

Sistem Operasi

Bahan Kuliah IKI-20230

**Gabungan Kelompok Kerja 21–28 IKI-20230 Semester
Genap 2002/2003**

Sistem Operasi: Bahan Kuliah IKI-20230

oleh Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003

\$Revision: 1.1 \$ Edisi

Diterbitkan 05 Mei 2003

Hak Cipta © 2003 oleh Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003

Silakan menyalin, mengedarkan, dan/atau, memodifikasi dokumen ini sesuai dengan ketentuan "*GNU Free Documentation License* versi 1.1" atau versi selanjutnya dari FSF (*Free Software Foundation*); tanpa bagian "*Invariant*", tanpa teks "*Front-Cover*", dan tanpa teks "*Back-Cover*".

Catatan Revisi

Revisi 21.4 05-05-2003 Revised by: Kelompok 21

Perapihan berkas dan penambahan entity

Revisi 21.3 29-04-2003 Revised by: Kelompok 21

Perubahan dengan menyempurnakan nama file

Revisi 21.2 24-04-2003 Revised by: Kelompok 21

Merubah Kata Pengantar

Revisi 21.1 21-04-2003 Revised by: Kelompok 21

Menambahkan Daftar Pustaka dan Index

Revisi 21.0 26-03-2003 Revised by: Kelompok 21

Memulai membuat tugas kelompok kuliah Sistem Operasi

Persembahan

Buku ini dipersembahkan *dari* Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003
oleh Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003 *untuk* Gabungan
Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003

Daftar Isi

Kata Pengantar	i
1. PENDAHULUAN.....	1
1.1. Seputar sistem Operasi	1
1.1.1. Apa itu Sistem Operasi	1
1.1.2. Fungsi dasar sistem operasi	1
1.1.3. Tujuan Mempelajari Sistem Operasi	1
1.1.4. Sasaran Sistem Operasi	1
1.1.5. Sejarah Sistem Operasi	2
1.1.5.1. Generasi Pertama	2
1.1.5.2. Generasi Kedua	2
1.1.5.3. Generasi Ketiga	2
1.1.5.4. Generasi Keempat	2
1.1.6. Layanan Sistem Operasi	2
1.2. Struktur Komputer.....	3
1.2.1. Sistem Operasi Komputer	3
1.2.2. Struktur I/O.....	4
1.2.2.1. Interupsi I/O	4
1.2.2.2.	4
1.2.3. Struktur Penyimpanan	4
1.2.3.1. <i>Main Memory</i>	4
1.2.3.2. <i>Magnetic Disk</i>	5
1.2.4. <i>Storage Hierarchy</i>	5
1.2.5. Proteksi Perangkat Keras	5
1.2.5.1. Operasi Dual Mode	6
1.2.5.2. Proteksi I/O	6
1.2.5.3. Proteksi Memori.....	6
1.3. Struktur Sistem Komputer.....	7
1.3.1. Komponen-komponen Sistem	7
1.3.2. Manajemen Proses	7
1.3.3. Manajemen <i>Main-Memory</i>	8
1.3.4. Manajemen <i>Secondary-Storage</i>	8
1.3.5. Manajemen Sistem I/O	8
1.3.6. Manajemen File	8
1.3.7. Sistem Proteksi	9
1.3.8. Jaringan.....	9
1.3.9. <i>Command-Interpreter System</i>	9
1.3.10. <i>Layanan Sistem Operasi</i>	9
1.3.11. <i>System Calls</i>	10
1.3.12. <i>Virtual Machines</i>	10
1.3.13. Perancangan Sistem dan Implementasi	11
1.3.14. <i>System Generation(SYSGEN)</i>	11
1.4. Rangkuman	11
1.5. Pertanyaan	13
1.6. Referensi	14

2. PROSES 1	16
2.1. Proses 11	16
2.2. Proses 12	16
2.3. Proses 13	16
2.4. Proses 14	16
2.5. Proses 15	16
2.6. Proses 16	16
2.7. Proses 17	16
2.8. Proses 18	17
3. PROSES 2	18
3.1. Proses 21	18
3.2. Proses 22	18
4. MEMORI	19
4.1. Memori 41	19
4.2. Memori 42.....	19
4.3. Memori 43.....	19
4.4. Memori 44.....	19
4.5. Memori 45.....	19
4.6. Memori 46.....	19
4.7. Memori 47.....	19
4.8. Memori 48.....	20
4.9. Memori 49.....	20
4.10. Memori 4A	20
4.11. Memori 4B	20
4.12. Memori 4C	20
4.13. Memori 4D.....	20
5. BERKAS	21
5.1. Berkas 51.....	21
5.2. Berkas 52.....	21
5.3. Berkas 53.....	21
5.4. Berkas 54.....	21
5.5. Berkas 55.....	21
5.6. Berkas 56.....	21
5.7. Berkas 57.....	21
5.8. Berkas 58.....	22
5.9. Berkas 59.....	22
5.10. Berkas 5A.....	22
5.11. Berkas 5B.....	22
5.12. Berkas 5C.....	22
5.13. Berkas 5D.....	22
6. LINUX	23
6.1. Linux 71	23
6.2. Linux 72	23
6.3. Linux 73	23
6.4. Linux 74	23
6.5. Linux 75	23

