

Tugas Sistem Operasi Sesi 12

Aldi Maulana Iqbal – 20210801222

MANAJEMEN I/O

Pengertian

Manajemen I/O (Input/Output), juga dikenal sebagai manajemen perangkat, adalah proses koordinasi operasi input dan output dari sistem komputer. Ini melibatkan penggunaan "driver perangkat", yang merupakan program khusus yang memungkinkan sistem operasi untuk berkomunikasi dengan perangkat keras seperti hard drive, drive CD-ROM, dan drive floppy.

Driver perangkat bertindak sebagai perantara antara sistem operasi dan perangkat keras, memungkinkan sistem operasi melakukan operasi input dan output pada perangkat secara konsisten. Sebagai contoh, pengguna dapat menggunakan rangkaian operasi input/output yang sama untuk membaca file dari hard drive, CD-ROM, atau floppy disk, karena driver perangkat untuk masing-masing perangkat ini telah dirancang agar sesuai dengan antarmuka standar.

Manajemen I/O penting karena memungkinkan sistem operasi mengabstraksi detail perangkat keras individual dan menyediakan serangkaian operasi input/output yang konsisten untuk semua perangkat. Ini memudahkan pengguna untuk bekerja dengan berbagai jenis perangkat, dan bagi pengembang perangkat lunak untuk membuat aplikasi yang dapat berjalan di berbagai platform perangkat keras.

Komponen Sistem Operasi Untuk Sistem I/O

Komponen sistem operasi untuk sistem I/O adalah bagian dari sistem operasi yang bertanggung jawab untuk menangani input dan output (I/O) di komputer. Berikut ini adalah beberapa komponen sistem operasi untuk sistem I/O yang umumnya ditemukan:

1. **Buffer:** Buffer merupakan area memori sementara yang digunakan untuk menyimpan data yang sedang ditransfer antara perangkat I/O dan prosesor. Ini membantu mengurangi kebutuhan untuk menunggu sambungan I/O yang lambat, seperti pembacaan atau penulisan ke disk, sehingga meningkatkan efisiensi sistem.
2. **Spooling:** Spooling adalah teknik yang digunakan sistem operasi untuk mengelola pemakaian I/O secara efisien dengan menjadwalkan operasi I/O dalam antrian. Ini memungkinkan prosesor untuk terus menjalankan proses lain ketika operasi I/O sedang dilakukan, sehingga meningkatkan efisiensi sistem.
3. **Driver:** Driver adalah perangkat lunak yang menyediakan interface antara sistem operasi dan perangkat keras I/O tertentu. Driver memungkinkan sistem operasi untuk melakukan operasi rinci pada perangkat keras I/O, seperti mengontrol kecepatan, mode, dan sebagainya.
4. **Manajemen perangkat I/O:** Manajemen perangkat I/O adalah bagian dari sistem operasi yang bertanggung jawab untuk mengelola semua perangkat I/O yang terpasang pada komputer. Ini termasuk mengontrol akses ke perangkat, mengatur prioritas, dan menangani kesalahan yang mungkin terjadi. Manajemen perangkat I/O merupakan aspek yang kompleks dan luas dari sistem operasi karena ada begitu banyak perangkat dan aplikasi yang tersedia.

Fungsi Manajemen I/O

Manajemen I/O adalah bagian dari sistem operasi yang bertanggung jawab untuk mengelola semua input dan output (I/O) yang terjadi pada komputer. Berikut ini adalah beberapa fungsi utama manajemen I/O:

1. Mengirim perintah ke perangkat I/O: Sistem operasi mengirim perintah ke perangkat I/O untuk menyediakan layanan kepada sistem. Contohnya, sistem operasi dapat mengirim perintah ke printer untuk mencetak dokumen, atau ke disk untuk menyimpan atau membaca data.
2. Menangani interupsi perangkat I/O: Saat perangkat I/O selesai melakukan operasi, ia dapat mengirim interupsi ke sistem operasi untuk memberitahukannya. Sistem operasi kemudian dapat menangani interupsi tersebut dengan memproses data yang telah diterima atau melanjutkan operasi I/O yang sedang berlangsung.
3. Menangani kesalahan perangkat I/O: Sistem operasi juga harus dapat menangani kesalahan yang mungkin terjadi pada perangkat I/O, seperti kesalahan pembacaan atau penulisan data. Ini bisa dilakukan dengan menyediakan mekanisme error-checking dan recovery, atau dengan mengirim perintah ke perangkat I/O untuk mengulangi operasi yang gagal.
4. Menyediakan interface ke pemakai: Sistem operasi juga harus menyediakan interface ke pemakai agar dapat mengakses perangkat I/O secara langsung. Ini bisa berupa perintah-perintah di command line atau antarmuka grafis yang disediakan oleh sistem operasi.

