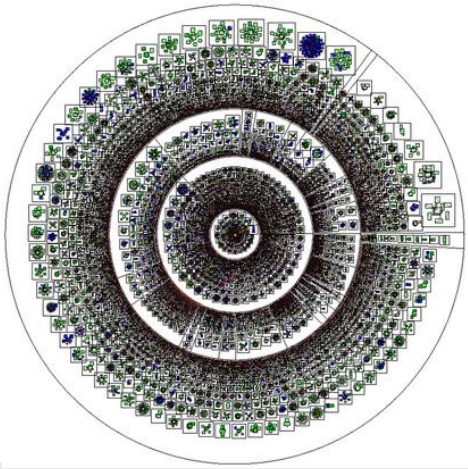
Pengantar Kernel

Core of a system the operating system (OS)



Yudho Yudhanto
yyudhanto@gmail.com
http://rumahstudio.com

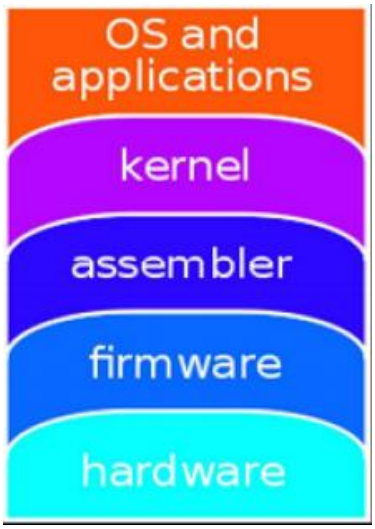
Lisensi Dokumen:
Copyright © 2003-2016 IlmuKomputer.Com
Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

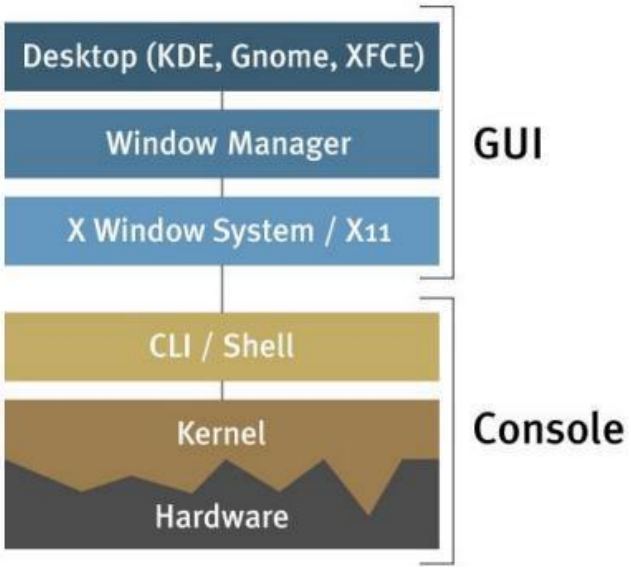
1. Pendahuluan

Apakah *kernel* itu? *Kernel* merupakan komponen inti dari sebuah *software* sistem operasi yang berjalan dalam sebuah perangkat, misalnya komputer, laptop atau handphone. Tugas utamanya adalah untuk mengatur pemakaian sumber daya sistem yang tersedia dan bertanggung jawab atas komunikasi antara komponen *hardware* dan *software*.

Jika GUI (Graphical User Interface) adalah mengatur komunikasi komputer dengan user pengguna, maka Kernel mengatur komunikasi software dengan hardwarenya (mesin).

Bayangkan jika sumber daya *hardware* terbatas, sementara semua program aplikasi meminta dijalankan dengan baik dalam waktu bersamaan dan meminta dengan kecepatan yang sama maka jika Kernel dengan sigap akan melayani sampai semua selesai kepada masing-masing bagiannya. Jika lebih dari satu program yang harus dilayani dalam waktu yang bersamaan, maka kernel juga bertugas untuk mengatur kapan dan berapa lama suatu program dapat menggunakan satu bagian perangkat keras tersebut. Inilah yang sisebut dengan *multiplexing*.





Akses kepada perangkat keras secara langsung merupakan masalah yang kompleks, ini pernah dilakukan oleh komputer generasi awal sehingga salah satu efeknya adalah tidak bisa menjalankan program secara bersamaan. Kemudian munculah *kernel* sebuah bagian penting yang mengimplementasikan sekumpulan *abstraksi hardware*. Abstraksi merupakan sebuah cara untuk menyembunyikan atau meringkas kompleksitas yang ada, sehinnga memungkinkan akses kepada perangkat keras menjadi lebih mudah dan teratur. Contoh berkat abstraksi adalah kemudahan pekerjaan programer. Dia tidak perlu membuat koding biner untuk memerintah hardware, tetapi cukup melakukan koding sesuai bahasa pemrograman yang dikuasai. Kernel diibaratkan sebagai nyawa, sedangkan Sistem Operasi sebagai Jasmani. Sehingga kedua elemen itu saling membutuhkan untuk bekerja bersama-sama.

Untuk menjalankan sebuah komputer kita tidak harus menggunakan kernel sistem operasi. Sebuah program dapat saja langsung diload dan dijalankan di atas mesin 'telanjang' komputer, yaitu bilamana pembuat program ingin melakukan pekerjaannya tanpa bantuan abstraksi perangkat keras atau bantuan sistem operasi. Teknik ini digunakan oleh komputer generasi awal, sehingga bila kita ingin berpindah dari satu program ke program lain, kita harus mereset dan meload kembali program-program tersebut. Sebagai pengetahuan, berikut ini adalah generasi bahasa pemrograman komputer.

Generasi ke-1: Machine Language
Pemrograman dengan bahasa mesin sudah jarang di gunakan pada era sekarang dikarenakan sangat kental dengan bahasa biner (0 dan 1).

Generasi ke-2: Assembly Language
Pemrograman dengan bahasa pemrograman assembly merupakan generasi ke dua setelah generasi bahasa mesin, bahasa pemrograman assembly memiliki keyword yang lebih kompleks dari pada bahasa mesin yang hanya berisi karakter “0” dan “1” saja sehingga relatif lebih mudah di mengerti oleh programmer. Assembler adalah nama bahasa pemrograman pada saat itu. Koding pada bahasa assembly cenderung berupa singkatan yang di kenal dengan sebutan kode mnemonic seperti MOV, SUB, CMP, JMP, JGE, JL, LOOP, dan lain sebagainya. Mirip dengan kode command line terminal atau kode command prompt.

Generasi ke-3: Procedural Programming

Copyright © 2003-2016 IlmuKomputer.Com

Bahasa pemrograman prosedural yang mengadopsi bahasa yang mirip dengan bahasa manusia namun masih mencampurkan unsur simbol/karakter khusus seperti {, }, ?, <<, >>, &&, ||, ; dan lain sebagainya, kemudian pada generasi ini juga mulai di kembangkan otomatisasi pengkodean agar programmer fokus pada fungsi utama program yang di kembangkan, unsur simbol / karakter khusus pun mulai di kurangi bahkan pada beberapa bahasa pemrograman telah di hilangkan.

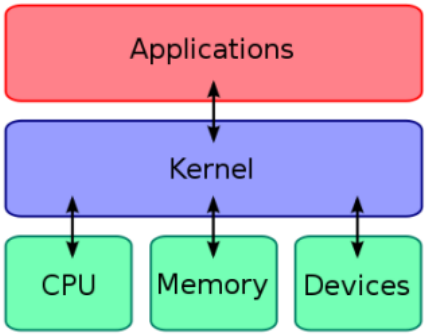
Generasi ke-4: 4 GL (Fourth Generation Language)
Generasi bahasa query terstruktur (SQL, Structured Query Language), SQL sebenarnya merupakan bahasa pemrograman namun penerapannya saat ini lebih banyak pada basis data misalnya MySQL, Oracle Database, SQL Server, PostgreSQL, SQLite dan masih banyak lagi. Pada bidang pemrograman, SQL umumnya di gunakan sebagai bahasa sekunder dengan library built-in atau terpisah, query yang di sematkan dalam kode suatu bahasa pemrograman tujuannya adalah untuk mendefinisikan basis data dan memanipulasi basis data, atau dikenal dengan DDL (Data Definition Language) dan DML (Data Manipulation Language).

