

BAB

Manajemen Memory

Tujuan memory utama harus diatur sebaik mungkin adalah :

1. Meningkatkan utilitas CPU sebesar-besarnya,
2. Data dan instruksi dapat diakses dengan cepat oleh CPU,
3. Memory utama memiliki kapasitas yg terbatas, sehingga pemakaian harus seefisien mungkin,
4. Transfer data dari/ke memory utama ke/dari CPU dapat efisien.

Manajemen memory mempunyai fungsi berikut :

1. Mengelola memory yg dipakai dan tidak terpakai,
2. Mengalokasikan memory ke proses yg memerlukan,
3. Mendelekasi memory dari proses yg telah selesai,
4. Mengelola swapping antara memory utama dan disk.

Syarat pengelolaan memory :

1. Relokasi, CPU dan SO harus dapat mentranslasikan memory referensi ke alamat fisik yg mengalokasikan program dalam memory utama,
2. Proteksi, user tidak boleh mengakses beberapa bagian dari wilayah Sistem Operasi (SO),
3. Sharing, manajemen memory harus dapat mengontrol sharing area pada memory utama,
4. Organisasi logika, So dan hardware diusahakan untuk dapat berhubungan dengan user program dalam satu modul,
5. Organisasi fisik, ada pengaturan yg jelas antara memory utama dengan memory sekunder.

Dynamic Loading, yaitu bahwa tidak semua bagian program diambil ke memory, hanya bagian-bagian yg diperlukan saja yg harus tetap tinggal dalam memory.

Dynamic Linking.

Overlay, membagi program yg besar menjadi bagian-bagian yg lebih kecil dan dapat dimuat dalam memory utama.

Swapping, suatu proses dialihkan sementara dari Ram ke penyimpanan permanen dan dipanggil lagi jika akan melanjutkan eksekusi.

Pencatatan Pemakaian memori

1. Peta Bit (*bit map*),
2. *Linked list*, setiap node memuat informasi tentang proses (P) atau hole (H), lokasi awal, dan panjang lokasi.

MANAJEMEN MEMORI PEMARTISIAN STATIS

A. Manajemen Memori pada Sistem Multiprogramming

Pada sistem multiprogramming maka keberadaan ruang alamat memory akan dipakai secara bersama. Sistem operasi harus mendukung dua kebutuhan yang saling bertentangan, yaitu:

1. Pemisahan ruang-ruang alamat
2. Pemakaian bersama memori

Manajemen memory harus dapate menyediakan adanya isolasi ruang ruang alamat untuk tiap proses agar tidak terjadi suatu proses yang menempati alokasi proses lain.

B. Manajemen Memory Berdasarkan Keberadaan Swapping

Terbagi menjadi 2, yaitu:

1. Manajemen memori tanpa swapping, yaitu manajemen memory tanpa memindahkan citra proses antara memori utama dan disk selama eksekusi.
2. Manajemen memori dengan swapping, yaitu manajemen memory dengan memindahkan citra proses antara memori utama dan disk selama eksekusi.

1. Manajemen Memory Tanpa Swapping

Terdiri dari 2, yaitu:

- a. Monoprogramming.
- b. Multiprogramming dengan Pemartisian Statis.

Add. a. Monoprogramming,

Merupakan manajemen memori paling sederhana. Hanya satu program yang berjalan pada satu waktu. Sehingga sumber daya dikuasasi sepenuhnya oleh program yang sedang berjalan.

Ciri-ciri manajemen memori monoprogramming adalah:

- ✓ Hanya satu proses yang berjalan pada satu waktu.
- ✓ Satu proses menggunakan semua memori.
- ✓ Pemakai memuatkan program ke seluruh memori dari Disk.
- ✓ Program mengambil kendali seluruh mesin.



a.1. Embedded System.

Teknik monoprogramming dapat digunakan pada sistem embedded, yaitu program kecil yang menyediakan fungsi yang spesifik. Memuat didalamnya processor dengan kapasitas memori yang kecil. Penerapan pada mobil seperti:

- ✓ Pengendalian pengapian,
- ✓ Pengendalian pengeluaran bahan bakar,
- ✓ Pengendalian pengereman,
- ✓ Pengendalian suspensi, dll

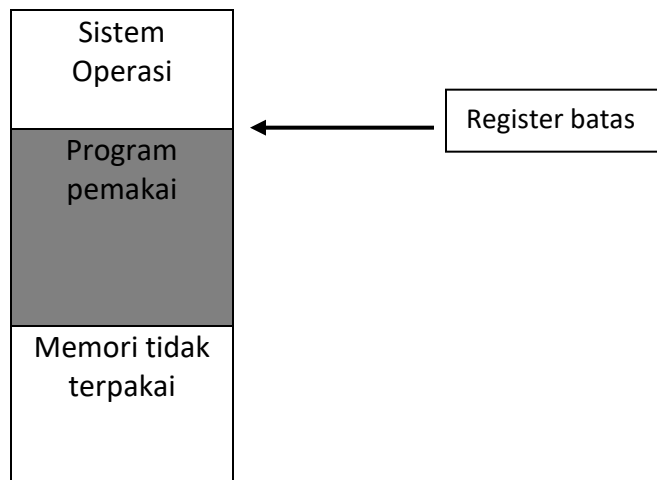
a.2. Proteksi monoprogramming sederhana

pada monoprogramming, pemakai mempunyai kendali penuh terhadap seluruh memori utama. Memori terbagi menjadi 3 bagian, yaitu:

1. Bagian memori yang berisi rutin sistem operasi.
2. Bagian memori yang berisi program pemakai.
3. Bagian memori yang tidak digunakan.

Pada bagian yang berisi sistem operasi harus dilakukan proteksi agar tidak termodifikasi dari program pemakai. Cara proteksi ini di realisasikan dengan menggunakan *register* batas di pemroses. Setiap kali program pemakai meminta alokasi alamat memori maka akan dibandingkan dengan *register* batas, hal ini untuk memastikan bahwa proses pemakai tidak masuk pada wilayah bagian memori rutin sistem operasi.

Register batas berisi alamat memori tertinggi yang dipakai sistem operasi. Jika ada program pemakai yang akan memasuki wilayah sistem operasi, maka instruksi di intersepsi terjadi ilegal akses, terjadi fault proses dan job diakhiri dan diberikan pesan kesalahan.



Add. a. Multiprogramming dengan Pemartisian Statis,

Memori dibagi menjadi sejumlah partisi tetap. Didalam partisi tersebut proses-proses akan dialokasikan.

Berdasarkan ukurannya, model partisi statis terbagi menjadi 2, yaitu:

1. Pemartisian dalam partisi dengan ukuran sama, yaitu semua ukuran partisi memori dengan besaran sama. Sehingga penempatan proses ke memori dilakukan dengan dipilih sembarang partisi yang kosong.

Simulas:

Partisi 1 (200 kbps)	150	350
Partisi 2 (200 kbps)		
Partisi 3 (200 kbps)	100	
Partisi 4 (200 kbps)		
Partisi 5 (200 kbps)	175	
Partisi 6 (200 kbps)		
Partisi 7 (200 kbps)		

2. Pemartisian dalam partisi dengan ukuran berbeda, yaitu semua ukuran partisi memori dengan besaran berbeda-beda.

Strategi penempatan program ke partisi:

- a. Satu antrian untuk tiap partisi,
- b. Satu antrian untuk seluruh partisi.