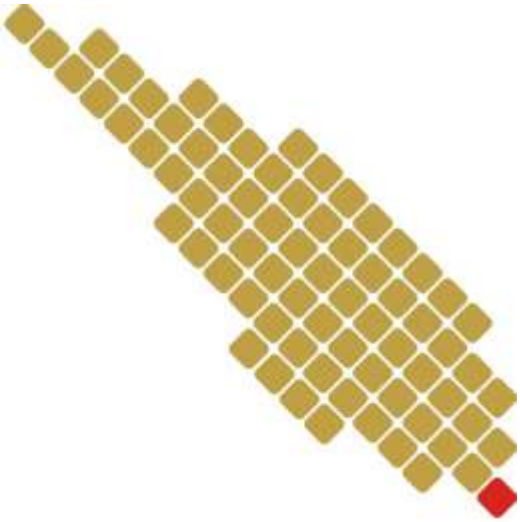
TUGAS SISTEM OPERASI



DISUSUN OLEH :

Tri Aji Bagaskara | 119140214

KELAS :

RC

INSTITUT TEKNOLOGI SUMATERA

JURUSAN TEKNIK ELEKTRO, INFORMATIKA, DAN SISTEM FISIKA

PROGRAM STUDI TEKNIK INFORMATIKA

TAHUN 2021

Materi-Materi Sistem Operasi :

- CHAPTER 6 : SINKRONISASI PROSES
- CHAPTER 7 : MAIN MEMORY
- CHAPTER 8 : MEMORI VIRTUAL
- CHAPTER 9 : MANAJEMEN DISK
- CHAPTER 10 : PERANGKAT KERAS I/O
- CHAPTER 11 : MANAJEMEN SISTEM FILE

1. CHAPTER 6 : SINKRONISASI PROSES

Pada tujuan perkuliahaan pada materi ke-6 memperkenalkan critical section problem,dimana solusinya dapat digunakan untuk memastikan konsisten dari shared data,memperkenalkan solusi software dan hardware atas permasalahan critical section

Proses yang bersifat simultan (concurrent) yang dijalankan pada sistem operasi dapat dibedakan menjadi proses independen dan proses kooperatif.Suatu proses dikatakan independen/kooperatif apabila proses tersebut tidak dapat terpengaruh atau dipengaruhi oleh proses lain yang sedang dijalankan pada sistem.

Semua proses yang tidak membagi data apa pun (baik sementara/ tetap) dengan proses lain adalah independent.Proses kooperatif adalah proses yang dapat dipengaruhi atau pun terpengaruhi oleh proses lain yang sedang dijalankan dalam sistem. Dengan kata lain, proses dikatakan kooperatif bila proses dapat membagi datanya dengan proses lain.

Alasan untuk penyedian sebuah lingkungan yang memperbolehkan terjadinya proses kooperatif yaitu pembagian informasi dan kecepatan perhitungan/komputasi.Kerjasama antar proses membutuhkan suatu mekanisme yang memperbolehkan proses-proses untuk mengkomunikasi data dengan yang lain dan mengsyncronize kerja mereka sehingga tifak ada yang saling menghalangi.Pemeliharaan konsisten data membutuhkan sebuah mekanisme yang memastika eksekusi berurutan atas proses-proses yang seling berhubungan.

Ada dua model yang fundamental dalam hubungan antar proses yaitu :

- Shared memori Dalam model ini proses saling berbagi memori.Untuk menjaga konsitensi data,perlu diatur proses mana yang dapat mengakses memori pada suatu waktu.
- Massange Passing.Pada model ini proses berkomunikasi lewat saling mengirimkan pesan.

Pada contoh kita hendak menyediakan solusi atas permasalahan consumer-producer yang menempati seluruh buffer.Kita dapat melakukan dengan mengeset bilangan integer yang melakukan tracking seluruh buffer secara penuh.Awalnya bilangan integer di set-0 kemudian dilakukan increment oleh producer setelah memproduksi buffer baru dan dilakukan decrement oleh consumer setelah mengkonsumsi buffer.

Langkah-Langkah Sinkronisasi

- Producer

```
while (true) {

        /* produce an item and put in nextProduced */
         while (counter == BUFFER_SIZE)
                ; // do nothing
              buffer [in] = nextProduced;
              in = (in + 1) % BUFFER_SIZE;
              counter++;
}
```

- Consumer

```
while (true) {
      while (counter == 0)
            ; // do nothing
            nextConsumed = buffer[out];
            out = (out + 1) % BUFFER_SIZE;
            counter--;

              /* consume the item in nextConsumed
}
```

- Race Condition

counter++ dapat diimplementasikan sebagai

```
register1 = counter
register1 = register1 + 1
counter = register1
```

counter-- dapat diimplementasikan sebagai

```
register2 = counter
register2 = register2 - 1
count = register2
```

Perhatikan ketika eksekusi ini pada count = 5:

S0: producer execute register1 = counter {register1 = 5}
S1: producer execute register1 = register1 + 1 {register1 = 6}
S2: consumer execute register2 = counter {register2 = 5}
S3: consumer execute register2 = register2 - 1 {register2 = 4}
S4: producer execute counter = register1 {count = 6 }
S5: consumer execute counter = register2 {count = 4}

Race Condition adalah proses-proses berbagi data,yang dimana bahwa data dibagi secara Bersama yang mengalami tidak konsisten dikarenakan adanya kemungkinan proses tersebut melakukan akses secara bersamaan yang menyebabkan data tersebut berubah.

Untuk mencegahnya dengan membutuhkan **mutual exclusion**,untuk sebuah jalan yang menjamin jika sebuah proses sedang menggunaan variable atau berkas yang digunakan Bersama sama,proses lain akan dikeluarkan dari pekerjaan yang sama

Untuk menghindar dari race condition adalah dengan cara menjamin jikas suatu proses sedang menjalankan critical section,maka proses lain tidak boleh masuk kedalam critical section tersebut.

❖ Critical Section

Critical Section adalah segmen kode yang mengakses data yang digunakan proses secara Bersama-sama yang dapat membawa proses tersebut ke bahaya race condition.

