

Manajemen Memori (1)

Manajemen Memori

Memori utama harus diatur sebaik mungkin agar :

- meningkatkan utilitas CPU yang sebesar-besarnya
- data dan instruksi dapat diakses dengan cepat oleh CPU
- memori utama memiliki kapasitas yang sangat terbatas, sehingga pemakaiannya harus seefisien mungkin
- transfer data dari/ke memori utama ke/dari CPU dapat efisien.

Manajemen Memori

- Manajemen memori berkaitan dengan memori utama sebagai sumber daya yang harus dialokasikan dan dipakai bersama diantara sejumlah proses yang aktif.
- Manajemen memori juga berkaitan dengan usaha agar pemrogram atau pemroses tidak dibatasi oleh kapasitas memori fisik yang terdapat pada sistem komputer.

Fungsi Manajemen Memori

- mengelola informasi memori yang terpakai dan yang tidak terpakai
- mengalokasikan memori ke proses yang memerlukan
- mendealokasikan memori dari proses telah selesai
- mengelola swapping antara memori utama dan disk

Swapping

- Suatu proses dapat dialihkan sementara dari memori ke suatu tempat penyimpanan dan dipanggil kembali ke memori jika akan melanjutkan eksekusi.
- Manajemen memori berdasarkan keberadaan swapping :
 - manajemen tanpa swapping
 - manajemen dengan swapping

Monoprogramming

- Merupakan manajemen memori paling sederhana. Sistem komputer hanya mengizinkan 1 program/pemakai berjalan pada 1 waktu.
- Semua sumber daya sepenuhnya dikuasai proses yang sedang berjalan.
- Ciri-ciri manajemen memori monoprogramming :
 - hanya 1 proses pada 1 saat
 - hanya 1 proses menggunakan semua memori
 - pemakai memuatkan program ke seluruh memori dari disk / tape
 - program mengambil kendali seluruh mesin
- Karena hanya terdapat 1 proses dan menguasai seluruh sistem maka alokasi memori dilakukan secara berurutan. 6

Monoprogramming

Memori utama

Sistem operasi di RAM
Program pemakai di RAM
Memori tak dipakai

Multiprogramming dengan Pemartisian Statis

- Multiprogramming = banyak proses pada memori utama pada saat bersamaan.
- Alasan menggunakan multiprogramming :
 - mempermudah pemrogram karena pemrogram dapat memecah program menjadi 2 proses/lebih
 - agar dapat memberi layanan interaktif ke beberapa orang secara simultan
 - efisiensi penggunaan sumber daya
 - eksekusi lebih murah jika proses besar dipecah menjadi beberapa proses kecil
 - dapat mengerjakan sejumlah job secara simultan

Partisi Statis

- memori dibagi menjadi sejumlah partisi tetap. Pada partisi tersebut proses-proses ditempatkan.
- Berdasarkan ukurannya, dibagi 2 :
 - Pemartisian dengan partisi berukuran sama (Ukuran semua partisi memori adalah sama. Beberapa proses yang ukurannya kurang/sama dengan ukuran partisi dimasukkan ke sembarang partisi yang tersedia.)
 - Pemartisian dengan partisi berukuran berbeda (ukuran semua partisi memori berbeda)

Pemartisian dengan Partisi Berukuran Sama

Kelemahan :

- Bila program berukuran lebih besar dibanding partisi yang tersedia, maka tidak dapat dimuatkan dan dijalankan. Pemrogram harus mempersiapkan overlay. Overlay adalah program dipecah menjadi bagian-bagian yang dapat dimuat ke memori. Sehingga hanya bagian program yang benar-benar dieksekusi yang dimasukkan ke memori utama dan saling bergantian. Untuk overlay diperlukan sistem operasi yang mendukung swapping.
- Bila program lebih kecil daripada ukuran partisi yang tersedia, maka akan ada ruang yang tak dipakai, yang disebut fragmentasi internal → pemborosan memori. Kelemahan ini dapat dikurangi dengan membuat partisi tetap dengan ukuran yang berbeda.

