

**MODUL ONLINE 1
MATA KULIAH ORGANISASI DAN ARSITEKTUR KOMPUTER
KODE MATA KULIAH CC0120**

**DISUSUN OLEH
NIZIRWAN ANWAR & TEAM**

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL
JAKARTA
20181**

Sistem Memori Komputer

Dalam pengetahuan komputer, penggunaan istilah memory biasanya merujuk pada media atau tempat untuk menyimpan data. Dapat dikatakan bahwa memory adalah perangkat khas yang digunakan untuk menyimpan data atau informasi dan dapat dibaca atau diambil kembali saat diperlukan.

Pada komputer, program (software) yang sedang dijalankan (di-run) dan data yang sedang diproses, disimpan di dalam memory selama program tadi masih aktif bekerja. Memori itu sendiri sebenarnya terdiri dari kotak kotak untuk menyimpan data (karakter). Kotak kotak itu sendiri memiliki alamat atau address sendiri. dengan menggunakan address inilah data keluar masuk memory biasanya sepadan dengan kecepatan kerja CPU itu sendiri. CPU dan memori merupakan bagian yang tak terpisahkan dari sebuah komputer. Secara fisik (hardware), kebanyakan memory berupa chip semikonduktor. Komputer utama dalam sistem komputer adalah ALU, Control Unit, Storage Space dan piranti Input Output. Tanpa memori, komputer hanya berfungsi sebagai piranti pemroses sinyal digital saja, contohnya kalkulator atau media player. Kemampuan memori untuk menyimpan data, instruksi dan informasi yang membuat komputer dapat disebut sebagai komputer multifungsi. Komputer merupakan piranti digital, maka informasi disajikan dengan sistem bilangan biner. Teks, angka, gambar, suara dan video dikonversikan menjadi sekumpulan bilangan biner. Sekumpulan bilangan biner dikenal dengan istilah bait, dimana 1 byte = 8 bit. Semakin besar ukuran memorinya maka semakin banyak pula informasi yang dapat disimpan di dalam komputer. Untuk mempelajari sistem memory secara keseluruhan, harus mengetahui karakteristik-karakteristik kuncinya. Karakteristik yang penting pada sistem memory ditunjukkan pada table ini,

Tabel 1 Karakteristik Memory

| No | Karakteristik | Macam/Keterangan |
|----|--------------------------|---|
| 1. | Lokasi | CPU Intrnal (main) Eksternal (secondary) |
| 2. | Kapasitas | Ukuran word Jumlah word |
| 3. | Satuan transfer | Word Block |
| 4. | Metode akses | Sequential acces Direct acces Random acces Assosiative acces |
| 5. | Kinerja | Acces time Cycle time Transfer rate |
| 6. | Tipe fisik | Semikonduktor Magnetic Optical |
| 7. | Karakteristik tipe fisik | Volatile/Nonvolatile Erasable/Nonerasable |

Dilihat dari lokasi, memori dibedakan menjadi beberapa jenis, yaitu register, memori internal dan memori eksternal. Register merupakan tempat penyimpanan berbagai data tidak tetap di dalam processor yang diolah secara langsung dengan kecepatan yang sangat tinggi. Register ini berada di dalam processor dengan jumlah yang sangat terbatas, berfungsi sebagai tempat perhitungan/ komputasi. Memori internal adalah memori yang berada di luar processor namun diakses secara langsung oleh processor. Memori internal dibedakan menjadi memori utama dan cache memori. Memori eksternal dapat diakses oleh prosesor melalui piranti I/O, memori ini dapat berupa disk atau pita.

Cache merupakan mekanisme dan tempat untuk penyimpanan data sekunder berkecepatan tinggi yang digunakan untuk menyimpan data/ instruksi yang sering di akses. Cache umumnya menggunakan memori static (SRAM) yang harganya mahal, sedangkan memori utama menggunakan memori dynamic (DRAM) yang jauh lebih murah.

