

 <p>MODUL 11 CCS 210 SISTEM OPERASI</p>		
Judul	MANAJEMEN I/O	
Penyusun	Distribusi	Perkuliahan
Nixon Erzed	FASILKOM UNIVERSITAS ESA UNGGUL	Pertemuan OL – 12

Materi	<ul style="list-style-type: none"> • Komponen Sistem Operasi untuk sistem I/O • Klasifikasi perangkat I/O • Evolusi fungsi perangkat I/O • Prinsip Manajemen Perangkat I/O • Hirarki manajemen perangkat I/O • Buffering I/O
Tujuan	Mahasiswa mengenal dan memahami pengelolaan media masukan keluaran (I/O) pada sistem operasi

MANAJEMEN 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 :

1. Buffer : menampung sementara data dari/ke perangkat I/O.
2. Spooling : melakukan penjadwalan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
3. Menyediakan “driver” untuk dapat melakukan operasi “rinci” untuk perangkat keras I/O tertentu.
4. Manajemen perangkat masukan/keluaran merupakan aspek perancangan sistem operasi terluas dan kompleks karena sangat beragamnya perangkat dan aplikasinya.

Beberapa fungsi manajemen input/ouput (I/O) :

1. Mengirim perintah ke perangkat I/O agar menyediakan layanan.
2. Menangani interupsi perangkat I/O.
3. Menangani kesalahan perangkat I/O.
4. Menyediakan interface ke pemakai.

KLASIFIKASI PERANGKAT I/O

Perangkat I/O dapat dikelompokkan berdasarkan :

A. Sifat aliran datanya, yang terbagi atas :

1. Perangkat berorientasi blok.

Yaitu menyimpan, menerima, dan mengirim informasi sebagai blok-blok berukuran tetap yang berukuran 128 sampai 1024 byte dan memiliki alamat tersendiri, sehingga memungkinkan membaca atau menulis blok-blok secara independen, yaitu dapat membaca atau menulis sembarang blok tanpa harus melewati blok-blok lain. Contoh : disk,tape,CD ROM, optical disk.

2. Perangkat berorientasi aliran karakter.

Yaitu perangkat yang menerima, dan mengirimkan aliran karakter tanpa membentuk suatu struktur blok. Contoh : terminal, line printer, pita kertas, kartu-kartu berlubang, interface jaringan, mouse.

B. Sasaran komunikasi, yang terbagi atas :

1. Perangkat yang terbaca oleh manusia.

Perangkat yang digunakan untuk berkomunikasi dengan manusia. Contoh : VDT (video display terminal) : monitor, keyboard, mouse.

2. Perangkat yang terbaca oleh mesin

Perangkat yang digunakan untuk berkomunikasi dengan perangkat elektronik. Contoh : Disk dan tape, sensor, controller.

3. Perangkat komunikasi.

Perangkat yang digunakan untuk komunikasi dengan perangkat jarak jauh. Contoh : Modem.

Faktor-faktor yang membedakan antar perangkat :

- Kecepatan transmisi data (data rate).
- Jenis aplikasi yang digunakan.
- Tingkat kerumitan dalam pengendalian.
- Besarnya unit yang ditransfer.
- Representasi atau perwujudan data.
- Kondisi-kondisi kesalahan. Teknik pemrograman perangkat I/O

C. Teknik Pemrograman

Terdapat 3 teknik pemrograman, yaitu :

1. I/O terprogram atau polling system.

Ketika perangkat I/O menangani permintaan, perangkat men-set bit status di register status perangkat. Perangkat tidak memberitahu ke pemroses saat tugas telah selesai dilakukan sehingga pemroses harus selalu memeriksa register tersebut secara periodik dan melakukan tindakan berdasar status yang dibaca. Software pengendali perangkat (driver) dipemroses harus mentransfer data ke/dari pengendali. Driver mengeksekusi perintah yang

berkomunikasi dengan pengendali (adapter) di perangkat dan menunggu sampai operasi yang dilakukan perangkat selesai.

Driver berisi kumpulan instruksi :

- a. Pengendalian.
- b. Berfungsi mengaktifkan perangkat eksternal dan memberitahu yang perlu dilakukan. Contoh : unit tape magnetik diinstruksikan untuk kembali ke posisi awal, bergerak ke record berikut, dan sebagainya.
- c. Pengujian.
- d. Berfungsi memeriksa status perangkat keras berkaitan dengan perangkat I/O.
- e. Pembacaan/penulisan
- f. Berfungsi membaca/menulis untuk transfer data antara register pemroses dan perangkat eksternal. Masalah utama I/O terprogram adalah pemroses diboroskan untuk menunggu dan menjagai operasi I/O. Diperlukan teknik lain untuk meningkatkan efisiensi pemroses.