Pemartisian dengan Partisi Berukuran Sama

Sistem operasi	
Partisi - 1	256 Kbyte
Partisi -2	256 Kbyte
Partisi -2	256 Kbyte

Pemartisian dengan Partisi Berukuran Berbeda

Sistem operasi	
Partisi - 1	100 Kbyte
Partisi -2	256 Kbyte
Partisi -3	350 Kbyte

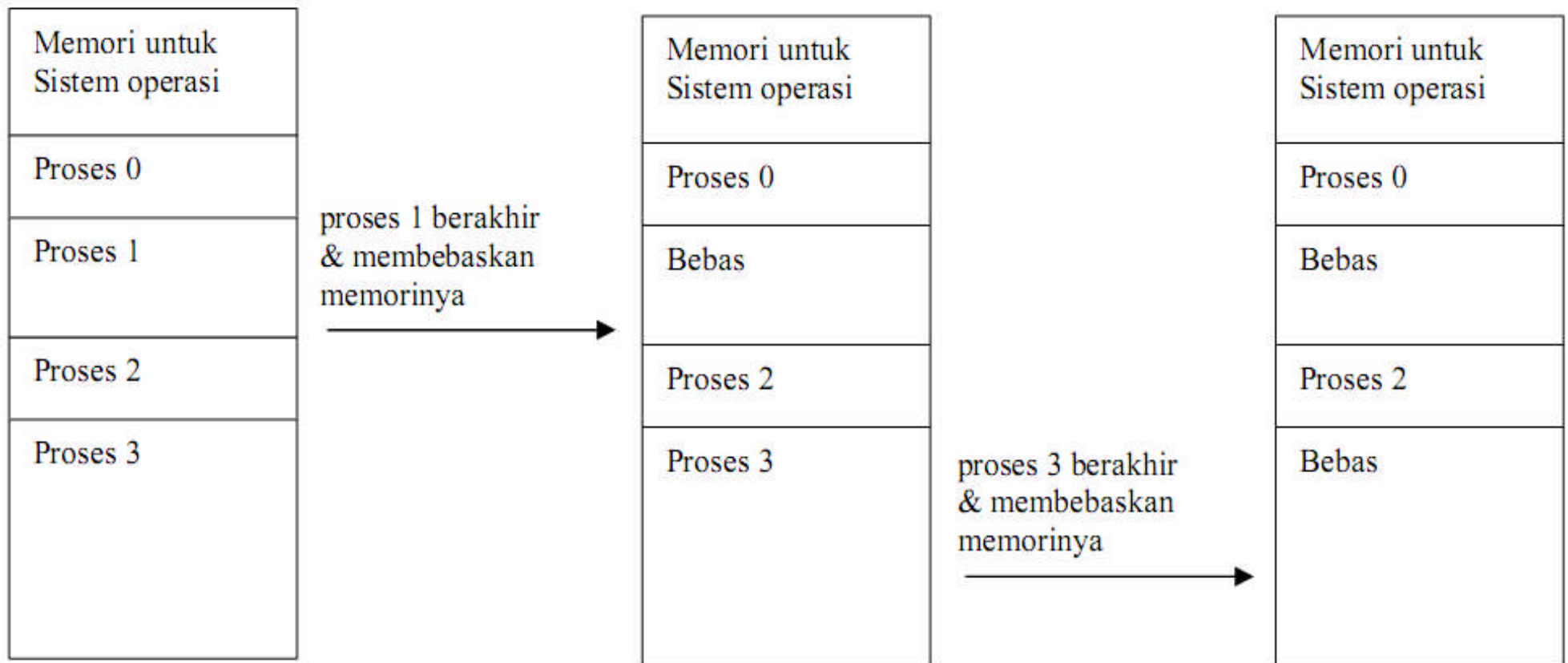
Fragmentasi Pada Partisi Statis

- Fragmentasi = penyiaian/pemborosan memori yang terjadi pada setiap organisasi penyimpanan.
 - Fragmentasi internal : proses tidak mengisi penuh partisi yang telah ditetapkan untuk proses
 - Fragmentasi eksternal : partisi tidak dapat digunakan karena ukuran partisi lebih kecil dibanding ukuran proses yang menunggu di antrian.