Generasi ke-5: Programming Language Based Object Oriented & Web Development
Lebih menekankan pada aspek efisiensi dan penggunaan kembali (re-use-able) atau juga dikenal dengan nama "intellegent programming" (pemrograman kecerdasan) yang menekankan aspek otomatisasi dalam setiap prosesnya.

2. Definisi Kernel

Kernel adalah suatu perangkat lunak yang berisi semua layanan yang disediakan oleh sistem operasi untuk pemakai. Berisi beberapa komponen paling mendasar sistem operasi. Kernel mengelola penjadwalan dan context switching, exception handling dan interrupt handling serta sinkronisasi multiprocessor.
Betul sekali! Fungsi utama kernel adalah untuk mengelola sumber daya komputer dan memungkinkan program-program lain untuk menjalankan dan menggunakan sumber daya komputer tersebut. Sumber daya komputer terdiri dari: CPU, Memori, I/O (device), Hardisk dan Network

1. Central Processing Unit (CPU).
   Sebagai bagian utama komputer, CPU bertugas dalam mengeksekusi program atau proses. Dalam hal ini, kernel bertugas menentukan setiap saat berapa banyak program yang harus dialokasikan ke CPU. Atau disebut scheduling ataumanajemen proses.
2. Memori Komputer.
   Memori digunakan untuk menyimpan instruksi program dan juga data. Keduanya dibutuhkan oleh memori agar suatu program dapat dieksekusi. Biasanya beberapa program akan berusaha mengakses memori, bahkan seringkali meminta memori melebihi yang dimiliki komputer. Disinilah kernel bertugas untuk menentukan porsi pemakaian memori yang dapat dipakai oleh tiap program dan menentukan apa yang



Copyright © 2003-2016 IlmuKomputer.Com

akan dikerjakan ketika tidak cukup tersedia memori.

3. Perangkat I/O.
   Beberapa perangkat I/O atau device yang umumnya dipakai, misal alat input berupa keyboard, mouse dan pen. Sedangkan alat output seperti monitor dan printer dsb. Kernel melanjutkan request dari suatu aplikasi yang ingin melakukan suatu aksiI/O kepada perangkat yang sesuai dan menyediakan metode yang sesuai untuk memakai perangkat tersebut.

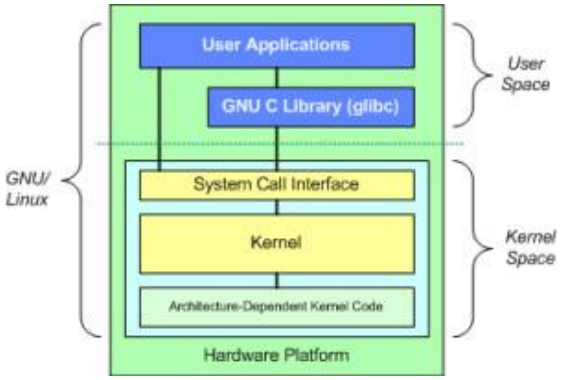
4. Media simpan
   Manajemen penyimpanan dilakukan agar data tidak hilang. Sumber daya ini digunakan untuk menyimpan data dalam memori yang tidak mudah hilang. Contoh media simpan adalah hard-disk. Data tersimpan dalam bentuk file. Jadi subsistem dalam kernel yang berhubungan dengan pembuatan / penghapusan dan pengelolaan file adalah Filesystem. Tugas kernel ini adalah apa yang kita sebut Manajemen Penyimpanan.

5. Network (jaringan)
   Untuk berkomunikasi antar sistem, memerlukan jaringan. Jadi, apa yang perlu dikelola dalam jaringan? Ada tumpukan protokol jaringan dan kartu antarmuka jaringan (NIC). Tugas mengelola jaringan adalah apa yang disebut Manajemen Jaringan. Kernel juga biasanya menyediakan metode untuk sinkronisasi dan komunikasi antara proses (disebut antar-proses komunikasi atau IPC inter-process communication). Kernel dapat mengimplementasikan fitur-fitur ini sendiri, atau bergantung pada beberapa proses dijalankan untuk menyediakan fasilitas untuk proses lainnya, meskipun dalam hal ini harus menyediakan beberapa cara untuk memungkinkan proses IPC untuk mengakses fasilitas yang disediakan oleh masing-masing lain. Akhirnya, kernel harus memberikan program yang berjalan dengan metode untuk membuat permintaan untuk mengakses fasilitas ini. Singkatnya tugas Kernel adalah :
   - Memberi ijin para software atau program aplikasi untuk mengakses sistem hardware yang ada di komputer dengan cara aman
   - Membagi resource komputer dan mengatur komunikasinya
   - Berperan sebagai alat pengatur, saat beberapa program software menggunakan akses hardware bersamaan, maka kernel akan melakukan multiplexing (mengatur waktu dan durasi untuk satu software dalam mengakses hardware tersebut)

3. Layer Kernel

Abstraction Layer adalah suatu cara untuk menyembunyikan detail implementasi dari sekumpulan fungsi. Model software yang menggunakan layer-layer abstraction seperti ini misalnya model OSI layer untuk protokol jaringan komputer serta library grafik pada OpenGL.