6.6. Linux 76	23
6.7. Linux 77	23
6.8. Linux 78	24
6.9. Linux 79	24
6.10. Linux 7A	24
6.11. Linux 7B	24
6.12. Linux 7C	24
6.13. Linux 7D	24
6.14. Linux 7E.....	24
6.15. Linux 7F.....	25
6.16. Linux 7G	25
6.17. Linux 7H	25
Daftar Pustaka	26
A. APPENDIX A	27
B. APPENDIX B	28
Indeks.....	29

Bab 1. PENDAHULUAN

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

1.1. Seputar sistem Operasi

Sistem operasi adalah sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Sebelum ada sistem operasi orang hanya menggunakan komputer dengan menggunakan sinyal analog dan sinyal digital. Seiring dengan berkembangnya pengetahuan dan teknologi manusia, sistem operasi ikut berkembang, sehingga pada saat ini terdapat berbagai sistem operasi dengan keunggulan masing-masing. Untuk lebih memahami sistem operasi maka sebaiknya kita mengetahui terlebih dahulu beberapa konsep dasar mengenai sistem operasi itu sendiri.

1.1.1. Apa itu Sistem Operasi

Pengertian secara umum adalah pengelola seluruh sumberdaya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*sistem calls*) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumberdaya sistem komputer.

1.1.2. Fungsi dasar sistem operasi

Sistem komputer pada dasarnya terdiri dari empat komponen utama, yaitu perangkat- keras, program aplikasi, sistem-operasi, dan pengguna. Pada suatu sistem komputer, sistem operasi berfungsi untuk mengatur dan mengawasi penggunaan perangkat keras yang digunakan dalam berbagai program aplikasi dan beberapa pengguna

Sistem operasi berfungsi ibarat pemerintah dalam suatu negara, artinya sistem operasi membuat kondisi komputer agar dapat menjalankan program komputer secara benar. Untuk menghindari konflik yang terjadi pada saat pengguna menggunakan sumberdaya yang sama, sistem operasi mengatur pengguna mana yang dapat mengakses suatu sumberdaya. Oleh karena itu sistem operasi juga sering disebut *resource allocator* . Satu lagi fungsi penting sistem operasi adalah sebagai pengontrol program yang bertujuan untuk menghindari *error* dan penggunaan komputer yang tidak perlu.

1.1.3. Tujuan Mempelajari Sistem Operasi

Tujuan mempelajari sistem operasi antara lain adalah agar dapat merancang sendiri dan dapat memodifikasi sistem yang telah ada sesuai dengan kebutuhan kita, agar dapat memilih alternatif sistem operasi, memaksimalkan penggunaan sistem operasi dan agar konsep dan teknik sistem operasi dapat diterapkan pada aplikasi-aplikasi lain.

1.1.4. Sasaran Sistem Operasi

Sistem operasi mempunyai 3 sasaran utama yaitu kenyamanan (membuat penggunaan komputer menjadi lebih nyaman), efisiensi (penggunaan sumberdaya sistem komputer secara efisien) dan mampu berevolusi (sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru).

1.1.5. Sejarah Sistem Operasi

Setelah komputer diciptakan, menurut TANENBAUM Sistem Operasi mengalami perkembangan yang sangat pesat, dan bisa dibagi kedalam empat generasi :

1.1.5.1. Generasi Pertama

Generasi Pertama, antara tahun 1945-1955. Generasi ini adalah 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 error. Pada generasi ini belum ada sistem operasi, maka sistem komputer diberi instruksi yang harus dikerjakan secara langsung.

1.1.5.2. Generasi Kedua

Generasi Kedua, antara tahun 1955-1965 generasi ini adalah masanya *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 adalah FMS dan IBSYS

1.1.5.3. Generasi Ketiga

Generasi Ketiga, antara tahun 1965-1980. Pada generasi ini perkembangan sistem operasi dikembangkan untuk melayani banyak pemakai sekaligus, dimana pemakai-pemakai interaktif berkomunikasi lewat terminal secara on-line ke komputer, maka sistem operasi menjadi *multiuser* (di gunakan banyak orang sekaligus) dan *multiprogramming*(melayani banyak program sekaligus).

1.1.5.4. Generasi Keempat

Generasi Keempat, Sistem Operasi berbasis Jaringan, antara tahun 1980 - 199x. Sistem operasi dipergunakan untuk jaringan komputer dimana pemakai menyadari keberadaan komputer-komputer yang saling terhubung satu sama lainnya. Pada masa ini user juga telah dinyamankan dengan *Graphical User Interface* yaitu antarmuka komputer yang berbasis grafis yang sangat nyaman, pada masa ini juga dimulai era komputasi tersebar dimana komputasi-komputasi tidak lagi berpusat di satu titik, tetapi dipecah dibanyak komputer sehingga tercapai kinerja yang lebih baik.

1.1.6. Layanan Sistem Operasi

Sistem Operasi yang baik menurut TANENBAUM harusnya memiliki pelayanan sebagai berikut : Pembuatan Program, Eksekusi Program, Pengaksesan *I/O Device*, Pengaksesan terkendali terhadap file, Pengaksesan sistem, Deteksi dan Pemberian tanggapan pada error, Akunting.