Structure umum dari proses pi adalah

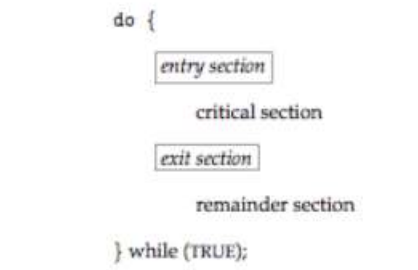
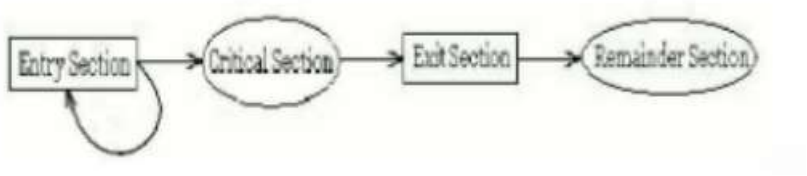


Figure 6.1 General structure of a typical process P_i.

Permasalahan Critical Section :

- Setiap proses memiliki critical section kode segmen
  - Proses dapat mengubah variable,update tabel,menulis file,etc
  - Ketika satu proses dalam critical section yang lain tidak mungkin berada dalam kondisi yang sama.
- Permasalahan critical section membutuhkan sebuah protocol untuk penyelasaiannya
- Setiap proses harus meminta izin untuk memasuki critical section dalam **entry section,**kemudian mengikuti critical section dengan **exit section** kemudian **remaider section**

Ilustrasi critical section



Solusi permasalahan critical section

- Mutual Exclusion : jika proses pi dieksekusi pada kondisi critical section,maka tidak ada proses lain yang dapat dieksekusi pada kondisi critical section
- Progress : Jika tidak ada proses yang dieksekusi pada kondisi critical section dan ada beberapa proses yang hendak memasuki kondisi critical section,maka pemilihan proses yang akan memasuki kondisi critical section,berikutnya tidak dapat ditunda tanpa batas waktu.
- Bounded Waiting : Sebuah pembatas harus ada diatara waktu-waktu dimana proses yang lai diizinkan untuk memasukin kondisi critical section setelah proses mengirimkan request untuk memasukin kondisi critical section dan sebelum request tersebut dikabulkan.

❖ Semaphore

Tool untuk sinkronisasi.Nilai awal dari semaphore menunjukan beberapa banyak proses yang boleh memasuki critical section dari suatu program,biasanaya untuk mendukung sifat mutual exclusive nilai ini diberi 1

- Semaphore sebagai tool sinkronisasi umum
  - Counting Semaphore adalah Nilai integer memiliki kisaran pada domai yang tak terbatas
  - Binary Semaphore adalah Nilai integer memiliki kisaran antara 0 dan 1;Implementasi lebih sederhana,sering disebut dengan mutex locks
  - Menyediakan Mutual exclusion
  - Nilai Semphore diset menjadi 0
  - Jika proses berjalan lebih cepat,maka outputan/keluaran menghasilkan “satu” kemudian diikuti oleh “dua”,dikarenakan jika proses 2 berjalan terlebih dahulu,maka proses tersebut akan menunggu (nilai semaphore=0)sampai proses 1 memanggil signal.
  - Jika proses 1 berjalan terlebih dahulu,maka proses tersebut akan memanggil signal untuk memberikan jalan terlebih dahulu kepada proses 2.

❖ Implementasi Semaphore

Pada Implementasi semaphore bahwa tidak ada dua proses yang dapat mengeksekusi wait () dan signal () pada semaphore dan waktu yang sama.Karena Implementasi menjadi permasalahan critical section dimana kode wait dan signal yang terletak pada kondisi critical section.

Didaalam implementasi critical section dapat memiliki busy waiting.Busy waiting akan terjadi sedikit jika critical section jarang ditempati.Dan busy waiting memiliki kode

implementasi yang pendek.Pada aplikasi banyak menghabiskan waktu dalam critical section dan kondisi ini bukan merupakan solusi yang baik.

Cara proses menunggu dapat dibagi menjadi dua : **Tipe Spinlock** dan **Tipe Non-Spinlock**

- **Tipe Spinlock**

  Proses melakukan iterasi (looping) secara terus menerus tanpa menjalankan perintah apapun.Proses terus berada dirunning state.Tipe Spinlock yang biasa disebut busy waiting.Spinlock waiting berarti proses tersebut menunggu dengan cara menajalankan perintah perintah yang tidak ada,dengan kata lain proses tersebut masih running state didalam spinlock waiting.

- **Tipe Non-Spinlock**

  Memblock dirinya sendiri dan secara otomatis masuk kedalam waiting quee.Pada waiting quee proses tidak aktif dan menunggu sampai ada proses lain yang membagunkannya dan membawa ready quee

❖ Implementasi Semaphore tanpa busy waiting

Pada setiap semaphore terasosiasi dengan antrian waiting,setiap entri dalam antrian waiting memiliki dua data yaitu : nilai,pointer ke enri berikutnya dalam list.
Memiliki 2 operasi yaitu

- Block : Menempatkan proses untuk memanggil operasi pada antrian waiting yang sesuai
- Wakeup : Menghapus salah satu proses dalam antrian waiting dan ditempatkan dalam antrian ready.

❖ Deadlock dan Starvation

- Deadlock adalah Proses yang memilih dua ata lebih yang menunggu tanpa Batasan waktu sebuah event sebenarnya disebabkan oleh perilaku salah satu proses tersebut.
- Starvation adalah pemblokiran tanpa batas waktu.Dimana proses yang tidak di hapus dari antrian semaphore maka di tangguhkan.
- Priorty Inversion adalah permasalahan dalam penjadwalan kerika proses dengan prioritas lebih rendah memegang kunci yang dibutuhkan oleh proses dengan prioritas lebih tinggi.

❖ Permasalahan Klasik Sinkronisasi

Permasalahan ini digunakan untuk mengetes skema sinkronisasi baru yang diusulkan yang terdiri dari :

- Bounded-Buffer Problem
- Readers and writers problem
- Dinning Philosophers problem

. Bounded Buffer problem memiliki contoh dari penggunaannya yaitu pada proses produsen-konsumen.Produsen terebut akan menghasilkan suatu barang,produsen akan menaruh barang itu di bounded buffer.setelah itu konsumen akan mengkonsumsi barang yang dihasilkan oleh produsen dan jika konsumen membutuhkan suatu barang maka diaakam mengambil dari bounded buffer

❖ Permasalahan Bounded-Buffer

- Semaphore mutex diinisiasikan dengan nilai 1

- Semaphore full diinisiasikan dengan nilai 0
- Semaphore empty diinisiasikan dengan nilai N

Mutex digunakan untuk menjamin hanya ada satu proses yang boleh berjalan mengakses buffer pada suatu waktu. Empty digunakan untuk menghitung jumlah buffer yang kosong.

