

BAB 8

PENGENALAN KONTROL INPUT/OUTPUT

Definisi dan Persyaratan Kontrol I/O

Sebuah sistem kontrol I/O bertujuan untuk memberikan bantuan kepada user untuk memungkinkan mereka mengakses berkas, tanpa memperhatikan detail dari karakteristik dan waktu penyimpanan. Kontrol I/O menyangkut manajemen berkas dan peralatan manajemen yang merupakan bagian dari sistem operasi.

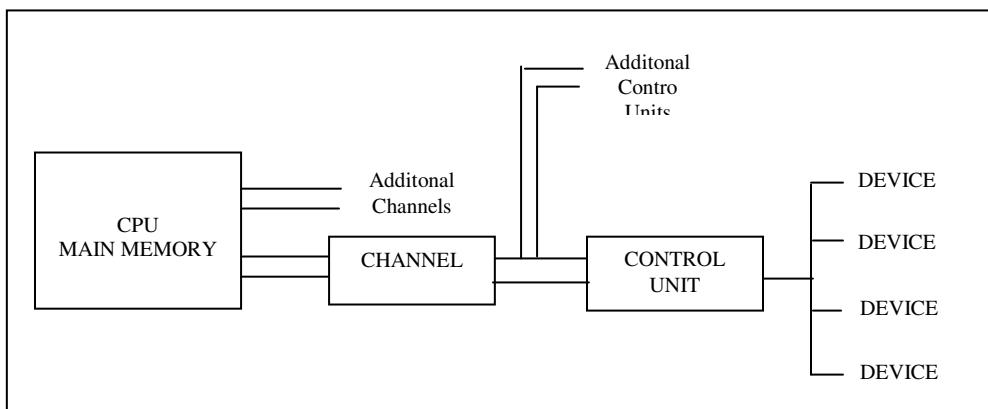
Tugas dari Sistem Kontrol I/O adalah :

- 1) Memelihara direktori dari berkas dan lokasi informasi
- 2) Menentukan jalan bagi aliran data antara main memory dan alat penyimpanan sekunder
- 3) Mengkoordinasi komunikasi antara CPU dan alat penyimpanan sekunder
- 4) Menyiapkan berkas penggunaan input atau output telah selesai

Channel

Pada kebanyakan sistem komputer, CPU tidak dibebani menangani tugas yang berhubungan dengan I/O. Tetapi tanggung jawab untuk kontrol peralatan diserahkan pada prosesor I/O, yang dikenal sebagai **saluran I/O** (I/O channel).

Saluran I/O itu sendiri merupakan prosesor yang sudah diprogram. Program-program yang di-execute ini disebut **channel program**. Channel program ini menentukan operasi, yang diperlukan untuk akses peralatan dan mengontrol jalur data (data pathway).



Macam-macam Channel

1. Selector Channel;

Dapat mengatur aliran data antara memori utama dengan sebuah peralatan pada saat tersebut. Karena saluran merupakan processor-processor yang cepat maka saluran selektor biasanya hanya menggunakan peralatan I/O dengan kecepatan tinggi, seperti **disk**. Penggunaan peralatan dengan kecepatan rendah, misal **card reader**.

2. Multiplexor Channel;

Dapat mengatur aliran data antara memori utama dengan beberapa peralatan. Saluran Multiplexor lebih efektif bila menggunakan peralatan dengan kecepatan rendah, dibandingkan dengan selector channel. Dengan saluran multiplexor, beberapa peralatan dapat diaktifkan secara serentak, tetapi saluran harus melengkapinya saluran program untuk satu peralatan sebelum memulai dengan saluran program lain.

3. Block Multiplexor Channel;

Mengatur aliran data ke berbagai peralatan. Block Multiplexor Channel dapat mengeksekusi satu instruksi dari saluran program untuk satu peralatan, kemudian dapat mengalihkan instruksi-instruksi dari saluran program itu ke peralatan yang lain.

Macam-macam Device

1. Dedicated Device;

Digunakan untuk pengaksesan oleh satu orang pada setiap saat.

Contoh : Terminal.

