



**MODUL SISTEM OPERASI
(CCI210)**

**MODUL SESI 1
KONSEP DAN STRUKTUR SISTEM OPERASI**

**DISUSUN OLEH
ADI WIDIANTONO, SKOM, MKOM.**

Universitas
Esa Unggul

**UNIVERSITAS ESA UNGGUL
2020**

PENGANTAR

A. Kemampuan Akhir Yang Diharapkan

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

1. Menguraikan visi dan misi Universitas Esa Unggul
2. Merinci topik-topik perkuliahan Sistem Operasi
3. Mengidentifikasi buku referensi serta komponen dan proporsi penilaian mata kuliah Sistem Operasi.

B. Uraian dan Contoh

1. Visi dan Misi

Universitas Esa Unggul mempunyai visi menjadi perguruan tinggi kelas dunia berbasis intelektualitas, kreatifitas dan kewirausahaan, yang unggul dalam mutu pengelolaan dan hasil pelaksanaan Tridarma Perguruan Tinggi.

Untuk mewujudkan visi tersebut, maka Universitas Esa Unggul menetapkan misi-misi sebagai berikut :

- a. Menyelenggarakan pendidikan tinggi yang bermutu dan relevan
- b. Menciptakan suasana akademik yang kondusif
- c. Memberikan pelayanan prima kepada seluruh pemangku kepentingan

2. Topik Perkuliahan

Dalam materi perkuliahan ini, diharapkan mahasiswa dapat memahami konsep, sejarah dan perkembangannya, serta teknologi dari Sistem Operasi. Sehingga dari penguasaan tersebut mahasiswa dapat memanfaatkan dalam pengembangan sebuah sistem, mengetahui tantangan yang dihadapi dan dapat memanfaatkan teknologi dari sistem operasi.

Topik mata kuliah Sistem Operasi terdiri dari topik-topik tentang perkembangan, cara kerja dan struktur dari sistem operasi. Adapun topik-topik perkuliahan system operasi adalah :

- a. Topik 01 - Pengantar dan konsep Sistem Operasi

- b. Topik 02 – Manajemen Proses
- c. Topik 03 - Thread
- d. Topik 04 - Teknik Scheduling
- e. Topik 05 – Manajemen Memori
- f. Topik 06 – Teknik Swapping
- g. Topik 07 – Virtual Memori
- a. Topik 08 – Algoritma Paging
- b. Topik 09 – System File
- c. Topik 10 – Manajemen File
- d. Topik 11 – Dead Lock System
- e. Topik 12 – *Embedded System*
- f. Topik 13 - Keamanan Sistem Operasi
- g. Topik 14 – Tren Teknologi Sistem Operasi (*Virtual Operating System*)

Sistem operasi dimana melibatkan perkembangan teknologi perangkat keras dan perangkat lunak. Bergerak pada perangkat pribadi seperti PC, Laptop, bahkan perangkat mobile (Smartphone misalnya). Tetapi juga meliputi system computer besar yang meliputi pengelolaan berbagai resource/sumber daya hingga muncul platform virtual.

3. Buku Referensi dan Komponen Penilaian

Mata kuliah Sistem Operasi memiliki tujuan perkuliahan yang harus diwujudkan dalam satu semester perkuliahan. Adapun tujuan perkuliahan yang dimaksud adalah :

Setelah selesai pembelajaran diharapkan mahasiswa mampu :

- a. Mahasiswa mengenali Sistem Operasi
- b. Mahasiswa mengenali sejarah dan perkembangan teknologi Sistem Operasi
- c. Mahasiswa memahami cara kerja Sistem Operasi
- d. Mahasiswa memahami pemanfaatan dan tantangan dalam memanfaatkan Sistem Operasi

Untuk mencapai tujuan tersebut, mata kuliah Sistem Operasi menggunakan berbagai buku referensi tentang Sistem Operasi dan Teknologinya. Ada beberapa buku yang direkomendasikan untuk dipelajari, yakni :

1. Andrew S. Tanenbaum, Modern Operating Systems, Prentice Hall, 2007.

2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts With Java, Wiley.
3. Andrew S. Tanenbaum, Operating Systems Design and Implementation, 3th Edition New Jersey: Prentice Hall, Inc., 2000
4. Stalling, W. Operating Systems Internals and Design Principles 7th Edition. Prentice Hall. 2012.
5. Strawan Draft. Computer Science, Computing Curricula 2013. IEEE Computer Society. 2012

Untuk penilaian akhir, komponen nilai yang digunakan terdiri dari kehadiran, UTS, UAS dan penugasan. Dalam kuliah *online* komponen penugasan ditambah dengan kuis, sedangkan komponen kehadiran tidak diperhitungkan karena ditekankan pada aspek aktivitas di *website*. Adapun proporsi penilaiannya sebagai berikut :

- a. UTS = 40 %
- b. UAS = 30 %
- c. Kuis/Tugas 1 = 15 % (sebelum UTS)
- d. Kuis/Tugas 2 = 15 % (sesudah UTS)



Universitas
Esa Unggul

KONSEP SISTEM OPERASI

A. Kemampuan Akhir Yang Diharapkan

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

1. Memahami konsep dasar Sistem Operasi.
2. Memahami sejarah dan perkembangan teknologi Sistem Operasi
3. Mamahami tantangan yang dihadapi dalam mengembangkan Sistem Operasi

B. Uraian dan Contoh

1. Sistem Operasi

1.1. Apakah Sistem Operasi itu?

Sistem Operasi (OS) adalah program yang mengendalikan pelaksanaan program aplikasi dan bertindak sebagai antarmuka antara aplikasi dan perangkat keras komputer. Sistem operasi juga bertindak sebagai antarmuka anatar user (manusia) dengan perangkat keras (computer).

