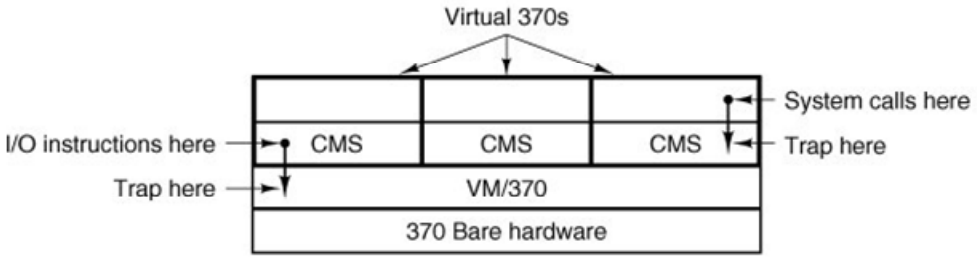
Konsep dasar dari mesin virtual ini tidak jauh berbeda dengan pendekatan terlapis, hanya saja konsep ini memberikan sedikit tambahan berupa antarmuka yang menghubungkan perangkat keras dengan kernel untuk tiap-tiap proses, Gambar 4-13 menunjukkan konsep tersebut. Mesin virtual menyediakan antar muka yang identik untuk perangkat keras yang ada. Sistem operasi membuat ilusi untuk beberapa proses, masing-masing mengeksekusi prosessor masing-masing untuk memori (virtual) masing masing.



Konsep Mesin Virtual a) Tanpa Mesin Virtual , b) DenganMesin Virtual

Meskipun konsep ini cukup baik, namun sulit untuk diimplementasikan, ingat bahwa system menggunakan metode *dual-mode*. Mesin virtual hanya dapat berjalan pada *monitor-mode* jika berupa sistem operasi, sedangkan mesin virtual itu sendiri berjalan dalam bentuk *user-mode*. Konsekuensinya, baik virtual *monitor-mode* maupun virtual *user-mode* harus dijalankan melalaui *physical user mode*. Hal ini menyebabkan adanya transfer dari *user-mode* ke *monitor-mode* pada mesin nyata, yang juga akan menyebabkan adanya transfer dari virtual *user-mode* ke virtual *monitor-mode* pada mesin virtual.

Sumber daya (*resource*) dari computer fisik dibagi untuk membuat mesin virtual. Penjadwalan CPU dapat membuat penampilan bahwa user mempunyai prosessor sendiri. *Spooling* dan system file dapat menyediakan *card reader* virtual dan *line printer* virtual. Terminal time sharing pada user melayani sebagai *console* operator mesin virtual. Contoh sistem operasi yang memakai mesin virtual adalah IBM S/370 dan IBM VM/370.



Struktur VM/370 dengan CMS

Keuntungan dan kerugian konsep mesin virtual adalah sebagai berikut:

- 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.

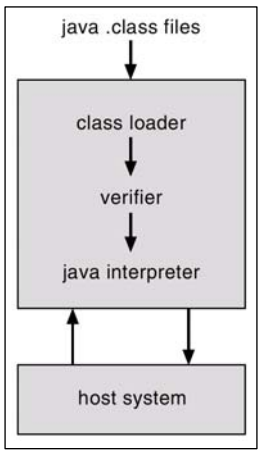
- Konsep mesin virtual sangat sulit untuk mengimplementasikan kebutuhan dan duplikasi yang tepat pada mesin yang sebenarnya.

Teknik ini berkembang menjadi operating system emulator, shingga system operasi dapat menjalankan aplikasi-aplikasi untuk system operasi lain.

- Sistem operasi MS-Windows NT dapat menjalankan aplikasi untuk MS-DOS, OS/2 mode teks dan aplikasi Win16. aplikasi tersebut dijalankan sebagai input bagi subsistem di MS-Windows NT yang mengemulasikan system calls yang dipanggil aplikasi dengan Win32 API (Sistem Call di MS-Windows NT)
- WABI yang dikembangkan oleh IBM yang bisa mengemulasikan Win32 API sehingga system operasi yang menjalankan WABI bisa menjalankan aplikasi-aplikasi yang diperuntukan bagi MS-Windows
- DOSEMU dikembangkan di system operasi Linux, DOSEMU berfungsi agar aplikasi MW-DOS dapat dijalankan di lunux,
- WINE juga dikembangan di system
- P-Code Machine digagas oleh Niklaus Wirth agar hasil kompilasi program pascal dapat dijalankan di mesin manapun yang mempunyai P-Code machine, dan gagasan ini dilanjutkan oleh JVM (Java Virtual Machine) yang hasil kompilasinya berupa kelas java (file dengan ekstensi.class) dapat dijalankan dimanapun yang memiliki JVM

Contoh sistem operasi yang menggunakan mesin virtual adalah java virtual mesin :

