

Topik 2 ► Pendahuluan

TUJUAN PEMBELAJARAN

Kemampuan

- Mahasiswa mengidentifikasi konsep sistem operasi.
- Mahasiswa memahami peranan, dan fungsi sistem operasi.

Indikator

- Mahasiswa menerangkan definisi sistem operasi.
- Mahasiswa menyebutkan kedudukan sistem operasi dalam sistem organisasi komputer.
- Mahasiswa menceritakan peranan sistem operasi.
- Mahasiswa menyebutkan fungsi sistem operasi.

2.1. Pengantar Sistem Operasi

2.1.1. Konsep Sistem Operasi Dasar

Apa itu Sistem Operasi?

Sistem Operasi pada dasarnya adalah perantara antara pengguna dan perangkat keras komputer.

- Mengelola sumber daya komputer (perangkat keras, sumber daya abstrak, perangkat lunak)
- Pengalokasi sumber daya.
- Sistem operasi juga digunakan untuk mengontrol program untuk mencegah kesalahan dan penggunaan komputer yang tidak tepat.
- Berdasarkan interupsi

Manfaat Sistem Operasi¶

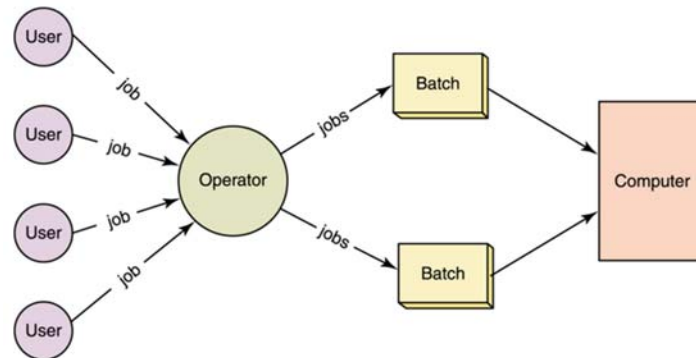
- Menyederhanakan kontrol perangkat keras untuk aplikasi
- Penegak berbagi, keadilan dan keamanan dengan tujuan kinerja keseluruhan yang lebih baik
 - Trade-off antara kelayakan dan kinerja
 - Trade-off antara algoritma optimal dan algoritma buruk – overhead pada OS.
- Menyediakan sumber daya abstrak
- Soket
- Komunikasi antar proses

2.2. Sejarah Sistem Operasi

- Ketika awal diciptakan komputer awal tidak memiliki Sistem Operasi
 - Aplikasi dimuat secara manual
 - Pengguna adalah pakar perangkat keras
- Perangkat Lunak Sistem Pertama adalah pustaka kode (library) untuk mengelola perangkat keras.
- Ini berkembang menjadi sistem pemrosesan batch, di mana beberapa berfokus pada pemrograman aplikasi dan beberapa pada pemrograman sistem.

2.2.1. Pemrosesan Batch

Komputer pada 1960-an dan 70-an tipikalnya adalah mesin besar. Pengolahannya dikelola oleh operator manusia. Operator akan mengatur berbagai pekerjaan dari banyak pengguna menjadi beberapa kelompok.



Operator akan mengatur pekerjaan menjadi beberapa kelompok

2.2.2. Timesharing

Sistem timesharing memungkinkan banyak pengguna berinteraksi dengan komputer pada saat yang bersamaan

Multiprogramming memungkinkan beberapa proses untuk aktif sekaligus, yang memunculkan kemampuan bagi programmer untuk berinteraksi dengan sistem komputer secara langsung, sambil tetap berbagi sumber dayanya

Dalam sistem timesharing, setiap pengguna memiliki mesin virtualnya sendiri, di mana semua sumber daya sistem (berlaku) tersedia untuk digunakan

2.3. Topik Penelitian Sistem Operasi Saat Ini

Multiprocessing simetris

Memungkinkan untuk beberapa CPU untuk memproses banyak pekerjaan pada saat yang bersamaan. CPU bersifat independen satu sama lain, tetapi masing-masing memiliki akses ke sistem operasi.