Sistem komputer akan bekerja sangat cepat apabila seluruh sistem memori utamanya menggunakan memori static, hanya saja harga sistem komputer akan menjadi sangat mahal. Selain itu, karena hamburan panas pada memori static lebih besar, sistem komputer yang menggunakan memori static yang besar akan menghasilkan panas yang berlebihan. Dua tipe cache yang terdapat di komputer adalah memory caching dan disk caching. Memori caching ini yang sering disebut dengan kas memori yang tersusun dari komputer jenis SRAM yang berkecepatan tinggi. Sedangkan disk caching menggunakan sebagian dari memori sekunder komputer. Istilah penting sekitar cache:

- ❖ Cache hit, jika data yang diminta oleh unit yang lebih tinggi yang ada dalam cache maka disebut hit. Permintaan dapat dilayani dengan cepat.
- ❖ Maksud urutan unit dari rendah hingga tinggi adalah streamer-harddisk memory-second level-first lever-CPU cache.
- ❖ Cache mist, bila data yang diminta tidak ada dalam cache harus diambil dari unit di bawahnya. Yang cukup memakan waktu. Ini disebut mist (gagal).
- ❖ Burst mode, dalam modus cepat ini cache mengambil banyak data sekaligus dari unit di bawahnya. Ia mengambil lebih dari yang dibutuhkan dengan asumsi bahwa data yang diminta berikutnya akan diletakkan berdekatan.
- ❖ LRU, algoritma pengganti cache
- ❖ COAST, adalah bentuk khusus L2 cache yang dapat diganti-ganti seperti RAM seperti RAM dan ditempatkan pada modul.
- ❖ First Level Cache L1, tingkat cache teratas dalam hirarki dengan kapasitas memori terkecil, termahal dan tercepat.
- ❖ L2, cache level ini memiliki kapasitas lebih besar dari L1 tetapi lebih lambat. Cache L2 masih lebih cepat dibanding RAM. Dahulu L2 cache disimpan di luar processor dan dapat ditambahkan akan tetapi seiring menurunnya biaya produksi serta untuk meningkatkan biaya kinerja, akhirnya cache ditanamkan di processor.
- ❖ Write Back (WB), cache digunakan tidak hanya saat membaca tapi juga dalam proses menulis.
- ❖ WT, cache hanya digunakan saat membaca, sedangkan untuk menulis akan ditunggu hingga memori yang dituju selesai menulis.

Modul Online 1

Karakteristik lainnya adalah kapasitas. Kapasitas memori internal maupun eksternal biasanya dinyatakan dalam bait (1bait = 8bit) atau word. Panjang word umumnya 8, 16, 32, 64 bit. Memori eksternal umumnya lebih besar kapasitasnya daripada memori internal, hal ini disebabkan karena teknologi dan sifat penggunaannya yang berbeda. Karakteristik selanjutnya adalah satuan transfer. Bagi memori internal, satuan transfer samadengan jumlah saluran data yang masuk dan keluar dari modul memori. Jumlah saluran ini seringkali sama dengan panjang word, tapi dimungkinkan juga sama. Tiga konsep yang berhubungan adalah:

- ❖ Word, merupakan satuan alami organisasi komputer. Ukuran word biasanya sama dengan jumlah bit yang digunakan untuk representasi bilangan dan panjang instruksi.
- ❖ Addressable unit. Pada sejumlah sistem, addressible unit adalah word. Namun terdapat sistem dengan pengalamatan pada tingkatan bait. Pada semua kasus hubungan anantara panjang A suatu alamat dan jumlah N addressable unit adalah $2A=N$.
- ❖ Unit of transfer, adalah jumlah bit yang dibaca atau dituliskan kedalam memori pada suatu saat. Pada memori eksternal, transfer data biasanya lebih besar dari 1 word, yang disebut block.

Perbedaan yang besar yang terdapat pada sejumlah memori adalah metode aksesnya. Terdapat 4 macam memori akses, yaitu :

- ❖ Sequential acces, memori di organisasi menjadi unit unit data yang disebut record. Akses harus dibuat dalam bentuk urutan linear yang spesifik. Informasi pengalamatan yang disimpan dipakai untuk memisahkan record-record dan untuk membsntu proses pencarian. Terdapat write mechanism untuk penulisan dan pembacaan memorinya. Pita magnetic merupakan memori yang menggunakan metode sequential acces.
- ❖ Direct acces, sama dengan sequential acces dimana terdapat shared read. Setiap block dan record memiliki alamat yang unit berdasarkan lokasi fisiknya. Akses dilakukan pada alamat memori. Disk adalah contoh direct acces.
- ❖ Random acces. Setiap lokasi memory dipilih secara random dan diakses dan dialamati secara langsung. Contohnya random acces adalah memori utama.
- ❖ Asosiative access, merupakan jenis random access yang memungkinkan perbandingan lokasi bit yang diinginkan untuk pencocokan. Jadi data dicari berdasarkan isinya, bukan alamatnya dalam memori. Contohnya: cache memori.