- Program Java yang di-compile adalah *platform-neutral bytecodes* yang dieksekusi oleh Java Virtual Machine (JVM).
- JVM terdiri dari :
  - *class loader*
  - *class verifier*
  - *runtime interpreter*
- *Just-In-Time (JIT) compiler* meningkatkan kemampuan



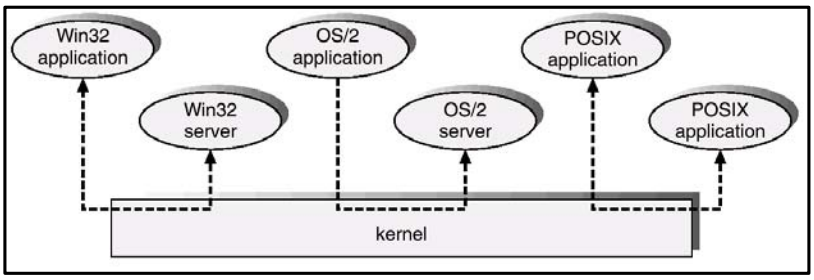
Java Mesin Virtual

Java merupakan system yang menggunakan implementasi mesin virtual. Untuk mengkompilasi program Java maka digunakan kode bit yang disebut *platform-neutral bytecode* yang dieksekusi oleh Java *Virtual Machine* (JVM). JVM terdiri dari class loader, class verifier dan runtime interpreter

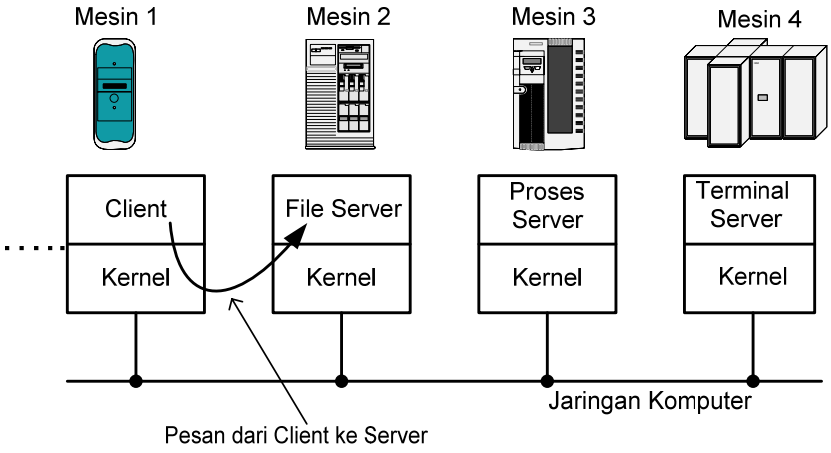
Sistem Client-Server

Sistem operasi modem memiliki kecendrungan untuk memindahkan kode ke lapisan yang lebih tinggi dan menghapus sebanyak mungkin, kode-kode tersebut dari sistem operasi sehingga akan meninggalkan keruel yang minimal. Konsep ini biasa diimplementasikan dengan dengan cara menjadikan fungsi-fungsi yang ada pada sistem operasi menjadi user proses. Jika satu proses minta untuk dilayani, misalnya satu blok file, maka user proses {disini dinamakan: Client proses} mengirim permintaan tersebut ke user proses. Server proses akan melayani permintaan tersebutkemudian mengirimkan jawabannya kembali. Semua pekerjaan keruel dilakukan pada pengendalian komunikasi antara client dan server.

Dengan membagi sistem operasi menjadi beberapa lapisan, dimana tiap-tipa bagian mengendalikan satu segi sistem, seperti pelayanan file, pelayanan proses, pelayanan terminal, atau pelayanan memori, maka tiap-tiap bagian menjadi lebih sederhana dan dapat diatur selain itu, oleh karena semua server berjalan pada user mode proses, dan bukan merupakan monitor mode, maka server tidak dapat mengakses hardware secara lansung. Akibatnya, jika terjadi kerusakan pada file server, maka pelayanan file akan terganggu. Namun hal ini tidak akan sampai menganggu sistem lainnya.