Pembuatan Program yaitu Sistem Operasi menyediakan fasilitas dan layanan untuk membantu pemrogram untuk menulis program, Eksekusi Program yang berarti Instruksi-instruksi dan data-data harus dimuat ke main-memory, perangkat-parangkat masukan/pengeluaran dan files harus diinisialisasi, serta sumber-daya yang ada harus disiapkan, semua itu harus di tangani oleh sistem operasi, Pengaksesan *I/O Device* artinya Sistem Operasi harus mengambil alih sejumlah instruksi yang rumit dan sinyal kendali menjengkelkan agar pemrogram dapat berfikir sederhana dan perangkatpun dapat beroperasi, Pengaksesan terkendali terhadap file yang artinya Disediakkannya mekanisme proteksi terhadap file untuk mengendalikan pengaksesan terhadap file, Pengaksesan sistem artinya pada pengaksesan dipakai bersama (*shared system*), fungsi pengaksesan harus menyediakan proteksi terhadap sejumlah sumberdaya dan data dari pemakai tak terdistorsi serta menyelesaikan konflik-konflik dalam perebutan sumberdaya, Deteksi dan Pemberian tanggapan pada error yaitu Jika muncul permasalahan muncul di sistem komputer maka sistem operasi harus memberikan tanggapan yang menjelaskan error yang terjadi serta dampaknya terhadap aplikasi yang sedang berjalan dan Akunting yang artinya Sistem Operasi yang bagus mengumpulkan data statistik penggunaan beragam sumberdaya dan memonitor parameter kinerja.

1.2. Struktur Komputer

Adapun Struktur Sistem Komputer adalah :

- Sistem Operasi Komputer
- Struktur I/O
- Struktur Penyimpanan
- *Storage Hierarchy*
- Proteksi Perangkat Keras

1.2.1. Sistem Operasi Komputer

Dewasa ini sistem computer multiguna terdiri dari CPU (*central processing unit*) sejumlah *device controller* yang dihubungkan melalui *bus* yang menyediakan akses ke memori . Setiap *device controller* bertugas mengatur perangkat yang spesifik (contohnya *disk drive*, *audio device*, dan *video display*). CPU dan *device controller* dapat dijalankan secara bersamaan, namun demikian diperlukan mekanisme sinkronisasi untuk mengatur akses ke memori.

Pada saat pertama kali dijalankan atau di-*reboot*, terdapat sebuah program awal yang mesti dijalankan. Program awal ini disebut program *bootstrap*. Program ini berisi semua aspek dari sistem computer, mulai dari register CPU, *device controller*, sampai isi memori.

Interupsi merupakan bagian penting dari sistem arsitektur computer. Setiap sistem computer memiliki mekanisme yang berbeda-beda. Interupsi bisa terjadi apabila perangkat keras (*hardware*) atau perangkat lunak (*software*) minta "dilayani" oleh prosesor. Apabila terjadi interupsi maka prosesor menghentikan proses yang sedang dikerjakannya, kemudian beralih mengerjakan *service routine* untuk melayani interupsi tersebut. Setelah selesai mengerjakan *service routine* maka prosesor kembali melanjutkan proses yang tertunda.

1.2.2. Struktur I/O

Dalam bab ini kita berbicara tentang struktur I/O, interupsi I/O, dan DMA. Serta perbedaan dalam penanganan interupsi.

1.2.2.1. Interupsi I/O

Untuk memulai operasi I/O, CPU me-load register yang bersesuaian ke *device controller*. Sebaliknya *device controller* memeriksa isi register untuk kemudian menentukan operasi apa yang harus dilakukan. Pada saat operasi I/O dijalankan ada dua kemungkinan, yaitu *synchronous I/O* dan *asynchronous I/O*. Pada *synchronous I/O*, kendali dikembalikan ke proses pengguna setelah proses I/O selesai dikerjakan. Sedangkan pada *asynchronous I/O*, kendali dikembalikan ke proses pengguna tanpa menunggu proses I/O selesai. Sehingga proses I/O dan proses pengguna dapat dijalankan secara bersamaan.

1.2.2.2.

Direct Memory Acces (DMA) suatu metode penanganan I/O dimana *device controller* langsung berhubungan dengan memori tanpa campur tangan CPU. Setelah men-set *buffers*, *pointers*, dan *counters* untuk perangkat I/O, *device controller* mentransfer blok data langsung ke penyimpanan tanpa campur tangan CPU. DMA digunakan untuk perangkat I/O dengan kecepatan tinggi. Hanya terdapat satu interupsi setiap blok, berbeda dengan perangkat yang mempunyai kecepatan rendah dimana interupsi terjadi untuk setiap *byte* (word).

1.2.3. Struktur Penyimpanan

Program computer harus berada di *main memory* (biasanya RAM) untuk dapat dijalankan. *Main memory* adalah satu-satunya tempat penyimpanan yang dapat diakses secara langsung oleh prosesor. Idealnya program dan data secara keseluruhan dapat disimpan dalam *main memory* secara permanen. Namun demikian hal ini tidak mungkin karena:

- Ukuran *main memory* relatif kecil untuk dapat menyimpan data dan program secara keseluruhan.
- *Main memory* bersifat *volatile*, tidak bisa menyimpan secara permanen, apabila komputer dimatikan maka data yang tersimpan di *main memory* akan hilang.

1.2.3.1. Main Memory

Hanya *main memory* dan register merupakan tempat penyimpanan yang dapat diakses secara langsung oleh prosesor. Oleh karena itu instruksi dan data - data yang akan dieksekusi harus disimpan di *main memory* atau register.

Untuk mempermudah akses perangkat I/O ke memori, pada arsitektur komputer menyediakan fasilitas pemetaan memori ke I/O. Dalam hal ini sejumlah alamat di memori dipetakan dengan *device register* . Membaca dan menulis pada alamat memori ini menyebabkan data ditransfer dari dan ke *device register*. Metode ini cocok untuk perangkat dengan waktu respon yang cepat seperti *video controller*.