Copyright © 2003-2016 IlmuKomputer.Com



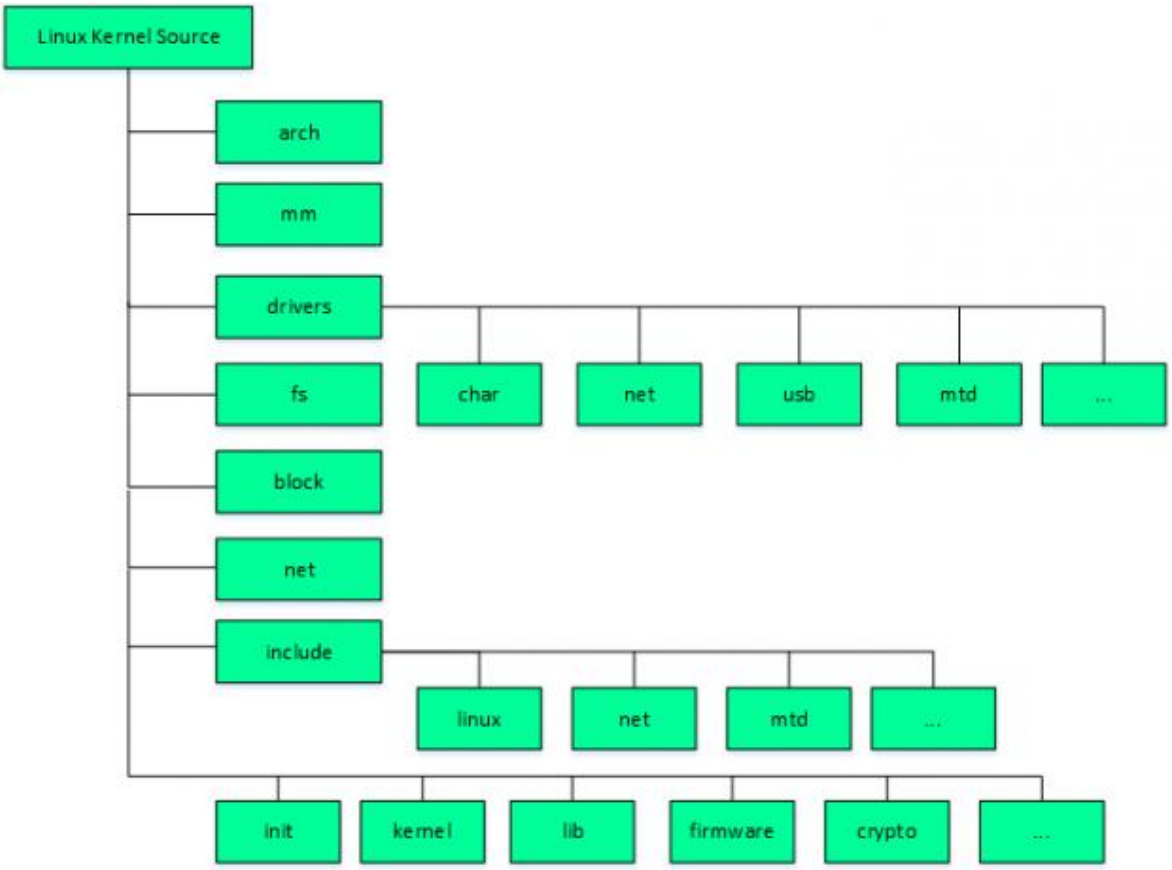
Gbr. Kernel Linux

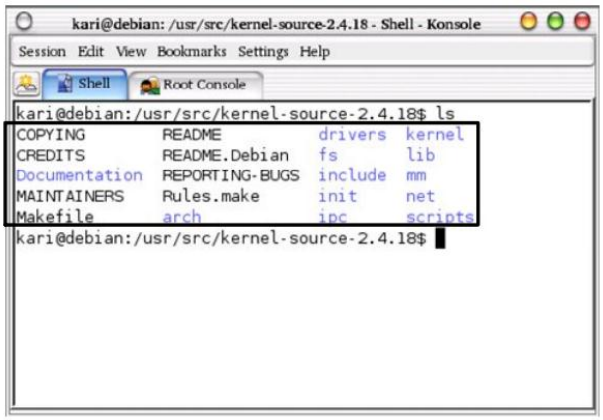
Kebanyakan Sistem Operasi modern menggunakan konsep kernel ini. Keberadaan kernel merupakan konsekuensi dari desain sistem komputer sebagai rangkaian abstracion layer, di mana tiap layer bergantung pada fungsi dari layer di bawahnya. Jika kita tidak ingin menggunakan kernel, maka kita harus mendesain semua software dalam sistem tanpa menggunakan abstraction layer; akibatnya, tentu saja akan menambah kompleksitas desain.

4. Direktori Kernel

Sebuah source kernel Linux biasanya tersedia dalam format kompresi *.gz atau *.bz2. Saat kernel tersebut telah diekstrak, akan terdapat sebuah direktori baru berisi source kernel yang telah terekstrak (Misal: kernel-source-2.4.18). Di dalam direktori tersebut terdapat beberapa file dan subdirektori. setiap file/direktori yang terdapat di dalam kernel memiliki fungsi khusus untuk menangani kompilasi, konfigurasi, serta instalasi kernel Linux ke sistem.

Copyright © 2003-2016 IlmuKomputer.Com





Gambar Tampilan file/direktori /usr/src/kernel-source-2.4.8

Copyright © 2003-2016 IlmuKomputer.Com

Berikut penjelasan masing-masing direktori yang terdapat di dalam direktori kernel-source Linux.

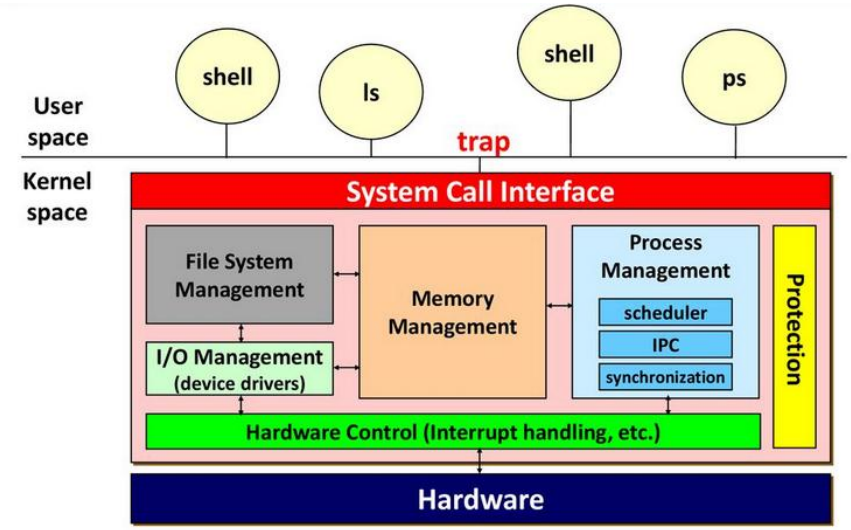
Documentation	Berisi informasi tentang kernel Linux dan subsistemnya. Beberapa informasi diperuntukkan bagi end user (pengguna). Selain itu, terdapat juga beberapa file yang khusus diperuntukkan bagi developer Linux.
arch	Tipe dari arsitektur processor komputer yang didukung oleh kernel Linux. Isi dari sub direktori tersebut antara lain:alpha cris ia64 mips parisc s390 sh sparc64 arm i386 m68k mips64 ppc s390x sparc.
drivers	Berisi kode-kode program driver perangkat keras seperti driver USB drive, kartu jaringan, video card, dll
fs	Berisi kode program yang digunakan untuk mendukung file system yang ada (misal: ntfs, ext2, ext3, msdos, minix, dll.).
include	Berisi header file (*.h) yang akan digunakan oleh source kernel.
init	Kode inisialisasi kernel
ipc	Kode-kode yang diimplementasikan pada System V InterProcess Communication APIs (misal: semaphores, message queues, dan shared memory).
kernel	Berisi kode-kode dasar dari kernel Linux. Berisi kode program untuk proses penjadwalan (scheduling), pengalokasian resource, modul-modul kernel, dll.
lib	Standar pustaka C (misal: ctype, sprintf, dll)
mm	Berisi kode sumber untuk sistem manajemen memori Linux.
net	Berisi kode-kode program untuk diimplementasikan pada protokol jaringan seperti ipv6, ipv4, dll.
scripts	Berisi script-script untuk konfigurasi kernel (misal: xconfig, dll).

5. Fasilitas Kernel

Tujuan utama dari adanya kernel adalah untuk mengatur pembagian sumbersumber daya dan program mana yang boleh mengakses dan menggunakan sumber daya tersebut.

Pada umumnya, sumber daya tersebut antara lain adalah CPU, memori, Perangkat I/O (Device), hardisk dan network.

Copyright © 2003-2016 IlmuKomputer.Com



Sedangkan fasilitas yang disediakan dan diatur pada Kernel seperti pada gambar diatas yakni :

1. **Pengaturan Proses (Process Management)**

Tugas utama kernel adalah membantu eksekusi aplikasi dan mendukungnya dengan fitur seperti abstraksi hardware. Suatu proses mendefinisikan seberapa porsi memori yang dapat diakses oleh suatu aplikasi. Pengaturan proses kernel harus memperhitungkan perangkat yang dibawa oleh hardware untuk perlindungan memori. Untuk menjalankan aplikasi, sebuah kernel pertama kali harus menyediakan space address untuk aplikasi, lalu me-load file yang berisi kode aplikasi ke dalam memori, mempersiapkan stack untuk program dan percabangan ke lokasi lain dalam program, dan kemudian baru memulai eksekusi program.

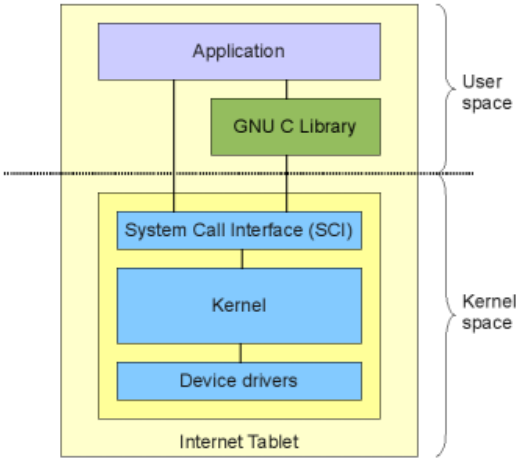
Dalam sebuah sistem pre-emptive multitasking, kernel akan memberi slot waktu kepada setiap program dan berpindah dari satu proses ke proses yang lain dengan cepat hingga terlihat seperti proses-proses ini dieksekusi secara bersamaan. Kernel menggunakan algoritma penjadwalan utnuk menentukan proses mana yang akan dikerjakan berikutnya dan berapa lama waktu yang akan diberikan. Algoritma yang dipilih mungkin akan mengijinkan beberapa proses memiliki prioritas yang lebih tinggi daripada yang lain. Umumnya kernel juga menyediakan cara agar proses-proses ini dapat saling berkomunikasi; hal ini dikenal sebagai Inter-Process Communication (IPC). Pendekatan utama IPC ini adalah untuk shared memory, message passing dan remote procedure call.