Model Client Server



Model Client-Server Pada Sistem Terdistribusi

Masalah yang sering terjadi pada system client –server adalah tidak semua tugas dapat dijalankan di tingkat pemakai, tapi kesulitan ini dapat di atas dengan:

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

Keuntungan dari model client server ini adalah

- Dapat diadaptasikan pada sistem terdistribusi.
- Jika suatu client berkomunikasi dengan server dengan cara mengirimkan pesan, maka server tidak perlu tahu apakah pesan itu dikirim oleh dan dari mesin itu sendiri {local} atau dikirim oleh mesin yang lain melalui jaringan.
- Pengembangan dapat dilakukan secara modular
- Kesalahan pada suatu subsistem tidak menganggu subsistem lain sehingga tidak mengakibatkan system mati secara keseluruhan

Sedangkan kelemahan dari system client-server adalah :

- Pertukaran pesan dapat menjadi bottleneck
- Layanan dilakukan secara "lambat" karena harus memlalui pertukaran pesan antar client-server

Sistem Berorientasi Objek

Layanan Sistem operasi sebagai kumpulan proses untuk menyelesaikan pekerjaannya, yang sering disebut dengan system operasi bermodel proses, sedangkan layanan system operasi sebagai objek disebut dengan system operasi berorentasi objek. Pendekatan objek dimaksudkan untuk mengadopsi keunggulan dari teknolgi berorientasi objek

Pada system operasi berorientasi objek, layanan diimplementasikan sebagai kumpulan objek, masing-masing objek diberi tipe yang menandai property objek seperti proses, dirktori, berkas, dan sebagainya. Dengan memanggil operasi yang didefinisikan di objek, data yang berada dalam objek tersebut dapat diakses dan dimodifikasi

Contoh dari system operasi berorentasi objek adalah:

- Eden
- Choices
- X-kernel
- Medusa
- Clunds
- Amoeba
- Muse
- Dan lain sebagainya

Sistem operasi MS-Windows NT mengadopsi beberapa teknologi berorientasi objek tapi tidak secara keseluruhannya

Daftar Pustaka

Ariyus,Dony,2006, "*Computer Security*", Andi Offset, Yogyakarta

Ariyus, Dony,2005," kamus hacker", Andi offset, Yogyakarta

Bob DuCharme, 2001," *The Operating System Handbook or, Fake Your Way Through Minis and Mainframes*" Singapore: McGraw-Hill Book Co

Bill Venners.1998. "*Inside the Java Virtual Machin"e* . McGraw-Hill.

Deitel, Harvey M, 2004 " *operating systems" 3th Edition*, Massachusetts: Addison-Wesley Publshing Company

Gary B. Shelly, 2007**,** "*Discovering Computers: Fundamentals*" Thomson

Gollmann, Dieter,1999 "*Computer Security*" Jhon Willey & Son Inc, Canada

Grosshans,D. 1986," *File system: design and implementation*", Englewwod Cliffs, New Jersey : Prentice-Hall Inc.

Harvey M Deitel dan Paul J Deitel. 2005. *Java How To Program*. Sixth Edition. Prentice Hall.

Hoare, C.A.R. 1985" *Communication sequential processes*"Englewood Cliffs, New Jersey, Prentice Hall Inc

Jean Bacon, Tim Harris, 2003 "*Operating Systems: Concurrent and Distributed Software Design*" Massachhussets. Addison Wesley

Kenneth H Rosen. 1999. "*Discrete Mathematics and Its Application"*. McGraw Hill.

Madnick,Stuart E dan John J. Donovan, 1974 " *operating system*", Singapore: McGraw-Hill Book Co

Michael Kifer and Scott A. Smolka, 2007 *Introduction to Operating System Design and Implementation The OSP 2 Approach*, Springer-Verlag London

Microsoft 1999. *Microsoft Windows User Experience*. Microsoft Press.

Milenkovie, Milan. 1992. "*Operationg system: Concepts and Design*", Singapore: McGraw-Hill Book Co

Randall Hyde. 2003. *The Art of Assembly Language*. First Edition. No Strach Press

Robert betz, 2001 "*Intoduction to Real-time operation system*", Department of Electrical and Computer Engineering University of Newcastle, Australia

Robert Love. 2005. *Linux Kernel Development* . Second Edition. Novell Press

Ron White,1998, *How Computers Work, Fourth Edition*, Que corporation, A Division of Macmillan Computer Publishing, USA

Shay, William A. 1993, " *Introduction to Operationg System*" New York: HarperCollins College Publishers

Silberschatz, Peter Galvin, dan Grag Gagne. 2000. "*Applied Operating System, 1st"* John Wile & Hiil Book Co

Silberschatz, A., dan Galvin, P.2003, "Operating Sistem Concept. Sixth Edition". Massachhussets. Addisson- Wasley

Silberschatz, Peter Galvin, dan Grag Gagne. 2005. "*Operating Systems Concepts"*. Seventh Edition. John Wiley & Sons.

Stalling, William, 1995, "*Operating Sistems"*. New Jersey. Prentice – Hall

Stalling, William, 1996" *Computer Organization and Architecture*". New Jersey. Prentice – Hall

Stalling William, 1995, "*Network and Internetwork Security"* Prentice-Hall, USA

Tanenbaum, Andrew S, 1992 "*Modern Operating Sistems*". New Jersey. Prentice – Hall

Taenbaum, Andrew S, 2006, "*Operating Systems Design and Implementation, Third Edition"* Massachusetts