Register yang terdapat dalam prosesor dapat diakses dalam waktu 1 *clock cycle*. Hal ini menyebabkan register merupakan media penyimpanan dengan akses paling cepat dibandingkan dengan *main memory* yang membutuhkan waktu relatif lama. Untuk mengatasi perbedaan kecepatan, dibuatlah suatu *buffer* penyimpanan yang disebut *cache*.

1.2.3.2. Magnetic Disk

Magnetic Disk berperan sebagai *secondary storage* pada sistem komputer modern. *Magnetic Disk* disusun dari piringan-piringan seperti CD. Kedua permukaan piringan diselimuti oleh bahan-bahan magnetik. Permukaan dari piringan dibagi-bagi menjadi *track* yang memutar, yang kemudian dibagi lagi menjadi sektor.

1.2.4. Storage Hierarchy

Dalam *storage hierarchy structure*, data-data yang sama bisa tampil dalam level berbeda dari sistem penyimpanan. Sebagai contoh integer A berlokasi pada file B yang ditambahkan 1, dengan asumsi file B terletak pada *magnetic disk*. Operasi penambahan diproses dengan pertama kali mengeluarkan operasi I/O untuk menduplikat disk block pada A yang terletak pada *main memory*. Operasi ini diikuti dengan kemungkinan penduplikatan A ke dalam *cache* dan penduplikatan A ke dalam internal register. Sehingga penduplikatan A terjadi di beberapa tempat. Pertama terjadi di internal register dimana nilai A berbeda dengan yang di sistem penyimpanan. Dan nilai di A akan kembali sama ketika nilai baru ditulis ulang ke *magnetic disk*

Pada kondisi multi prosesor, situasi akan menjadi lebih rumit. Hal ini disebabkan masing-masing prosesor mempunyai . Dalam kondisi seperti ini hasil duplikat dari A mungkin hanya ada di beberapa *cache*. Karena CPU(register-register) dapat dijalankan secara bersamaan maka kita harus memastikan perubahan nilai A pada satu *cache* akan mengubah nilai A pada semua *cache* yang ada. Hal ini disebut sebagai *Cache Coherency*.

1.2.5. Proteksi Perangkat Keras

Sistem komputer terdahulu berjenis *programmer-operated systems*. Ketika komputer dioperasikan dalam konsol mereka(pengguna) harus melengkapi sistem terlebih dahulu. Akan tetapi setelah sistem operasi lahir maka hal tersebut diambil alih oleh sistem operasi. Sebagai contoh pada monitor yang proses I/O sudah diambil alih oleh O/S, padahal dahulu hal ini dilakukan oleh pengguna.

Untuk meningkatkan utilisasi sistem, O/S akan membagi sistem sumber daya sepanjang program secara simultan. Pengertian *spooling* adalah suatu program dapat dikerjakan walaupun I/O masih mengerjakan proses lainnya dan disk secara bersamaan menggunakan data untuk banyak proses. Pengertian *multi programming* adalah kegiatan menjalankan beberapa program pada memori pada satu waktu.

Pembagian ini memang menguntungkan sebab banyak proses dapat berjalan pada satu waktu akan tetapi mengakibatkan masalah-masalah baru. Ketika tidak di *sharing* maka jika terjadi kesalahan hanyalah akan membuat kesalahan program. Tapi jika *disharing* jika terjadi kesalahan pada satu proses/program akan berpengaruh pada proses lainnya

Sehingga diperlukan pelindung(proteksi).Tanpa proteksi jika terjadi kesalahan maka hanya satu saja program yang dapat dijalankan atau seluruh output pasti diragukan

Banyak kesalahan pemrograman dideteksi oleh perangkat keras. Kesalahan ini biasanya ditangani oleh sistem operasi.Jika terjadi kesalahan program, perangkat keras akan meneruskan kepada sistem operasi dan sistem operasi akan menginterupsi dan mengakhirinya. Pesan kesalahan disampaikan, dan memori dari program akan dibuang. Tapi memori yang terbuang biasanya tersimpan pada disk agar *programmer* bisa membetulkan kesalahan dan menjalankan program ulang.

1.2.5.1. Operasi Dual Mode

Untuk memastikan operasi berjalan baik kita harus melindungi O/S, program, dan data dari program-program yang *error*. Proteksi ini memerlukan *share resources*. Hal ini bisa dilakukan O/S dengan cara menyediakan pendukung perangkat keras yang memungkinkan kita membedakan mode pengeksekusian program.

Mode yang kita butuhkan ada dua mode operasi yaitu

- Mode Monitor
- Mode Pengguna

Pada perangkat keras akan ada bit atau Bit Mode yang berguna untuk membedakan mode apa yang sedang digunakan dan apa yang sedang dikerjakan. Jika Mode Monitor maka akan bernilai 0, dan jika Mode Pengguna maka akan bernilai 1.

Pada saat *boot time*, perangkat keras bekerja pada mode monitor dan setelah sistem operasi di-*load* maka akan mulai masuk ke mode pengguna. Ketika terjadi *trap* atau interupsi, perangkat keras akan *switch* lagi keadaan dari mode pengguna menjadi mode monitor (terjadi perubahan state menjadi bit 0). Dan akan kembali menjadi mode pengguna jikalau O/S mengambil alih proses dan kontrol komputer (state akan berubah menjadi bit 1)

1.2.5.2. Proteksi I/O

Pengguna bisa mengacaukan sistem operasi dengan melakukan instruksi I/O ilegal dengan mengakses lokasi memori untuk O/S atau dengan cara hendak melepaskan diri dari prosesor. Untuk mencegahnya kita menganggap semua instruksi I/O sebagai *privaladge instruction* sehingga mereka tidak bisa mengerjakan instruksi I/O secara langsung ke memori tapi harus lewat O/S terlebih dahulu. Proteksi I/O dikatakan selesai jika pengguna dapat dipastikan tidak akan menyentuh mode monitor. Jika hal ini terjadi proteksi I/O dapat dikompromikan.

