**IO FILE**

Sistem input/output (I/O)

I/O Sistem Operasi

I/O System merupakan bagian untuk menangani inputan dan outputan dari DCS. Inputan dan outputan tersebut bisa analog atau digital. Inputan/outputan digital seperti sinyal-sinyal ON/OFF atau Start/Stop. Kebanyakan dari pengukuran proses dan outputan terkontrol merupakan jenis analog.

Pengertian Input  
Input adalah semua data dan perintah yang dimasukkan ke dalam memori komputer untuk selanjutnya diproses lebih lanjut oleh prosesor. Sebuah perangkat input adalah komponen piranti keras yang memungkinkan user atau pengguna memasukkan data ke dalam komputer, atau bisa juga disebut sebagai unit luar yang digunakan untuk memasukkan data dari luar ke dalam mikroprosesor.

Pengertian Output  
Output adalah data yang telah diproses menjadi bentuk yang dapat digunakan. Artinya komputer memproses data-data yang diinputkan menjadi sebuah informasi. Yang disebut sebagai perangkat output adalah semua komponen piranti keras yang menyampaikan informasi kepada orang-orang yang menggunakannya.

I/O system terdiri dari beberapa bagain penting yaitu:

a.    I/O Hardware

b.    Application I/O Interface

c.    Kernel I/O Subsystem

d.    I/O Requests to Hardware Operations

e.    Streams

f.    Performance

A.I/O Hardware

   Secara umum, I/O Hardware terdapat beberapa jenis seperti device penyimpanan  
   (disk,tape),

   transmission device (network card, modem), dan human-interface device (screen, keyboard,mouse). Device tersebut dikendalikan oleh instruksi I/O. Alamat-alamat yang dimiliki device akan digunakan oleh direct I/O instruction dan memory-mapped I/O. Beberapa konsep yang umum digunakan ialah port, bus (daisy chain/ shared direct access), dan controller  (host adapter).

* Port adalah koneksi yang digunakan oleh device untuk berkomunikasi dengan mesin.
* Bus adalah koneksi yang menghubungkan beberapa device menggunakan kabel-kabel.
* Controller adalah alat-alat elektronik yang berfungsi untuk mengoperasikan port, bus, dan device.

B.Application I/O Interface

   Merupakan suatu mekanisme untuk mempermudah pengaksesan, sehingga sistem operasi melakukan standarisasi cara pengaksesan peralatan I/O.

   Interface aplikasi I/O melibatkan abstraksi, enkapsulasi, dan software layering. Device driver mengenkapsulasi tiap-tiap peralatan I/O ke dalam masing-masing 1 kelas yang umum (interface standar). Tujuan dari adanya lapisan device driver ini adalah untuk menyembunyikan  perbedaan-perbedaan yang ada pada device controller dari subsistem I/O pada  kernel. Karena hal ini, subsistem I/O dapat bersifat independen dari hardware. Beberapa hal yang berhubungan dengan Application I/O Interface adalah:

1.Peralatan Block dan Karakter:

* Perangkat Block termasuk disk drive
* Perintah termasuk baca, tulis dan cari
* Raw I/O atau akses file-sistem
* Pemetaan memori untuk pengaksesan file
* Perangkat karakter termasuk keyboad, mouse dan serial port
* Perintahnya seperti get, put
* Library layered  dalam proses pengeditan

2.Peralatan Jaringan

   Adanya perbedaan pengalamatan dari jaringan I/O, maka sistem operasi memiliki interface I/O yang berbeda dari baca, tulis dan pencarian pada disk. Salah satu yang banyak digunakan pada sistem operasi adalah interface socket. Socket berfungsi untuk menghubungkan komputer ke jaringan. System call pada socket interface dapat memudahkan suatu aplikasi untuk membuat local  socket, dan menghubungkannya ke remote socket.Dengan menghubungkan komputer ke socket, maka komunikasi antar komputer dapat dilakukan.

3.Jam dan Timer

   Jam dan timer pada hardware komputer, memiliki tiga fungsi :

* memberi informasi waktu saat ini
* memberi informasi lamanya waktu sebuah proses
* sebagai trigger untuk suatu operasi pada suatu waktu.

   Fungsi ini sering digunakan oleh sistem operasi. Akan tetapi, system call untuk pemanggilan fungsi  ini tidak di-standarisasi antar sistem operasi. Hardware yang mengukur waktu dan melakukan operasi trigger dinamakan programmable interval timer yang dapat di set untuk menunggu waktu tertentu dan kemudian melakukan interupsi.

C.Kernel I/O subsystems  
   Kernel menyediakan banyak service yang berhubungan dengan I/O. Pada bagian ini, kita akan mendeskripsikan beberapa service yang disediakan oleh kernel I/O subsystem, dan kita akan membahas bagaimana caranya membuat infrastruktur hardware dan device-driver. Service yang akan kita bahas adalah I/O scheduling, buffering, caching, pooling, reservasi device, error handling.

1.I/O Scheduling  
   Untuk menjadualkan sebuah set permintaan I/O, kita harus menetukan urutan yang bagus untuk mengeksekusi permintaan tersebut. Scheduling dapat meningkatkan kemampuan sistem secara keseluruhan, dapat membagi device secara rata di antara proses-proses, dan dapat mengurangi waktu tunggu rata-rata untuk menyelesaikan I/O.

2. Buffering  
   Buffer adalah area memori yang menyimpan data ketika mereka sedang dipindahkan antara dua device atau antara device dan aplikasi. Buffering dilakukan untuk tiga buah alasan. Alasan pertama adalah untuk men-cope dengan kesalahan yang terjadi karena perbedaan kecepatan antara produsen dengan konsumen dari sebuah stream data.