Berdasarkan karakteristik unjuk kerja, memori memiliki 3 parameter utama pengukuran unjuk kerja, yaitu:

- ❖ Acces time
- ❖ Memori cache time
- ❖ Transfer rate

Keandalan Memori

Memori harus mampu mengikuti kecepatan CPU sehingga terjadi sinkronisasi kerja antar CPU dan memori tanpa adanya waktu tunggu karena komponen lain belum

Modul Online 1

selesai prosesnya. Mengenai harga, sangatlah relatif. Bagi produsen selalu mencari harga produksi paling murah tanpa mengorbankan kualitasnya untuk memiliki daya saing di pasaran. Hubungan harga, kapasitas dan waktu akses adalah :

- ❖ Semakin kecil waktu akses, semakin besar harga per bitnya.
- ❖ Semakin besar kapasitas, semakin kecil harga per bitnya.
- ❖ Semakin besar kapasitas, semakin besar waktu aksesnya.

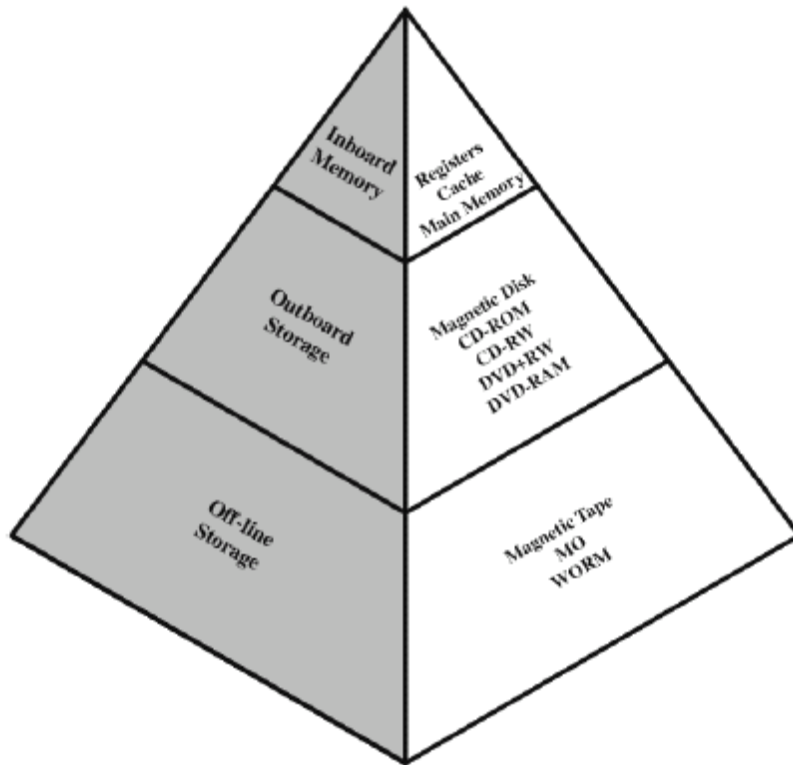
Dilema yang dihadapi para perancang adalah keinginan menerapkan teknologi untuk kapasitas memori yang besar karena harga per bit yang murah namun hal itu dibatasi oleh teknologi dalam memperoleh waktu akses yang cepat. Salah satu pengorganisasian masalah ini adalah menggunakan hirarki memori. Seperti terlihat pada tabel 2, bahwa semakin menurunnya hirarki maka hal berikut akan terjadi :

- ❖ Penurunan harga/bit
- ❖ Peningkatan kapasitas
- ❖ Peningkatan waktu akses
- ❖ Penurunan frekuensi akses memori oleh CPU.

Kunci keberhasilan hirarki ini pada penurunan frekuensi aksesnya. Semakin lambat memori maka keperluan CPU untuk mengaksesnya semakin sedikit. Secara keseluruhan system komputer akan tetap cepat namun kebutuhan kapasitas memori besar terpenuhi.

Tabel 2 Spesifikasi Memori

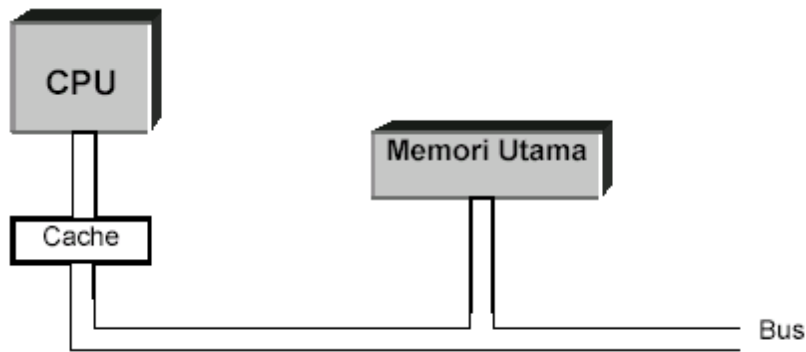
| Tipe memori | Teknologi | Ukuran | Waktu akses |
|---------------|----------------------|--------------|--------------------|
| Cache Memory | semikonduktor RAM | 128 – 512 KB | 10 ns |
| Memori Utama | semikonduktor RAM | 4 – 128 MB | 50 ns |
| Disk magnetik | Hard Disk | Gigabyte | 10 ms, 10MB/det |
| Disk Optik | CD-ROM | Gigabyte | 300ms, 600KB/det |
| Pita magnetik | Tape | 100 MB | Det -mnt, 10MB/mnt |



