

SISTEM OPERASI

Oleh :
Ir. I Gede Made Karma, MT

Pendahuluan (1)

- Sistem operasi merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer.
- Sebelum ada sistem operasi, komputer dioperasikan dengan menggunakan sinyal analog dan sinyal digital.
- Seiring dengan berkembangnya pengetahuan dan teknologi, pada saat ini terdapat berbagai sistem operasi dengan keunggulan masing-masing.
- Untuk lebih memahami sistem operasi maka sebaiknya perlu diketahui terlebih dahulu beberapa konsep dasar mengenai sistem operasi itu sendiri.

Pendahuluan (2)

- Pengertian sistem operasi secara umum ialah pengelola seluruh sumber-daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*system calls*) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumber-daya sistem komputer.

Fungsi Dasar (1)

- Sistem komputer pada dasarnya terdiri dari empat komponen utama, yaitu :
 1. perangkat-keras
 2. program aplikasi
 3. sistem-operasi, dan
 4. para pengguna.
- Sistem operasi berfungsi untuk mengatur dan mengawasi penggunaan perangkat keras oleh berbagai program aplikasi serta para pengguna.

Fungsi Dasar (2)

- Sistem operasi berfungsi ibarat pemerintah dalam suatu negara, dalam arti membuat kondisi komputer agar dapat menjalankan program secara benar.
- Untuk menghindari konflik yang terjadi pada saat pengguna menggunakan sumber-daya yang sama, sistem operasi mengatur pengguna mana yang dapat mengakses suatu sumber-daya.
- Sistem operasi juga sering disebut *resource allocator*.
- Satu lagi fungsi penting sistem operasi ialah sebagai program pengendali yang bertujuan untuk menghindari kekeliruan (*error*) dan penggunaan komputer yang tidak perlu.

Tujuan Mempelajari Sistem Operasi

1. Agar dapat merancang sendiri serta dapat memodifikasi sistem yang telah ada sesuai dengan kebutuhan kita,
2. Agar dapat memilih alternatif sistem operasi,
3. Memaksimalkan penggunaan sistem operasi, dan
4. Agar konsep dan teknik sistem operasi dapat diterapkan pada aplikasi-aplikasi lain.

Sasaran Sistem Operasi

1. *Kenyamanan* -- membuat penggunaan komputer menjadi lebih nyaman,
2. *Efisien* -- penggunaan sumber-daya sistem komputer secara efisien, serta
3. Mampu *berevolusi* -- sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru.

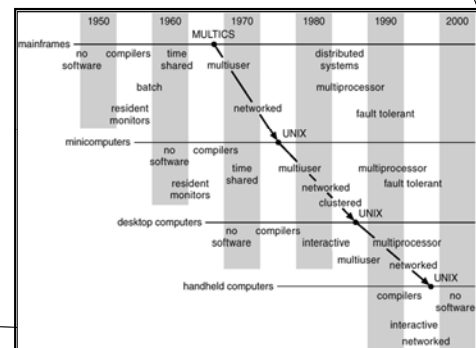
Sejarah Sistem Operasi (1)

- Generasi Pertama (1945-1955)
Generasi pertama merupakan awal perkembangan sistem komputasi elektronik sebagai pengganti sistem komputasi mekanik, hal itu disebabkan kecepatan manusia untuk menghitung terbatas dan manusia sangat mudah untuk membuat kecerobohan, kekeliruan bahkan kesalahan. Pada generasi ini belum ada sistem operasi, maka sistem komputer diberi instruksi yang harus dikerjakan secara langsung.
- Generasi Kedua (1955-1965)
Generasi kedua memperkenalkan *Batch Processing System*, yaitu Job yang dikerjakan dalam satu rangkaian, lalu dieksekusi secara berurutan. Pada generasi ini sistem komputer belum dilengkapi sistem operasi, tetapi beberapa fungsi sistem operasi telah ada, contohnya fungsi sistem operasi ialah FMS dan IBSYS.