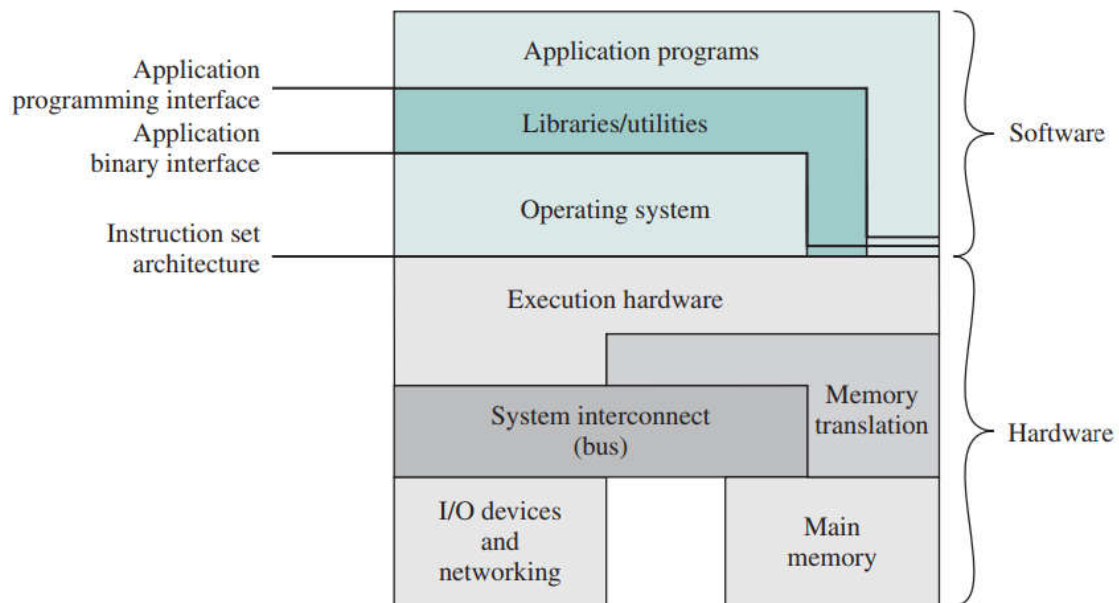
Sistem operasi mempunyai tiga tujuan utama yaitu:

- Kenyamanan: OS membuat komputer lebih nyaman digunakan.
- Efisiensi: OS memungkinkan sumber daya sistem komputer digunakan dengan cara yang efisien.
- Kemampuan untuk berkembang: OS harus dibangun sedemikian rupa untuk mengizinkan pengembangan, pengujian, dan pengenalan fungsi sistem baru yang efektif tanpa mengganggu fungsinya sebagai system operasi.

1.2. Sistem Operasi sebagai Antarmuka Pengguna / Komputer

Perangkat keras dan perangkat lunak yang digunakan dalam menyediakan aplikasi kepada pengguna dapat dilihat dalam model berlapis atau hierarkis, seperti yang digambarkan pada Gambar 1.1. Pengguna aplikasi, pengguna akhir, umumnya tidak peduli dengan detail perangkat keras computer. Dengan demikian, pengguna akhir memandang sistem komputer dalam kaitannya dengan satu set aplikasi. Suatu

aplikasi dapat diekspresikan dalam bahasa pemrograman dan dikembangkan oleh programmer.



Gambar 1.1. Struktur perangkat keras dan perangkat lunak computer.

Secara singkat, OS biasanya menyediakan layanan di area berikut:

Program Pengembangan: OS menyediakan berbagai fasilitas dan layanan, seperti editor dan debugger, untuk membantu programmer dalam membuat program. Layanan ini dalam bentuk program utilitas, bukan dari inti OS, yang disertakan dalam OS dan dirujuk sebagai alat pengembangan program aplikasi.

Program Eksekusi: Sejumlah langkah perlu dilakukan untuk mengeksekusi sebuah program. Instruksi dan data harus dimuat ke dalam memori utama, perangkat I/O dan file harus diinisialisasi, dan sumber daya lainnya harus disiapkan. OS menangani tugas penjadwalan ini untuk pengguna.

Akses ke perangkat I/O: Setiap perangkat I/O memerlukan rangkaian instruksi atau sinyal kontrolnya sendiri untuk pengoperasian. OS menyediakan antarmuka yang seragam dan menyembunyikan detail ini sehingga pemrogram dapat mengakses perangkat tersebut dengan membaca dan menulis secara sederhana.

Akses terkontrol ke file: Untuk akses file, OS harus mencerminkan pemahaman mendetail tidak hanya tentang sifat perangkat I/O (disk drive, tape drive) tetapi juga struktur data yang terdapat pada file pada media penyimpanan. Dalam kasus sistem dengan banyak pengguna, OS dapat memberikan mekanisme mengontrol akses ke file.

Akses sistem: Untuk sistem bersama atau publik, OS mengontrol akses ke sistem secara keseluruhan dan sumber daya sistem tertentu. Fungsi akses harus memberikan perlindungan sumber daya dan data dari pengguna yang tidak sah dan harus menyelesaikan konflik yang terjadi dalam penggunaan sumber daya.