Gambar 1 Hirarki Memory

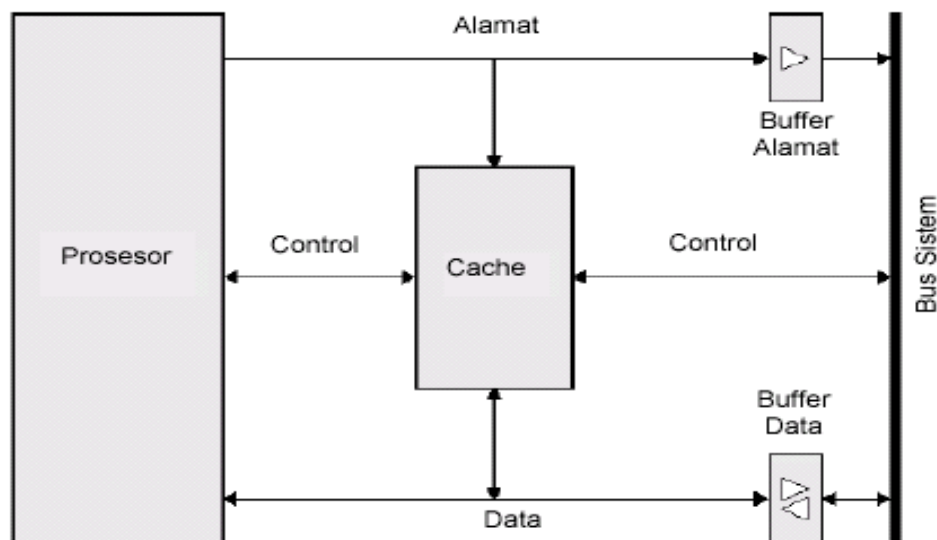
Prinsip Cache Memori

Cache memori difungsikan mempercepat kerja memori sehingga mendekati kecepatan prosesor. Dalam organisasi komputer, memori utama lebih besar kapasitasnya namun lambat operasinya, sedangkan cache memori berukuran kecil namun lebih cepat. Cache memori berisi salinan memori utama. Pada saat CPU membaca sebuah word memori, maka dilakukan pemeriksaan untuk mengetahui apakah word tersebut berada dalam cache memori. Bila ada dalam cache memori maka dilakukan pengiriman ke CPU, bila tidak dijumpai maka dicari dalam memori utama, selanjutnya blok yang berisi sejumlah word tersebut dikirim ke cache memori dan word yang diminta CPU dikirimkan ke CPU dari cache memori. Karena fenomena lokalitas referensi, ketika blok data diberikan ke dalam cache memori, terdapat kemungkinan bahwa word-word berikutnya yang berada dalam satu blok akan diakses oleh CPU. Konsep ini yang menjadikan kinerja memori lebih baik.



Gambar 2 Hubungan cache memori

Sehingga dapat disimpulkan bahwa kerja cache adalah antisipasi terhadap permintaan data memori yang akan digunakan CPU. Apabila data diambil langsung dari memori utama bahkan memori eksternal akan memakan waktu lama yang menyebabkan status tunggu pada prosesor. Ukuran cache memori adalah kecil, semakin besar kapasitasnya maka akan memperlambat proses operasi cache memori itu sendiri, disamping harga cache memori yang sangat mahal.