Partisi Dinamis

- partisi statis menyebabkan memori terlalu banyak diboroskan dengan proses-proses yang lebih kecil dibanding partisi yang ditempatinya.
- Dengan partisi dinamis maka jumlah, lokasi, dan ukuran proses di memori dapat beragam sepanjang waktu secara dinamis. proses yang akan masuk ke memori segera dibuatkan partisi untuknya sesuai kebutuhannya. Teknik ini meningkatkan utilitas memori.
- Kelemahan :
 - dapat terjadi lubang-lubang kecil memori di antara partisi-partisi yang dipakai
 - merumitkan alokasi dan dealokasi memori

Contoh



Contoh

- Setelah proses 3 berakhir memori dipenuhi lubang-lubang memori yang tidak terpakai.
- Dapat diatasi dengan cara pemadatan memori (memory compaction) yaitu operasi penggabungan semua lubang kecil menjadi lubang besar dengan memindahkan semua proses agar saling berdekatan.

Contoh

- Contoh diatas setelah pemadatan memori :

Memori untuk Sistem operasi
Proses 0
Proses 2
Bebas

Partisi Dinamis

- Masalah lain pada partisi dinamis adalah proses dapat tumbuh berkembang. Solusinya adalah bila proses bersebelahan dengan lubang memori tak dipakai, proses tumbuh memakai lubang itu.
- Masalah menjadi lebih parah bila proses bersebelahan dengan proses-proses lain.

Partisi Dinamis

- Alternatif penyelesaian :
 - bila masih terdapat lubang besar yang dapat memuat proses, maka proses dipindah ke lubang memori yang dapat memuat
 - 1 proses atau lebih diswap ke disk agar memberi lubang cukup besar untuk proses yang berkembang
 - jika proses tidak dapat tumbuh di memori dan daerah swap di disk telah penuh, proses harus menunggu/disingkirkan.

Strategi Alokasi Memori

- Alokasi harus mencari sekumpulan blok memori yang ukurannya mencukupi untuk memuat proses, dimana lubang kosong yang sama atau lebih besar dibanding ukuran memori yang diperlukan oleh proses.
- Ada beberapa strategi alokasi memori

Algoritma First Fit

- Pencarian dimulai dari awal dan akan berhenti jika ditemukan lokasi pertama yang cukup besar untuk menempatkan proses tersebut.
- Contoh :
- terdapat partisi kosong pada memori dengan urutan dan ukuran : 4 Kb, 3 Kb, 2 Kb, 6 Kb
- bila datang data yang berukuran 3 Kb maka akan menempati partisi ukuran 4 Kb

Algoritma Next-fit

- Sama dengan first-fit hanya saja pencarian tidak dimulai dari awal, tapi dari lokasi terakhir kali menemukan segmen yang cocok dan akan berhenti jika ditemukan lokasi pertama yang cukup besar untuk menempatkan proses tersebut
- Contoh :
- terdapat partisi pada memori dengan urutan dan ukuran : 4 Kb, 3 Kb, 2 Kb, 6 Kb
- bila datang data yang berukuran 3 Kb dan pencarian partisi dimulai dari urutan ketiga karena sebelumnya posisi terakhir pencarian di partisi kedua, maka data tersebut akan menempati partisi ukuran 6 Kb.

Algoritma Best Fit

- Pencarian dimulai dari awal dan akan berhenti jika ditemukan lokasi terkecil pertama yang cukup untuk menempatkan proses tersebut.
- Contoh :
- terdapat partisi kosong pada memori dengan urutan dan ukuran : 4 Kb, 3 Kb, 2 Kb, 6 Kb
- bila datang data yang berukuran 3 Kb maka akan menempati partisi ukuran 3 Kb.