Multiprocessing asimetris

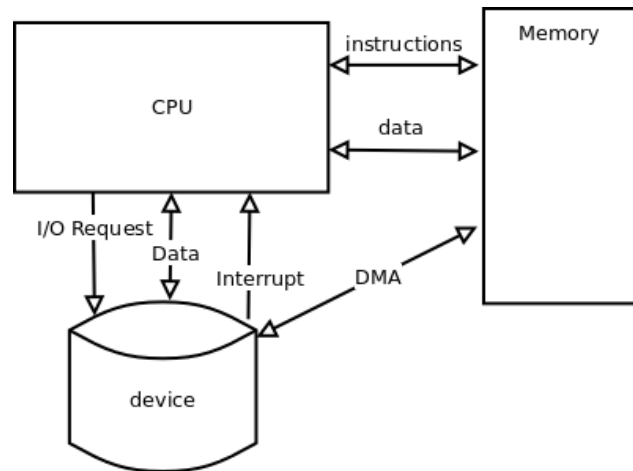
Beberapa fungsi sistem operasi ditugaskan untuk mensubordinasikan prosesor, yang mengambil instruksi mereka dari CPU utama.

Pengolahan terdistribusi

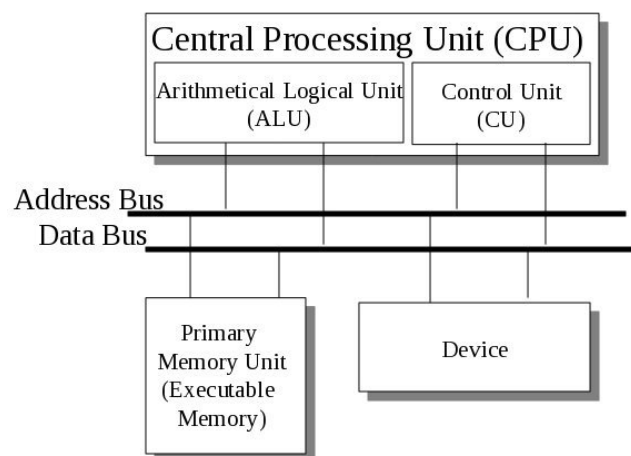
Prosesor ditempatkan di lokasi terpencil dan terhubung satu sama lain melalui perangkat telekomunikasi. Berbeda dari sistem multiprocessing simetris karena mereka tidak berbagi memori. Komputasi dapat tersebar di antara beberapa prosesor.

2.3.1. Arsitektur computer

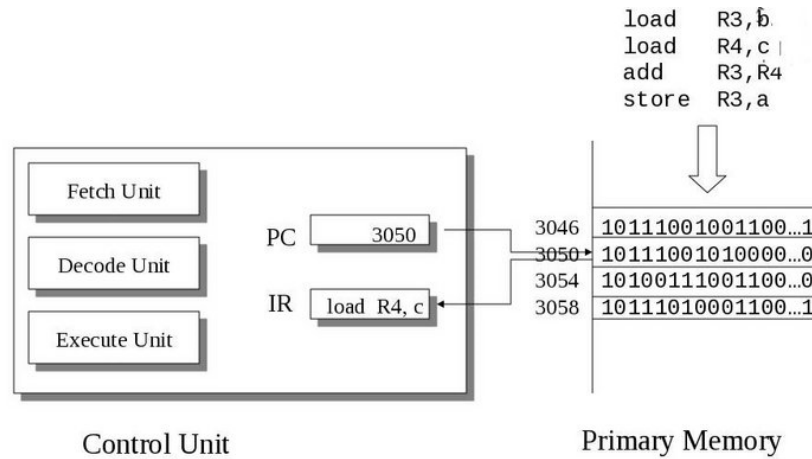
Komputer adalah suatu sistem komunikasi. Data secara konstan dipindahkan antara CPU, memori, dan berbagai perangkat. CPU menggunakan alamat I / O untuk mengarahkan data ke perangkat tertentu. Perangkat pada gilirannya menggunakan interupsi untuk memberitahukan CPU dan sistem operasi kebutuhan mereka.