Deteksi dan respons kesalahan: Berbagai kesalahan dapat terjadi saat menggunakan computer sistem saat berjalan. Dalam setiap kasus, OS harus memberikan respons dengan menghilangkan kesalahan dengan sedikit berdampak pada aplikasi yang sedang berjalan. Responsnya dapat berupa mengakhiri program yang menyebabkan kesalahan, mencoba kembali beroperasi, hingga hanya melaporkan kesalahan pada aplikasi.

Akuntansi: OS yang baik akan mengumpulkan statistik penggunaan untuk berbagai sumber dan pantau parameter kinerja seperti waktu respons. Pada sistem multi-pengguna, informasi dapat digunakan untuk tujuan perhitungan penggunaan.

Gambar 1.1 juga menunjukkan tiga antarmuka utama dalam sistem computer:

Arsitektur set instruksi (ISA): ISA mendefinisikan kumpulan mesin instruksi bahasa yang dapat diikuti komputer. Antarmuka ini adalah batas antara perangkat keras dan perangkat lunak.

Application binary interface (ABI): ABI mendefinisikan standar biner untuk portabilitas di seluruh program. ABI mendefinisikan antarmuka panggilan sistem ke sistem operasi dan sumber daya dan layanan perangkat keras yang tersedia di sebuah sistem melalui pengguna ISA.

Antarmuka pemrograman aplikasi (API): API memberikan akses program ke sumber daya dan layanan perangkat keras yang tersedia dalam sistem melalui pengguna ISA dilengkapi dengan panggilan pustaka bahasa tingkat tinggi (HLL). Panggilan sistem umumnya dilakukan melalui *library*.

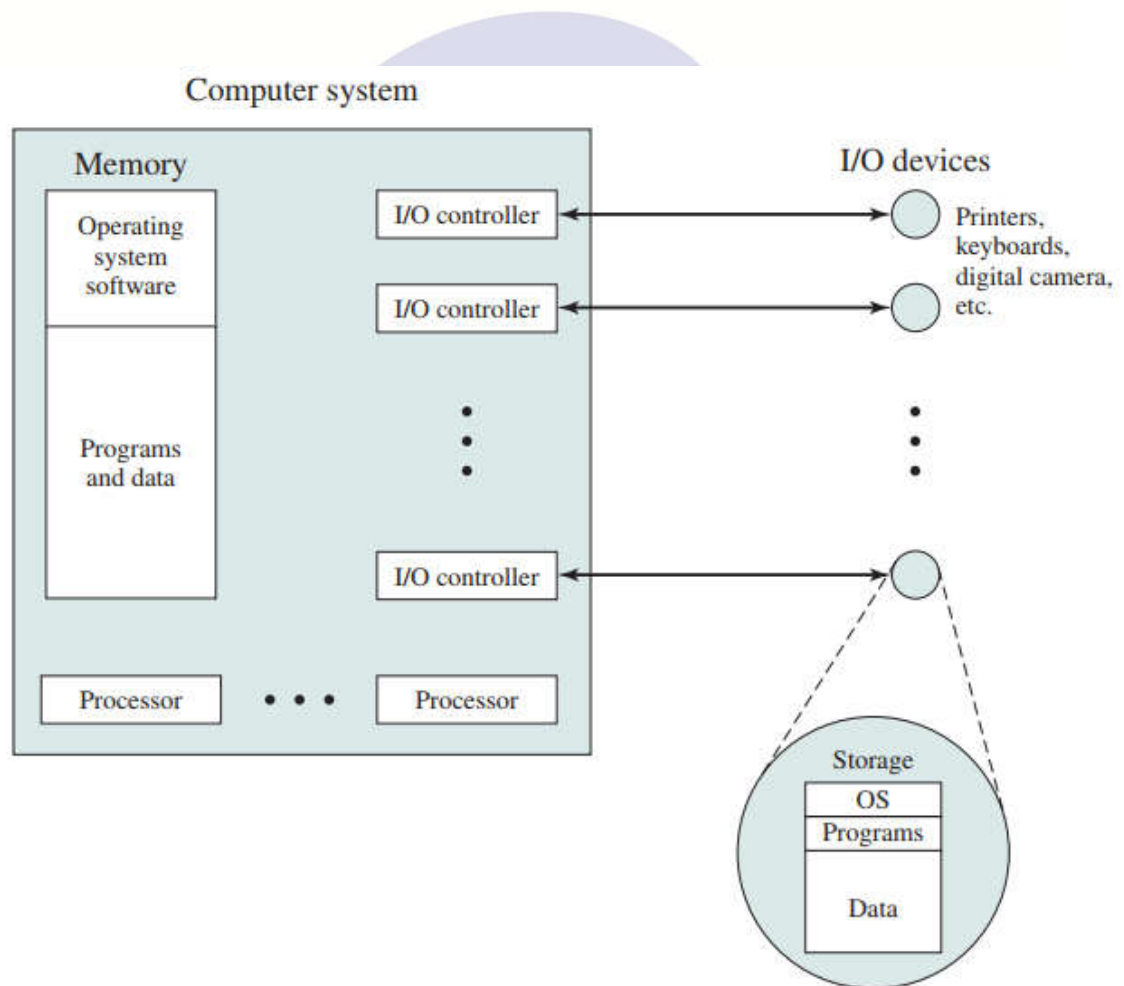
1.3. Sistem Operasi sebagai Manajer Sumber Daya

Komputer adalah sekumpulan sumber daya untuk operasional, penyimpanan, dan pemrosesan data dan untuk mengontrol fungsi-fungsi ini. OS bertanggung jawab untuk mengelola sumber daya. Dengan mengelola sumber daya, OS mengendalikan fungsi dasar computer dalam dua hal:

- OS berfungsi dengan cara yang sama seperti perangkat lunak komputer biasa, yaitu sebuah program atau rangkaian program yang dijalankan oleh prosesor.