2. I/O dikendalikan interupsi.

Teknik I/O dituntun interupsi mempunyai mekanisme kerja sebagai berikut :

- a. Pemroses memberi instruksi ke perangkat I/O kemudian melanjutkan melakukan pekerjaan lainnya.
- b. Perangkat I/O akan menginterupsi meminta layanan saat perangkat telah siap bertukar data dengan pemroses.
- c. Saat menerima interupsi perangkat keras (yang memberitahukan bahwa perangkat siap melakukan transfer), pemroses segera mengeksekusi transfer data.

Keunggulan :

Pemroses tidak disibukkan menunggu dan menjaga perangkat I/O untuk memeriksa status perangkat.

Kelemahan :

- Rate transfer I/O dibatasi kecepatan menguji dan melayani operasi perangkat.
- Pemroses terikat ketat dalam mengelola transfer I/O. Sejumlah intruksi harus dieksekusi untuk tiap transfer I/O.

3. Dengan DMA (*direct memory access*).

DMA berfungsi membebaskan pemroses menunggu transfer data yang dilakukan perangkat I/O. Saat pemroses ingin membaca atau menulis data, pemroses memerintahkan DMA controller dengan mengirim informasi berikut:

- Perintah penulisan/pembacaan.
- Alamat perangkat I/O.
- Awal lokasi memori yang ditulis/dibaca.
- Jumlah word (byte) yang ditulis/dibaca.

Setelah mengirim informasi-informasi itu ke DMA controller, pemroses dapat melanjutkan kerja lain. Pemroses mendelegasikan operasi I/O ke DMA. DMA mentransfer seluruh data yang diminta ke/dari memori secara langsung tanpa melewati pemroses. Ketika transfer data selesai, DMA mengirim sinyal interupsi ke pemroses. Sehingga pemroses hanya dilibatkan pada awal dan akhir transfer data. Operasi transfer antara perangkat dan memori utama dilakukan sepenuhnya oleh DMA lepas dari pemroses dan hanya melakukan interupsi bila operasi telah selesai.

Keunggulan

- Penghematan waktu pemroses.
- Peningkatan kinerja I/O.

EVOLUSI FUNGSI PERANGKAT I/O

Sistem komputer mengalami peningkatan kompleksitas dan kecanggihan komponen-komponennya, yang sangat tampak pada fungsi-fungsi I/O sebagai berikut :

1. Pemroses mengendalikan perangkat I/O secara langsung
Masih digunakan sampai saat ini untuk perangkat sederhana yang dikendalikan mikroprosesor sehingga menjadi perangkat berintelijen (intelligent device).
2. Pemroses dilengkapi pengendali I/O (I/O controller).
Pemroses menggunakan I/O terprogram tanpa interupsi, sehingga tak perlu memperhatikan rincian-rincian spesifik antarmuka perangkat.

3. Perangkat dilengkapi fasilitas interupsi.
Pemroses tidak perlu menghabiskan waktu menunggu selesainya operasi I/O, sehingga meningkatkan efisiensi pemroses.
4. I/O controller mengendalikan memori secara langsung lewat DMA.
Pengendali dapat memindahkan blok data ke/dari memori tanpa melibatkan pemroses kecuali diawal dan akhir transfer.
5. Pengendali I/O menjadi pemroses terpisah.
Pemroses pusat mengendalikan/memerintahkan pemroses khusus I/O untuk mengeksekusi program I/O di memori utama. Pemroses I/O mengambil dan mengeksekusi intruksi-intruksi ini tanpa intervensi pemroses pusat. Dimungkinkan pemroses pusat menspesifikasikan barisan aktivitas I/O dan hanya diinterupsi ketika seluruh barisan intruksi diselesaikan.
6. Pengendali I/O mempunyai memori lokal sendiri.
Perangkat I/O dapat dikendalikan dengan keterlibatan pemroses pusat yang minimum.

Arsitektur ini untuk pengendalian komunikasi dengan terminal-terminal interaktif. Pemroses I/O mengambil alih kebanyakan tugas yang melibatkan pengendalian terminal.