Gambar 3 Struktur/Organisasi Cache Memori

Elemen Perancangan cache memori

Elemen pada memori Cache, terdiri dari ;

- (1) Ukuran Cache
- (2) Fungsi Pemetaan
- (3) Algoritma Penggantian
- (4) Ukuran Blok

Penjelasan ;

(1) Ukuran cache

Menentukan ukuran memori cache sangatlah penting untuk mendongkrak kinerja komputer. Dari segi harga cache sangatlah mahal tidak seperti memori utama. Semakin besar kapasitas cache tidak berarti semakin cepat prosesnya, dengan ukuran besar akan terlalu banyak gate pengalamatannya sehingga akan memperlambat proses. Kita bisa melihat beberapa merek prosesor di pasaran beberapa waktu lalu. AMD mengeluarkan prosesor K5 dan K6 dengan cache yang besar (1MB) tetapi kinerjanya tidak bagus. Kemudian Intel pernah mengeluarkan prosesor tanpa cache untuk alasan harga yang murah, yaitu seri Intel Celeron pada tahun 1998-an hasil kinerjanya sangat buruk terutama untuk operasi data besar, floating point, 3D. Intel Celeron versi berikutnya sudah ditambah cache sekitar 128KB. Lalu berapa idealnya kapasitas cache? Sejumlah penelitian telah menganjurkan bahwa ukuran cache antara 1KB dan 512KB akan lebih optimum [STA96]. Ukuran cache disesuaikan dengan kebutuhan untuk membantu kerja memori. Semakin besar ukuran cache semakin lambat karena semakin banyak jumlah gerbang dalam pengalamatan cache.

(2) Fungsi Pemetaan (Mapping Function)

Pemetaan langsung

Pemetaan langsung adalah teknik yang paling sederhana, yaitu teknik ini memetakan blok memori utama hanya ke sebuah saluran cache saja.

$$i = j \text{ modulus } m \text{ dan } m = 2r$$

Dimana :

i = nomer saluran cache

j = nomer blok memori utama

m = jumlah saluran yang terdapat dalam cache

Fungsi pemetaan diimplementasikan dengan menggunakan alamat, yang terdiri dari tiga field (tag, line, word). w = word, adalah bit paling kurang berarti yang mengidentifikasi word atau byte unik dalam blok memori utama. s = byte sisa word yang menspesifikasi salah satu dari 2^s blok memori utama. Cache logik menginterpretasikan bit – bit S sebagai suatu tag s – r bit (bagian paling berarti dalam alamat) dan field saluran r bit.

Efek pemetaan tersebut adalah blok – blok memori utama diberikan ke saluran cache seperti berikut ini:

Jadi dalam metode ini pemetaan adalah bagian alamat blok memori utama sebagai nomer saluran cache. Ketika suatu blok data sedang diakses atau dibaca terhadap saluran yang diberikan, maka perlu memberikan tag bagi data untuk membedakannya dengan blok–blok lain yang dapat sesuai dengan saluran tersebut.

Disajikan contoh pemetaan langsung dengan m = 16K, maka pemetaannya:

Perlu diketahui bahwa tidak ada dua buah blok yang dipetakan ke nomer saluran yang sama memiliki tag sama. Sehingga 000000, 010000,, FF0000 masing-masing memiliki tag 00, 01,, FF.

Contoh pemetaan langsung:

Teknik pemetaan ini sederhana dan mudah diimplementasikan, namun kelemahannya adalah terdapat lokasi cache yang tetap bagi sembarang blok-blok yang diketahui. Dengan demikian, apabila suatu program berulang-ulang melakukan word referensi dari dua blok yang berbeda memetakan saluran yang sama maka blok-blok itu secara terus-menerus akan di-swap ke dalam cache sehingga hit rasionya akan rendah.

Pemetaan Asosiatif

Pemetaan asosiatif mengatasi kekurangan pemetaan langsung dengan cara setiap blok memori utama dapat dimuat ke sembarang saluran cache. Alamat memori utama diinterpretasikan dalam field tag dan field word oleh kontrol logika cache. Tag secara unik mengidentifikasi sebuah blok memori utama. Mekanisme untuk mengetahui suatu blok dalam cache dengan memeriksa setiap tag saluran cache oleh kontrol logika cache. Dengan pemetaan ini didapat fleksibilitas dalam penggantian blok baru yang ditempatkan dalam cache. Algoritma penggantian dirancang untuk memaksimalkan hit ratio, yang pada pemetaan langsung terdapat kelemahan dalam bagian ini. Kekurangan pemetaan asosiatif adalah kompleksitas rangkaian sehingga mahal secara ekonomi.