1.2.5.3. Proteksi Memori

Salah satu proteksi perangkat keras adalah dengan proteksi memori yaitu dengan pembatasan penggunaan memori. Disini diperlukan beberapa istilah yaitu:

- Base Register yaitu alamat memori fisik awal yang dialokasikan/boleh digunakan oleh pengguna
- Limit Register yaitu nilai batas dari alamat memori fisik awal yang dialokasikan/boleh digunakan oleh pengguna
- Proteksi Perangkat Keras

Sebagai contoh sebuah pengguna dibatasi mempunyai base register 300040 dan mempunyai limit register 420940 maka pengguna hanya diperbolehkan menggunakan alamat memori fisik antara 300040 hingga 420940 saja

1.3. Struktur Sistem Komputer

1.3.1. Komponen-komponen Sistem

Pada kenyataannya tidak semua sistem operasi mempunyai struktur yang sama. Namun menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne umumnya sistem operasi modern mempunyai komponen sebagai berikut:

- Manajemen Proses
- Manajemen *Main-Memory*
- Manajemen *Secondary-Storage*
- Manajemen Sistem I/O
- Manajemen File
- Sistem Proteksi
- Jaringan
- *Command-Interpreter system*

1.3.2. Manajemen 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, file-file, dan perangkat-perangkat I/O.

Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti :

- Pembuatan dan penghapusan proses pengguna dan sistem proses.
- Menunda atau melanjutkan proses.
- Menyediakan mekanisme untuk proses sinkronisasi.
- Menyediakan mekanisme untuk proses komunikasi.
- Menyediakan mekanisme untuk penanganan *deadlock*

1.3.3. Manajemen *Main-Memory*

Main-memory 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. *Main-memory* berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. *Main-memory* 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 :

- Menjaga *track* dari memori yang sedang digunakan dan siapa yang menggunakannya.
- Memilih program yang akan di-*load* ke memori.
- Mengalokasikan dan meng-*dealokasikan* *memory-space* sesuai kebutuhan.

1.3.4. Manajemen *Secondary-Storage*

Data yang disimpan dalam *main-memory* 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, penjadwalan disk

1.3.5. Manajemen 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 file pada *hard-disk*, CD-ROM dan *floppy disk*.

Komponen Sistem Operasi untuk sistem I/O :

- *Buffer* : menampung sementara data dari/ke perangkat I/O.
- *Spooling* : melakukan penjadwalan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
- Menyediakan "*driver*" untuk dapat melakukan operasi "rinci" untuk perangkat keras I/O tertentu.

1.3.6. Manajemen File

File adalah kumpulan informasi yang berhubungan (sesuai dengan tujuan pembuat file tsb). File dapat mempunyai struktur yang bersifat hirarkis (direktori, volume dll). Sistem Operasi bertanggung jawab:

- Pembuatan dan penghapusan file.
- Pembuatan dan penghapusan direktori.
- Mendukung manipulasi file dan direktori.
- Memetakan file ke *secondary storage*.
- Mem-*backup* file ke media penyimpanan yang permanen (*nonvolatile*).

1.3.7. Sistem Proteksi

Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program, prosesor, atau pengguna ke sistem sumber daya.

Mekanisme proteksi harus :

- membedakan antara penggunaan yang sudah diberi izin dan yang belum.
- *specify the controls to be imposed.*
- *provide a means of enforcement.*

1.3.8. 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 resource sistem .

Akses tersebut menyebabkan :

- *Computation speed-up.*
- *Increased data availability.*
- *Enhanced reliability.*

1.3.9. 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, shell (in UNIX)*

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) etc.

1.3.10. 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 file adalah kemampuan program untuk operasi pada file (membaca, menulis, membuat, and menghapus file). 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 maupun operasi.

Efisiensi penggunaan sistem :

- *Resource allocator* adalah mengalokasikan sumberdaya ke beberapa pengguna atau *job* yang jalan pada saat yang bersamaan.
- Proteksi menjamin akses ke sistem sumber daya dikendalikan (pengguna dikontrol aksesnya ke sistem).
- *Accounting* adalah merekam kegiatan pengguna, jatah pemakaian sumber daya (keadilan atau kebijaksanaan).

1.3.11. 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 file.

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:

- Melalui registers (sumber daya di CPU).
- Menyimpan parameter pada data struktur (table) di memori, dan alamat table tsb ditunjuk oleh *pointer* yang disimpan di register.
- *Push (store)* melalui "*stack*" pada memori dan OS mengambilnya melalui *pop* pada *stack* tsb.

1.3.12. Virtual Machines

Sebuah *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.