Evolusi bertujuan meminimalkan keterlibatan pemroses pusat, sehingga pemroses tidak disibukkan dengan tugas I/O dan dapat meningkatkan kinerja sistem.

PRINSIP MANAJEMEN PERANGKAT I/O

Terdapat dua sasaran perancangan I/O, yaitu :

1. Efisiensi.
Aspek penting karena operasi I/O sering menimbulkan bottleneck.
2. Generalitas (device independence).
Manajemen perangkat I/O selain berkaitan dengan simplisitas dan bebas kesalahan, juga menangani perangkat secara seragam baik dari cara proses memandang maupun cara sistem operasi mengelola perangkat dan operasi I/O.

Software diorganisasikan berlapis. Lapisan bawah berurusan menyembunyikan kerumitan perangkat keras untuk lapisan-lapisan lebih atas. Lapisan lebih atas berurusan memberi antar muka yang bagus, bersih, nyaman dan seragam ke pemakai.

Masalah-masalah manajemen I/O adalah :

1. Penamaan yang seragam (uniform naming)
Nama berkas atau perangkat adalah string atau integer, tidak bergantung pada perangkat sama sekali.
2. Penanganan kesalahan (error handling).
Umumnya penanganan kesalahan ditangani sedekat mungkin dengan perangkat keras.
3. Transfer sinkron vs asinkron
Kebanyakan I/O adalah asinkron. Pemroses mulai transfer dan mengabaikan untuk melakukan kerja lain sampai interupsi tiba. Program pemakai sangat lebih mudah ditulis jika operasi I/O berorientasi blok. Setelah perintah read, program kemudian ditunda secara otomatis sampai data tersedia di buffer.
4. Sharable vs dedicated
Beberapa perangkat dapat dipakai bersama seperti disk, tapi ada juga perangkat yang hanya satu pemakai yang dibolehkan memakai pada satu saat. Contoh : printer.

HIRARKI MANAJEMEN PERANGKAT I/O

Hirarki manajemen perangkat I/O :

1. Interrupt handler.
Interupsi harus disembunyikan agar tidak terlihat rutin berikutnya. Device driver di blocked saat perintah I/O diberikan dan menunggu interupsi. Ketika interupsi terjadi, prosedur penanganan interupsi bekerja agar device driver keluar dari state blocked.
2. Device drivers.
Semua kode bergantung perangkat ditempatkan di device driver. Tiap device driver menangani satu tipe (kelas) perangkat dan bertugas

menerima permintaan abstrak perangkat lunak device independent di atasnya dan melakukan layanan permintaan.

Mekanisme kerja device driver :

- Menerjemahkan perintah abstrak menjadi perintah konkret.
- Setelah ditentukan perintah yang harus diberikan ke pengendali, device driver mulai menulis ke register-register pengendali perangkat.
- Setelah operasi selesai dilakukan perangkat, device driver memeriksa status kesalahan yang terjadi.
- Jika berjalan baik, device driver melewati data ke perangkat lunak device independent.
- Kemudian device driver melaporkan status operasinya ke pemanggil.

3. Perangkat lunak device independent.

Bertujuan membentuk fungsi-fungsi I/O yang berlaku untuk semua perangkat dan memberi antarmuka seragam ke perangkat lunak tingkat pemakai. Fungsi-fungsi lain yang dilakukan :

- Sebagai interface seragam untuk seluruh device driver.
- Penamaan perangkat.
- Proteksi perangkat.
- Memberi ukuran blok perangkat agar bersifat device independent.
- Melakukan buffering.
- Alokasi penyimpanan pada block devices.
- Alokasi dan pelepasan dedicated devices.
- Pelaporan kesalahan.

4. Perangkat lunak level pemakai.

Kebanyakan perangkat lunak I/O terdapat di sistem operasi. Satu bagian kecil berisi pustaka-pustaka yang dikaitkan pada program pemakai dan berjalan diluar kernel. System calls I/O umumnya dibuat sebagai prosedur-prosedur pustaka. Kumpulan prosedur pustaka I/O merupakan bagian sistem I/O. Tidak semua perangkat lunak I/O level pemakai berupa prosedur-prosedur pustaka. Kategori penting adalah sistem spooling. Spooling adalah cara khusus berurusan dengan perangkat I/O yang harus didedikasikan pada sistem multiprogramming.