Struktur dari proses procedur

```
do {

        // produce an item in nextp

    wait (empty);
    wait (mutex);

        // add the item to the buffer

    signal (mutex);
    signal (full);
} while (TRUE);
```

Struktur dari proses consumer

```
do {
    wait (full);
    wait (mutex);

        // remove an item from buffer to nextc

    signal (mutex);
    signal (empty);

        // consume the item in nextc

} while (TRUE);
```

❖ Permasalahan Readers-Writers
  - Readers :hanya membaca dataset,tidak melakukan update
  - Writers : dapat membaca dan menulis

  Permasalahan memungkinkan beberapa readers membaca dalam satu waktu

  Hanya satu writer yang dapat mengakses shared data pada satu waktu.

Shared Data :

- Dataset
- Semaphore mutex diinisiasi dengan nilai 1
- Semaphore wrt diinisiasi dengan nilai 1
- Integer readcount diinisiasi dengan nilai 0

Struktur dari proses writer

```
do {
        wait (wrt) ;

           //    writing is performed

        signal (wrt) ;
} while (TRUE);
```

Struktur dari proses reader

```
do {
            wait (mutex) ;
            readcount ++ ;
            if (readcount == 1)
                        wait (wrt) ;
            signal (mutex)

                  // reading is performed

             wait (mutex) ;
             readcount  - - ;
             if (readcount  == 0)
                        signal (wrt) ;
             signal (mutex) ;
      } while (TRUE);
```

Pada proses readers dan writers harus memiliki kondsi yang dipenuhi :

- Hanya ada satu writer yang mengubah data,pada saat writer mengubah data maka tidak boleh reader yang diperbolehkan untuk membaca data.
- Diperbolehkan lebih dari satu reader membaca data .Ketika sedang dibaca,maka tidak ada writer yang boleh mengubah data

Untuk memecahkan masalah readers dan writers harus memenuhi kondisi :

- Reader yang baru mendapatkan prioritas dari pada writers yang sedang menunggu

- Jika lebih dari satu reader didalam critical section,maka reader diijinkan untuk memasuki critical section.
- Jika writers sedang menunggu dapat mengakses data, tetapi tidak ada writers tersebut berada didalam sistem.
- Jika writers sedang menjalankan sistem,maka semua readers yang sedang menunggu lebih mempunyai prioritas untuk dijalankan ketimbang writers yang sedang mengantri.
- Jika readers dating terus menerus makan writers tidak akan mendapatkan giliran untuk mengakses data (starvation).

❖ Permasalahan Dining-Philosophers

Filshuf mencoba mengambil sumpit dengan melakukan operasi wait () pada semaphore tersebut,lalu melepaskan sumpit dengan menjalankan operasi signal pad semaphore yang sesuai.

Struktur dari filshuf

```
do {
        wait ( chopstick[i] );
       wait ( chopStick[ (i + 1) % 5] );

            // eat

        signal ( chopstick[i] );
        signal (chopstick[ (i + 1) % 5] );

            // think

} while (TRUE);
```

❖ Permasalahan Deadlock

Sekumpulan Proses yang saling memblockir,masing-masing memegang resource dan menunggu resource yang dipegang oleh proses lain dalam satu data set.

❖ Karakteristik Deadlock

- Mutual Exclussion :hanya satu proses yang dapat menggunakan resource pada satu waktu
- Hold and Wait : proses yang memegang satu resource menunggu untuk memperoleh resource yang dipegang proses lain
- No preemption :Resouce hanya dapat dilepas secara sukarela oleh proses setelah seluruh tugasnya selesai
- Circular wait :proses proses yang saling menunggu diana P0 menunggu resource yang dipengang oleh p1.

❖ Resource Allocation Graph

Sumberdaya akan digunakan oleh proses proses yang membutuhkan nya.mekanisme hubungan dari proses-proses dan sumber daya yang dibutuhkan/digunakan dapat diwakili oleh graph

❖ Penghindaran Deadlock

- Model yang paling sederhana dan paling berguna mensyaratkan setiap proses mendeklarasian maximum number dari setiap tipe resource yang dibutuhkan
- Mengecek secara dinamis status alokasi resource untuk memastikan bahwa kondisi circular wait tidak terjadi
- Status alokasi resource didefinisikan dengan jumlah resource tersedia dan resource yang sudah dialokasikan,serta permintaan maksimum dari seluruh proses.

2. CHAPTER 7 : MAIN MEMORY

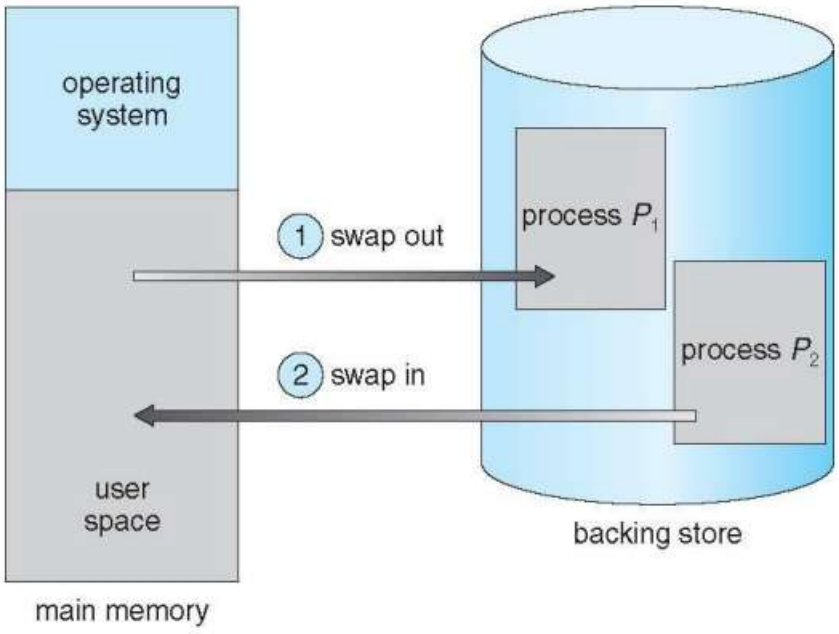
- Register Basis dan Batas
  Sepasang register basis dan batas menentukan ruang alamat.Cpu harus memerikasa setiap akses memory yang dihasilkan dala mode pengguna untu memastikan antara basis dan batas dari pengguna.
- Alamat Binding
  Sebuah program ditempatkan dalam disk dalam bentuk berkas biner yang dapat dieksekusi.Sebelum dieksekusi program harus ditempatkan di memory terlebih dahulu.Pada kumpulan proses yang ada pada didisk harus menunggu dalam antrian sebelum dibawa memori dan di eksekusi.
  Alamat Binding adalah Pemetaan alamat suatu variable program ke alamat memori tertentu yang dapat terjadi pada saat :
  - Complication Time
  - Load Time
  - Execution Time
- Ruang Alamat Logika dan Fisik
  - Alamat Logika adalah alamat yang dihasilkan oleh CPU,disebut juga alamat virtual.
  - Alamat Fisik adalah alamat memori yang sebenarnya
  - Ruang Alamat virtual adalah kumpulan alamat virtual yang dibuat oleh CPU
  - Ruang Alamat fisik adalah kumpulan alamat difisik yang berkorespodensi dengan alamat virtual.
- Memory-Management unit (MMU)
  Memory management unit (MMU) yang bertujuan untuk mengubah alamat virtual ke alamat fiisik.
  Pada MMU nilai pada regiter relokasi akan ditambahkan setiap alamat logika yang dihasilkan oleh CPU dan pada waktu yang sama alamat ini dikirimkan ke memori
- Relokasi Dinamis menggunakan register relokasi
  Physical address = logical address +relocation register
- Pemanggilan Dinamis
  Untuk mendapatkan utilisasi ruang memori yang baik (pemanfaatan ruang memori yang lebih baik), kita melakukan pemanggilan dinamis, sebuah rutin tidak akan dipanggil sampai diperlukan