Klasifikasi Perangkat I/O

Perangkat I/O dapat dikelompokkan berdasarkan berbagai kriteria, di antaranya:

1. Sifat Aliran Datanya
 - Perangkat berorientasi blok: Perangkat ini menyimpan, menerima, dan mengirim informasi sebagai blok-blok berukuran tetap yang memiliki alamat tersendiri. Hal ini memungkinkan membaca atau menulis blok-blok secara independen, yaitu dapat membaca atau menulis sembarang blok tanpa harus melewati blok-blok lain. Contoh perangkat berorientasi blok adalah disk, tape, CD ROM, dan optical disk.
 - Perangkat berorientasi aliran karakter: Perangkat ini menerima, dan mengirimkan aliran karakter tanpa membentuk suatu struktur blok. Contoh perangkat berorientasi aliran karakter adalah terminal, line printer, pita kertas, kartu-kartu berlubang, interface jaringan, dan mouse.
2. Sasaran Komunikasi
 - Perangkat yang terbaca oleh manusia: Perangkat ini digunakan untuk berkomunikasi dengan manusia melalui antarmuka yang dapat dibaca oleh manusia. Contohnya adalah video display terminal (VDT) seperti monitor, keyboard, dan mouse.

- Perangkat yang terbaca oleh mesin: Perangkat ini digunakan untuk berkomunikasi dengan perangkat elektronik lainnya. Contohnya adalah disk dan tape, sensor, dan controller.
- Perangkat komunikasi: Perangkat ini digunakan untuk berkomunikasi dengan perangkat jarak jauh, seperti modem. Modem memungkinkan komputer untuk terhubung ke internet atau jaringan lain melalui koneksi telepon atau jaringan nirkabel.

Faktor-faktor yang membedakan antar perangkat I/O biasanya meliputi:

- Kecepatan transmisi data (data rate): Ini mengacu pada kecepatan yang dapat dicapai oleh perangkat I/O dalam mentransfer data. Misalnya, kecepatan transmisi data untuk perangkat I/O seperti hard disk dapat berbeda-beda tergantung pada spesifikasi perangkat tersebut.
- Jenis aplikasi yang digunakan: Perangkat I/O yang digunakan bisa berbeda-beda tergantung pada aplikasi yang digunakan. Misalnya, perangkat I/O seperti printer dan scanner biasanya digunakan untuk aplikasi perkantoran, sedangkan perangkat I/O seperti joystick dan gamepad biasanya digunakan untuk aplikasi gaming.
- Tingkat kerumitan dalam pengendalian: Beberapa perangkat I/O mungkin lebih rumit untuk dikendalikan daripada yang lain. Misalnya, perangkat I/O seperti printer atau scanner mungkin memerlukan lebih banyak konfigurasi dan pengaturan daripada perangkat I/O seperti keyboard atau mouse.
- Besarnya unit yang ditransfer: Ukuran unit data yang ditransfer oleh perangkat I/O juga bisa berbeda-beda. Misalnya, perangkat I/O seperti hard disk dapat mentransfer data dalam unit yang lebih besar daripada perangkat I/O seperti printer atau scanner.
- Representasi atau perwujudan data: Perangkat I/O juga bisa berbeda dalam cara mereka mewakili atau menyimpan data. Misalnya, perangkat I/O seperti hard disk menyimpan data dalam bentuk digital, sedangkan perangkat I/O seperti printer mencetak data dalam bentuk fisik.
- Kondisi-kondisi kesalahan: Perangkat I/O juga bisa berbeda dalam cara mereka menangani kesalahan yang mungkin terjadi saat mentransfer data. Misalnya, beberapa perangkat I/O mungkin memiliki mekanisme error-checking dan recovery yang lebih canggih daripada yang lain.