2. Shared Device;

Digunakan untuk pengaksesan oleh banyak pemakai secara bersamaan.

Contoh : Disk.

Aktivitas I/O untuk shared device adalah sangat kompleks dibanding aktivitas I/O pada dedicated device. Dua fungsi yang sangat penting dari shared device adalah **alokasi tempat** dan **pemberian akses yang tepat**.

Aktifitas Saluran

Tujuan dari saluran I/O adalah sebagai perantara antara CPU-main memory dengan unit pengontrol penyimpanan. CPU berkomunikasi dengan saluran melalui beberapa perintah yang sederhana.

Beberapa saluran akan memberi perintah :

- ❑ Test I/O, untuk menentukan apakah jalur (pathway) yang menuju peralatan sedang sibuk.
- ❑ Start I/O, pada peralatan tertentu.
- ❑ Halt I/O, pada peralatan tertentu.

Saluran biasanya berkomunikasi dengan CPU melalui cara **interupsi**. Interupsi akan terjadi, jika keadaan error terdeteksi, misalnya instruksi CPU yang salah atau jika aktifitas I/O telah diakhiri.

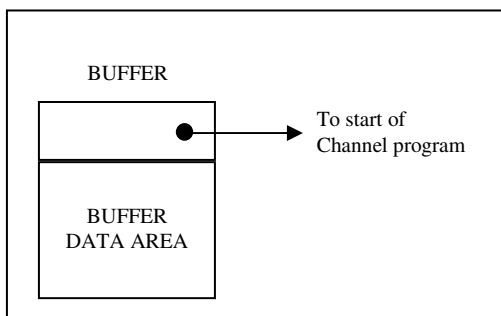
Jika interupsi terjadi, kontrol akan bercabang melalui rutin pengendali interupsi (interrupt-handler routine), dimana kontrol akan menentukan penyebab dari interupsi, melakukan kegiatan yang tepat, kemudian mengembalikan kontrol pada pemanggil (caller).

Manajemen Buffer

Terbagi menjadi 4 jenis manajemen yaitu :

1. Single Buffering;
2. Anticipatory Buffering;
3. Double Buffering;
4. Three Buffers;

1. Single Buffering;



Gambar di atas menunjukkan struktur data dari buffer dalam bentuk yang sederhana, yang terdiri dari satu record per-block dan satu buffer per-berkas, dimana buffer ini berfungsi mengisikan permintaan dari sebuah program. Struktur buffer ini berisi sebuah pointer pada alamat awal & channel program untuk berkas.

Struktur dasar dari channel program untuk mengisi buffer adalah :

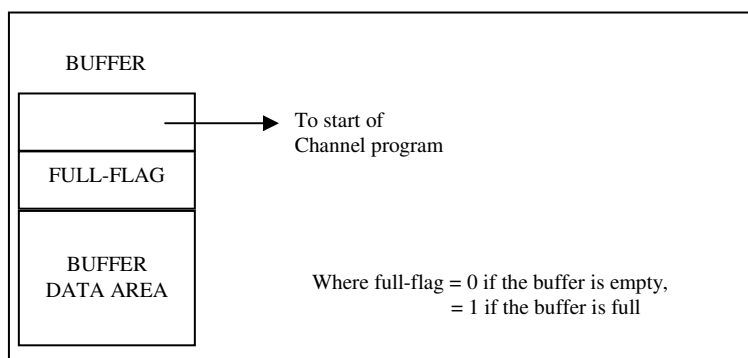
- Tunggu instruksi READ dari program
 - Memberitahukan instruksi start I/O ke unit kontrol
 - Tunggu hingga buffer dikosongkan
 - Memberitahukan interupsi pada program sehingga dapat mulai membaca dari buffer
- Masalah yang timbul di sini adalah pemakai program menganggur pada saat menunggu buffer diisi.

2. Anticipatory Buffering;

Pendekatan lain yang dapat menghilangkan beberapa hal yang mungkin untuk menunggu CPU adalah dengan menggunakan Anticipatory Buffering.