keuntungan dari pemanggilan dinamis adalah rutin yang tidak digunakan tidak pernah dipanggil

- Swapping

  Sebuah proses dapat ditukar sementara dari memori utam ke penyimpanan cadangan dan kemudian dibawa Kembali ke memori utama untuk eksekusi lanjutan.
  Bagian utama dari waktu swap adalah waktu transfer; total waktu transfer berbanding lurus dengan jumlah memori yang ditukar

  Tampilan Skema Swapping



- Waktu Context Switch Termasuk Swapping

  Jika proses selanjutnya yang akan diletakan di CPU tidak ada memori,perlu swap out proses dan swap in proses target
  Waktu context switch bisa jadi sangat tinggi
  Proses 100MB swapping ke hard disk dengan transfer rate 50MB/sec
  - Waktu swap out 2000 ms
  - Ditambah proses swap in dalam ukuran yang sama
  - Total waktu komponen context switch swapping 4000ms (4 detik)
  - Swap hanya jika memori yang kosong sangat rendah
- Alokasi Memori Berurutan

  Alokasi proses pengguana pada memori berupa single partition allocation atau multiple partition allocation.Keuntungan : sederhana,cepat,mendukung proteksi memori.

- Single Partition Allocation

  Alamat memori yang akan dialokasikan untuk proses adalah alamat memori pertama setelah pengalokasian sebelumnya .Atau dapat juga diasumsikan sistem operasi ditempatkan di memori kecil dan proses pengguna dieksekusi di memori tinggi. Proteksi memori dapat dilakukan dengan menggunakan register relokasi dan register

limit .Register relokasi berisi alamat fisik terkecil, register limit berisi alamat logika yang kurang dari register limit . MMU memetakan alamat logika secara dinamis dengan menambahkan nilai pada register relokasi

- Multiple Partition Allocation

  Sistem Operasi menyimpan informasi tentang semua bagian memori yang tersedia untuk dapat diisi oleh proses-proses.Dibagi menjadi sistem partisi tetap dan sistem partisi dinamis.Sistem partisi tetap,memori dipartisi menjadi blok-blok yang ukuran tetap yang ditentukan oleh dari awal sedangkan sistem partisi dinamis,memori dipartisi menjadi bagian bagian dengan jumlah dan besar yang ditentu.

  Sistem operasi menyimpan sebuah tabel yang menunjukan bagian mana dari memori yang memungkinkan untuk menyimpan proses tersebut dan mempersiapkan sisanya untuk menampung proses proses yang akan dating kemudian,selain itu sistem operasi juga dapat mengatur antrian masukan berdasarkan algoritma penjadwalan yang digunakan.

- Permasalahan Alokasi Penyimpanan Dinamis
  - First-Fit : Mengalokasikan lubang pertama ditemukan yang besarnya mencukupi
  - Best-Fit : Mengalokasikan lubang dengan besar minimum yang mencukupi permintaan;Strategi ini memerlukan pencarian keseluruh lobang kecuali bila ukuran sudah terurut.
  - Wort-Fit Next-Fit : Mengalokasikan lubang terbesar yang ada;strategi ini memerlukan pencarian keseluruh lubang kecuali bila ukuran sudah terurut
- Alokasi Ruang Swap pada Disk

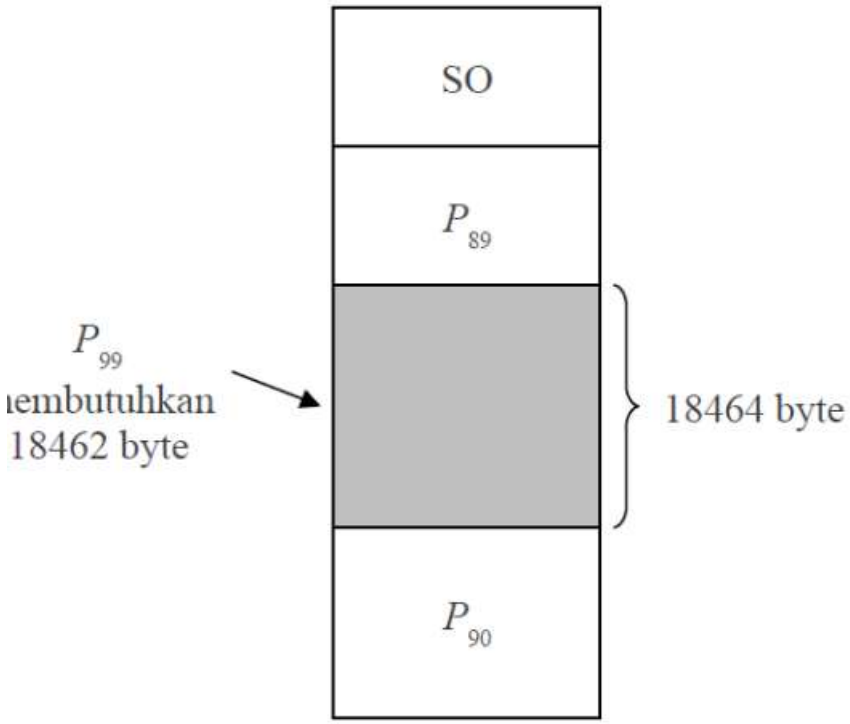
  Terdapat 2 strategi utama penempatan proses yang dikeluarkan dari memori ke disk :
  - Ruang Disk tempat swap dialokasikan begitu diperlukan
  - Ruang Disk tempat swap dialokasikan lebih dulu
- Fragmentation
  - External Fragmentation : Jumlah keseluruhan memori kosong yang tersedia memang mencukupi untuk menampung permintaan tempat dari proses,tetapi letaknya tidak berkesinambungan atau terpecah menjadi beberapa bagian kecil sehingga proses tidak dapat masuk.
  - Internal Fragmentation : Jumlah memori yang diberikan oleh penjadwalan CPU untuk ditempat proses lebih besar dari pada yang diminta proses karena adanya selisih antara permintaan proses dengan alokasi lubang yang sudah ditetapkan.