3. Teknik Pemrograman

Teknik pemrograman adalah cara yang digunakan oleh sistem operasi untuk mengelola input dan output (I/O) pada komputer. Berikut ini adalah tiga teknik pemrograman yang umum digunakan:

- I/O terprogram atau Polling system

Teknik ini menggunakan prosesor untuk secara berkala memeriksa status perangkat I/O. Jika perangkat I/O siap untuk melakukan operasi, prosesor akan memproses data atau memberikan perintah kepada perangkat I/O untuk melanjutkan operasi. Teknik ini efektif, tetapi dapat memakan waktu yang cukup lama jika ada banyak perangkat I/O yang terhubung ke sistem.

- I/O dikendalikan interupsi

Teknik ini menggunakan interupsi perangkat I/O untuk memberi tahu sistem operasi bahwa perangkat I/O siap untuk melakukan operasi. Saat perangkat I/O mengirim interupsi, sistem operasi akan menangani interupsi tersebut dan memberikan perintah kepada perangkat I/O untuk melanjutkan operasi. Teknik ini lebih efisien daripada polling system karena tidak memerlukan prosesor untuk secara berkala memeriksa status perangkat I/O.

- Dengan DMA (direct memory access)

Teknik ini menggunakan perangkat DMA (direct memory access) untuk mengirim data langsung dari perangkat I/O ke memori tanpa melalui prosesor. Ini memungkinkan perangkat I/O untuk memindahkan data ke memori dengan cepat, sehingga mengurangi beban pada prosesor dan meningkatkan efisiensi sistem. Namun, teknik ini hanya dapat digunakan pada perangkat I/O yang dilengkapi dengan perangkat DMA.

Evolusi Fungsi Perangkat I/O

Evolusi fungsi perangkat I/O merujuk pada perkembangan sistem I/O yang terjadi seiring dengan perkembangan teknologi komputer. Berikut ini adalah beberapa tahapan evolusi fungsi perangkat I/O yang telah terjadi:

1. Pemroses mengendalikan perangkat I/O secara langsung: Pada sistem komputer pertama, pemroses (processor) secara langsung mengendalikan perangkat I/O. Ini berarti bahwa pemroses harus menunggu sambungan I/O yang lambat selesai sebelum dapat melanjutkan proses lain.
2. Pemroses dilengkapi pengendali I/O (I/O controller): Pada tahap selanjutnya, sistem komputer dilengkapi dengan pengendali I/O (I/O controller) yang bertugas mengontrol perangkat I/O. Ini memungkinkan pemroses untuk terus menjalankan proses lain saat perangkat I/O sedang digunakan.
3. Perangkat dilengkapi fasilitas interupsi: Kemudian, perangkat I/O dibekali dengan fasilitas interupsi (interrupt) yang memungkinkannya untuk mengirim sinyal ke sistem operasi saat operasi selesai. Ini mempercepat proses I/O karena sistem operasi tidak perlu lagi menunggu sambungan I/O yang lambat untuk selesai.
4. I/O controller mengendalikan memori secara langsung lewat DMA: Selanjutnya, I/O controller dibekali dengan kemampuan Direct Memory Access (DMA), yang memungkinkannya untuk mengakses memori secara langsung tanpa harus melalui pemroses. Ini meningkatkan efisiensi sistem karena pemroses tidak perlu terus-menerus mengontrol operasi I/O.
5. Pengendali I/O menjadi pemroses terpisah: Pada tahap terakhir, pengendali I/O menjadi pemroses terpisah yang memiliki memori lokal sendiri. Ini memungkinkan sistem operasi untuk melakukan operasi I/O secara paralel dengan proses lain yang sedang berlangsung di pemroses utama.

Arsitektur pengendalian komunikasi dengan terminal-terminal interaktif merupakan sebuah desain sistem yang digunakan untuk mengelola komunikasi antara terminal-terminal interaktif dan pemroses pusat. Dalam arsitektur ini, pemroses I/O (input/output processor)

bertanggung jawab untuk mengambil alih kebanyakan tugas yang melibatkan pengendalian terminal.

Pemroses I/O memiliki peran penting dalam arsitektur ini karena ia menangani sebagian besar tugas I/O yang melibatkan terminal. Hal ini memungkinkan pemroses pusat untuk terbebas dari tugas-tugas I/O sehingga dapat mengoptimalkan kinerjanya. Dengan demikian, sistem dapat bekerja lebih efisien dan cepat.