Algoritma Worst-fit

- Pencarian dimulai dari awal dan akan berhenti jika ditemukan lokasi yang paling besar yang cukup untuk menempatkan proses tersebut.
- Contoh :
- terdapat partisi kosong pada memori dengan urutan dan ukuran : 4 Kb, 3 Kb, 2 Kb, 6 Kb
- bila datang data yang berukuran 3 Kb maka akan menempati partisi ukuran 6 Kb.

Sistem Buddy

- Mengalokasikan memori berdasarkan kelipatan 2 (2^k ; $k = 0, 1, 2, \dots$).
- Pada sistem buddy, semula semua ruang pada memori kerja didefinisikan sebagai satu kesatuan utuh. Bagian ini baru dibelah apabila ada data yang menempati ruang tersebut. Sistem pembelahan sesuai dengan pangkat bilangan biner, yaitu : $2^0, 2^1, 2^2, 2^3, \dots$

Sistem Buddy

- Tentukan batas atas (u) dan batas bawah (l)
- Contoh : memori berukuran 2000k, batas atasnya (u) dari k adalah 20 karena 2^{20} adalah 1024, sisa 976k akan dialokasikan pada blok yang lebih kecil. Batas bawah umumnya bisa antara 12-16, jika terlalu kecil akan membebani system karena harus mengingat terlalu banyak block yg dialokasikan dan yg tidak, misalkan batas bawahnya adalah 16 jadi $2^{16} = 64k$

Sistem Buddy

Contoh :

1=Program A meminta 34k

2=Program B meminta 66k

3=Program C meminta 35k

4=Program D meminta 67k

5=Program C melepaskan memori

6=Program A melepaskan memori

7=Program B melepaskan memori

8=Program D melepaskan memori

Sistem Buddy

Langkah-langkah :

- Memori akan dialokasikan :
 - Cari slot memori yang cukup (blok minimal 2^k yang lebih besar atau sama dengan memori yang diminta)
 - Jika ditemukan, alokasikan ke program
 - Jika tidak, bagi slot memori yang bebas menjadi 2
- Memori dibebaskan :
 - Jika slot di sebelahnya juga kosong, maka gabungkan

	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K
$t = 0$	1024K															
$t = 1$	A-64K	64K	128K		256K			512K								
$t = 2$	A-64K	64K	B-128K		256K			512K								
$t = 3$	A-64K	C-64K	B-128K		256K			512K								
$t = 4$	A-64K	C-64K	B-128K		D-128K		128K	512K								
$t = 5$	A-64K	64K	B-128K		D-128K		128K	512K								
$t = 6$	128K		B-128K		D-128K		128K	512K								
$t = 7$	256K				D-128K		128K	512K								
$t = 8$	1024K															

Virtual Memory dan Overlay

- Program yang dijalankan harus dimuat di memori utama. Masalah muncul ketika program lebih besar dibanding memori utama yang tersedia. Solusinya :
 - Overlay
 - Memori maya (virtual memory)

Overlay

- program dipecah menjadi bagian-bagian yang dapat dimuat memori.
- Overlay yang belum diperlukan (tidak sedang dieksekusi) disimpan di disk. Overlay ini dimuatkan ke memori begitu diperlukan.

Memori Maya (virtual memory)

- Adalah kemampuan mengamati ruang memori melebihi memori utama yang tersedia.
- Gagasan memori maya adalah ukuran gabungan program, data dan stack melampaui jumlah memori fisik yang tersedia.
- Sistem operasi menyimpan bagian-bagian proses yang sedang digunakan di memori utama dan sisanya di disk.
- Bagian-bagian di disk diperlukan maka bagian memori yang tidak diperlukan disingkirkan diganti bagian di disk yang diperlukan itu.

Memory Maya (virtual memory)

- Contoh :
- Program 10 Mbyte dapat berjalan di mesin 2 Mbyte, yaitu memilih bagian proses sebesar 2 Mbyte secara hati-hati dan ditaruh di memori.
- Bagian-bagian proses di-swap antara disk dan memori saat diperlukan secara otomatis oleh sistem operasi.

Memori Maya (virtual memory)

- Memori maya dapat dilakukan dengan 2 cara yaitu :
 - paging
 - segmentasi