  Fragmentation eksternal dapat diatasi dengan cara :

  - Pemadatan,yaitu mengatur Kembali isi memori agar memori yang kosong diletakan Bersama disuatu bagian yang besar sehingg proses dapat masuk keruang memori kosong tersebut.
  - Segmentasi
  - Paging

  Fragmentasi internal tidak dapat dihindarkan apabila kita menggunakan sistem partisi tetap,mengingat besar lubang yang disediakan selalu tetap.

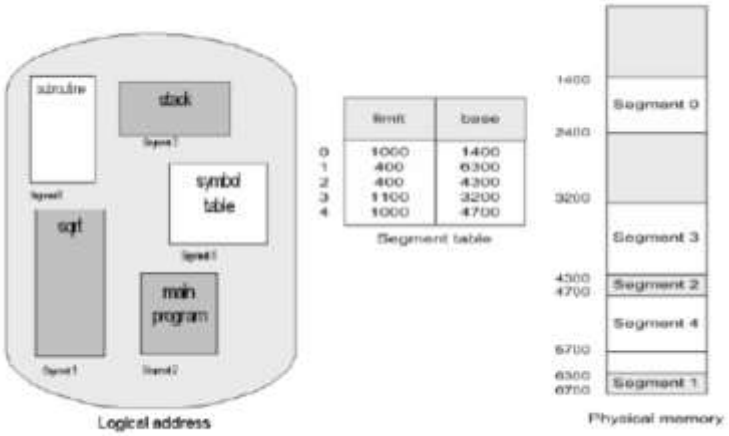
Contoh Fragmentasi Internal :



- Segmentasi

  Segmentasi merupakan skema manajumen memori yang mendukung cara pandang seorang pragramer terhadap memori.Dalam segmentasi terdapat ruang alamat logika yang merupakan sekumpulan dari segmen segmen dan masing masing segment yang mempunyai Panjang dan sama.

  Contoh Segmentasi :



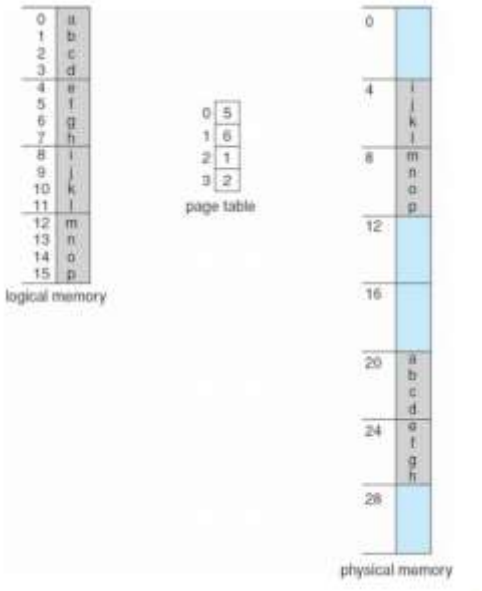
❖ Paging

Paging merupakan suatu metoda yang memungkinkan suatu alamat fisik memori yang tersedia dapat tidak berurutan.

Dalam paging terdapat :

- frame yaitu proses membagi memori fisik menjadi blok berukuran tetap,dimana ukurannya adalah pangkat 2 antara 512 byte dan 16 mybte
- Pages yaitu proses membagi memori logika menjadi blok-blok berukuran sama dan page table yang digunakan untuk menterjemahkan alamat logika ke alamat fisik

Contoh Paging :



❖ Untung/Rugi Paging

Jika kita membuat ukuran dari masing masing page menajdi lebih besar :

- Keuntungan : Akses memori akan relatif lebih cepat
- Kerugian : Kemungkinan terjadinya fragmentasi internal sangat besar.

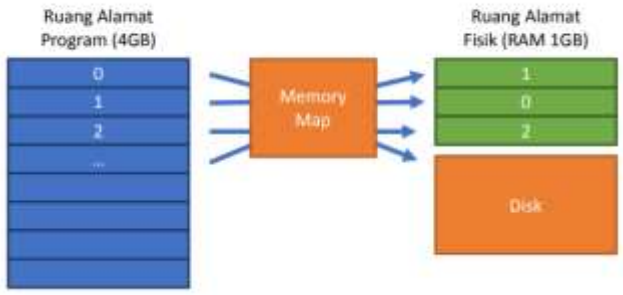
Jika kita membuat ukuran dari masing masing page menjadi lebih kecil :

- Keuntungan : Kemungkinan terjadinya fragmentasi internal menjadi lebih kecil
- Kerugian : Akses memori akan relatif lebih lambat.

3. CHAPTER 8 : MEMORI VIRTUAL

Memori Virtual adalah suatu teknik yang memisahkan antara memori logis dan memori fisiknya,Sehingga menjadikan memori sebagai ruang virtual (virtual address space) dan menaruh beberapa bagian dari memori virtual yang berada di memori logis.

Contoh :



Keuntungan :

- Program bisa dieksekusi hanya dengan menggunakan Sebagian fungsi saja
- Ruang memori fisik menjadi lebih leluasa
- Lebih banyak program program yang bisa dijalankan secara bersamaaan.
- Berkurangnya proses I/O
- Meningkatkan respon,karena menurunya beban I/O dan memori.