2. Pengaturan Memori (Memory Management)

Kernel memiliki akses penuh ke memori sistem dan harus memberikan ijin kepada proses-proses untuk mengakses memori secara aman ketika membutuhkan.
Langkah pertama yang dilakukan untuk melakukan hal ini di antaranya adalah virtual addressing. Virtual addressing mengijinkan kernel untuk menjadikan suatu alamat fisik untuk tampil sebagai alamat lain, alamat virtual. Space alamat virtual mungkin berbeda untuk masing-masing proses; memori yang

Copyright © 2003-2016 IlmuKomputer.Com

diakses suatu proses pada alamat virtual tertentu mungkin berbeda dengan memori yang diakses proses lain pada alamat yang sama.
Hal ini menyebabkan tiap program seakan menjadi satu-satunya program yang berjalan dan karenanya dapat mencegah terjadinya crash antar program. Dalam kebanyakan sistem, alamat virtual suatu program dapat berupa data yang sebenarnya tidak ada dalam memori. Layer abstraksi yang disediakan virtual addressing mengijinkan Sistem Operasi untuk menggunakan penyimpan data, seperti Hard Disk, untuk menyimpan apa yang mestinya ada di memori utama (RAM).



Gbr. Kernel space dan User space

Akibatnya, Sistem Operasi dapat mengijinkan program untuk memakai memori melebihi memori fisik yang yang dimiliki oleh komputer.Virtual addressing juga mengijinkan dibuatnya partisi vritual memori di dua area yang terpisah, satu dipakai oleh kernel (kernel space) dan yang lain untuk aplikasi (user space). Aplikasi tidak diijinkan oleh CPU untuk memakai alamat kernel, yang akhirnya hal ini akan mencegah agar aplikasi tidak merusak kernel.

3. Pengaturan perangkat

Untuk menjalankan fungsinya, aplikasi membutuhkan akses ke perangkat-perangkat yang terhubung ke komputer, yang dikontrol oleh kernel melalui device driver. Sebagai contoh, untuk menampilkan sesuatu di layar, sebuah aplikasi harus membuat request ke kernel. Kernel lalu akan melanjutkan request tersebut ke driver display yang kemudian akan bertanggung jawab mem-plot karakter/pixel. Kernel harus memelihara sejumlah perangkat yang tersedia. Daftar perangkat ini dapat berupa plug and play (dideteksi oleh Sistem Operasi saat perangkat dijalankan), di-configure oleh user, atau bahkan advance (misal dalam system embedded di mana kernel akan diubah jika hardware diubah).

4. System Calls

Copyright © 2003-2016 IlmuKomputer.Com

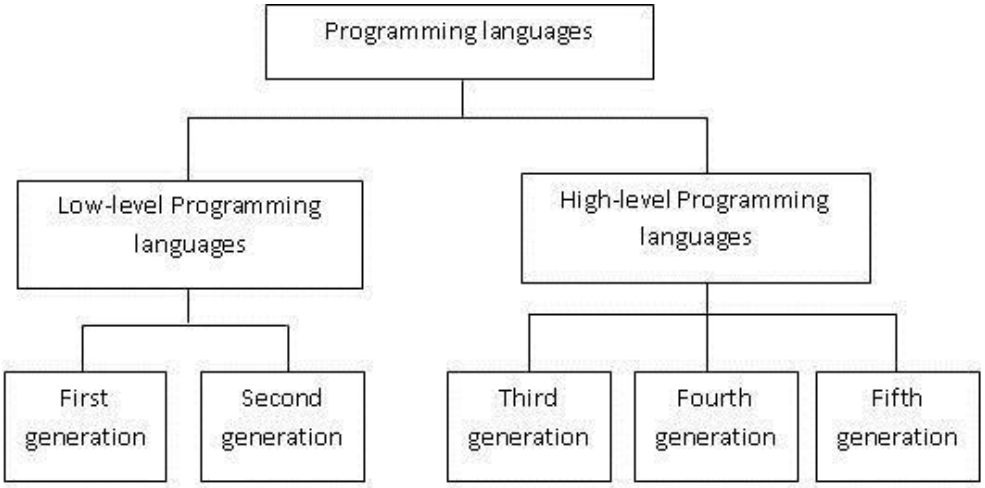
Untuk mengakses layanan yang disediakan oleh kernel, biasanya kernel menyediakan library C atau API (Application Programming Interface) yang selanjutnya akan melibatkan fungsi kernel yang berhubungan. Metode untuk melibatkan fungsi kernel berbeda-beda antar kernel. Jika isolasi memori sedang dipakai, tidak mungkin bagi proses untuk memanggil kernel secara langsung, karena hal itu merupakan pelanggaran aturan akses kontrol dari prosesor. Beberapa kemungkinan antara lain;

- Menggunakan interrupt software-simulated. Metode ini tersedia dalam hampir semua hardware, karenanya sangat umum.
- Menggunakan call gate. Call gate adalah address khusus di mana alamat tersebut ditambahkan dalam daftar yang disimpan dalam memori kernel yang diketahui prosesor. Ketika prosesor mendeteksi call ke lokasi tersebut, prosesor kemudian mengalihkannya ke lokasi target tanpa menyebabkan pelanggaran akses.
- Menggunakan instruksi system call khusus.
- Menggunakan memory-based queue. Sebuah aplikasi yang membuat request dalam jumlah besar tapi tidak perlu menunggu hasilnya dapat menambahkan detil request-nya ke suatu area memori yang di-scan secara priodik oleh kernel untuk menemukan request.

6. Desain Kernel

Sebuah kernel sistem operasi tidak harus ada dan dibutuhkan untuk menjalankan sebuah komputer. Program dapat langsung dijalankan secara langsung di dalam sebuah mesin (contohnya adalah CMOS Setup) sehingga para pembuat program tersebut membuat program tanpa adanya dukungan dari sistem operasi atau hardware abstraction.
Cara kerja seperti ini, adalah Cara kerja yang digunakan pada zaman awal-awal dikembangkannya komputer (pada sekitar tahun 1950). Kerugian dari diterapkannya metode ini adalah pengguna harus melakukan reset ulang komputer tersebut dan memuatkan program lainnya untuk berpindah program, dari satu program ke program lainnya.

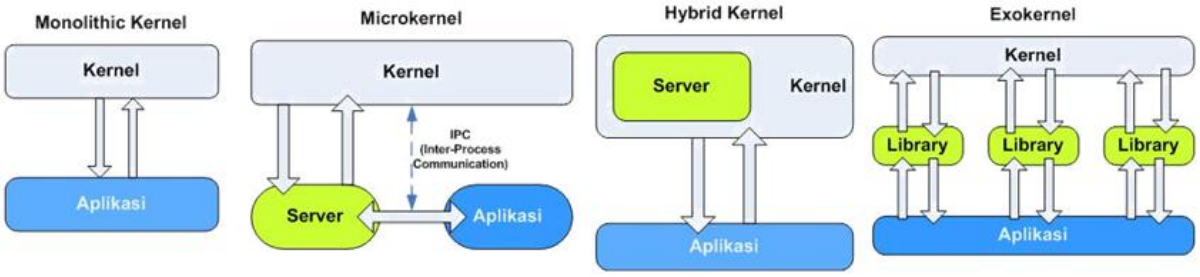
Copyright © 2003-2016 IlmuKomputer.Com



Gbr. Generasi Programming

Selanjutnya, para pembuat program tersebut membuat beberapa komponen program yang sengaja ditinggalkan di dalam komputer, seperti halnya loader atau debugger, atau dimuat dari dalam ROM (Read-Only Memory). Seiring dengan perkembangan zaman komputer yang mengalami akselerasi yang signifikan, metode ini selanjutnya membentuk apa yang disebut dengan kernel sistem operasi.