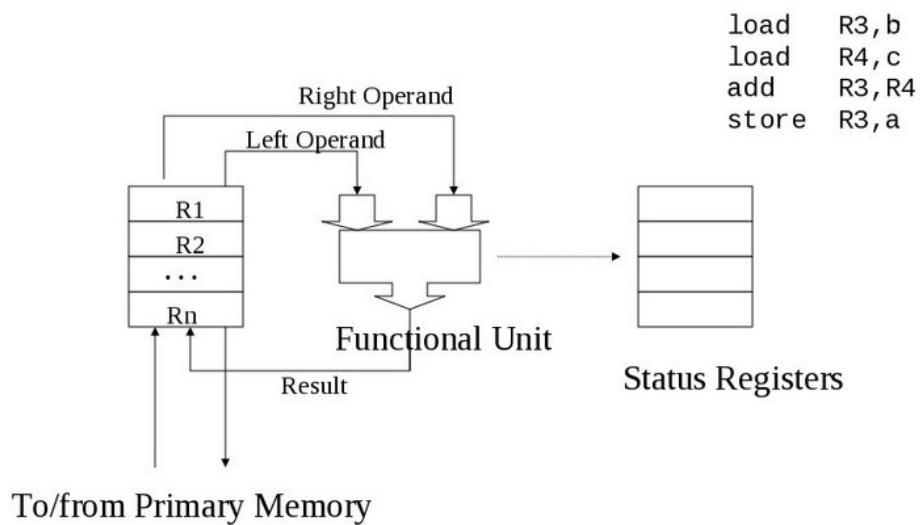
2.3.2. Arsitektur von Neumann



2.3.4. Control Unit



2.3.5. Arithmetic Logic Unit (ALU)



2.3.6. The von Neumann Loop

Fetch Phase

Ambil instruksi selanjutnya dan simpan di daftar instruksi

Execute Phase

Unit ALU atau I / O menjalankan instruksi

- ALU melakukan perhitungan
- I / O unit memuat atau menyimpan data antara memori utama dan register

2.4. Dasar-dasar tentang Cara Kerja Sistem Operasi

2.4.1 Peran Interupsi

Interupsi adalah sinyal yang dikirim ke CPU oleh perangkat eksternal, biasanya perangkat I / O. Mereka memberi tahu CPU untuk menghentikan aktivitasnya saat ini dan menjalankan bagian yang sesuai dari sistem operasi.

Ada tiga jenis interupsi:

- **Hardware Interrupts** dihasilkan oleh perangkat keras untuk memberi sinyal bahwa mereka membutuhkan perhatian dari OS. Mereka mungkin baru saja menerima beberapa data (misalnya, penekanan tombol pada keyboard atau data pada kartu ethernet); atau mereka baru saja menyelesaikan tugas yang sebelumnya diminta oleh sistem operasi, seperti mentransfer data antara hard drive dan memori.
- **Software Interrupts** dihasilkan oleh program ketika mereka ingin meminta panggilan sistem dilakukan oleh sistem operasi.
- **Trap Interrupts** Perangkat dihasilkan oleh CPU itu sendiri untuk menunjukkan bahwa beberapa kesalahan atau kondisi terjadi yang memerlukan bantuan dari sistem operasi.

Interupsi penting karena memberi pengguna kontrol yang lebih baik terhadap komputer. Tanpa interupsi, pengguna mungkin harus menunggu aplikasi yang memiliki prioritas yang lebih tinggi selesai dijalankan oleh CPU. Interupsi ini memastikan bahwa CPU akan segera menangani prosesnya.

2.4.2 Mode Eksekusi CPU

Ada dua mode eksekusi, yang dikenal sebagai mode pengguna dan mode kernel atau supervisor. Mode pengguna dibatasi karena instruksi tertentu tidak dapat dijalankan, register tertentu tidak dapat diakses, dan perangkat I / O tidak dapat diakses. Mode kernel tidak memiliki batasan ini. Panggilan sistem akan mengatur CPU ke mode kernel, begitu pula dengan perangkat dan interupsi. Program aplikasi tidak bisa melakukan ini.

Mode bit: Supervisor atau mode Pengguna

- Mode pengawas
 - Dapat menjalankan semua instruksi mesin
 - Dapat mereferensikan semua lokasi memori
- Mode pengguna
 - Hanya dapat mengeksekusi sebagian instruksi
 - Hanya dapat mereferensikan sebagian lokasi memori

2.4.3. Tanggapan CPU terhadap Interupsi

Kunci utama untuk memahami cara kerja sistem operasi adalah memahami apa yang dilakukan CPU ketika terjadi interupsi. Perangkat keras CPU melakukan hal yang sama untuk setiap interupsi, yang memungkinkan sistem operasi untuk mengambil kendali dari proses pengguna yang sedang berjalan. Peralihan proses yang berjalan untuk mengeksekusi kode dari kernel OS disebut sebagai konteks switch.

CPU bergantung pada data yang terdapat dalam register untuk menangani interupsi dengan benar. Satu register menyimpan pointer ke blok kontrol proses dari proses yang sedang berjalan. Register ini diatur setiap kali proses dimuat ke dalam memori. Register lainnya menyimpan pointer ke tabel yang berisi petunjuk ke petunjuk di kernel OS untuk penanganan interupsi dan panggilan sistem. Nilai dalam daftar dan isi dari tabel ini ditetapkan ketika sistem operasi diinisialisasi pada saat boot.