- OS sering melepaskan kontrol dan bergantung pada prosesor sebagai pengendali.

Seperti program komputer lainnya, OS menyediakan instruksi untuk prosesor. Perbedaan utamanya terletak pada tujuan program. OS mengarahkan prosesor dalam penggunaan sumber daya sistem lain dan dalam waktu pelaksanaannya dengan program lain. Agar prosesor dapat melakukan hal ini, prosesor harus berhenti menjalankan program OS dan menjalankan program lain. Dengan demikian, OS melepaskan mengontrol prosesor untuk melakukan beberapa pekerjaan "berguna", kemudian melanjutkan kontrol lama dan cukup untuk mempersiapkan prosesor untuk melakukan pekerjaan selanjutnya.



Gambar 1.2. Sistem Operasi sebagai pengelola sumber daya

2. Evolusi Sistem Operasi

2.1. Faktor yang mempengaruhi Evolusi Sistem Operasi

OS utama akan berkembang seiring waktu karena sejumlah alasan:

- **Peningkatan kemampuan perangkat keras ditambah jenis perangkat keras baru:** Misalnya, processor tanpa teknologi paging -> ada teknologi paging, monitor terminal teks menjadi grafis (teks command -> graphical user interface).
- **Layanan baru:** Sebagai tanggapan atas permintaan pengguna atau sebagai tanggapan terhadap kebutuhan manajer sistem, OS memperluas untuk menawarkan layanan baru. Misalnya, jika ditemukan kesulitan dalam mempertahankan kinerja yang baik untuk pengguna dengan alat yang ada, alat baru pengukuran dan kontrol dapat ditambahkan ke OS.
- **Perbaikan:** OS apa pun memiliki kesalahan. Ini ditemukan seiring berjalannya waktu dan perbaikan dilakukan. Tetapi perbaikan tersebut dapat juga menyebabkan kesalahan baru.

Kebutuhan untuk mengubah OS secara teratur menempatkan persyaratan tertentu pada desainnya. Mengharuskan system menjadi modular dalam konstruksi, dengan antarmuka yang terdefinisi dengan jelas antara modul, dan harus didokumentasikan dengan baik.

2.2. Evolusi Cara kerja Sistem Operasi

Untuk memahami evolusi system operasi, perkembangan dapat dibagi berdasarkan perubahan fitur utama dari OS kontemporer yang digunakan dalam mempertimbangkan cara pengoperasian sistem selama bertahun-tahun sbb:

- **Pengolahan Serial**

Dengan komputer paling awal, dari akhir 1940-an hingga pertengahan 1950-an, programmer berinteraksi langsung dengan perangkat keras komputer; tidak ada OS. Komputer ini dijalankan dari konsol yang terdiri dari lampu layar, sakelar sakelar, beberapa bentuk perangkat input, dan printer. Program dalam kode mesin dimuat melalui input perangkat (misalnya, pembaca kartu). Jika kesalahan menghentikan program, kondisi kesalahan itu ditunjukkan oleh lampu. Jika program melanjutkan ke penyelesaian normal, output muncul di printer. Sistem awal ini menghadirkan dua masalah utama:

- **Penjadwalan:** bekerja sesuai dengan urutan waktu, salah satu masalah yang lain menunggu.
- **Waktu penyetelan:** membutuhkan waktu dalam pemasangan dan penyetelan program, jika terjadi kesalahan harus diulang dari awal kembali mengatur kembali urutan pengaturan.

Mode operasi ini dapat disebut pemrosesan serial, yang mencerminkan fakta bahwa pengguna memiliki akses ke komputer secara serial.