Selanjutnya, para arsitek sistem operasi mengembangkan kernel sistem operasi yang pada akhirnya terbagi menjadi 4 bagian yang secara desain berbeda, sebagai berikut:



1. Monolithic Kernel.

Adalah struktur kernel yang melingkupi banyak fungsi yang ada di dalam kernel dan menyediakan lapisan pada perangkatkeras yang berada pada sistem operasi di ruang kernel

2. Mikrokernel.

Adalah struktur kernel yang menyediakan sedikit sumber daya perangkat keras dan menggunakan aplikasi yang sedang ber!alan untuk melakukan beberapa fungsionalitas lainnya.

Copyright © 2003-2016 IlmuKomputer.Com

3. Hybrid Kernel

Adalah kernel mirip dengan mikrokernel tetapi pada kernel ini mempunya kelebihan yang dimana memasukkan beberapa kode atau tambahan kode di dalam ruangan kernel untuk meningkatkan performanya

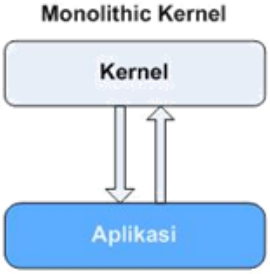
4. Exokernel.

Adalah kernel yang menyediakan program untuk dapat mengakses hardware secara langsung dan dapat melakukan abstraksi yang mirip dengan abstraksi yang dilakukan dalam desain monolithic kernel.

Penjelasannya masing-masing jenis kernel adalah sebagai berikut :

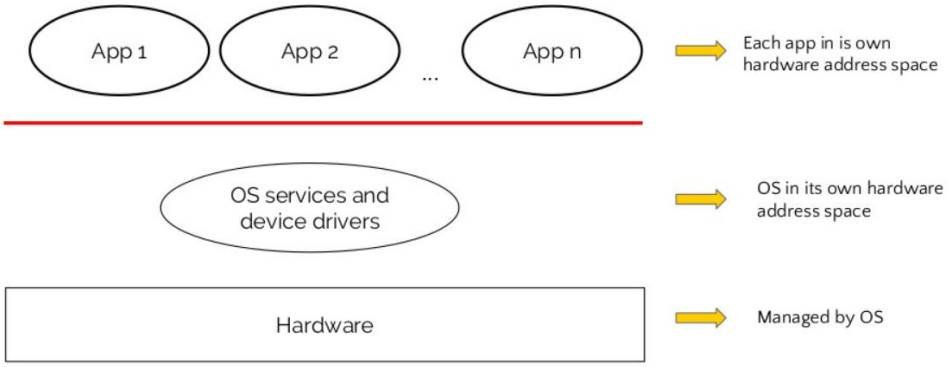
1. Monolitik Kernel

Kernel monolitik merupakan suatu arsitektur kernel yang melingkupi keseluruhan dari sistem operasi untuk berjalan pada ruang kernel dalam modus supervisor. Berbeda dengan arsitektur kernel lainnya (mikrokernel, kernel hibrida), kernel monolitik menyediakan layananan virtual atas perangkat keras secara penuh pada level tingkat tinggi, disertai dengan serangkaian layanan pada level tingkat bawah yang bersifat primitif sebagai layanan basis sistem operasi seperti manajemen proses, konkurensi, dan manajemen ingatan dalam satu atau beberapa modul.



Meskipun jika setiap modul memiliki layanan operasi-operasi tersebut terpisah dari modul utama, integrasi kode yang terjadi di dalam monolithic kernel sangatlah kuat, dan karena semua modul berjalan di dalam address space yang sama, sebuah bug dalam salah satu modul dapat merusak keseluruhan sistem. Akan tetapi, ketika implementasi dilakukan dengan benar, integrasi komponen internal yang sangat kuat tersebut justru akan mengizinkan fitur-fitur yang dimiliki oleh sistem yang berada di bawahnya dieksploitasi secara efektif, sehingga membuat sistem operasi dengan monolithic kernel sangatlah efisie meskipun sangat sulit dalam pembuatannya.

Copyright © 2003-2016 IlmuKomputer.Com



Concept diagram of Monolithic kernel

Pada sistem operasi modern yang menggunakan monolithic kernel, seperti halnya Linux, FreeBSD, Solaris, dan Microsoft Windows, dapat memuat modul-modul yang dapat dieksekusi pada saat kernel tersebut dijalankan sehingga mengizinkan ekstensi terhadap kemampuan kernel sesuai kebutuhan, dan tentu saja dapat membantu menjaga agar kode yang berjalan di dalam ruangan kernel (kernel-space) seminim mungkin.

Di bawah ini ada beberapa sistem operasi yang menggunakan Monolithic kernel:
1. Varian Unix :
    - Linux
    - Syllable
    - BSD (FreeBSD, NetBSD, OpenBSD), Solaris dan OpenSolaris, IBM AIX
2. DOS :
    - DR-DOS
    - MS-DOS
    - Windows 9x (95, 98, Windows 98SE, Me)

3. Mac OS, sampai dengan Mac OS 8.6
4. OpenVMS
5. XTS-400

Kelebihan :
- Program dapat lebih langsung mengakses ke perangkat keras.
- Lebih mudah bagi pemrosesan untuk berkomunikasi satu sama lain.
- Dapat terus bekerja tanpa instalasi tambahan.
- Proses bereaksi lebih cepat karena tidak ada antrian untuk waktu prosesor.

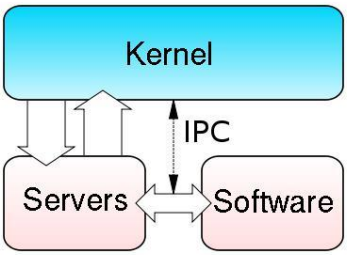
Copyright © 2003-2016 IlmuKomputer.Com

Kekurangan :

- Kapasitas hardisk yang lebih besar
- Kapasitas memori yang lebih besar
- Kurang aman karena semuanya berjalan dalam mode supervisor

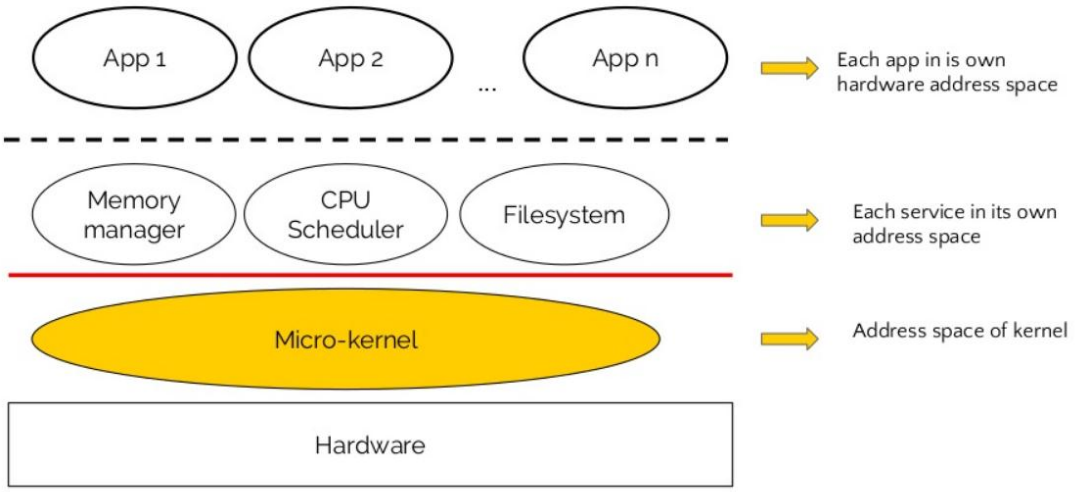
2. Mikrokernel

Pendekatan mikrokernel berisi sebuah abstraksi yang sederhana terhadap hardware, dengan sekumpulan primitif atau system call yang dapat digunakan untuk membuat sebuah sistem operasi agar dapat berjalan, dengan layanan-layanan seperti manajemen thread, komunikasi antar address space, dan komunikasi antar proses. Layanan-layanan lainnya, yang biasanya disediakan oleh kernel, seperti halnya dukungan jaringan, pada pendekatan microkernel justru diimplementasikan di dalam ruangan pengguna (user-space), dan disebut dengan server.