Evolusi arsitektur ini bertujuan untuk meminimalkan keterlibatan pemroses pusat dalam tugas I/O, sehingga dapat meningkatkan kinerja sistem secara keseluruhan. Arsitektur ini sangat berguna dalam sistem yang memiliki banyak terminal interaktif yang terhubung ke pemroses pusat, karena memungkinkan pemroses pusat untuk terbebas dari tugas-tugas I/O yang tidak perlu dan fokus pada tugas-tugas yang lebih penting.

Prinsip Manajemen Perangkat I/O

Prinsip manajemen perangkat I/O adalah panduan yang digunakan dalam merancang sistem operasi untuk mengelola perangkat input dan output (I/O). Berikut ini adalah dua prinsip utama manajemen perangkat I/O:

1. Efisiensi: Prinsip ini bertujuan untuk memastikan bahwa sistem operasi dapat mengelola perangkat I/O secara efisien, sehingga dapat meningkatkan kinerja sistem secara keseluruhan. Contohnya, sistem operasi dapat menggunakan buffer untuk menyimpan data yang sedang ditransfer antara perangkat I/O dan prosesor, atau menggunakan spooling untuk menjadwalkan operasi I/O secara efisien.
2. Generalitas (device independence): Prinsip ini bertujuan untuk memastikan bahwa sistem operasi dapat bekerja dengan berbagai jenis perangkat I/O tanpa tergantung pada spesifikasi teknis masing-masing perangkat. Contohnya, sistem operasi dapat menggunakan driver untuk mengontrol perangkat I/O secara rinci, sehingga dapat bekerja dengan berbagai jenis perangkat tanpa harus mengerti spesifikasi teknis masing-masing perangkat.

Software diorganisasikan berlapis adalah sebuah cara penyusunan software yang membagi software ke dalam beberapa lapisan yang saling terkait. Setiap lapisan memiliki peran dan tanggung jawab yang berbeda, sehingga membantu mengelola kerumitan dan kompleksitas software secara lebih efektif.

Lapisan bawah merupakan lapisan yang berurusan dengan perangkat keras. Lapisan ini menyembunyikan kerumitan perangkat keras dari lapisan-lapisan di atasnya, sehingga memudahkan pengembangan software di lapisan-lapisan di atasnya. Lapisan bawah biasanya terdiri dari driver perangkat keras, rutin-rutin sistem operasi, dan sebagainya.

Lapisan di atas lapisan bawah merupakan lapisan yang memberikan antarmuka yang bersih, nyaman, dan seragam kepada pemakai. Lapisan ini bertanggung jawab untuk menyediakan fitur-fitur yang diinginkan oleh pemakai, seperti antarmuka grafis, perintah-perintah command line, dan sebagainya. Lapisan ini mengandalkan lapisan bawah untuk mengakses perangkat keras dan mengelola tugas-tugas I/O.

Dengan mengorganisasikan software berlapis, sistem dapat bekerja lebih efisien dan mudah dipertahankan. Lapisan-lapisan tersebut membantu mengelola kompleksitas software

dan memudahkan pengembangan software di lapisan-lapisan di atasnya. Selain itu, lapisan-lapisan tersebut juga membantu menyediakan antarmuka yang bersih, nyaman, dan seragam kepada pemakai.

Masalah–Masalah Manajemen I/O

Masalah-masalah manajemen I/O adalah pertanyaan-pertanyaan yang harus dijawab oleh sistem operasi dalam mengelola input dan output (I/O) di komputer. Berikut ini adalah beberapa masalah manajemen I/O yang umum:

1. Penamaan yang seragam (uniform naming): Sistem operasi harus memiliki cara yang seragam untuk menyebut perangkat I/O. Penamaan yang seragam memudahkan pengguna untuk mengakses perangkat I/O dan membantu mengelola perangkat I/O secara efektif.
2. Penanganan kesalahan (error handling): Sistem operasi harus dapat menangani kesalahan yang mungkin terjadi pada perangkat I/O, seperti kesalahan pembacaan atau penulisan data. Ini bisa dilakukan dengan menyediakan mekanisme error-checking dan recovery, atau dengan mengirim perintah ke perangkat I/O untuk mengulangi operasi yang gagal.
3. Transfer sinkron vs asinkron: Sistem operasi harus memutuskan apakah transfer data dengan perangkat I/O harus dilakukan secara sinkron atau asinkron. Transfer sinkron artinya prosesor harus menunggu sampai operasi I/O selesai sebelum melanjutkan ke proses lain, sedangkan transfer asinkron artinya prosesor dapat terus menjalankan proses lain saat operasi I/O sedang dilakukan.
4. Sharable vs dedicated: Sistem operasi harus memutuskan apakah perangkat I/O harus dibagi (sharable) atau khusus (dedicated) untuk suatu proses. Perangkat I/O yang dibagi dapat digunakan oleh lebih dari satu proses, sedangkan perangkat I/O yang khusus hanya dapat digunakan oleh satu proses saja.

Hierarki Manajemen Perangkat I/O

Hierarki manajemen perangkat I/O adalah struktur yang digunakan oleh sistem operasi untuk mengelola perangkat I/O (input/output) di komputer. Hierarki manajemen perangkat I/O terdiri dari beberapa lapisan yang saling terkait, yaitu:

1. Interrupt handler: Interrupt handler adalah lapisan terendah dari hirarki manajemen perangkat I/O. Lapisan ini bertanggung jawab untuk menangani interupsi yang dikirim oleh perangkat I/O saat operasi I/O selesai. Interrupt handler juga bertanggung jawab untuk menangani kesalahan yang mungkin terjadi pada perangkat I/O.
2. Device drivers: Device drivers adalah lapisan yang berada di atas interrupt handler. Lapisan ini bertanggung jawab untuk menyediakan interface antara sistem operasi dan perangkat I/O tertentu. Device drivers memungkinkan sistem operasi untuk melakukan operasi rinci pada perangkat I/O, seperti mengontrol kecepatan, mode, dan sebagainya.

3. Perangkat lunak device independent: Lapisan ini berada di atas device drivers dan merupakan lapisan yang independen terhadap perangkat I/O. Lapisan ini bertanggung jawab untuk menyediakan antarmuka yang seragam kepada perangkat lunak level pemakai, tidak peduli apa perangkat I/O yang digunakan.
4. Perangkat lunak level pemakai: Lapisan ini merupakan lapisan tertinggi dari hirarki manajemen perangkat I/O. Lapisan ini bertanggung jawab untuk menyediakan fitur-fitur yang diinginkan oleh pemakai, seperti antarmuka grafis, perintah-perintah command line, dan sebagainya. Lapisan ini mengandalkan lapisan-lapisan di bawahnya untuk mengakses perangkat I/O dan mengelola tugas-tugas I/O.

Buffering I/O

Buffering I/O adalah teknik yang digunakan untuk menampung sementara data dari atau ke perangkat I/O sebelum diteruskan ke sistem operasi atau pemakai. Buffering bertujuan untuk melembutkan lonjakan-lonjakan kebutuhan akses I/O, sehingga meningkatkan efisiensi dan kinerja sistem operasi. Berikut ini adalah beberapa jenis buffering:

Single Buffering

Single buffering merupakan teknik buffering yang paling sederhana. Pada single buffering, data hanya dapat ditampung oleh satu buffer saja. Ini berarti saat data sedang ditampung oleh buffer, tidak dapat menerima data lain.

Single buffering dapat diterapkan dalam dua mode, yaitu:

1. Mode line at a time, menyimpan satu baris data di bufer sebelum mengirimkannya ke perangkat I/O.
2. Mode byte at a time, menyimpan satu byte data di bufer sebelum mengirimkannya ke perangkat I/O.

Mode line at a time biasanya lebih efisien daripada mode byte at a time karena memungkinkan perangkat I/O untuk memproses data secara utuh sekaligus. Namun, mode line at a time juga membutuhkan lebih banyak memori untuk menyimpan data di bufer. Mode byte at a time lebih sesuai untuk perangkat I/O yang memproses data secara parsial, seperti printer atau pembaca kartu magnetik.

Keunggulan:

3. Implementasinya lebih sederhana dibandingkan dengan teknik buffering lainnya.
4. Membutuhkan ruang memori yang lebih kecil dibandingkan dengan teknik buffering lainnya.
5. Lebih cepat daripada teknik buffering lainnya karena tidak perlu menunggu sampai buffer lain selesai digunakan.