❖ Implementasi Memori Virtual

- Demand Paging

  Permintaan pemberian halaman (demand paging) adalah salah satu implementasi dari memori virtual prinsipnya hamper sama dengan permintaan halaman (paging),hanya saja halaman (page) tidak akan dibawa ke dalam memori fisik sampai benar-benar diperlukan,Ada tiga kemungkinan kasus yang dapat terjadi pada saat dilakukan.

  Pengecekan pada halaman yang dibutuhkan :

- Valid (“1”) : Halaman dan sudah berada di memori
- Tidak Valid (“0”) : Halaman ada tetapi masih berada di disk atau belum berada di memori
- Invalid reference : Halaman tidak ada di memori maupun disk

❖ Kelebihan dan kekurangan Demand Paging

- Kelebihan
  - Memori virtual yang besar
  - Pengguanaan memori yang lebih efesien
  - Meningkatkan derajat multiprogramming
  - Penggunaan I/O yang lebih sedikit
- Kekurangan
  - Processor Overhead
  - Thrasing

❖ Kinerja Demand Paging

Waktu akses memori yang lambat akibat perlunya penangan kesalahan halaman yang disebut dengan halaman fault time.Ada tiga factor utama yang mempengaruhi fault time:

- Melayani Interupsi dari kesalahan halaman
- Pembacaan halaman
- Pengulangan Instruksi

Contoh :

Diketahui waktu pengaksesan memori (ma) = 100 ns, waktu kesalahan halaman (halaman fault time) = 20 ms. Berapakah Effective Access Time-nya?

EAT = (1-p) x ma + p x halaman fault time
= (1-p) x 100 + p x 20.000.000
= 100 - 100p + 20.000.000p
= 100 + 19.999.900p nanosecond

❖ Persyaratan Perangkat Keras

- Locality of references

  Jika terjadi banyak kesalahan halaman maka efesiensi sistem akan menurun.oleh karena itu,kemungkinan terjadinya kesalahan halaman harus dikurangi atau dibatasi dengan memakai prinsip lokalitasi ini:
  - Temporal : lokasi yang sekarang ditunjuk kemungkinan akan ditunjuk lagi
  - Spatial : Kemungkinan lokasi yang dekat dengan lokasi yang sedang ditunjuk akan ditunjuk juga.
- Pure Demand Paging

  Penjalanan (running) sebuah program dimulai dengan membawa hanya satu halaman awal memori.Setelah itu,setiap kali terjadi permintaan halaman,halaman

tersebut baru akan dimasukan ke dalam memori.Halaman akan dimasukan ke dalam memori hanya bila diperlukan.

- Multiple Page Fault

  Kesalahan halaman yang terjadi karena satu instruksi memerlukan pengaksesan beberapa halaman yang tidak ada memori utama.Pemberian nomor halaman melibatkan dukungan perangkat keras,sehingga ada persyaratan perangkat keras yang harus dipenuhi perangkat perangkat keras tersebut sama dengan yang digunakan untuk Paging dan Swapping yaitu:
  - Tabel halaman “bit valid-tidak valid”
  - Memori sekunder
- Restart Instruction

  Salah satu penangan jika terjadi kesalahan halaman adalah kebutuhan akan pengualangan instruksi,misalnya jika kesalahaan terjadi saat pengambilan instruksi maka pengulangan proses dilakukan dengan mengambil instruksi itu lagi,dengan mengambil intruksi itu lagi,mendekode kan,baru mengambil operan.

4. CHAPTER 9 : MANAJEMEN DISK

Karakteristik mass/secondary storage

- Non-Volatile
- Biasanya berukuran lebih besar dari main memory

Bebearapa mass storage berdasarkan jenis media fisik yaitu :

- Magnetic disk (Harddisk)
- Solid state disk
- Magnetic tape

Fungsi manajemen disk

- Merupakan salah satu piranti I/O
- Berfungsi sebagai media penyimpanan utama
- Disk yang umum adalah disk cakram magnetis (Harddisk)

❖ Struktur Disk

Secara fisik,disk cakram magnetis terdiri atas cakram yang tersusun secara vertical
Memiliki struktur 3 dimensi yaitu: silinder,Track,Sector.

❖ Pengalamatan Disk

Disk diamati secara logika sebagai array satu dimensi.Unit terkecilnya adalah blok,baik untuk operasi read atau write.
Urutan Penomeran alamat logika disk mengikuti aturan :

- Alamat paling awal yaitu sektor 0 adalah sektor pertama dari track pertama pada silinder paling luar
- Proses pemetaan dilakukan secara berurutan dari sektor 0 lalu keseluruh track dari silinder tersebut,lalu keseluruh silinder mulai dari silinder terluat sampai silider terdalam

❖ Penanganan Disk Request

Mekanisme penanganan disk request adalah sebagai berikut :

- Suatu proses yang membutuhkan transfer data dari dan ke disk,maka proses akan memanggil system call SO

- System Call akan memicu SO memblok proses bersangkutan karena operasi I/O disk akan memakan waktu.Disk request akan ditangani oleh device driver yang sesuai dengan piranti I/O yang hendak diakses.
- Device driver akan memerikasa status disk.Jika sedang sibuk,maka akan dimasukan pada antrian disk bersangkutan.
- Jika disk tidak digunakan,maka disk request tersebut akan dilayani dan alamat disk dikirimkan ke disk controller.
- Operasi writer : data yang akan dibaca,akan disalin ke buffer disk controller terlebih dahulu baru disalin ke memori utama.

❖ Waktu Penanganan Disk Request

Waktu yang dibutuhkan untuk memproses disk request terdiri atas :

- Overhead time : waktu yang dihabiskan SO untuk menangani disk request.
- Transfer time : waktu untuk mentransfer data dari atau ke lokasi disk (ukuran data/transfer rate).
- Latency :waktu yang dibutuhkan untuk menempatkan head ke lokasi yang hendak diakses.Latency terdiri atas 2 komponen yaitu seek time dan rotational latency.