Server atau disebut sebagai peladen adalah sebuah program, seperti halnya program lainnya. Server dapat mengizinkan sistem operasi agar dapat dimodifikasi hanya dengan menjalankan program atau menghentikannya. Sebagai contoh, untuk sebuah mesin yang kecil tanpa dukungan jaringan, server jaringan (istilah server di sini tidak dimaksudkan sebagai komputer pusat pengatur jaringan) tidak perlu dijalankan. Pada sistem operasi tradisional yang menggunakan monolithic kernel, hal ini dapat mengakibatkan pengguna harus melakukan rekompilasi terhadap kernel, yang tentu saja sulit untuk dilakukan oleh pengguna biasa yang awam.

Copyright © 2003-2016 IlmuKomputer.Com



Concept diagram of Micro-kernel

Dalam teorinya, sistem operasi yang menggunakan microkernel disebut jauh lebih stabil dibandingkan dengan monolithic kernel, karena sebuah server yang gagal bekerja, tidak akan menyebabkan kernel menjadi tidak dapat berjalan, dan server tersebut akan dihentikan oleh kernel utama. Akan tetapi, dalam prakteknya, bagian dari system state dapat hilang oleh server yang gagal bekerja tersebut, dan biasanya untuk melakukan proses eksekusi aplikasi pun menjadi sulit, atau bahkan untuk menjalankan server-server lainnya.

Sistem operasi yang menggunakan microkernel umumnya secara dramatis memiliki kinerja di bawah kinerja sistem operasi yang menggunakan monolithic kernel. Hal ini disebabkan oleh adanya overhead yang terjadi akibat proses input/output dalam kernel yang ditujukan untuk mengganti konteks (context switch) untuk memindahkan data antara aplikasi dan server.

Beberapa sistem operasi yang menggunakan microkernel:

- IBM AIX, sebuah versi UNIX dari IBM
- Amoeba, sebuah kernel yang dikembangkan untuk tujuan edukasi
- Kernel Mach, yang digunakan di dalam sistem operasi GNU/Hurd, NexTSTEP, OPENSTEP, dan Mac OS/X
- Minix, kernel yang dikembangkan oleh Andrew Tanenbaum untuk tujuan edukasi
- Symbian OS, sebuah sistem operasi yang populer digunakan pada hand phone, handheld device, embedded device, dan PDA Phone.

Kelebihan

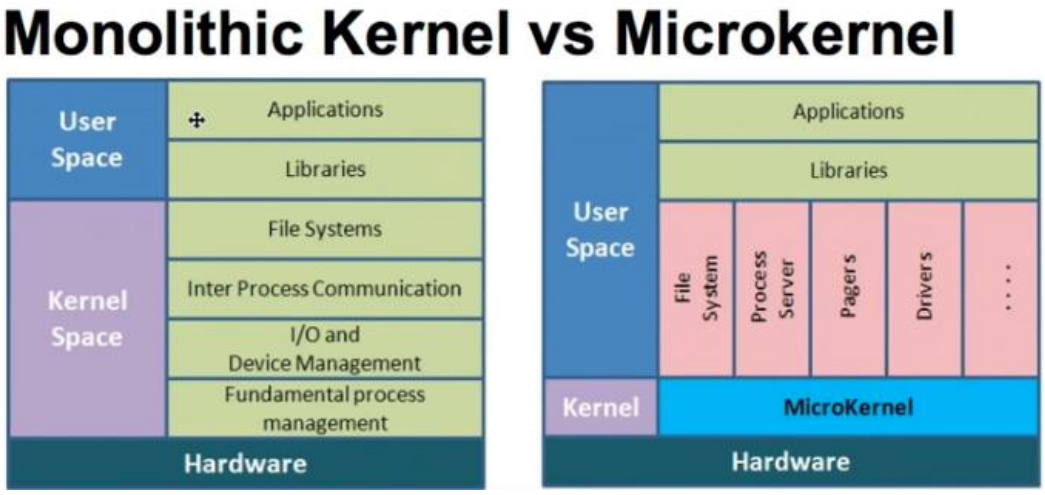
- Portabilitas

Copyright © 2003-2016 IlmuKomputer.Com

- Kapasitas hardisk yang kecil
- Kapasitas memori yang kecil
- Keamanan

Kekurangan

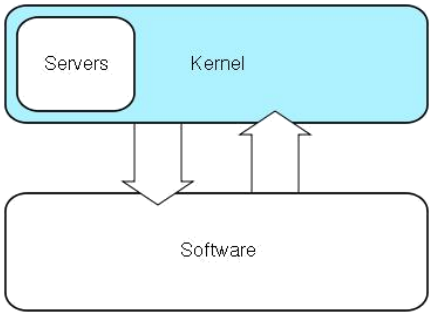
- Hardware harus bekerja melalui driver
- Hardware mungkin bereaksi lebih lambat karena driver dalam user mode
- Proses harus menunggu dalam antrian untuk mendapatkan informasi
- Proses yang tidak bisa mendapatkan akses ke proses lainnya tanpa menunggu



3. Hybrid Kernel

Gagasan di belakang kernel hibrid adalah membuat suatu struktur kernel yang serupa seperti mikrokernel, namun diimplementasikan sebagai kernel monolitik. Berbeda dengan mikrokernel, semua (atau setidaknya hampir semua) layanan sistem terdapat pada ruang kernel dan bukan pada ruang pengguna. Seperti halnya pada kernel monolitik, tidak ada beban penurunan unjuk kerja seperti pada mikrokernel atas mekanisme penyampaian pesan, dan mekanisme proses lainnya, juga memiliki pemahaman yang sama seperti kernel monolitik, tidak ada keuntungan apapun atas penyediaan layanan pada ruang pengguna.

Copyright © 2003-2016 IlmuKomputer.Com



Beberapa orang banyak yang bingung dalam membedakan antara kernel hibrida dan kernel monolitik yang dapat memuat modul kernel setelah proses booting, dan cenderung menyamakannya. Antara kernel hibrida dan kernel monolitik jelas berbeda. Kernel hibrida berarti bahwa konsep yang digunakannya diturunkan dari konsep desain kernel monolitik dan mikrokernel. Kernel hibrida juga memiliki secara spesifik memiliki teknologi pertukaran pesan (message passing) yang digunakan dalam mikrokernel, dan juga dapat memindahkan beberapa kode yang seharusnya bukan kode kernel ke dalam ruangan kode kernel karena alasan kinerja.

Di bawah ini adalah beberapa sistem operasi yang menggunakan kernel hibrida:

- BeOS, sebuah sistem operasi yang memiliki kinerja tinggi untuk aplikasi multimedia.
- Novell NetWare, sebuah sistem operasi yang pernah populer sebagai sistem operasi jaringan berbasis IBM PC dan kompatibelnya.
- Microsoft Windows NT (dan semua keturunannya).
- Darwin (MacOs X)
- Dragonfly BSD

Kelebihan

- Kapasitas hardisk yang sedang
- Kapasitas memori yang sedang
- Pengembang dapat memilih dan memilih apa yang berjalan dalam user mode dan apa yang berjalan dalam mode supervisor.
- Lebih fleksibel dibandingkan model lain

Kekurangan

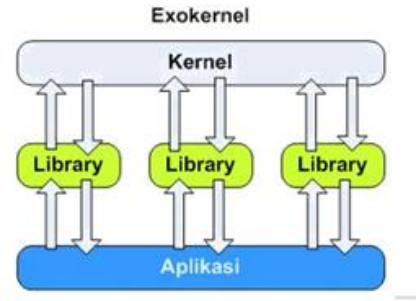
- Bisa menderita proses antrian yang sama seperti mikrokernel
- Device driver harus dikelola oleh pengguna (biasanya).