CPU melakukan tindakan berikut sebagai tanggapan atas interupsi:

- a) Menggunakan pointer ke blok kontrol proses saat ini, state dan semua nilai daftar untuk proses disimpan untuk digunakan ketika proses ini kemudian dimulai kembali.
- b) Bit mode CPU dialihkan ke mode pengawasan.
- c) Dengan menggunakan pointer ke tabel interrupt handler dan vektor interupsi, lokasi kode kernel yang akan dieksekusi ditentukan. Vektor interupsi adalah IRQ untuk interupsi perangkat keras (dibaca dari register pengontrol interupsi) dan argumen dalam bahasa assembly untuk interupsi perangkat lunak.
- d) Pemrosesan beralih ke bagian kernel yang sesuai.

2.5. Bagian dari Sistem Operasi

Ada 4 tugas utama yang dilakukan oleh sistem operasi.

a) **Manajemen proses**

Suatu proses adalah program eksekusi. Ini memiliki kode, data, seperangkat sumber daya tertentu yang dialokasikan untuk itu, dan satu atau lebih thread eksekusi melalui kode. OS mengelola alokasi sumber daya untuk proses-proses yang berjalan, dan juga menyediakan panggilan sistem untuk mengelola proses-proses yang berjalan.

b) **Manajemen memori**

Memori harus dibagi antara OS dan program aplikasi. OS harus mengatur alokasi memori untuk memproses dan mengontrol perangkat keras manajemen memori yang menentukan lokasi memori mana yang dapat diakses suatu proses.

c) **Manajemen Sistem File**

Komputer memproses informasi yang harus ditransmisikan, diproses, atau disimpan. Sistem file adalah kumpulan file objek sistem abstrak yang terorganisir. OS menyediakan aturan sederhana untuk memanipulasi objek-objek ini.

d) Manajemen perangkat

Informasi dikirim melalui perangkat input dan output komputer. Proses mengakses perangkat ini menggunakan antarmuka panggilan sistem. OS mencoba mengelola perangkat tersebut dengan cara yang membuat mereka secara efisien berbagi di antara semua proses yang membutuhkannya. Panggilan sistem adalah antarmuka pemrograman ke layanan yang disediakan oleh OS, biasanya ditulis dalam C / C ++.

2.6. Desain dan Implementasi Sistem Operasi

2.6.1. Tujuan Desain

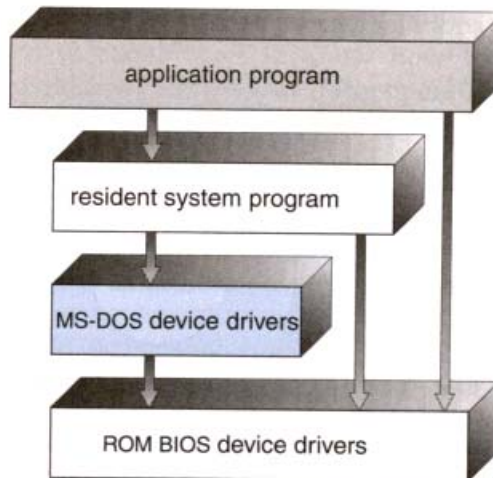
- Pada level tertinggi, desain sistem didominasi oleh pilihan perangkat keras dan tipe sistem.
- Di luar tingkat ini, persyaratan dapat dibagi menjadi dua kelompok: tujuan pengguna, dan sasaran sistem.
 - Tujuan pengguna termasuk kenyamanan, keandalan, keamanan, dan kecepatan.
 - Sasaran sistem termasuk kemudahan desain, implementasi, pemeliharaan, fleksibilitas, dan efisiensi.

2.6.2. Implementasi

- Pada awalnya, sistem operasi ditulis Bahasa assembly, tetapi sekarang Bahasa yang umum digunakan adalah C / C ++.
- Blok-blok kecil kode assembly masih diperlukan, terutama yang berkaitan dengan beberapa fungsi I / O tingkat rendah di driver perangkat, menyalakan dan mematikan interupsi, dan Test dan Set Instruction untuk Fasilitas Sinkronisasi.
- Menggunakan bahasa tingkat yang lebih tinggi memungkinkan kode ditulis lebih cepat. Ini juga membuat OS lebih mudah untuk di-port ke platform perangkat keras yang berbeda.

2.7. Struktur Sistem Operasi

2.7.1. Struktur Sederhana



Dalam system MS-DOS, aplikasi dapat mem-bypass sistem operasi.