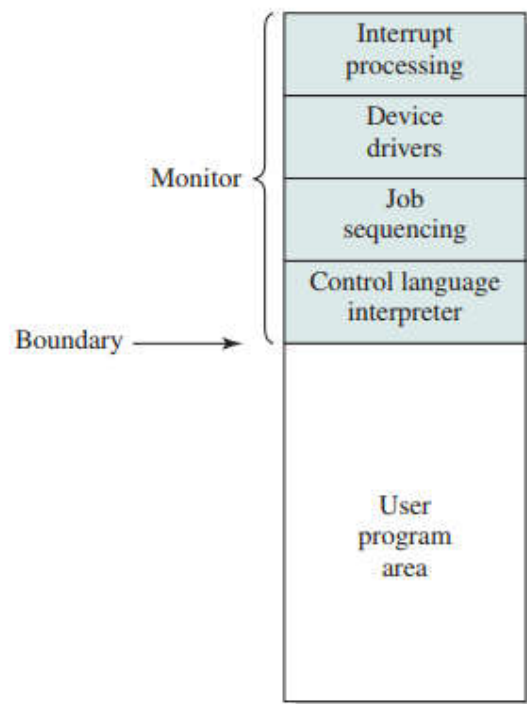
Sistem Batch Sederhana

OS batch pertama dikembangkan pada pertengahan 1950-an oleh General Motors untuk digunakan pada IBM 701. Konsep itu kemudian disempurnakan dan diimplementasikan pada IBM 704 oleh sejumlah pelanggan IBM. Oleh awal 1960-an, sejumlah vendor telah mengembangkan sistem operasi batch untuk mereka sistem komputer. IBSYS, OS IBM untuk komputer 7090/7094, khususnya terkenal karena pengaruhnya yang luas pada sistem lain. Ide sentral di balik skema pemrosesan batch sederhana adalah penggunaan sebuah perangkat lunak yang dikenal sebagai monitor. Dengan jenis OS ini, pengguna tidak lagi memilikinya akses langsung ke prosesor. Sebaliknya, pengguna mengirimkan pekerjaan di kartu atau pita ke sebuah operator komputer, yang mengumpulkan pekerjaan bersama-sama secara berurutan dan menempatkan keseluruhannya batch pada perangkat input, untuk digunakan oleh monitor. Setiap program dibangun untuk kembali ke monitor setelah selesai diproses, di mana monitor secara otomatis mulai memuat program berikutnya.

Cara kerja system batch dapat dilihat dalam sudut pandang berikut ini:

- Sudut pandang monitor:
Monitor mengontrol urutan kejadian. Untuk itu, sebagian besar monitor harus selalu berada di memori utama dan tersedia untuk eksekusi (Gambar 2.1). Porsi itu disebut sebagai *resident monitor*. Bagian monitor lainnya terdiri dari utilitas dan fungsi umum yang dibutuhkan dimuat sebagai subrutin ke program pengguna di awal pekerjaan.
- Sudut pandang prosesor:
Pada titik tertentu, prosesor menjalankan instruksi dari bagian memori utama yang berisi monitor. Ini instruksi menyebabkan pekerjaan berikutnya dibaca ke bagian lain dari utama Penyimpanan. Setelah pekerjaan dibaca, prosesor akan menemukan cabang instruksi di monitor yang menginstruksikan prosesor untuk

melanjutkan eksekusi di awal program pengguna. Prosesor kemudian akan menjalankan instruksi dalam program pengguna hingga menemui kondisi akhir atau kesalahan.



Gambar 2.1. Layout memori dari *resident monitor*

Monitor, atau OS batch, hanyalah sebuah program komputer. Ini bergantung pada kemampuan prosesor untuk mengambil instruksi dari berbagai bagian ke memori utama secara bergantian mengambil alih dan melepaskan kendali. Fitur perangkat keras tertentu lainnya juga digunakan:

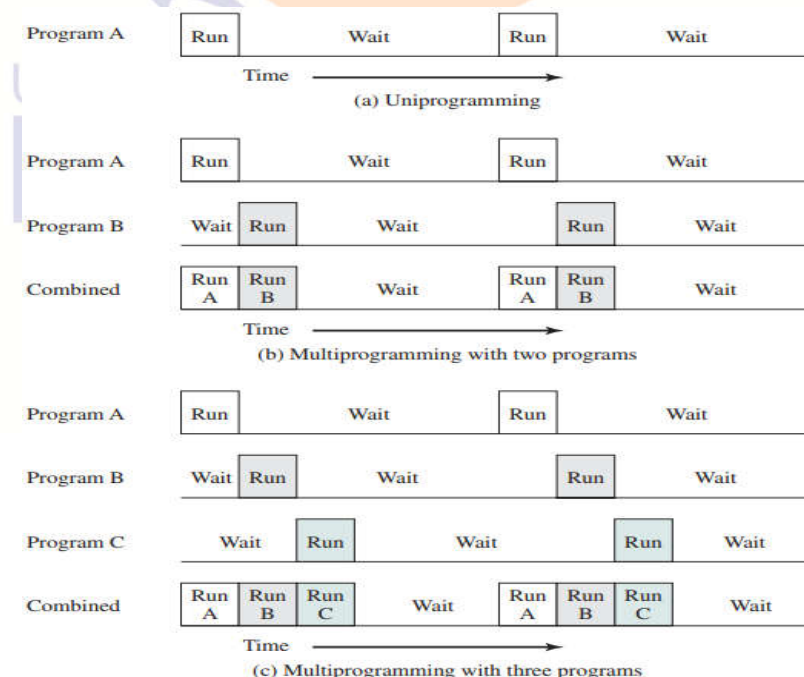
- **Proteksi memori:** Saat program dijalankan, tidak boleh mengubah area memori yang berisi monitor.
- **Timer:** Timer digunakan untuk mencegah satu pekerjaan memonopoli sistem. Pengatur waktu disetel di awal setiap pekerjaan. Jika pengatur waktu berakhir, program pengguna dihentikan, dan kontrol kembali ke monitor.
- **Instruksi dengan hak istimewa:** Instruksi tingkat mesin tertentu ditetapkan dengan hak istimewa dan hanya dapat dijalankan oleh monitor. Di antara instruksi istimewa adalah instruksi I/O, sehingga monitor tetap mengontrol semua perangkat I/O. Ini mencegah, misalnya, program pengguna dari tidak sengaja membaca instruksi kontrol pekerjaan dari pekerjaan berikutnya. Jika program pengguna ingin melakukan I/O, itu harus meminta monitor melakukan pengoperasiannya.

- **Interupsi:** Model komputer awal tidak memiliki kemampuan ini. Fitur ini memberi OS lebih banyak fleksibilitas dalam melepaskan kendali dan mendapatkan kembali kendali dari program pengguna.

Dengan OS batch, waktu prosesor digunakan secara bergantian antara eksekusi program pengguna dan eksekusi monitor, hal ini meningkatkan pemanfaatan komputer.