4. Exokernel

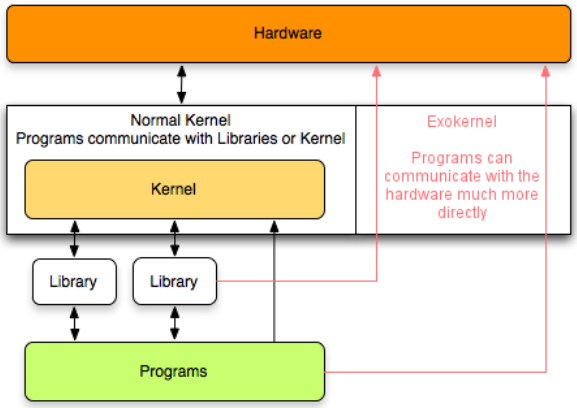
Sebenarnya, Exokernel bukanlah pendekatan kernel sistem operasi yang umum seperti halnya microkernel atau monolithic kernel yang populer, melainkan sebuah struktur sistem operasi yang

Copyright © 2003-2016 IlmuKomputer.Com

disusun secara vertikal.



Ide di balik exokernel adalah untuk memaksa abstraksi yang dilakukan oleh developer sesedikit mungkin, sehingga membuat mereka dapat memiliki banyak keputusan tentang abstraksi hardware. Exokernel biasanya berbentuk sangat kecil, karena fungsionalitas yang dimilikinya hanya terbatas pada proteksi dan penggandaan sumber daya. Kernel-kernel klasik yang populer seperti halnya monolithic dan microkernel melakukan abstraksi terhadap hardware dengan menyembunyikan semua sumber daya yang berada di bawah hardware abstraction layer atau di balik driver untuk hardware. Sebagai contoh, jika sistem operasi klasik yang berbasis kedua kernel telah mengalokasikan sebuah lokasi memori untuk sebuah hardware tertentu, maka hardware lainnya tidak akan dapat menggunakan lokasi memori tersebut kembali.



Exokernel mengizinkan akses terhadap hardware secara langsung pada tingkat yang rendah: aplikasi dan abstraksi dapat melakukan request sebuah alamat memori spesifik baik itu berupa lokasi alamat physical memory dan blok di dalam hard disk. Tugas kernel hanya memastikan bahwa sumber daya yang diminta itu sedang berada dalam keadaan kosong—belum digunakan oleh yang lainnya dan tentu saja mengizinkan aplikasi untuk mengakses sumber daya tersebut. Akses hardware pada tingkat rendah ini mengizinkan para programmer untuk mengimplementasikan sebuah abstraksi yang dikhususkan untuk sebuah aplikasi tertentu, dan tentu saja mengeluarkan sesuatu yang tidak perlu dari kernel agar membuat kernel lebih kecil, dan tentu saja meningkatkan performa. Exokernel biasanya menggunakan library yang disebut dengan libOS untuk melakukan abstraksi. libOS memungkinkan para pembuat aplikasi

Copyright © 2003-2016 IlmuKomputer.Com

untuk menulis abstraksi yang berada pada level yang lebih tinggi, seperti halnya abstraksi yang dilakukan pada sistem operasi tradisional, dengan menggunakan cara-cara yang lebih fleksibel, karena aplikasi mungkin memiliki abstraksinya masing-masing. Secara teori, sebuah sistem operasi berbasis Exokernel dapat membuat sistem operasi yang berbeda seperti halnya Linux, UNIX, dan Windows dapat berjalan di atas sistem operasi tersebut.
Contoh pengguna : MIT Exokernel, Nemesis

7. Versi Kernel

Kernel Linux adalah kernel yang digunakan dalam sistem operasi GNU/Linux. Kernel ini merupakan turunan dari keluarga sistem operasi UNIX, dirilis dengan menggunakan lisensi GNU General Public License (GPL), dan dikembangkan oleh pemrogram di seluruh dunia. Linux merupakan contoh utama dari perangkat lunak OS yang bebas dengan sumber terbuka.

Kernel Linux dirilis dibawah Lisensi Publik Umum GNU versi 2 (GPLv2) (dengan bermacam-macam Firmware dengan lisensi tidak gratis yang bervariatif), dan dikembangkan oleh kontributor dari seluruh dunia. Diskusi perkembangan dari hari ke hari dilakukan di Mailing List Linux Kernel.

Kernel Linux pada awalnya dibayangkan dan kemudian dibuat oleh murid Ilmu Komputer berkebangsaan Finlandia, Linus Torvalds pada tahun 1991. Pengembang dan Pengguna Linux bertambah dengan cepat, lalu mereka mengadaptasi kode dari proyek-proyek software gratis untuk digunakan sebagai Sistem Operasi baru. Linux kernel telah menerima kontribusi dari ribuan programmer. Banyak Distribusi Linux telah dirilis berdasarkan Kernel Linux. Dan berikut merupakan versi kernel linux

- Pada April 1991, Linus Torvalds, seorang mahasiswa berusia 21 tahun dari Universitas Helsinski, Finlandia mulai memikirkan sesuatu ide sederhana untuk suatu Sistem Operasi. Ia memulai dengan task switcher dalam Bahasa Rakitan x86 pada Intel 80386 dan sebuah driver Terminal.

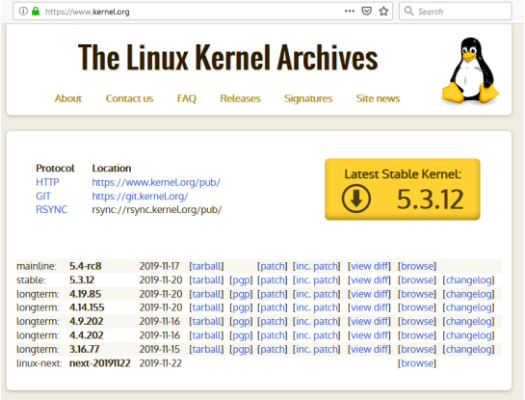
Setelah itu, banyak orang berkontribusi pada proyek ini. Pada awalnya, komunitas MINIX berkontribusi kode dan ide pada Kernel Linux. Pada saat itu, Proyek GNU telah membuat banyak komponen yang diperlukan untuk sebuah sistem operasi gratis, tapi kernel mereka yakni GNU Hurd, pada saat itu masih belum sempurna dan belum tersedia. Sistem operasi BSD pada waktu itu juga belum membebaskan diri dari masalah legalitas. Meskipun pada versi awal fungsinya masih sangat terbatas, pengembang dan pengguna Linux berkembang cepat.

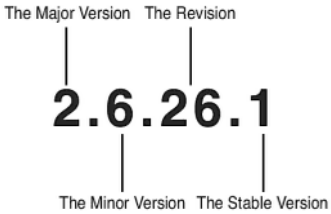
- Pada September 1991, Linux versi 0.01 telah dirilis di server FTP ftp.funet.fi milik Finnish University and Research Network (FUNET). Rilis pertama tersebut memiliki 10,239 baris kode.
- Pada Oktober 1991, Linux versi 0.02 dirilis.
- Pada Desember 1991, Linux 0.11 dirilis, versi ini adalah versi pertama yang di host oleh Linus Torvalds sendiri. Linuz 0.11 dapat di kompilasi oleh komputer yang menjalankan Linux 0.11.
- Saat ia merilis versi 0.12 di bulan Februari 1992, Torvalds mengadopsi GNU General Public License (GPL), dan berhenti menggunakan lisensi yang ia buat sendiri, yang tidak boleh didistribusikan secara komersil.Newsgroup dengan nama alt.os.linux juga dibuat, dan pada 19 Januari 1992, post pertama pada alt.os.linux dibuat. Pada 31 Maret 1992, alt.os.linux berubah menjadi comp.os.linux.X

Copyright © 2003-2016 IlmuKomputer.Com

Window System juga segera di port ke Linux.