Sejarah Sistem Operasi (2)

- Generasi Ketiga (1965-1980)
Pada generasi ini perkembangan sistem operasi dikembangkan untuk melayani banyak pemakai sekaligus, dimana para pemakai interaktif berkomunikasi lewat terminal secara on-line ke komputer, maka sistem operasi menjadi *multi-user* (di gunakan banyak pengguna sekali gus) dan *multi-programming* (melayani banyak program sekali gus).
- Generasi Keempat (Pasca 1980an)
Dewasa ini, sistem operasi dipergunakan untuk jaringan komputer dimana pemakai menyadari keberadaan komputer-komputer yang saling terhubung satu sama lainnya. Pada masa ini para pengguna juga telah dinyamankan dengan *Graphical User Interface* yaitu antar-muka komputer yang berbasis grafis yang sangat nyaman, pada masa ini juga dimulailah era komputasi tersebar dimana komputasi-komputasi tidak lagi berpusat di satu titik, tetapi dipecah dibanyak komputer sehingga tercapai kinerja yang lebih baik.

Sejarah Sistem Operasi (3)



Layanan Sistem Operasi (1)

- Sebuah sistem operasi yang baik harus memiliki layanan :
 1. pembuatan program,
 2. eksekusi program,
 3. pengaksesan I/O Device,
 4. pengaksesan terkendali terhadap berkas pengaksesan sistem,
 5. deteksi dan pemberian tanggapan pada kesalahan, serta
 6. akunting.

Layanan Sistem Operasi (2)

1. Pembuatan program yaitu sistem operasi menyediakan fasilitas dan layanan untuk membantu para pemrogram untuk menulis program;
2. Eksekusi Program yang berarti Instruksi-instruksi dan data-data harus dimuat ke memori utama, perangkat-parangkat masukan/ keluaran dan berkas harus di-inisialisasi, serta sumber-daya yang ada harus disiapkan, semua itu harus di tangani oleh sistem operasi;
3. Pengaksesan I/O Device, artinya Sistem Operasi harus mengambil alih sejumlah instruksi yang rumit dan sinyal kendali menjengkelkan agar pemrogram dapat berfikir sederhana dan perangkat pun dapat beroperasi;
4. Pengaksesan terkendali terhadap berkas yang artinya disediakan mekanisme proteksi terhadap berkas untuk mengendalikan pengaksesan terhadap berkas;

Layanan Sistem Operasi (3)

- Pengaksesan sistem artinya pada pengaksesan digunakan bersama (*shared system*);
- Fungsi pengaksesan harus menyediakan proteksi terhadap sejumlah sumber-daya dan data dari pemakai tak terdistorsi serta menyelesaikan konflik-konflik dalam perebutan sumber-daya;
- 5. Deteksi dan Pemberian tanggapan pada kesalahan, yaitu jika muncul permasalahan muncul pada sistem komputer maka sistem operasi harus memberikan tanggapan yang menjelaskan kesalahan yang terjadi serta dampaknya terhadap aplikasi yang sedang berjalan; dan
- 6. Akunting yang artinya Sistem Operasi yang bagus mengumpulkan data statistik penggunaan beragam sumber-daya dan memonitor parameter kinerja.

Komponen Sistem Operasi

- Manajemen Proses.
- Manajemen Memori Utama.
- Manajemen *Secondary-Storage*.
- Manajemen Sistem I/O.
- Manajemen Berkas.
- Sistem Proteksi.
- Jaringan.
- *Command-Interpreter system*.

Managemen Proses

- Proses adalah keadaan ketika sebuah program sedang di eksekusi. Sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. sumber daya tersebut dapat berupa *CPU time*, memori, berkas-berkas, dan perangkat-perangkat I/O.
- Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:
 1. Pembuatan dan penghapusan proses pengguna dan sistem proses.
 2. Menunda atau melanjutkan proses.
 3. Menyediakan mekanisme untuk proses sinkronisasi.
 4. Menyediakan mekanisme untuk proses komunikasi.
 5. Menyediakan mekanisme untuk penanganan *deadlock*.