Sistem Batch Multiprogram

Walaupun dengan urutan pekerjaan otomatis yang disediakan oleh OS batch sederhana, prosesor sering kali menganggur. Masalahnya adalah perangkat I/O lambat dibandingkan dengan prosesor. Gambar 2.2 (a) mengilustrasikan hal ini, situasi di mana ada satu program, disebut sebagai uniprogramming. Prosesor menghabiskan sejumlah waktu untuk mengeksekusi, hingga mencapai instruksi I/O. Kemudian harus menunggu sampai instruksi I/O selesai sebelum melanjutkan (menjadi tidak efisien). Jika ada cukup memori untuk menampung OS (monitor residen) dan dua program pengguna. Ketika satu pekerjaan perlu menunggu I/O, prosesor dapat beralih ke pekerjaan lain, yang kemungkinan besar tidak menunggu I/O (Gambar 2.2 b). Lebih jauh, kita mungkin memperluas memori untuk menampung tiga, empat, atau lebih program dan beralih di antara semuanya (Gambar 2.5c). Pendekatan tersebut dikenal sebagai multiprogramming, atau multitasking.



Gambar 2.2. Multiprogramming Batch

Seperti sistem batch sederhana, sistem batch multiprogramming harus bergantung pada fitur perangkat keras komputer tertentu. Fitur tambahan yang paling menonjol dan berguna untuk multiprogramming adalah hardware yang mendukung interupsi dan I/O DMA (akses memori langsung). Dengan I/O atau DMA yang digerakkan oleh interupsi, prosesor dapat mengeluarkan perintah I/O untuk satu pekerjaan dan melanjutkan dengan eksekusi pekerjaan lain, pekerjaan sementara I/O dilakukan oleh pengontrol perangkat. Saat operasi I/O dilakukan selesai, prosesor diinterupsi dan kontrol diteruskan ke penanganan interupsi program di OS. OS kemudian akan meneruskan kontrol ke pekerjaan lain.

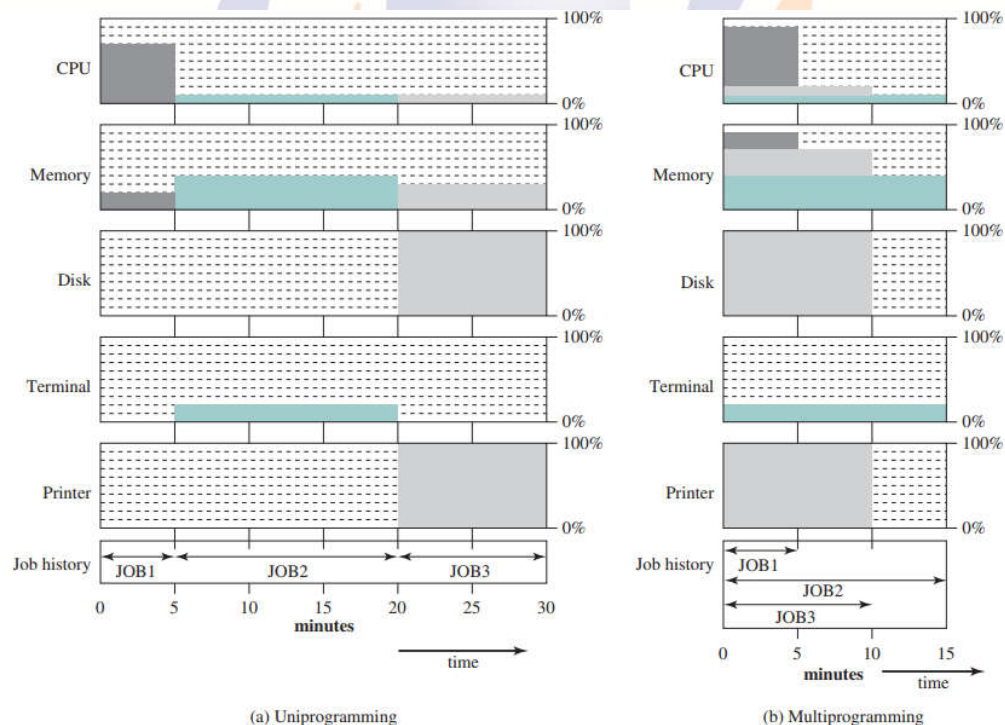
Sistem operasi multiprogramming cukup canggih dibandingkan program tunggal, atau uniprogramming, sistem. Untuk memiliki beberapa pekerjaan yang siap dijalankan, mereka harus disimpan dalam memori utama, membutuhkan beberapa bentuk manajemen memori. Selain itu, jika beberapa pekerjaan siap dijalankan, prosesor harus memutuskan yang mana untuk menjalankannya, keputusan ini membutuhkan algoritma untuk penjadwalan.