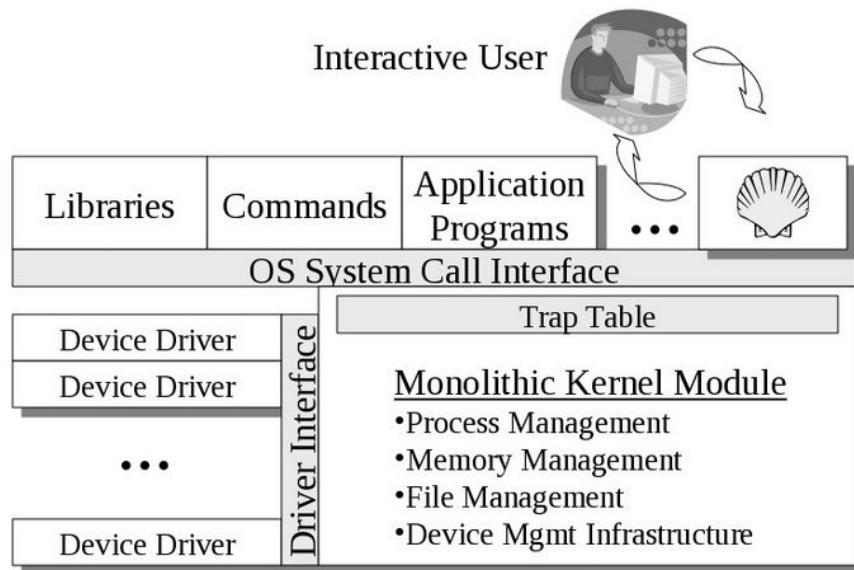
Sistem operasi seperti MS-DOS dan UNIX asli tidak memiliki struktur yang terdefinisi dengan baik.

Tidak ada Mode Eksekusi CPU (pengguna dan kernel), sehingga kesalahan dalam aplikasi dapat menyebabkan seluruh sistem crash.

2.7.2. Pendekatan Monolitik

Fungsionalitas OS dipanggil dengan pemanggilan fungsi sederhana di dalam kernel, yang merupakan satu program besar.

Driver perangkat dimuat ke kernel yang sedang berjalan dan menjadi bagian dari kernel.

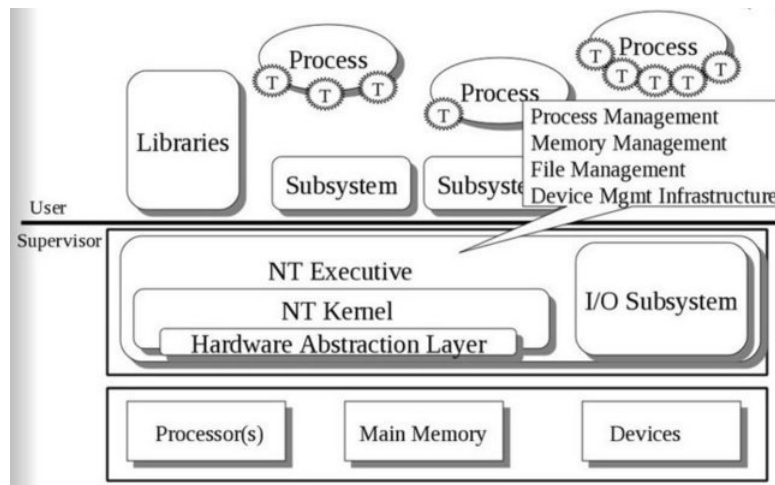


Kernel monolitik, seperti Linux dan sistem Unix lainnya.

2.7.3. Layered Approach

Pendekatan ini memecah sistem operasi menjadi lapisan yang berbeda.

- Hal ini memungkinkan pelaksana untuk mengubah cara kerja bagian dalam, dan meningkatkan modularitas.
- Selama antarmuka eksternal dari rutinitas tidak berubah, pengembang memiliki lebih banyak kebebasan untuk mengubah kerja bagian dalam rutinitas.
- Dengan pendekatan berlapis, lapisan bawah adalah perangkat keras, sedangkan lapisan tertinggi adalah antarmuka pengguna.
 - *Keuntungan* utama adalah kesederhanaan konstruksi dan debugging.
 - *Kesulitan* utama adalah mendefinisikan berbagai lapisan.
 - *Kerugian* utama adalah bahwa OS cenderung kurang efisien daripada implementasi lainnya.



Sistem Operasi Microsoft Windows NT. Level terendah adalah kernel monolitik, tetapi banyak komponen OS berada pada level yang lebih tinggi, tetapi masih merupakan bagian dari OS.