Dengan anticipatory buffering, sistem kontrol I/O akan berusaha mendahulukan kebutuhan program akan data. Diusahakan agar buffer selalu penuh. Channel selalu menguji flag ini. Jika buffer mendekati kosong, karena pemakai program telah membaca isinya maka flag itu akan direset dan channel program akan menginitates pengisian kembali buffer.

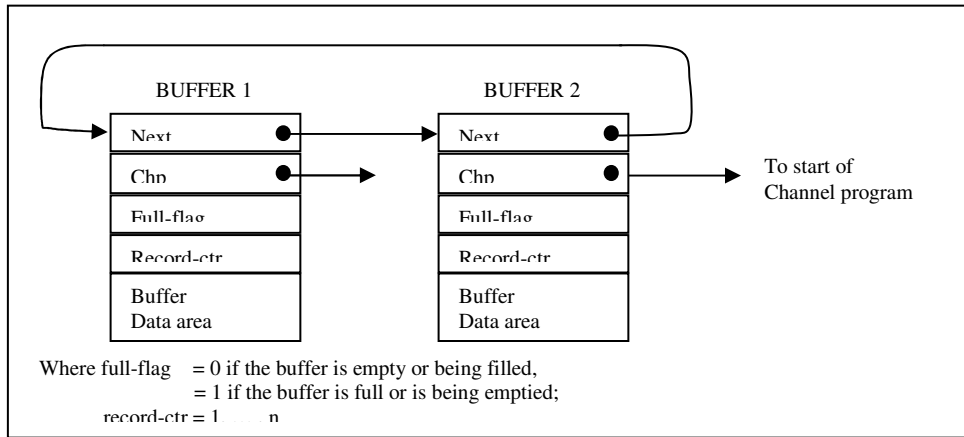
Struktur dasar channel program untuk mengisi sebuah buffer dengan anticipatory buffer diperlihatkan pada gambar di bawah ini :



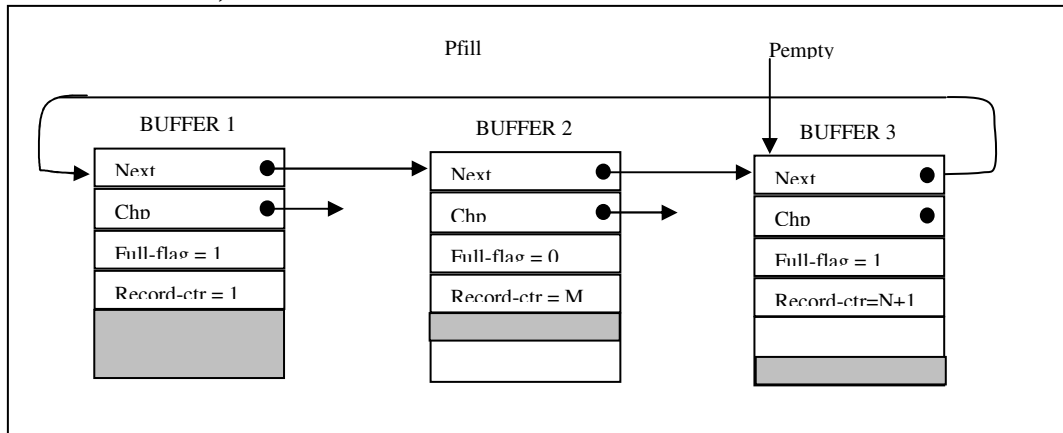
3. Double Buffering;

Untuk mengurangi kemungkinan dari program menunggu, maka double buffer dapat digunakan. Dua dari tempat buffer yang ada, hanya satu yang ditetapkan untuk berkas.

Ide dasar dari double buffering adalah jika consumer mengosongkan salah satu buffer, maka producer dapat mengisi ke dalam buffer yang lain, pada saat buffer pertama sudah kosong, maka buffer yang kedua harus dalam keadaan penuh. Kemudian consumer dapat mengosongkan buffer yang kedua, pada saat producer mengisi buffer yang pertama, demikian seterusnya.



4. Three Buffers;



pfill : yang menunjukkan buffer berikutnya akan diisi atau sedang diisi

pempty : yang menunjukkan buffer berikutnya akan dikosongkan atau sedang dikosongkan