❖ Penjadwalan Disk Request

Penjadwalan disk request terjadi pada sistem multitasking.Penjadwalan request ditunjukan untuk meminimalkan total access time pada operasi transfer data

Contoh Algoritma penjadwalan disk antara lain :

- FCFS (first come first serve)
  Berdasarkan urutan masuknya di antrian.umumnya menghasilkan total akses time yang buruk dan terlalu tinggi.
- SSTF (shortest seek time first)
  disk request yang memiliki seek distance yang paling dekat dengan posisi head terkini, akan dilayani lebih dahulu algoritma ini meminimalkan pergerakan head.
- Elevator/Scan
  Mengasumsikan head bergerak satu arah.jika head sudah mencapai bagian terluar atau terdalam dari cakram,maka arah gerak head dibalik.
- One way elevator/C-Scan
  Mirip dengan elevator/scan.bedanya head tidak melakukan pembalikan arah.
- LOOK
  Mirip dengan scan.bedanya head tidak perlu melakukan perjalanan penuh dari bagian terluar sampai terdalam bila sudah tidak ada disk request lagi.
- C-LOOK
  Mirip dengan elevator/C-SCAN bedanya A head bergerak paling jauh hanya permintaan terakakhir pada masing-masing arah pergerakannya.kemudian langsung berbalik arah tanpa harus menuju ujung disk.

❖ Organisasi Disk

- Pemformatan fisik (low-level formatting)
  - Membagi disk ke dalam sektor-sektor sehingga disk controller dapat membaca atau menulis
  - Operasi ini umumnya dilakukan dari pabrik pembuat disk
- Pemartisian

- Membagi disk menjadi salah satu atau lebih partisi dimana dapat dipandang secara logika sebagai disk yang terpisah
- Suatu partisi belum dapat ditulis sebelum di format secar logika

- Pemformatan secara logika
  - Membagun struktur pengolahan berkas sebelum data atau berkas dapat disimpan ke suatu partisi disk
  - Menentukan unit alokasi berkas atau blok terkecil untuk alokasi berkas
- Alokasi Blok Booting
  - Membagun struktur untuk melakukan operasi booting yang biasanya diwal disk.Boostrap yaitu program kecil untuk inisiasi booting SO
  - Biasa tersimpan di ROM BIOS pada motherboard
    - Manajemen Bad Sector

Mengelola,mencatat dan mengalihkan bad block,yaitu satu atau lebih sector yang rusak pada disk ke tempat lain agar ditulisi data.

❖ Pengelolaan Ruang Kosong

Informasi ruang kosong akan diperbarui bila ada alokasi bekas baru atau penghapusan berkas

Teknik-Teknik untuk pencatatan ruang kosong :

- Teknik Bit-Vektor menggunakan satu bit untuk menyatakan kosong tidaknya setiap alamat blok media penyimpanan
- Teknik Link List Menggunakan blok-blok kosong di media penyimpanan untuk menyimpan pointer atau alamat blok kosong berikutnya
- Teknik Grouping untuk mengumpulkan informasi alamat blok kosong ke blok kosong pertama
- Teknik Counting untuk memperhitungkan rangkaian blok-blok kosong yang kontinu sebagai suatu segmen.

❖ Pengelolaan alokasi berkas

Ada berbagai cara alokasi berkas :

- Alokasi berurut : semua bagian dari berkas harus diletakkan secara berurut dan tidak boleh tersebar pada media penyimpanan
- Alokasi Berantai : Berkas dapat dialokasi ke blok-blo kosong secara tersebar di media penyimpanan
- Alokasi Berindeks : Pemakaian blok khusus untuk mencatata blok-blok yang ditempati berkas yang disebut blok indeks

❖ Struktur Penyimpanan Tersier

Ciri-ciri :

- Biaya produksi lebih murah
- Menggunakan removable mediaData yang disimpan bersifat permanen

Macam-macam :

- Floppy disk

- Optical disk
- Flash disk
- Tapes

5. CHAPTER 10 : PERANGKAT KERAS I/O

❖ Perangkat Keras I/O

Salah satu fungsi utama sistem operasi adalah mengatur operasi input/output (I/O)beserta perangkatnya

❖ Skema I/O

Secara umum,perangkat I/O dapat dibagi menjadi dua kategori :

- Perangkat Blok,perangkat yang menyimpan informasi dalam bentuk blok-blok berukuran tertentu dan setiap blok memiliki alamat masing-masing contoh disk.
- Perangkat karakter,perangkat yang mengirim atau menerima sebarisan karakter,tanpa menghiraukan struktur blok.contoh printer,network interface.

❖ Port & Bus

- Port adalah sebuah titik yang dilalui perangkat tersebut untuk bisa berhubungan dengan komputer.
- Bus adalah serangkaian penghubungan yang berfungsi sebagai saat pengiriman pesan/data berlangsung,jik satu atau lebih perangkat menggunakan serangkaian kabel atau penghubung yang sama.

Port I/O terdiri dari empat regiter :

- Status : menandakan statur apakah perintah I/O sempurna dilaksanakan perangkat ada bit di register data-in yang tersedia untuk dibaca ataupun apakah perangkat I/O yang eror
- Control : regiter yang ditulis oleh CPU untuk memulai perintah atau untuk mengganti modus perangkat.
- Data-in :register yang dibaca CPU sebagain input data
- Data-out : register yang dibaca CPU sebagai ouput data

❖ Komponen I/O

Unit I/O terdiri dari dua komponen :

- Komponen mekanisme yakni perangkat I/O itu sendiri contonya mouse,screen,keyboard,dll
- Komponen Elektronis disebut sebagai pengendali perangkat I/O (device controller)

❖ Interface I/O

Terdapat berbagai macam antar muka antara perangkat dengan pengendalinya antara lain

- ANSI
- IEEE
- ISO
- IDE (integrated Drive Electronics)
- SCSI (Small computer system interface)

❖ Penanganan I/O

Terdapat dua cara sistem operasi memberikan perintah dan data :

- Instruksi I/O merupakan instruksi CPU yang khusu menangani transger byte atau word ke sebuah port I/O
- M/K memory-mapped register register pengendali perangkat yang dipetakan ke ruang alamat prosesor.

❖ Polling

Polling/Busy waiting adalah ketika host mengalami looping yaitu membaca status register secara terus menerus sampai status busy di clear.