Sistem Pembagian Waktu (*Time Sharing System*)

Dengan penggunaan multiprogramming, pemrosesan batch bisa sangat efisien. Namun, untuk banyak pekerjaan, diinginkan untuk menyediakan mode di mana pengguna berinteraksi langsung dengan komputer. Untuk beberapa pekerjaan, seperti pemrosesan transaksi, sebuah mode interaktif sangat penting. Saat ini, kebutuhan untuk fasilitas komputasi interaktif dapat, dan seringkali dipenuhi dengan penggunaan komputer pribadi atau workstation khusus. Opsi itu adalah tidak tersedia pada tahun 1960-an, ketika kebanyakan komputer berukuran besar dan mahal. Sebaliknya, waktu berbagi dikembangkan. Sama seperti multiprogramming yang memungkinkan prosesor menangani beberapa pekerjaan batch sekaligus, multiprogramming juga dapat digunakan untuk menangani beberapa pekerjaan interaktif. Di kasus terakhir ini, teknik ini disebut sebagai berbagi waktu (*time-sharing*). karena waktu prosesor dibagikan di antara banyak pengguna. Dalam sistem pembagian waktu, banyak pengguna secara bersamaan mengakses sistem melalui terminal, dengan OS interleaving eksekusi masing-masing program pengguna dalam siklus singkat atau kuantum komputasi. Jadi, jika ada n pengguna aktif meminta

layanan pada satu waktu, setiap pengguna hanya akan melihat rata-rata $1/n$ kapasitas komputer efektif.

Salah satu sistem operasi time-sharing yang pertama kali dikembangkan adalah Sistem Berbagi Waktu yang Kompatibel (CTSS) [CORB62], dikembangkan di MIT oleh grup yang dikenal sebagai Project MAC (Machine-Aided Cognition, atau Multiple-Access Komputer). Sistem ini pertama kali dikembangkan untuk IBM 709 pada tahun 1961 dan kemudian ditransfer ke IBM 7094. Dahulu memori sangat terbatas pada setiap interval ada interupsi, OS mendapatkan kembali kendali dan dapat menetapkan prosesor ke pengguna lain. Teknik ini dikenal sebagai mengiris waktu (*time-slicing*). Jadi, pada interval waktu yang teratur, pengguna saat ini akan menjadi preempted dan pengguna lain dimuat. Untuk mempertahankan status program pengguna lama untuk kemudian dilanjutkan, program dan data pengguna lama ditulis ke disk sebelum file program dan data pengguna baru dibaca. Selanjutnya, kode program pengguna lama dan data dikembalikan ke memori utama ketika program itu diberi giliran berikutnya.



Gambar 2.3. Histogram Utilisasi

3. Tren Pengembangan Sistem Operasi Modern

Sistem operasi modern menanggapi perkembangan baru dalam perangkat keras, aplikasi baru, dan ancaman keamanan baru. Di antara driver perangkat keras utama adalah sistem multiprosesor, sangat pesat peningkatan dalam kecepatan prosesor, jaringan *overlay* berkecepatan tinggi, dan peningkatan ukuran dan berbagai perangkat penyimpanan memori. Di arena aplikasi, aplikasi multimedia, akses Internet dan Web, dan komputasi klien / server telah mempengaruhi rancangan OS. Berkenaan dengan keamanan, akses Internet ke komputer telah meningkatkan potensi ancaman dan serangan yang semakin canggih, seperti virus, worm, dan teknik peretasan, berdampak besar pada desain OS.

3.1. Kategori elemen desain OS

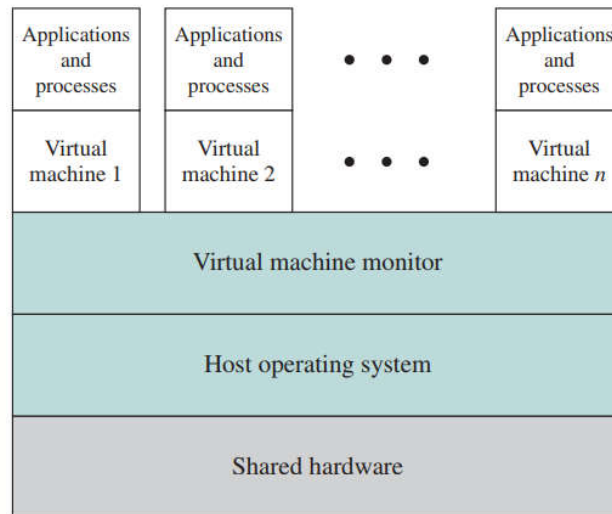
Tingkat perubahan tuntutan pada sistem operasi membutuhkan tidak hanya modifikasi dan peningkatan pada arsitektur yang ada tetapi cara baru untuk mengatur OS. Pendekatan dan elemen desain OS modern dikategorikan berikut:

- Arsitektur mikrokernell
- Multithreading
- Multiprosesing simetris
- Sistem operasi terdistribusi
- Desain berorientasi objek

Detail elemen desain OS akan dibahas pada materi berikutnya

3.2. Virtual Machine dan Virtualisasi

Secara tradisional, aplikasi berjalan langsung di OS di PC atau server. Setiap PC atau server hanya akan menjalankan satu OS pada satu waktu. Jadi, vendor harus menulis ulang bagiannya aplikasinya untuk setiap OS / platform yang akan mereka jalankan. Strategi yang efektif untuk mengatasi masalah ini dikenal sebagai virtualisasi. Teknologi virtualisasi memungkinkan satu PC atau server untuk secara bersamaan menjalankan beberapa sistem operasi atau beberapa sesi dari satu OS. Mesin dengan virtualisasi dapat menampung banyak aplikasi, termasuk yang berjalan pada sistem operasi yang berbeda, pada satu platform aplikasi. Intinya, sistem operasi host dapat mendukung sejumlah virtual mesin (VM), yang masing-masing memiliki karakteristik OS tertentu dan beberapa memiliki karakteristik platform perangkat keras tertentu.