2.7.4. Microkernels

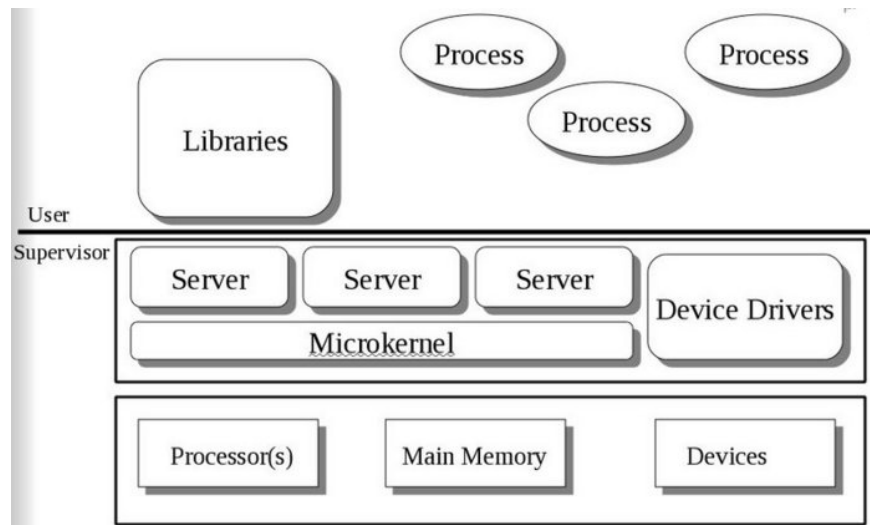
Ini struktur sistem operasi dengan menghapus semua bagian yang tidak penting dari kernel dan mengimplementasikannya sebagai sistem dan program tingkat pengguna.

- Umumnya mereka menyediakan proses minimal dan manajemen memori, dan fasilitas komunikasi.
- Komunikasi antar komponen OS disediakan oleh pengiriman pesan.

Manfaat dari mikrokernel adalah sebagai berikut:

- Memperluas sistem operasi menjadi lebih mudah.
- Setiap perubahan pada kernel cenderung lebih sedikit, karena kernel lebih kecil.
- Mikrokernel juga menyediakan lebih banyak keamanan dan keandalan.

Kerugian utama adalah kinerja yang buruk karena peningkatan overhead sistem dari pesan yang lambat.



Arsitektur Mikrokernel.

2.8. Boot Sistem

- Booting sistem dilakukan dengan memasukkan kernel ke dalam memori utama, dan memulai eksekusinya.
- CPU diberikan peristiwa reset, dan register instruksi dimuat dengan lokasi memori yang telah ditentukan, di mana eksekusi dimulai.
 - Program bootstrap awal ditemukan di memori read-only BIOS.
 - Program ini dapat menjalankan diagnostik, menginisialisasi semua komponen sistem, memuat dan memulai memuat Sistem Operasi. (Disebut booting strapping)
 - Program loader memuat dan memulai sistem operasi.
 - Ketika sistem Operasi dimulai, boot system menyiapkan struktur data yang diperlukan dalam memori, menetapkan beberapa register di CPU, dan kemudian membuat dan memulai program tingkat pengguna pertama. Dari titik ini, sistem operasi hanya berjalan sebagai respons terhadap interupsi. Lihat Tanggapan CPU untuk Interupsi

2.9. Lingkungan Sistem Operasi

2.9.1. Eksekusi Program

Sebuah OS harus dapat memuat program ke dalam memori dan menjalankan program itu. Juga, program ini harus dapat mengakhiri eksekusi, apakah normal atau tidak normal.

Operasi I / O

- Program yang sedang berjalan mungkin memerlukan I / O dalam bentuk file atau perangkat keluaran.
- Karena keamanan atau kebutuhan efisiensi, pengguna sering tidak mengontrol I / O, sehingga OS harus menyediakan sarana untuk I / O.

Manipulasi sistem file

- Program harus dapat membaca dan menulis ke file, juga menghapus, mengganti nama, dan menghapusnya.
- Juga perlu bahwa file memiliki fungsi izin untuk memberikan keamanan tambahan.

Komunikasi

- Sering kali satu proses harus berkomunikasi dengan proses lain.
- Komunikasi mungkin antara proses satu komputer, atau komputer yang terhubung melalui jaringan.
- Komunikasi dapat diimplementasikan melalui memori bersama atau melalui pengiriman pesan.