BUFFERING I/O

Buffering adalah melembutkan lonjakan-lonjakan kebutuhan pengaksesan I/O, sehingga meningkatkan efisiensi dan kinerja sistem operasi. Terdapat beragam cara buffering, antar lain :

1. Single buffering.

Merupakan teknik paling sederhana. Ketika proses memberi perintah untuk perangkat I/O, sistem operasi menyediakan buffer memori utama sistem untuk operasi. Untuk perangkat berorientasi blok. Transfer masukan dibuat ke buffer sistem. Ketika transfer selesai, proses memindahkan blok ke ruang pemakai dan segera meminta blok lain. Teknik ini disebut reading ahead atau anticipated input. Teknik ini dilakukan dengan harapan blok akan segera diperlukan. Untuk banyak tipe komputasi, asumsi ini berlaku. Hanya di akhir pemrosesan maka blok yang dibaca tidak diperlukan.

Keunggulan :

Pendekatan ini umumnya meningkatkan kecepatan dibanding tanpa buffering. Proses pemakai dapat memproses blok data sementara blok berikutnya sedang dibaca. Sistem operasi dapat menswap keluar proses karena operasi masukan berada di memori sistem bukan memori proses pemakai.

Kelemahan :

- Merumitkan sistem operasi karena harus mencatat pemberian buffer-buffer sistem ke proses pemakai.
- Logika swapping juga dipengaruhi. Jika operasi I/O melibatkan disk untuk swapping, maka membuat antrian penulisan ke disk yang sama yang digunakan untuk swap out proses. Untuk menswap proses dan melepas memori utama tidak dapat dimulai sampai operasi I/O selesai, dimana waktu swapping ke disk tidak bagus untuk dilaksanakan
- Buffering keluaran serupa buffering masukan. Ketika data transmisi, data lebih dulu dikopi dari ruang pemakai ke buffer sistem. Proses pengirim menjadi bebas untuk melanjutkan eksekusi berikutnya atau di swap ke disk jika perlu. (untuk perangkat berorientasi aliran karakter).

Single buffering dapat diterapkan dengan dua mode, yaitu :

a. Mode line at a time.

Cocok untuk terminal mode gulung (scroll terminal atau dumb terminal). Masukan pemakai adalah satu baris per waktu dengan enter menandai akhir baris. Keluaran terminal juga serupa, yaitu satu baris per waktu.

Contoh mode ini adalah printer. Buffer digunakan untuk menyimpan satu baris tunggal. Proses pemakai ditunda selama masukan, menunggu kedatangan satu baris seluruhnya. Untuk keluaran, proses pemakai menempatkan satu baris keluaran pada buffer dan melanjutkan pemrosesan. Proses tidak perlu suspend kecuali bila baris kedua dikirim sebelum buffer dikosongkan

b. Mode byte at a time.

Operasi ini cocok untuk terminal mode form, dimana tiap ketikan adalah penting dan untuk peripheral lain seperti sensor dan pengendali.

2. Double buffering.

Peningkatan dapat dibuat dengan dua buffer sistem. Proses dapat ditransfer ke/dari satu buffer sementara sistem operasi mengosongkan (atau mengisi) buffer lain. Teknik ini disebut double buffering atau buffer swapping. Double buffering menjamin proses tidak menunggu operasi I/O. Peningkatan ini harus dibayar dengan peningkatan kompleksitas. Untuk berorientasi aliran karakter.

Double buffering mempunyai 2 mode alternatif, yaitu :

a. Mode line at a time.

Proses pemakai tidak perlu ditunda untuk I/O kecuali proses secepatnya mengosongkan buffer ganda.

b. Mode byte at a time.

Buffer ganda tidak memberi keunggulan berarti atas buffer tunggal. Double buffering mengikuti model producer-consumer.

3. Circular buffering.

Seharusnya melembutkan aliran data antara perangkat I/O dan proses. Jika kinerja proses tertentu menjadi fokus kita, maka kita ingin agar operasi I/O mengikuti proses. Double buffering tidak mencukupi jika proses melakukan operasi I/O yang berturutan dengan cepat. Masalah sering dapat dihindari dengan menggunakan lebih dari dua buffer. Ketika lebih dari dua buffer yang digunakan, kumpulan buffer itu sendiri diacu sebagai circular buffer. Tiap buffer individu adalah satu unit di circular buffer.