- Pada Maret 1992, Linux versi 0.95 adalah versi pertama yang bisa menjalankan X. Lompatan versi yang jauh ini (dari 0.1x ke 0.9x) karena Torvalds merasa fitur-fiturnya sudah lumayan lengkap dan ia menganggap sudah layak untuk langsung ke versi 1.0. Namun, ternyata langkah ini terlalu berlebihan, karena dari tahun 1993 sampai awal 1994, muncul 15 versi perkembangan Linux 0.99.
- Pada 14 Maret 1994, Linux 1.0.0 dirilis, dengan 176,250 barus kode. Pada Maret 1995, Linux 1.2.0 dirilis (310,950 baris kode).
- Versi 2 Linux, dirilis pada 9 Juni 1996, dan diikuti oleh perubahan versi besar dibawah versi kepala 2.
- 25 Januari 1999 - Linux 2.2.0 dirilis (1,800,847 baris kode).
- 18 Desember 1999 - patch IBM mainframe untuk versi 2.2.13 diterbitkan, yang memperbolehkan Linux dijalankan di mesin bisnis.
- 4 Januari 2001 - Linux 2.4.0 dirilis (3,377,902 baris kode).
- 17 Desember 2003 - Linux 2.6.0 dirilis (5,929,913 baris kode).
- Mulai 2004, proses rilis dirubah dan kernel baru mulai dirilis rutin setiap 2-3 bulan, dengan nomor 2.60, 2.6.1, sampai 2.6.39.
- Pada 21 Juli 2011 Linus Torvalds mengumumkan rilis Linux 3.0. Lompatan versi ini bukan karena perubahan teknologi besar-besaran dibandingkan dengan Linux 2.6.39 tetapi menandakan ulang tahun ke 20 kernel linux. Proses waktu rilis tetap sama. Per tahun 2012, Linux 3.2 telah memiliki 14,998,651 baris kode.





Gbr. Website Kernel.org

Kita dapat mengakses kernel linux yang terbaru pada alamat berikut :
http://www.kernel.org
ftp://ftp.cdrom/pub/linux/sunsite/kernel
ftp://sunsite.unc.edu/pub/Linux/kernel
ftp://tsx-11.mit.edu/pub/linux/source/system

Copyright © 2003-2016 IlmuKomputer.Com

8. Kernel Error

Ketika Kernel Linux mendapatkan fatal error dan tidak mungkin melakukan recovery. Maka akan muncul beberapa opsi, seperti :

1. Kernel Panic

```
EIP: 0060: [<c02b2516>]  Not tainted VLI
EFLAGS: 00010286 (2.6.18)
EIP is at nf_queue+0x16/0x240
eax: 00000000 ebx: 00000001 ecx: 00000000 edx: 00000002
esi: dece0920 edi: c03a3aa8 ebp: c0277e70 esp: cf165e78
ds:  007b     es:  007b     ss: 0068

Process sh (pid:5278, threadinfo=cf164000 task=dc5b4ab0)
Stack:   cf165f00 dae29000 00000000 c0277e70 00000001 cf165f00 c03a3aa8 c0277e70
c02b1f7c cf165f00 dece0920 00000002 00000001 dae29000 00000000 c0277e70
00000000 dece0920 00000000 da641b40 da54bcde cc318abc c0277c10 00000002

Call Trace:
[<c0277e70>] ip_local_deliver_finish+0x0/0x230
[<c0277e70>] ip_local_deliver_finish+0x0/0x230
… deleted for space …
[<c012caa6>] do_softirq+0x26/0x30
[<c010528e>] do_IRQ+0x1e/0x30
[<c0103ada>] common_interrupt+0x1a/0x20
Code: c0 75 ec e9 bd e5 e6 ff 8d b6 00 00 00 00 8d bc 27 00 00 00 00 55 57 56 53
83 ec 10 8b 54 24 2c 8b 74 24 28 8b 04 9b c0 42 3a c0 <8b> 38 85 ff 0f 84 96 01
00 00 86 44 24 2c 8b 7c 24 2c c7 44 24
<0>Kernel panic - not syncing: Fatal exception in interrupt
```

```
Your PC ran into a problem and needs to restart. We're just
collecting some error info, and we'll restart for you.
If you'd like to know more, you can search online later for
this error: KERNEL_DATA_INPAGE_ERROR
KERNEL_DATA_INPAGE_ERROR
```

2. Kernel Oops

Copyright © 2003-2016 IlmuKomputer.Com

```
Modules linked in: arc4 ecb crypto_blkcipher ieee80211_crypt_wep vfat fat
   ipt_MASQUERADE iptable_nat nf_nat bridge bnep rfcomm l2cap bluetooth ib_iser
   rdma_cm ib_cm iw_cm ib_sa ib_mad ib_core ib_addr iscsi_tcp libiscsi
   scsi_transport_iscsi sunrpc ipt_REJECT nf_conntrack_ipv4 iptable_filter
   ip_tables ip6t_REJECT xt_tcpudp nf_conntrack_ipv6 xt_state nf_conntrack
   ip6table_filter ip6_tables x_tables cpufreq_ondemand acpi_cpufreq fuse loop
   dm_multipath ipv6 ppdev parport_pc parport video 8139cp output ac battery
   firewire_ohci fglrx(P) firewire_core crc_itu_t 8139too mii ipw2200 ieee80211
   ieee80211_crypt snd_intel8x0m asus_laptop snd_intel8x0 button snd_seq_dummy
   snd_ac97_codec ac97_bus snd_seq_oss snd_seq_midi_event joydev snd_seq pcspkr
   serio_raw snd_seq_device snd_pcm_oss snd_mixer_oss snd_pcm usb_storage
   iTCO_wdt snd_timer snd iTCO_vendor_support soundcore snd_page_alloc sr_mod
   cdrom sg dm_snapshot dm_zero dm_mirror dm_mod pata_acpi ata_generic ata_piix
   libata sd_mod scsi_mod ext3 jbd mbcache uhci_hcd ohci_hcd ehci_hcd [last
   unloaded: microcode]

Pid: 5488, comm: Xorg Tainted: P         (2.6.25.3-18.fc9.i686 #1)
EIP: 0060:[<c04d038d>] EFLAGS: 00213246 CPU: 0
EIP is at task_has_capability+0x48/0x76
EAX: 00000030 EBX: f6852030 ECX: f6a8af28 EDX: 00000000
ESI: f36e50e0 EDI: f354fec8 EBP: f354fed4 ESP: f354fe84
 DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
Process Xorg (pid: 5488, ti=f354f000 task=f35f0e90 task.ti=f354f000)
Stack: c06d48b5 f6852030 f35f0e90 00000003 f35f0e90 f6852030 00000000 00000000
       00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
       00000000 f6852030 f35f0e90 f66f8000 f354fee4 c04d03da f35f0e90 f8d52580
```

9. Sumber

https://knowstuffs.wordpress.com/2012/06/11/linux-kernel-and-architecture/
https://pt.slideshare.net/anil_pugalia/linux-kernel-overview-4566798/5
http://www.encyclopk.com/2017/04/what-is-kernel-in-software-language/
https://androbuntu.com/2018/11/19/pengertian-kernel/
http://blog.unsri.ac.id/download3/26435.pdf
http://id.wikipedia.org/wiki/Kernel_(ilmu_komputer)
http://sekolahteknik.blogspot.com/2011/12/apa-itu-linux-kernel-dan-apa.html
http://etix.wordpress.com/ Mengenal Arsitektur Linux _ Linux Distro & Opensource
Understanding the Linux Kernel (3rd Edition), Marco Cesati, Daniel Pierre Bovet

Biografi Penulis

Yudho Yudhanto. Alumni SD Negeri Surodadi Boyolali, SMPN 1 Boyolali, SMKN 2 Surakarta, S1 Informatika di UNIKOM dan S2 di MTI, Universitas AMIKOM Yogyakarta. SMS/WA 0818-43-82-89 / 0858-7903-7920.Saat ini sebagai Teknopreneur AIDC Barcode, RFID & Sidik Jari. Ketua Asosiasi Barcode RFID Indonesia."Dream - Create - Improve - Share"

Copyright © 2003-2016 IlmuKomputer.Com