Deteksi kesalahan

- Sebuah OS harus terus-menerus mencari kesalahan, dan tahu cara mengatasinya.
- Untuk setiap kesalahan, OS harus memiliki tindakan yang harus diambil untuk memastikan komputasi yang benar dan konsisten.
- Fasilitas debug meningkatkan penggunaan pengguna dan pengembangan sistem.

2.9.2 Fungsi Efisiensi OS

Alokasi Sumber Daya

- Beberapa pengguna atau pekerjaan membutuhkan sumber daya yang dialokasikan untuk masing-masing.
- Beberapa sumber daya memiliki kode alokasi khusus, sementara yang lain mungkin memiliki lebih banyak permintaan umum dan kode rilis.

Perlindungan dan Keamanan

- Seharusnya tidak mungkin untuk satu proses untuk mengganggu proses lain atau OS itu sendiri.
- Ini berarti bahwa semua akses ke sumber daya sistem dikontrol.

- Melindungi sistem dari ancaman luar dilakukan melalui otentikasi pengguna, dan untuk membela perangkat I / O eksternal.
- File log dapat membantu mendeteksi dan melacak upaya infiltrasi luar.

2.10. Antarmuka pengguna

Seperti disebutkan di atas, pengguna umumnya menggunakan antarmuka baris perintah atau GUI untuk berinteraksi dengan sistem operasi.

2.10.1. Command Interpreter

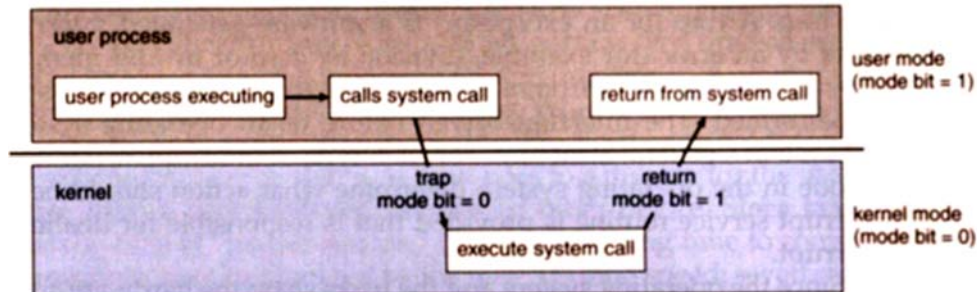
Beberapa sistem operasi termasuk penerjemah perintah di kernel. Beberapa, seperti sistem operasi Windows dan Linux yang populer, menggunakan penerjemah perintah sebagai program khusus yang berjalan saat pengguna masuk atau pekerjaan dimulai.

- di Windows, ini adalah prompt MS-DOS.
- Linux memiliki lebih banyak opsi. Interpreter perintah di Linux dikenal sebagai shell. Shell yang paling umum digunakan adalah shell Bash, tetapi yang lain seperti shell Korn, C shell, dan Bourne shell ada. Kebanyakan shell menyediakan fungsionalitas yang serupa, preferensi pribadi biasanya menentukan shell mana yang terbaik.
- Fungsi utama dari utilitas perintah adalah untuk menerima dan mengeksekusi perintah yang dibuat pengguna berikutnya.
 - Banyak perintah dimaksudkan untuk memanipulasi file.
 - Sistem operasi seperti UNIX mengimplementasikan perintah melalui program sistem. Seringkali program-program ini disimpan sebagai file teks, yang memungkinkan pemrogram menambahkan fungsionalitas tambahan ke utilitas.

2.10.2. Antarmuka Pengguna Grafis

- GUI menyediakan jendela berbasis mouse dan sistem menu sebagai antarmuka.
- Pengguna Windows lebih cenderung menggunakan GUI daripada antarmuka baris perintah MS-DOS, sementara pengguna UNIX umumnya lebih suka menggunakan antarmuka baris perintah shell daripada GUI.

2.11. Panggilan Sistem



Panggilan sistem menyediakan antarmuka ke layanan sistem operasi.