Managemen Memori Utama

- Memori utama atau lebih dikenal sebagai memori adalah sebuah *array* yang besar dari *word* atau *byte*, yang ukurannya mencapai ratusan, ribuan, atau bahkan jutaan. Setiap *word* atau *byte* mempunyai alamat tersendiri. Memori Utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama termasuk tempat penyimpanan data yang sementara (*volatile*), artinya data dapat hilang begitu sistem dimatikan.
- Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti:
 1. Menjaga *track* dari memori yang sedang digunakan dan siapa yang menggunakannya.
 2. Memilih program yang akan di-*load* ke memori.
 3. Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.

Managemen *Secondary-Storage*

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

Managemen Sistem I/O

- Sering disebut *device manager*. Menyediakan "*device driver*" yang umum sehingga operasi I/O dapat seragam (membuka, membaca, menulis, menutup). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada *hard-disk*, CD-ROM dan *floppy disk*.
- Komponen Sistem Operasi untuk sistem I/O:
 1. *Buffer*: menampung sementara data dari/ ke perangkat I/O.
 2. *Spooling*: melakukan penjadualan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
 3. Menyediakan *driver* untuk dapat melakukan operasi "rinci" untuk perangkat keras I/O tertentu.

Managemen Berkas

- ◎ Berkas adalah kumpulan informasi yang berhubungan sesuai dengan tujuan pembuat berkas tersebut. Berkas dapat mempunyai struktur yang bersifat hirarkis (direktori, volume, dll.). Sistem operasi bertanggung-jawab:
 1. Pembuatan dan penghapusan berkas.
 2. Pembuatan dan penghapusan direktori.
 3. Mendukung manipulasi berkas dan direktori.
 4. Memetakan berkas ke *secondary storage*.
 5. Mem-*backup* berkas ke media penyimpanan yang permanen (*non-volatile*).

Sistem Proteksi

- ◎ Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program, prosesor, atau pengguna ke sistem sumber daya. Mekanisme proteksi harus:
 1. membedakan antara penggunaan yang sudah diberi izin dan yang belum.
 2. *specify the controls to be imposed*.
 3. *provide a means of enforcement*.

Jaringan

- ◎ Sistem terdistribusi adalah sekumpulan prosesor yang tidak berbagi memori atau *clock*. Tiap prosesor mempunyai memori sendiri. Prosesor-prosesor tersebut terhubung melalui jaringan komunikasi. Sistem terdistribusi menyediakan akses pengguna ke bermacam sumber-daya sistem. Akses tersebut menyebabkan:
 1. *Computation speed-up*.
 2. *Increased data availability*.
 3. *Enhanced reliability*.

Command-Interpreter System

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

Layanan Sistem Operasi

- ◎ Eksekusi program adalah kemampuan sistem untuk "*load*" program ke memori dan menjalankan program. Operasi I/O: pengguna tidak dapat secara langsung mengakses sumber daya perangkat keras, sistem operasi harus menyediakan mekanisme untuk melakukan operasi I/O atas nama pengguna. Sistem manipulasi berkas adalah kemampuan program untuk operasi pada berkas (membaca, menulis, membuat, and menghapus berkas). Komunikasi adalah pertukaran data/informasi antar dua atau lebih proses yang berada pada satu komputer (atau lebih). Deteksi *error* adalah menjaga kestabilan sistem dengan mendeteksi "*error*", perangkat keras mau pun operasi.
- ◎ Efisiensi penggunaan sistem:
 1. *Resource allocator* adalah mengalokasikan sumber-daya ke beberapa pengguna atau *job* yang jalan pada saat yang bersamaan.
 2. Proteksi menjamin akses ke sistem sumber daya dikendalikan (pengguna dikontrol aksesnya ke sistem).
 3. *Accounting* adalah merekam kegiatan pengguna, jatah pemakaian sumber daya (keadilan atau kebijaksanaan).