Pemetaan Asosiatif Set

Pemetaan asosiatif set menggabungkan kelebihan yang ada pada pemetaan langsung dan pemetaan asosiatif. Memori cache dibagi dalam bentuk set-set. Pemetaan asosiatif set prinsipnya adalah penggabungan kedua pemetaan sebelumnya. Alamat memori utama diinterpretasikan dalam tiga field, yaitu: field tag, field set, dan field word. Hal ini mirip dalam pemetaan langsung. Setiap blok memori utama dapat dimuat dalam sembarang saluran cache. Dalam pemetaan asosiatif set, cache dibagi dalam v buah set, yang masing-masing terdiri dari k saluran. Hubungan yang terjadi adalah :

$$m = v \times k$$

dimana ;

$i = j$ modulus v dan $v = 2d$ dimana :

i = nomer set cache

j = nomer blok memori utama

m = jumlah saluran pada cache

contoh yang menggunakan pemetaan asosiatif set dengan dua saluran pada masing-masing set, yang dikenal sebagai asosiatif set dua arah. Nomor set mengidentifikasi set unik dua saluran di dalam cache. Nomor set ini juga memberikan jumlah blok di dalam memori utama, modulus 2. Jumlah blok menentukan pemetaan blok terhadap saluran. Sehingga blok-blok 000000, 00A000,..., FF1000 pada memori utama dipetakan terhadap set 0 cache. Sembarang blok tersebut dapat dimuatkan ke salah

satu dari kedua saluran di dalam set. Perlu dicatat bahwa tidak terdapat dua blok yang memetakannya terhadap set cache yang sama memiliki nomor tag yang sama. Untuk operasi read, nomor set dipakai untuk menentukan set dua saluran yang akan diuji. Kedua saluran di dalam set diuji untuk mendapatkan yang cocok dengan nomor tag alamat yang akan diakses.

Penggunaan dua saluran per set ($v = m/2$, $k = 2$), merupakan organisasi asosiatif set yang paling umum. Teknik ini sangat meningkatkan hit ratio dibandingkan dengan pemetaan langsung.

Asosiatif set empat arah ($v = m/4$, $k = 4$) memberikan peningkatan tambahan yang layak dengan penambahan harga yang relatif rendah. Peningkatan lebih lanjut jumlah saluran per set hanya memiliki efek yang sedikit.

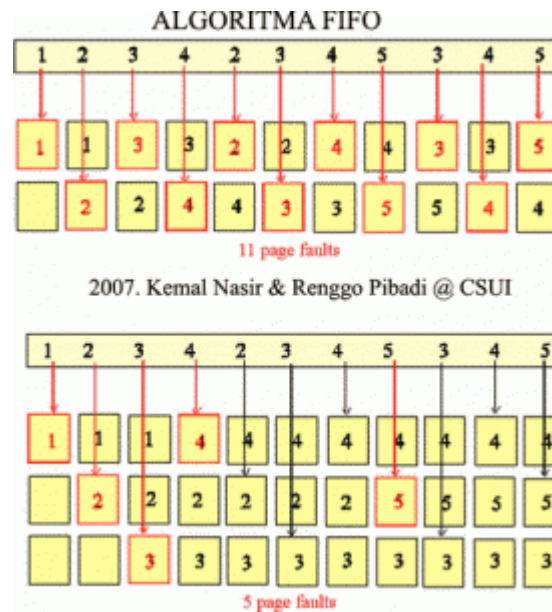
(3) Algoritma Penggantian

Yang dimaksud Algoritma Penggantian adalah suatu mekanisme pergantian blok-blok dalam memori cache yang lama dengan data baru. Dalam pemetaan langsung tidak diperlukan algoritma ini, namun dalam pemetaan asosiatif dan asosiatif set, algoritma ini mempunyai peranan penting untuk meningkatkan kinerja cache memori.

Banyak algoritma penggantian yang telah dikembangkan, namun dalam buku ini akan dijelaskan algoritma yang umum digunakan saja. Algoritma yang paling efektif adalah Least Recently Used (LRU), yaitu mengganti blok data yang terlama berada dalam cache dan tidak memiliki referensi. Algoritma lainnya adalah:

First In First Out (FIFO),

yaitu mengganti blok data yang awal masuk. Inti dari algoritma ini adalah simple / paling sederhana karena prinsipnya sama seperti prinsip antrian tak berprioritas. Page yang masuk terlebih dahulu maka yaitu yang akan keluar duluan juga. Untuk algoritma ini menggunakan structure data stack. Jadi kerjanya yaitu dimana kalau tidak ada frame yang kosong saat terjadi page fault maka korban yang dipilih adalah frame dengan stack paling bawah seperti halnya halaman yang sudah lama tersimpan didalam memory maka dari itu algoritma ini juga bisa memindahkan page yang sering digunakan.



Gambar 4 Contoh gambar dari Algoritma page FIFO

Utamanya algoritma ini di anggap cukup mengatasi pergantian page sampai pada tahun 70-an, pada saat itu juga Belady menemukan keganjalan pada algoritma ini dan dikenal dengan anomali Belady. Anomali Belady itu sendiri ialah keadaan dimana page fault rate meningkat seiring dengan pertambahannya jumlah frame.