Virtual Machine(VM) => control program yang minimal VM memberikan ilusi *multitasking*: seolah-olah terdapat prosesor dan memori eksklusif digunakan VM. VM memilah fungsi *multitasking* dan implementasi *extended machine* (tergantung proses pengguna) => flexible dan lebih mudah untuk

pengaturan. Jika setiap pengguna diberikan satu VM => pengguna bebas untuk menjalankan OS (kernel) yang diinginkan pada VM tersebut. Potensi lebih dari satu OS dalam satu komputer. Contoh : IBM VM370: menyediakan VM untuk berbagai OS: CMS (interaktif), MVS, CICS, dll. Masalah : Sharing disk => setiap OS mempunyai file sistem yang mungkin berbeda. IBM: virtual disk (minidisk) yang dialokasikan untuk pengguna melalui VM.

Konsep VM menyediakan proteksi yang lengkap untuk sumberdaya sistem, dikarenakan tiap VM terpisah dari VM yang lain. Namun, hal tersebut menyebabkan tidak adanya *sharing* sumberdaya secara langsung. VM merupakan alat yang tepat untuk penelitian dan pengembangan sistem operasi. Konsep VM susah untuk diimplementasi sehubungan dengan usaha yang diperlukan untuk menyediakan duplikasi dari mesin utama.

1.3.13. 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 didesain, diimplementasi, dan dimantain, sebagaimana fleksibel, *error* dan efisien.

Mekanisme dan Kebijakan :

- Mekanisme menjelaskan bagaimana melakukan sesuatu kebijakan memutuskan apa yang akan dilakukan. Pemisahan kebijakan dari mekanisme merupakan hal yang sangat penting; ini memungkinkan fleksibilitas yang tinggi bila kebijakan akan diubah nanti.
- Kebijakan memutuskan apa yang akan dilakukan.

Pemisahan kebijakan dari mekanisme merupakan hal yang sangat penting; ini memungkinkan fleksibilitas yang tinggi bila kebijakan akan diubah nanti.

Implementasi Sistem biasanya menggunakan bahas *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 dipindah ke perangkat keras yang lain bila ditulis dengan bahasa tingkat tinggi.

1.3.14. 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.

Booting : memulai komputer dengan me-load kernel.

Bootstrap program : kode yang disimpan di code ROM yang dapat menempatkan kernel, memasukkannya kedalam memori, dan memulai eksekusinya.

1.4. Rangkuman

Sistem operasi telah berkembang selama lebih dari 40 tahun dengan dua tujuan utama. Pertama, sistem operasi mencoba mengatur aktivitas-aktivitas komputasi untuk memastikan pendayagunaan yang baik dari sistem komputasi tersebut. Kedua, menyediakan lingkungan yang nyaman untuk pengembangan dan jalankan dari program.

Pada awalnya, sistem komputer digunakan dari depan konsol. Perangkat lunak seperti *assembler*, *loader*, *linker* dan *compiler* meningkatkan kenyamanan dari sistem pemrograman, tapi juga memerlukan waktu *set-up* yang banyak. Untuk mengurangi waktu *set-up* tersebut, digunakan jasa operator dan menggabungkan tugas-tugas yang sama (sistem *batch*). Sistem *batch* mengizinkan pengurutan tugas secara otomatis dengan menggunakan sistem operasi yang resident dan memberikan peningkatan yang cukup besar dalam utilisasi komputer. Komputer tidak perlu lagi menunggu operasi oleh pengguna. Tapi utilisasi CPU tetap saja rendah. Hal ini dikarenakan lambatnya kecepatan alat-alat untuk I/O relatif terhadap kecepatan CPU. Operasi *off-line* dari alat-alat yang lambat bertujuan untuk menggunakan beberapa sistem *reader-to-tape* dan *tape-to-printer* untuk satu CPU.

Untuk meningkatkan keseluruhan kemampuan dari sistem komputer, para developer memperkenalkan konsep *multiprogramming*. Dengan *multiprogramming*, beberapa tugas disimpan dalam memori dalam satu waktu; CPU digunakan secara bergantian sehingga menambah utilisasi CPU dan mengurangi total waktu yang dibutuhkan untuk menyelesaikan tugas-tugas tersebut. *Multiprogramming*, yang dibuat untuk meningkatkan kemampuan, juga mengizinkan *time sharing*. Sistem operasi yang bersifat *time-shared* memperbolehkan banyak pengguna untuk menggunakan komputer secara interaktif pada saat yang bersamaan. **Personal Komputer** adalah mikrokomputer yang dianggap lebih kecil dan lebih murah dibandingkan komputer *mainframe*. Sistem operasi untuk komputer-komputer seperti ini diuntungkan oleh pengembangan sistem operasi untuk komputer *mainframe* dalam beberapa hal. Namun, semenjak penggunaan komputer untuk keperluan pribadi, maka utilisasi CPU tidak lagi menjadi perhatian utama. Karena itu, beberapa desain untuk komputer *mainframe* tidak cocok untuk sistem yang lebih kecil.

Sistem parallel mempunyai lebih dari satu CPU yang mempunyai hubungan yang erat; CPU-CPU tersebut berbagi bus komputer, dan kadang-kadang berbagi memori dan perangkat yang lainnya. Sistem seperti itu dapat meningkatkan *throughput* dan reliabiliti. **Sistem hard real-time** seringkali digunakan sebagai alat pengontrol untuk aplikasi yang dedicated. Sistem operasi yang *hard real-time* mempunyai batasan waktu yang tetap yang sudah didefinisikan dengan baik. Pemrosesan harus selesai dalam batasan-batasan yang sudah didefinisikan, atau sistem akan gagal. **Sistem soft real-time** mempunyai lebih sedikit batasan waktu yang keras, dan tidak mendukung penjadwalan dengan menggunakan batas akhir. Pengaruh dari internet dan *World Wide Web* baru-baru ini telah mendorong pengembangan sistem operasi modern yang menyertakan *web browser* serta perangkat lunak jaringan dan komunikasi sebagai satu kesatuan.