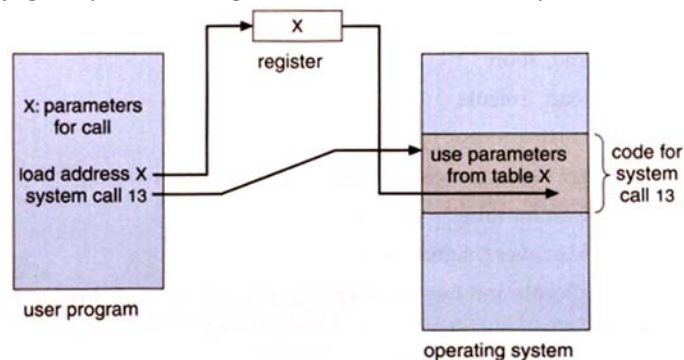
Pengembang aplikasi sering tidak memiliki akses langsung ke panggilan sistem, tetapi dapat mengaksesnya melalui antarmuka pemrograman aplikasi (API). Fungsi-fungsi yang termasuk dalam API memanggil panggilan sistem yang sebenarnya. Dengan menggunakan API, manfaat tertentu dapat diperoleh:

- Portabilitas: selama sistem mendukung API, program apa pun yang menggunakan API tersebut dapat dikompilasi dan dijalankan.
- Kemudahan Penggunaan: menggunakan API dapat secara signifikan lebih mudah kemudian menggunakan sistem panggilan yang sebenarnya.

2.11.1. Parameter Panggilan Sistem

Ada tiga metode umum untuk mengirimkan parameter ke OS:

- a) Parameter dapat diteruskan dalam register.
- b) Ketika ada lebih banyak parameter daripada register, parameter dapat disimpan dalam satu blok dan alamat blok dapat dilewatkan sebagai parameter ke register.
- c) Parameter juga dapat didorong atau dimatikan dari tumpukan oleh sistem operasi.



2.11.2. Jenis Panggilan Sistem

Ada 5 kategori berbeda dari panggilan sistem:

kontrol proses, manipulasi file, manipulasi perangkat, pemeliharaan informasi dan komunikasi.

Kontrol Proses

Program yang berjalan harus dapat menghentikan eksekusi baik secara normal atau tidak normal. Ketika eksekusi dihentikan secara tidak normal, sering kali dump memori diambil dan dapat diperiksa dengan debugger.

Manajemen File

Beberapa panggilan sistem umum adalah membuat , menghapus , membaca , menulis , mengubah posisi , atau menutup . Juga, ada kebutuhan untuk menentukan atribut file - dapatkan dan atur atribut file. Banyak kali OS menyediakan API untuk membuat panggilan sistem ini.

Manajemen Perangkat

Proses biasanya membutuhkan beberapa sumber daya untuk dieksekusi, jika sumber daya ini tersedia, mereka akan diberikan dan kontrol dikembalikan ke proses pengguna. Sumber daya ini juga dianggap sebagai perangkat. Beberapa bersifat fisik, seperti kartu video, dan lainnya bersifat abstrak, seperti file.

Program pengguna meminta perangkat, dan ketika selesai mereka melepaskan perangkat. Mirip dengan file, kita dapat membaca , menulis , dan memposisikan ulang perangkat.

Manajemen Informasi

Beberapa panggilan sistem ada semata-mata untuk mentransfer informasi antara program pengguna dan sistem operasi. Contoh dari ini adalah waktu , atau tanggal .

OS juga menyimpan informasi tentang semua prosesnya dan menyediakan panggilan sistem untuk melaporkan informasi ini.

Komunikasi

Ada dua model komunikasi interproses, model pengiriman pesan, dan model memori bersama.

- Pengiriman pesan menggunakan kotak pesan umum untuk mengirimkan pesan antar proses.
- Memori bersama menggunakan panggilan sistem tertentu untuk membuat dan mendapatkan akses untuk membuat dan mendapatkan akses ke wilayah memori yang dimiliki oleh proses lain. Kedua proses bertukar informasi dengan membaca dan menulis dalam data bersama.

2.12. Program Sistem

Program-program ini biasanya bukan bagian dari kernel OS, tetapi merupakan bagian dari keseluruhan sistem operasi.

2.12.1. Manajemen file

Program-program ini membuat, menghapus, menyalin, mengganti nama, mencetak, membuang, daftar, dan umumnya memanipulasi file dan direktori.

2.12.2. Informasi status

Beberapa program hanya meminta tanggal dan waktu, dan permintaan sederhana lainnya. Lainnya menyediakan informasi kinerja, pencatatan, dan debugging terperinci. Output dari file-file ini sering dikirim ke jendela terminal atau jendela GUI

Catatan Registri ada di kategori ini.

2.12.3. Modifikasi file

Program seperti editor teks digunakan untuk membuat, dan memodifikasi file.

2.12.4. Komunikasi

Program-program ini menyediakan mekanisme untuk membuat koneksi virtual di antara proses, pengguna, dan komputer lain. Email dan browser web adalah beberapa contoh.



Pertanyaan Review

1.



DISCUSSION