3. Caching  
   Sebuah cache adalah daerah memori yang cepat yang berisikan data kopian. Akses ke sebuah kopian yang di-cached lebih efisien daripada akses ke data asli. Sebagai contoh, instruksi-instruksi dari proses yang sedang dijalankan disimpan ke dalam disk, dan ter-cached di dalam memori physical, dan kemudian dicopy lagi ke dalam cache secondary and primary dari CPU. Perbedaan antara sebuah buffer dan ache adalah buffer dapat menyimpan satu-satunya informasi datanya sedangkan sebuah cache secara definisi hanya menyimpan sebuah data dari sebuah tempat untuk dapat diakses lebih cepat. Caching dan buffering adalah dua fungsi yang berbeda, tetapi terkadang sebuah daerah memori dapat digunakan untuk keduanya.

4.Spooling dan Reservasi Device  
   Sebuah spool adalah sebuah buffer yang menyimpan output untuk sebuah device, seperti printer, yang tidak dapat menerima interleaved data streams. Walau pun printer hanya dapat melayani satu pekerjaan pada waktu yang sama, beberapa aplikasi dapat meminta printer untuk mencetak, tanpa harus mendapatkan hasil output mereka tercetak secara bercampur. Sistem operasi akan menyelesaikan masalah ini dengan meng-intercept semua output kepada printer. Tiap output aplikasi sudah di-spooled ke disk file yang berbeda. Ketika sebuah aplikasi selesai mengeprint, sistem spooling akan melanjutkan ke antrian berikutnya. Di dalam beberapa sistem operasi, spooling ditangani oleh sebuah sistem proses daemon.

5.Error Handling  
   Sebuah sistem operasi yang menggunakan protected memory dapat menjaga banyak kemungkinan error akibat hardware mau pun aplikasi. Devices dan transfer I/O dapat gagal dalam banyak cara, bisa karena alasan transient, seperti overloaded pada network, mau pun alasan permanen yang seperti kerusakan yang terjadi pada disk controller. Sistem operasi seringkali dapat mengkompensasikan untuk kesalahan transient. Seperti, sebuah kesalahan baca pada disk akan mengakibatkan pembacaan ulang kembali dan sebuah kesalahan pengiriman pada network akan mengakibatkan pengiriman ulang apabila protokolnya diketahui. Akan tetapi untuk kesalahan permanent, sistem operasi pada umumnya tidak akan bisa mengembalikan situasi seperti semula.

6.Kernel Data Structure  
   Kernel membutuhkan informasi state tentang penggunakan komponen I/O. Kernel menggunakan banyak struktur yang mirip untuk melacak koneksi jaringan, komunikasi karakter-device, dan aktivitas I/O lainnya. UNIX menyediakan akses sistem file untuk beberapa entiti, seperti file user, raw devices, dan alamat tempat proses. Walau pun tiap entiti ini didukung sebuah operasi baca, semantics-nya berbeda untuk tiap entiti. Seperti untuk membaca file user, kernel perlu memeriksa buffer cache sebelum memutuskan apakah akan melaksanakan I/O disk. Untuk membaca sebuah raw disk, kernel perlu untuk memastikan bahwa ukuran permintaan adalah kelipatan dari ukuran sektor disk, dan masih terdapat di dalam batas sektor. Untuk memproses citra, cukup perlu untuk mengkopi data ke dalam memori. UNIX mengkapsulasikan perbedaan-perbedaan ini di dalam struktur yang uniform dengan menggunakan teknik object oriented.Beberapa sistem operasi bahkan menggunakan metode object oriented secara lebih extensif.

D.I/O Requests to Hardware Operations  
   Salah satu contohnya adalah:

a. Ilustrasi membuka sebuah  
–  Device mana tempat file yang akan dibuka  
–  Menerjemahkan \_nama\_ ke dalam device yang dimaksud   
–  Membaca secara fisik file yang hendak dibuka   
–  Data sudah siap untuk diakses   
–  Mengembalikan kontrol pada proses   
b.Ilustrasi lain pada waktu boot   
   Sistem mula-mula meminta bus piranti keras untuk menentukan device apa yang ada.

 E.Streams  
   I/O stream adalah suatu mekanisme pengiriman data secara bertahap dan terus menerus melalui suatu aliran data (dua arah).Biasa digunakan dalam network protocol dan menggunakan  message passingdalam men-transferdata Stream terdiri atas :

* sebuah stream head yang merupakan antarmuka dengan user process,
* sebuah driver end yang mengontrol device,
* dan nol atau lebih stream modules

F.Performance  
   Faktor utama dalam performa sistem :

* Permintaan CPU untuk menjalankan device driver, kode kernel I/O
* Keadaan/state untuk melayani interrupt
* Copy data
* Network traffic khususnya pada beban kinerja

Improving Perfomance:

* Menurunkan jumlah alih konteks.
* Mengurangi jumlah pengkopian data ke memori ketika sedang dikirimkan antara device dan aplikasi.
* Mengurangi frekuensi interupsi, dengan menggunakan ukuran transfer yang besar, smart controller, dan polling.
* Meningkatkan concurrency dengan controller atau channel yang mendukung DMA.
* Memindahkan kegiatan processing ke perangkat keras, sehingga operasi kepada device controller dapat berlangsung bersamaan dengan CPU.
* Menyeimbangkan antara kinerja CPU, memory subsystem, bus, dan I/O.