Multiprogramming dan *sistem time-sharing* meningkatkan kemampuan komputer dengan melampaui batas operasi (*overlap*) CPU dan I/O dalam satu mesin. Hal seperti itu memerlukan perpindahan data antara CPU dan alat I/O, ditangani baik dengan polling atau *interrupt-driven* akses ke *I/O port*, atau dengan perpindahan DMA. Agar komputer dapat menjalankan suatu program, maka program tersebut harus berada di memori utama (*main memory*). **Memori utama adalah** satu-satunya tempat penyimpanan yang besar yang dapat diakses secara langsung oleh prosessor, merupakan suatu *array* dari *word* atau *byte*, yang mempunyai ukuran ratusan sampai jutaan ribu. Setiap *word* memiliki alamatnya sendiri. memori utama adalah tempat penyimpanan yang *volatile*, dimana isinya hilang bila sumber energinya (energi listrik) dimatikan. Kebanyakan sistem komputer menyediakan *secondary storage* sebagai perluasan dari memori utama. Syarat utama dari *secondary storage* adalah dapat

menyimpan data dalam jumlah besar secara permanen. *Secondary storage* yang paling umum adalah disk magnetik, yang menyediakan penyimpanan untuk program maupun data. **Disk magnetik** adalah alat penyimpanan data yang *nonvolatile* yang juga menyediakan akses secara random. **Tape magnetik** digunakan terutama untuk *backup*, penyimpanan informasi yang jarang digunakan, dan sebagai media pemindahan informasi dari satu sistem ke sistem yang lain.

Beragam sistem penyimpanan dalam sistem komputer dapat disusun dalam hirarki berdasarkan kecepatan dan biayanya. Tingkat yang paling atas adalah yang paling mahal, tapi cepat. Semakin kebawah, biaya perbit menurun, sedangkan waktu aksesnya semakin bertambah (semakin lambat). Sistem operasi harus memastikan operasi yang benar dari sistem komputer. Untuk mencegah pengguna program mengganggu operasi yang berjalan dalam sistem, perangkat keras mempunyai dua mode : mode pengguna dan mode *monitor*. Beberapa perintah (seperti perintah I/O dan perintah halt) adalah perintah khusus, dan hanya dapat dijalankan dalam mode *monitor*. Memori juga harus dilindungi dari modifikasi oleh pengguna. *Timer* mencegah terjadinya pengulangan secara terus menerus (*infinite loop*). Hal-hal tersebut (dual mode, perintah khusus, pengamanan memori, *timer interrupt*) adalah blok bangunan dasar yang digunakan oleh sistem operasi untuk mencapai operasi yang sesuai.

Sistem operasi menyediakan banyak pelayanan. Di tingkat terendah, *sistem calls* mengizinkan program yang sedang berjalan untuk membuat permintaan secara langsung dari sistem operasi. Di tingkat tertinggi, *command interpreter* atau *shell* menyediakan mekanisme agar pengguna dapat membuat permintaan tanpa menulis program. *Command* dapat muncul dari file sewaktu jalankan *batch-mode*, atau secara langsung dari terminal ketika dalam mode interaktif atau *time-shared*. Program sistem disediakan untuk memenuhi kebanyakan dari permintaan pengguna. Tipe dari permintaan beragam sesuai dengan levelnya. Level *sistem call* harus menyediakan fungsi dasar, seperti kontrol proses serta manipulasi alat dan file. Permintaan dengan level yang lebih tinggi (*command interpreter* atau program sistem) diterjemahkan kedalam urutan *sistem call*.

Pelayanan sistem dapat dikelompokkan kedalam beberapa kategori : kontrol program, status permintaan dan permintaan I/O. Program *error* dapat dipertimbangkan sebagai permintaan yang implisit untuk pelayanan. Bila sistem pelayanan sudah terdefinisi, maka struktur dari sistem operasi dapat dikembangkan. Berbagai macam tabel diperlukan untuk menyimpan informasi yang mendefinisikan status dari sistem komputer dan status dari sistem tugas. Perancangan dari suatu sistem operasi yang baru merupakan tugas yang utama. Sangat penting bahwa tujuan dari sistem sudah terdefinisi dengan baik sebelum memulai perancangan. Tipe dari sistem yang diinginkan adalah landasan dalam memilih beragam *algoritma* dan strategi yang akan digunakan. Karena besarnya sistem operasi, maka modularitas adalah hal yang penting. Merancang sistem sebagai suatu urutan dari *layer* atau dengan menggunakan *mikrokernel* merupakan salah satu teknik yang baik. Konsep *virtual machine* mengambil pendekatan *layer* dan memperlakukan baik itu *kernel* dari sistem operasi dan perangkat kerasnya sebagai suatu perangkat keras. Bahkan sistem operasi yang lain dapat dimasukkan diatas *virtual machine* tersebut. Setiap sistem operasi yang mengimplemen JVM dapat menjalankan semua program java, karena JVM mendasari dari sistem ke program java, menyediakan arsitektur tampilan yang netral.

Didalam daur perancangan sistem operasi, kita harus berhati-hati untuk memisahkan pembagian kebijakan (*policy decision*) dengan detail dari implementasi (*mechanism*). Pemisahan ini membuat fleksibilitas yang maksimal apabila *policy decision* akan diubah kemudian. Sistem operasi sekarang ini hampir selalu ditulis dengan menggunakan bahasa tingkat tinggi. Hal ini meningkatkan implementasi, perawatan portabilitas. Untuk membuat sistem operasi untuk suatu konfigurasi mesin tertentu, kita harus melakukan *system generation*.