Gambar 5 Contoh gambar dari Anomali Belady Algoritma page Modifikasi FIFO

Algoritma FIFO murni jarang digunakan, tetapi dikombinasikan (modifikasi). Kelemahan FIFO yang jelas adalah algoritma dapat memilih memindahkan page yang sering digunakan yang lama berada di memori. Kemungkinan ini dapat dihindari dengan hanya memindahkan page tidak diacu Page ditambah bit R mencatat apakah page diacu atau tidak. Bit R bernilai 1 bila diacu dan bernilai 0 bila tidak diacu.

Variasi dari FIFO antara lain:

- ❖ Algoritma penggantian page kesempatan kedua (second chance page replacement algorithm)
- ❖ Algoritma penggantian clock page (clock page replacement algorithm)

Algoritma yang pertama adalah algoritma second chance. Algoritma second chance berdasarkan pada algoritma FIFO yang disempurnakan. Algoritma ini menggunakan tambahan berupa reference bit yang nilainya 0 atau 1. Jika dalam FIFO menggunakan stack, maka second chance menggunakan circular queue. Halaman yang baru di-load atau baru digunakan akan diberikan nilai 1 pada reference bit-nya. Halaman yang reference bit-nya bernilai 1 tidak akan langsung diganti walaupun dia berada di antrian paling bawah (berbeda dengan FIFO). Urutan langkah kerja algoritma second chance adalah sebagai berikut:

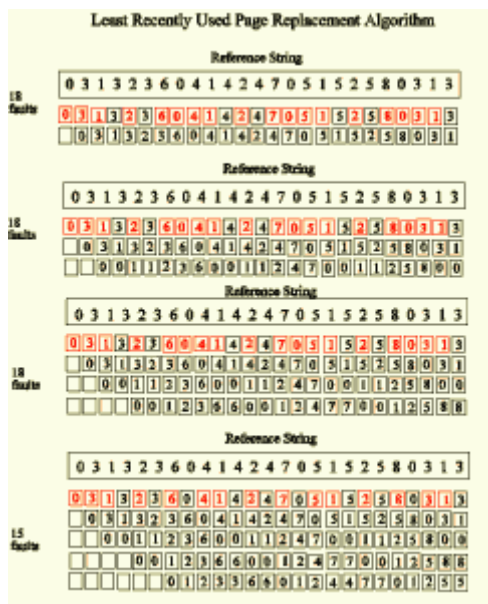
- (1) apabila terjadi page fault dan tidak ada frame yang kosong, maka akan dilakukan razia (pencarian korban) halaman yang reference bit-nya bernilai 0 dimulai dari bawah antrian (seperti FIFO).
- (2) Setiap halaman yang tidak di-swap (karena reference bit-nya bernilai 1), setiap dilewati saat razia reference bit-nya akan diset menjadi 0.



Gambar 6 Contoh gambar dari Algoritma page Modifikasi FIFO

Least Frequently Used (LFU)

Least Frequently Used (LFU) adalah mengganti blok data yang mempunyai referensi paling sedikit. Dikarenakan algoritma optimal sangat sulit dalam pengimplementasiannya, maka dibuatlah algoritma lain yang performance-nya mendekati algoritma optimal dengan sedikit cost yang lebih besar. Sama seperti algoritma optimal, algoritma LRU tidak mengalami anomali Belady. Algoritma ini memakai linked list untuk mendata halaman mana yang paling lama tidak terpakai. Linked list inilah yang membuat cost membesar, karena harus meng-update linked list tiap saat ada halaman yang di akses.



Gambar 7 Contoh gambar dari Algoritma page LRU

Teknik lain adalah **algoritma Random**, yaitu penggantian tidak berdasarkan pemakaian datanya, melainkan berdasar slot dari beberapa slot kandidat secara acak.