Kelemahan:

1. Tidak dapat menangani operasi I/O yang berjalan paralel.
2. Bisa terjadi bottle neck pada proses I/O jika terdapat banyak proses yang menunggu menggunakan buffer yang sama.
3. Menyebabkan proses lain terganggu jika buffer sedang digunakan untuk operasi I/O yang lama.

Double Buffering

Double buffering adalah teknik buffering yang menggunakan dua buffer. Saat satu buffer sedang dipakai untuk menampung data, buffer yang lain bisa digunakan untuk menerima data baru. Dengan demikian, proses I/O dapat berjalan lebih lancar.

Double buffering memiliki dua mode alternatif yaitu mode line at a time dan mode byte at a time. Pilihan mode tergantung pada jenis perangkat I/O yang digunakan.

1. Mode line at a time: Mode ini digunakan untuk perangkat I/O yang memproses data dalam bentuk baris, seperti printer atau terminal. Dalam mode ini, data ditulis ke buffer sementara baris demi baris selama operasi I/O sedang dilakukan.
2. Mode byte at a time: Mode ini digunakan untuk perangkat I/O yang memproses data dalam bentuk byte, seperti disk atau kartu jaringan. Dalam mode ini, data ditulis ke buffer sementara byte demi byte selama operasi I/O sedang dilakukan.

Keunggulan:

1. Menghindari tearing: Double buffering dapat menghindari tearing, yaitu fenomena yang terjadi saat sebagian gambar sudah muncul di layar tapi sebagian lainnya belum. Tearing dapat terjadi saat proses rendering gambar terjadi secara asinkron dengan pembaruan layar. Dengan double buffering, proses rendering dan pembaruan layar dilakukan secara sinkron, sehingga menghindari tearing.
2. Menghindari flickering: Double buffering juga dapat menghindari flickering, yaitu fenomena yang terjadi saat tampilan layar terus menerus berubah dengan cepat. Flickering dapat terjadi saat proses rendering gambar terjadi secara terlalu cepat dan menyebabkan tampilan layar terus menerus berubah. Dengan double buffering, proses rendering dilakukan di buffer terpisah sebelum ditampilkan di layar, sehingga menghindari flickering.

Kelemahan:

1. Membutuhkan ruang memori ekstra: Double buffering membutuhkan ruang memori ekstra untuk menyimpan buffer yang digunakan dalam proses rendering. Hal ini dapat menyebabkan penggunaan memori yang lebih tinggi dan dapat mengurangi performa sistem pada komputer dengan memori terbatas.
2. Menambah kompleksitas pemrograman: Double buffering juga dapat menambah kompleksitas pemrograman karena harus memperhitungkan buffer yang digunakan dalam proses rendering. Hal ini dapat menyebabkan waktu yang lebih lama dalam pengembangan dan pemeliharaan software yang menggunakan teknik ini.

Circular Buffering

Circular buffering adalah teknik buffering yang menggunakan buffer yang berbentuk lingkaran. Saat buffer penuh, data yang masuk ke buffer akan menggantikan data yang paling lama disimpan di buffer. Dengan demikian, buffer selalu terisi data yang baru-baru saja masuk.

Keunggulan:

1. Efisien dalam penggunaan memori: Circular buffering hanya menggunakan sebagian dari memori yang tersedia sebagai buffer, sehingga menghemat penggunaan memori.

2. Mudah diimplementasikan: Algoritma circular buffering sangat sederhana sehingga mudah diimplementasikan dan dipahami.
3. Cepat dalam menangani interupsi: Saat terjadi interupsi, circular buffering dapat dengan cepat menangani data yang diterima.

Kelemahan:

1. Bisa menyebabkan data hilang: Jika tidak ada perawatan yang tepat, circular buffering dapat menyebabkan data yang sedang diproses hilang.
2. Membutuhkan perawatan tambahan: Circular buffering membutuhkan perawatan tambahan untuk menjaga agar data tidak hilang atau teroverwrite.
3. Sulit diimplementasikan untuk buffer yang besar: Saat buffer menjadi lebih besar, circular buffering menjadi lebih sulit diimplementasikan karena perlu mengelola indeks yang lebih banyak.