Tugas 1: Algoritma Penggantian dalam Manajemen Cache

Algoritma penggantian dalam manajemen cache memainkan peran penting dalam menentukan blok mana yang harus diganti ketika cache penuh dan ada permintaan untuk data baru. Berikut adalah penjelasan mengenai beberapa algoritma penggantian utama dan dampaknya terhadap kinerja sistem:

Least Recently Used (LRU)

- Cara Kerja: LRU mengganti blok yang paling lama tidak digunakan.- Kelebihan: Efektif dalam menangani pola akses yang berulang karena menjaga blok yang sering digunakan di cache.- Kekurangan: Sulit diimplementasikan secara efisien karena memerlukan pembaruan terus-menerus untuk melacak blok terakhir yang digunakan.

First-In-First-Out (FIFO)

- Cara Kerja: FIFO mengganti blok yang paling awal dimasukkan ke cache.- Kelebihan: Implementasi sederhana dan mudah.- Kekurangan: Tidak efektif untuk pola akses yang berulang karena mungkin mengganti blok yang masih sering digunakan.

Least Frequently Used (LFU)

- Cara Kerja: LFU mengganti blok yang paling jarang digunakan.- Kelebihan: Lebih efektif untuk pola akses yang tidak teratur karena menjaga blok yang sering digunakan di cache.- Kekurangan: Memerlukan penghitungan frekuensi akses yang dapat menambah kompleksitas dan biaya implementasi.

Perbandingan dalam Lingkungan Komputasi yang Berbeda

- Dalam sistem dengan pola akses memori yang teratur, LRU cenderung lebih efektif daripada FIFO.- Dalam sistem dengan pola akses yang tidak teratur atau acak, LFU dapat memberikan kinerja yang lebih baik.- FIFO sering digunakan dalam sistem sederhana atau dengan batasan sumber daya karena kemudahan implementasinya.

Tugas 2: Pemetaan Cache

Untuk memori utama sebesar 4 GB, memori cache sebesar 16 MB, dan ukuran blok 1 KB, kita akan menunjukkan format alamat memori utama menggunakan pemetaan set-associative untuk cache 8-way associative.

Format Alamat Memori:

1. Ukuran Memori Utama: 4 GB = 2^32 byte2. Ukuran Cache: 16 MB = 2^24 byte3. Ukuran Blok: 1 KB = 2^10 byte4. Jumlah Set: 2^24 / (2^10 \* 8) = 2^11

Alamat memori akan dibagi menjadi tiga bagian:- Tag: Digunakan untuk membedakan antara blok yang berbeda yang dapat dimuat ke set yang sama.- Indeks: Digunakan untuk menentukan set mana di mana blok harus dimuat.- Offset: Digunakan untuk mengakses byte tertentu dalam blok.

- Offset: 10 bit (karena ukuran blok adalah 1 KB)- Indeks: 11 bit (karena jumlah set adalah 2^11)- Tag: 32 - 10 - 11 = 11 bit

Format alamat memori akan menjadi:- Tag: 11 bit- Indeks: 11 bit- Offset: 10 bit

Tugas 3: Contoh Pemetaan Alamat Memori

Misalnya, jika kita memiliki alamat memori 0x0040F3A8:

1. Konversi ke Biner: 0000 0000 0100 0000 1111 0011 1010 10002. Pembagian Alamat: - Tag: 0000 0000 010 - Indeks: 0000 1111 001 - Offset: 1010 1000

Dengan demikian, alamat 0x0040F3A8 akan dimuat ke set dengan indeks 0000 1111 001 di cache.Dengan penjelasan di atas, kita telah menjelaskan cara kerja algoritma penggantian dalam cache, format alamat memori untuk pemetaan cache, dan memberikan contoh pemetaan alamat memori ke cache.