WRITE POLICY

Kebijakan menulis cache yang menentukan bagaimana menangani menulis ke lokasi memori yang saat ini ditahan dalam cache. Dijelaskan lebih rinci di sini, kedua jenis kebijakan adalah:

- ❖ Write-Back Cache: yaitu teknik meminimasi penulisan dengan cara penulisan pada cache saja. Ketika sistem menulis ke lokasi memori yang saat ini dipegang dalam cache, itu hanya menulis informasi baru pada baris cache yang sesuai. Ketika garis cache akhirnya diperlukan untuk beberapa alamat memori lain, data yang diubah adalah "menulis kembali" ke memori sistem. Jenis cache yang memberikan performa yang lebih baik dari cache write-through, karena menghemat (memakan waktu) menulis siklus ke memori.
- ❖ Write-Through Cache: yaitu operasi penulisan melibatkan data pada memori utama dan sekaligus pada cache memori sehingga data selalu valid. Ketika sistem menulis ke lokasi memori yang saat ini dipegang dalam cache, itu menulis informasi baru baik ke baris cache yang tepat dan lokasi memori itu sendiri pada saat yang sama. Jenis cache menyediakan kinerja buruk dari menulis-kembali, tapi lebih sederhana untuk menerapkan dan memiliki keuntungan dari konsistensi internal, karena cache tidak pernah tidak sinkron dengan memori apa adanya dengan cache write-back. Kekurangan teknik ini adalah menjadikan lalu lintas data ke memori utama dan cachesangattinggi sehingga mengurangi kinerja sistem, bahkan bisa terjadi hang.
- ❖ Bus Watching with Write Through yaitu setiap cache controller akan memonitoring bus alamat untuk mendeteksi adanya operasi tulis. Apabila ada operasi tulis di alamat yang datanya digunakan bersama maka cache controller akan menginvalidasi data cache-nya.

Modul Online 1

- ❖ Hardware Transparency yaitu adanya perangkat keras tambahan yang menjamin semua updating data memori utama melalui cache direfleksikan pada seluruh cache yang ada.
- ❖ Non Cacheable Memory yaitu hanya bagian memori utama tertentu yang digunakan secara bersama. Apabila ada mengakses data yang tidak di share merupakan kegagalan cache.

Apabila suatu data telah diletakkan pada cache maka sebelum ada penggantian harus dicek apakah data tersebut telah mengalami perubahan. Apabila telah berubah maka data pada memori utama harus di-update. Masalah penulisan ini sangat kompleks, apalagi memori utama dapat diakses langsung oleh modul I/O, yang memungkinkan data pada memori utama berubah.

LINE SIZE

Dalam Tipografi Line size adalah lebar ditempati oleh blok mengeset teks diukur dalam inci, picas dan poin. Sebuah blok teks atau paragraf memiliki panjang garis maksimum yang sesuai dengan desain yang ditentukan. Panjang garis ditentukan oleh parameter tipografi berdasarkan formal jaringan dan template dengan beberapa tujuan dalam pikiran, keseimbangan dan fungsi untuk fit dan mudah dibaca dengan kepekaan terhadap gaya estetika dalam tipografi. Typographers menyesuaikan panjang baris untuk membantu keterbacaan atau pas copy. Teks dapat siram kiri dan kanan compang-camping, rata kanan kiri dan bergerigi, atau dibenarkan mana semua garis dengan panjang yang sama. Dalam panjang garis yang tepat pengaturan compang-camping bervariasi untuk membuat tepi kanan compang-camping garis yang bervariasi panjangnya. Kadang-kadang ini dapat secara visual memuaskan. Untuk pengaturan yang benar dibenarkan dan compang-camping typographers dapat menyesuaikan panjang baris untuk menghindari yang tidak diinginkan tanda hubung, sungai ruang putih, dan kata-kata yatim / karakter pada akhir baris (misalnya: "The", "Aku", "Dia", "Kami").

(4) Ukuran Blok

Elemen rancangan yang harus diperhatikan lagi adalah ukuran blok. Telah dijelaskan adanya sifat lokalitas referensi maka nilai ukuran blok sangatlah penting. Apabila blok berukuran besar ditransfer ke cache akan menyebabkan hit ratio mengalami penurunan karena banyaknya data yang dikirim disekitar referensi. Tetapi apabila terlalu kecil, dimungkinkan memori yang akan dibutuhkan CPU tidak tercakup. Apabila blok berukuran besar ditransfer ke cache, maka akan terjadi :

- ❖ Blok-blok yang berukuran lebih besar mengurangi jumlah blok yang menempati cache. Karena isi cache sebelumnya akan ditindih.
- ❖ Dengan meningkatnya ukuran blok maka jarak setiap word tambahan menjadi lebih jauh dari word yang diminta, sehingga menjadi lebih kecil kemungkinannya digunakan cepat.

Hubungan antara ukuran blok dan hit ratio sangat rumit untuk dirumuskan,

Modul Online 1

tergantung pada karakteristik lokalitas programnya dan tidak terdapat nilai optimum yang pasti telah ditemukan. Ukuran antara 4 hingga 8 satuan yang dapat dialamati (word atau byte) cukup beralasan untuk mendekati nilai optimum