Gambar 3.1. Konsep Memori Mesin Virtual

3.3. Pengembangan OS Multiprocessor/MultiCore

Beberapa factor pertimbangan dalam pengembangan OS Modern dengan perkembangan teknologi prosessor saat ini adalah:

Symmetric Multiprocessor OS (SMP)

Dalam sistem SMP, kernel dapat dijalankan pada prosesor apa pun, dan biasanya masing-masing prosesor melakukan penjadwalan sendiri dari kumpulan proses atau utas yang tersedia. Kernel dapat dibangun sebagai beberapa proses atau beberapa *thread*, memungkinkan bagian dari kernel untuk dieksekusi secara paralel. Sistem operasi SMP mengelola prosesor dan sumber daya komputer lainnya sehingga pengguna dapat melihat sistem dengan cara yang sama seperti multiprogramming sistem uniprocessor. Seorang pengguna dapat membangun aplikasi yang menggunakan banyak proses atau beberapa *thread* dalam proses tanpa memperhatikan apakah prosesor tunggal atau beberapa prosesor akan tersedia. Jadi, OS multiprosesor harus menyediakan semua fungsionalitas sistem multiprogramming ditambah fitur tambahan untuk mengakomodasi banyak prosesor.

Masalah utama desain SMP meliputi:

- Proses atau thread berjalan secara bersamaan.
- Penjadwalan.
- Sinkronisasi.
- Manajemen memori.
- Keandalan dan toleransi kesalahan.

MultiCore OS

Tantangan desain untuk sistem multicore banyak inti adalah untuk efisien memanfaatkan kekuatan pemrosesan multicore dan dengan cerdas mengelola substansial sumber daya on-chip secara efisien. Perhatian utama adalah bagaimana mencocokkan paralelisme inheren dari sistem banyak inti dengan persyaratan kinerja aplikasi.

Tingkatan paralelisme dalam multicore system kontemporer :

- paralelisme perangkat keras dalam setiap prosesor inti, yang dikenal sebagai paralelisme tingkat instruksi,
- multiprogramming dan eksekusi multithreaded dalam setiap prosesor.
- satu aplikasi untuk dieksekusi dalam proses atau utas bersamaan di beberapa inti.

Strategi pendekatan paralelisme dalam OS:

- **Paralelisme Dalam Aplikasi.** Kebanyakan aplikasi pada prinsipnya bisa dibagi menjadi beberapa tugas yang dapat dijalankan secara paralel, dengan tugas-tugas ini kemudian diimplementasikan sebagai beberapa proses, mungkin masing-masing dengan beberapa thread. OS setidaknya dapat mendukung proses desain ini dengan efisien mengalokasikan sumber daya di antara tugas-tugas paralel seperti yang ditentukan oleh pengembang.
- **Pendekatan Virtual Machine.**
Dalam pendekatan ini, OS bertindak lebih seperti hypervisor. Programnya sendiri yang menentukan beberapa tugas dengan manajemen sumber daya. OS menetapkan aplikasi ke sebuah prosesor dan beberapa memori, dan program itu sendiri, menggunakan metadata yang dihasilkan oleh kompilator, yang paling tahu cara menggunakan sumber daya ini.

4. Jenis-jenis OS

4.1. Mainframe OS

Komputer dengan kemampuan dan sumber daya yang jauh lebih dari PC membutuhkan OS tersendiri Memproses banyak job sekaligus dengan kapasitas I/O sangat besar. Tiga jenis mode pemrosesan: batch, transaction processing dan timesharing

4.2. Server OS

Bisa berupa PC/mainframe dengan berbasis konsep client-server. Terhubung via network, server berfungsi menyediakan layanan spesifik mis: mail server, web server, print server, dsb.

4.3. Multiprocessors OS

Saat ini sudah merambah ke microcomputer dengan multicore chip. Mencakup processor scheduling dan communication. Masalah: bagaimana mengoptimalkan parallel processing.

4.4. PC OS

Yang paling umum digunakan, seringkali tidak sadar bahwa ada OS lainnya selain PC OS. Saat ini sudah support multiprogramming.

4.5. Handheld computer OS

Komputer saku: PDA (Personal Digital Assistant) dan smartphone, OS harus menangani multimedia dan telepon, juga 3rd party software. Contoh: Symbian, Palm OS, Android, IOS dll.

4.6. Embedded OS

Embedded dalam perangkat keras yang tidak tampak seperti computer, Mis: microwave oven, mesin cuci, dvd player. Mirip dengan handheld computer OS, hanya lebih tertutup tidak perlu khawatir dengan software yang tidak terpercaya, Contoh: QNX dan VxWorks.

4.7. Sensor Node OS

Sensor node adalah komputer kecil dengan tenaga baterai yang biasa ditempatkan di tempat terbuka dilengkapi radio dan sensor. Contoh: TinyOS

4.8. Real Time OS

Waktu menjadi parameter kunci, respon komputer diharapkan terjadi dalam jangka waktu tertentu. Terbagi menjadi

- Hard real-time system: jika kendala waktu dilanggar mengakibatkan bahaya/kerusakan, contoh: sistem kontrol, pertahanan, transportasi
- Soft real-time system: jika kendala waktu dilanggar tidak diinginkan namun tidak berbahaya, contoh: audio/video/telepon digital

4.9. Smartcard OS

Sistem Operasi terkecil dan hanya menangani fungsi yang terbatas

C. Daftar Pustaka

1. Operating System, Internals and design Principles, William Stallings 7th Ed. 2012
2. Modern Operating System 3th Edition Andrew S Tanembaun 2009