❖ Mekanisme Interupsi

Langkah-langkah mekanisme interupsi yang disebabkan perangkat I/O yaitu :

- Perangkat I/O mengirim sinyal interupsi
- CPU menerima sinyal interupsi
- Penyimpanan informasi proses yang sedang dieksekusi
- CPU mengindentifikasi penyebab interupsi
- CPU mengkeskusi interupsi routine sampai return
- CPU melanjutkan proses yang sebelum ditunda

Ada tiga kemungkinan penyebab perangkat I/O mengirimkan interupsi :

- Input Ready
- Output Complete
- Error

Sistem Operasi mengggunakan mekanisme interupsi untuk beberapa hal diluar operasi yang berhubungan dengan I/O seperti :

- Menangani berbagai macam exception
- Mengatur virtual memori paging
- Menangaini software interupsi
- Mengatur alur control kernel

❖ Fitur Tambahan Mekanisme interupsi

Pada arsitektur komponen modern,tiga fitur disediakan oleh CPU untuk menangani interupsi :

- Interrupt request line
- Interrupt vector
- Interrupt Chaining

❖ Penyebab Interupsi

Interupsi dapat disebababkan berbagai hal :

- Exception
- Page fault
- Interupsi yang dikirimkan oleh pengendali perangkat
- System call

❖ Direct Memory Access (DMA)

DMA adalah proses khusu (special purpose processor) yang berguna untuk menghindari pembebanan CPU utama oleh program I/O (PIO).DMA memiliki sepasang kabel yaitu DMA request dan DMA acknowledge.

❖ Handshacking di DMA

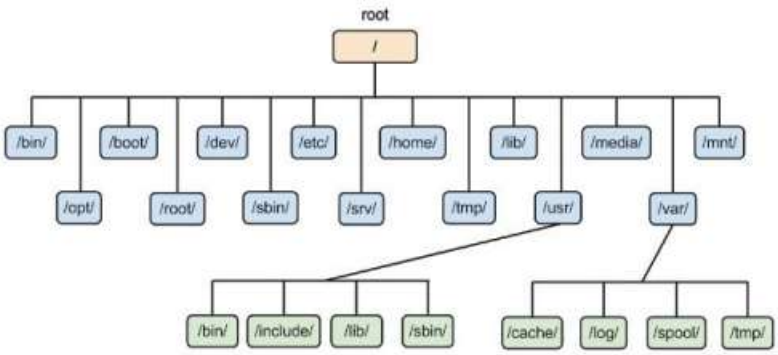
Handschaking adalah proses pada di DMA yang berlangsung secara berulang-ulang.

- Direct virtual Memort atau DVMA
  Penerjemahan alamat memori virtual menjadi alamat memori fisik

6. CHAPTER 11 : MANAJEMEN SISTEM FILE

- Konsep File
  File adalah sebuah koleksi informasi berkaitan yang diberi nama dan disimpan di dalam secondary storage
  Media Penyimpanan data/informasi : Magnetic disk,magnetic tape,optical disk
  Isi file : representasi program atau data yang terekam dalam secondary storage
  Attribut file terdiri dari :
  - Name : informasi yang disimpan dalam format yang dapat dibaca oleh pengguna
  - Type : informasi ini diperlukan untuk sistem sistem yang mendukung jenis file yang berbeda.
  - Location : informasi berupa penunjuk pada sebuah device dan lokasi berkas pada device tersebut.
  - Size : Ukuran file yang sedang digunakan.
  - Protection : Kontrol terhadapa pengguna siapa yang dapat melakukan baca,tulis dan eksekusi.
  - Time,date,dan user identification :Proteksi untuk pengamanan dan monitoring pengguna
- Operasi Operasi File
  Enam operasi dasar yang berkaitan dengan manajemen file sistem :
  - Create File
  - Write File
  - Read File
  - Reposition dalam file
  - Delete File
  - Truncate File
- Metoda Akses
  - Sequential Access
    - Akses dilakukan dengan satu arah pembacaan/penulisan (dari awal hingga akhir)
  - Direct Access (random access)
    - Akses dilakukan bisa pada posisi mana saja dalam file
- Struktur Direktori
  Direktori adalah kumpulan node yang berisi informasi dari semua file yang mengandung direktori lain file istimewa baik direktori maupun file terletak di disk.
- Informasi yang ada pada device direktori
  - Nama : nama dari direktori itu sendiri
  - Alamat : alamat dari direktori
  - Ukuran : besarnya ukuran direktori
  - Tanggal : berisi keterangan mengenai tanggal pembuatan dari direktori tersebut
  - ID pemilik : ID dari pemilik direktori tersebut
  - Proteksi : Atribut yang berguna sebagai proteksi

- Operasi Direktori
    - Pencarian File : mencari lewat struktur direktori untuk dapat menemukan suatu file tertentu
    - Pembuatan File : File baru perlu untuk dibuat dan ditambahkan ke dalam direktori baru
    - Penghapusan File : Saat suatu file tidak diperlukan lagi,file tersebut perlu dihapus dari direktori
    - Daftar Direktori : menampilkan daftar file yang ada di direktori dan semua isi direktori dari file dalam daftar tersebut
    - Penggantian nama file : Nama file harus dapat diubah-ubah ketika isi dan kegunaannya sudah beruab atau tidak sesuai lagi
- Organisasi Direktori
    - Effeciency : menempatkan file secara cepat
    - Naming : kenyamaanan pengguna
    - Grouping : pengelompokan file secara berdasarkan property



- Direktori Satu tingkat
  Hanya ada satu direktori untuk semua user
    - Naming problem
    - Grouping problem
- Direktori Dua Tingkat
  Pemisahan Direktori untuk setiap user
    - Share file
- Direktori Struktur Tree
  Lintasan relative dan lintasan mutlak
- Direktori Acyclic-Graph
  Saling berbagi diantara subdirektorri dan file
- File Sharing
  Sharing File pada sistem multi user sangat diharapkan,sharing dapat dilakukan melalui skema proteksi.pada sistem terdistribusi,file dapat dishare lintas jaringan.
  Network File System (NFS) adalah bentuk umum shating file terdistribusi.
- Proteksi
  Ketika suatu informasi disimpan dalam sistem komputer,kita harus menjaganya nya dari kerusakan fisik (relabilty) dan akses yang tidak diinginkan (protection)

Jenis Akses :

- Read :membaca file
- Write: Menulis isi file
- Excute : memasukan file ke memori dan dieksekusi
- Delete : menghapus file

❖ Securty Implementation

- Access Control List (ACL)
  - Menentukan jenis akses yang diperbolehkan untuk pengguna lain
  - Dilakukan dengan menghubungkan setiap berkas atau direktori yang berisi nama pengguna dan jenis akses yang diberikan kepada pengguan tersebut
- Tiga Klarifikasi pengguna :
  - Owner : pengguna yang telah membuat berkas tersebut
  - Group : Sekelompok pengguna yang saling berbagi berkas dan membutuhkan akses yang sama
  - Universe : keseluruhan pengguna.