System Calls

- ◎ *System call* menyediakan interface antara program (program pengguna yang berjalan) dan bagian OS. *System call* menjadi jembatan antara proses dan sistem operasi. *System call* ditulis dalam bahasa *assembly* atau bahasa tingkat tinggi yang dapat mengendalikan mesin (C). Contoh: UNIX menyediakan *system call*: *read*, *write* => operasi I/O untuk berkas.
- ◎ Sering pengguna program harus memberikan data (parameter) ke OS yang akan dipanggil. Contoh pada UNIX: *read(buffer, max_size, file_id)*;
- ◎ Tiga cara memberikan parameter dari program ke sistem operasi:
 1. Melalui registers (sumber daya di CPU).
 2. Menyimpan parameter pada data struktur (table) di memori, dan alamat table tsb ditunjuk oleh *pointer* yang disimpan di register.
 3. *Push (store)* melalui "*stack*" pada memori dan OS mengambilnya melalui *pop* pada *stack* tsb.

Mesin Virtual

- Sebuah mesin virtual (*Virtual Machine*) menggunakan misalkan terdapat sistem program => control program yang mengatur pemakaian sumber daya perangkat keras. Control program = trap *System call* + akses ke perangkat keras. Control program memberikan fasilitas ke proses pengguna. Mendapatkan jatah CPU dan memori. Menyediakan *interface* "identik" dengan apa yang disediakan oleh perangkat keras => *sharing devices* untuk berbagai proses.
- Mesin Virtual (MV) (MV) => control program yang minimal MV memberikan ilusi *multitasking*; seolah-olah terdapat prosesor dan memori eksklusif digunakan MV. MV memilah fungsi *multitasking* dan implementasi *extended machine* (tergantung proses pengguna) => *flexible* dan lebih mudah untuk pengaturan. Jika setiap pengguna diberikan satu MV => bebas untuk menjalankan OS (kernel) yang diinginkan pada MV tersebut. Potensi lebih dari satu OS dalam satu komputer. Contoh: IBM VM370; menyediakan MV untuk berbagai OS: CMS (interaktif), MVS, CICS, dll. Masalah: *Sharing disk* => OS mempunyai sistem berkas yang mungkin berbeda. IBM: virtual disk (minidisk) yang dialokasikan untuk pengguna melalui MV.
- Konsep MV menyediakan proteksi yang lengkap untuk sumberdaya sistem, dikarenakan tiap MV terpisah dari MV yang lain. Namun, hal tersebut menyebabkan tidak adanya *sharing* sumberdaya secara langsung. MV merupakan alat yang tepat untuk penelitian dan pengembangan sistem operasi. Konsep MV susah untuk diimplementasi sehubungan dengan usaha yang diperlukan untuk menyediakan duplikasi dari mesin utama.

Perancangan Sistem dan Implementasi

- Target untuk pengguna: sistem operasi harus nyaman digunakan, mudah dipelajari, dapat diandalkan, aman dan cepat. Target untuk sistem: sistem operasi harus gampang dirancang, diimplementasi, dan dipelihara, sebagaimana fleksibel, *error*, dan efisien.
- Mekanisme dan Kebijakan:
 1. Mekanisme menjelaskan bagaimana melakukan sesuatu kebijakan memutuskan apa yang akan dilakukan. Pemisahan kebijakan dari mekanisme merupakan hal yang sangat penting; ini mengizinkan fleksibilitas yang tinggi bila kebijakan akan diubah nanti.
 2. Kebijakan memutuskan apa yang akan dilakukan

Perancangan Sistem ... (2)

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

System Generation (SYSGEN)

- Sistem operasi dirancang untuk dapat dijalankan di berbagai jenis mesin; sistemnya harus di konfigurasi untuk tiap komputer. Program SYSGEN mendapatkan informasi mengenai konfigurasi khusus dari sistem perangkat keras.
 1. *Bootting*: memulai komputer dengan me-load kernel.
 2. *Bootstrap program*: kode yang disimpan di code ROM yang dapat menempatkan kernel, memasukkannya ke dalam memori, dan memulai eksekusinya.