1.5. Pertanyaan

1. Sebutkan tiga tujuan utama dari sistem operasi ?
2. Sebutkan keuntungan dari *multiprogramming* ?
3. Apakah perbedaan utama dari sistem operasi untuk komputer *mainframe* dan PC ?
4. Sebutkan kendala-kendala yang harus diatasi oleh *programmer* dalam menulis sistem operasi untuk lingkungan waktu nyata ?
5. Jelaskan perbedaan antara *symmetric* dan *asymmetric multiprocessing*. Sebutkan keuntungan dan kerugian dari sistem *multiprocessor* ?
6. Apakah perbedaan antara *trap* dan *interrupt* ? Sebutkan penggunaan dari setiap fungsi tersebut ?.
7. Untuk jenis operasi apakah **DMA** itu berguna ? Jelaskan jawabanmu !.
8. Sebutkan dua alasan mengapa *cache* berguna. Problem apakah yang dapat dipecahkan dan juga muncul dengan adanya *cache* ?
9. Beberapa **CPU** menyediakan lebih dari dua mode operasi. Sebutkan dua kemungkinan penggunaan dari mode tersebut ?.
10. Sebutkan lima kegiatan utama dari sistem operasi yang berhubungan dengan manajemen proses !
11. Sebutkan tiga kegiatan utama dari sistem operasi yang berhubungan dengan manajemen memori !
12. Sebutkan tiga kegiatan utama dari sistem operasi yang berhubungan dengan manajemen *secondary-storage* !
13. Sebutkan lima kegiatan utama dari sistem operasi yang berhubungan dengan manajemen file !
14. Apakah tujuan dari *command interpreter* ? Mengapa biasanya hal tersebut terpisah dengan *kernel* ?

1.6. Referensi

1. <http://cs-www.cs.yale.edu/homes/avi/os-book/osc/slide-dir/>
2. <http://www.ignou.ac.in/virtualcampus/adit/course/index-tr1.htm>
3. <http://www.cs.wpi.edu/~cs502/s99/>
4. <http://legion.virginia.edu/presentations/sc2000/sld001.htm>
5. <http://www.mcsr.olemiss.edu/unixhelp/concepts/history.html>
6. <http://www.osdata.com/kind/history.htm>
7. <http://agt.buka.org/concept.html>
8. <http://www.techrescue.net/guides/insthware.asp>
9. <http://www.chipcenter.com/circuitcellar/march02/c0302dc4.htm>

10. <http://www.hardware.fr/articles/338/page1.html>
11. <http://www.csc.uvic.ca/~mcheng/360/notes/NOTES2.html>
12. <http://kos.enix.org/pub/greenwald96synergy.pdf>
13. <http://www.cs.panam.edu/fox/CSCI4334/ch3.ppt>
14. <http://www.imm.dtu.dk/courses/02220/OS/OH/week7.pdf>
15. <http://www.hardware.fr/articles/338/page1.html>
16. <http://www.cs.technion.ac.il/~hagit/OSS98>
17. <http://www.cis.umassd.edu/~rbalasubrama/>

Daftar Pustaka

[Silberschatz2000] Avi Silberschatz, Peter Galvin, dan Grag Gagne, 2000, *Applied Operating Systems: First Edition*, Edisi Pertama, John Wiley & Sons.

[Walsh2002] Norman Walsch dan Leonard Muellner, Bob Stayton, 1999, 2000, 2001, 2002, *DocBook: The Definitive Guide*, Version 2.0.7, O'Reilly.

Lampiran B. APPENDIX B

bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla... bla...
bla... bla... bla... bla... bla... bla... bla... bla... bla... bla...

Indeks

Bad Blocks, ?
Blocking dan Nonblocking I/O, ?
Boot Block, ?
Buffering, ?
Caching, ?
Circular-SCAN, ?
Daftar Istilah, ?
DMA, ?
Error Handling, ?
First Come First Serve, ?
Fitur pada Komputer Modern, ?
Format Disk, ?
Handshaking, ?
I/O Scheduling, ?
Implementasi DMA, ?
Implementasi Fungsi I/O, ?
Implementasi Stable-Storage, ?
Interface Aplikasi I/O, ?
Interrupt Request Line, ?
Interrupt Vector dan Interrupt Chaining, ?
Jam dan Timer, ?
Kernel Data Structure, ?
Kernel I/O Subsystem, ?
Lokasi Swap-Space, ?
LOOK, ?
Mekanisme Dasar Interupsi, ?
Memilih Algoritma Penjadwalan Disk, ?
Meningkatkan Efisiensi I/O, ?
Penanganan Swap-Space, ?
Pengaruh I/O pada Performa, ?
Pengelolaan Swap-Space, ?
Penggunaan Swap-Space, ?
Penjadwalan Blok, ?
Penyebab Interupsi, ?
Peralatan Block dan Karakter, ?
Peralatan Jaringan, ?
Perangkat Keras I/O, ?
Permintaan I/O, ?
Polling, ?
Reabilitas Disk, ?
Referensi, ?
SCAN, ?
Shortest Seek Time First, ?
Soal Latihan, ?
Spooling dan Reservasi Device, ?

Struktur Disk, ?
Summary, ?
Transfer DMA, ?