

Sistem Operasi

(Kuis)

Nama : Aditia Putra Mahendra

Npm : 5220411234

Algoritma

1. Optimal (OPT)

Algoritma yang berfungsi untuk memilih halaman yang tidak akan digunakan untuk waktu yang paling lama di masa depan.

Kriteria :

Kesesuaian dengan Konteks, di katakan memiliki kriteria tersebut karena, lebih sesuai untuk situasi di mana prediksi akurat tentang akses halaman di masa mendatang dapat diperoleh.

Contoh :

sebuah soal dengan angka-angka di bawah ini :

1 2 3 4 1 2 5 1 2 3 4 5

lalu buat dengan 4 frame dan carilah page faultnya?

Jawab :

1 2 3 4 1 2 5 1 2 3 4 5

1 1 1 1 * 1 * 4

2 2 2 * 2 * 2

3 3 3 * 3

4 5 5 *

Caranya adalah masukan angka yang di soal ke masing2 page fault seperti yang diatas, jika ada kesamaan dengan yang di depannya maka tidak usah di hitung (yang di beri tanda bintang (*)), jika tidak ada maka anda analisa seperti konsep yang tadi di jelaskan, dan kemudian ganti angka yang di page fault dengan yang di soal.

2. Least Recently Used (LRU)

Algoritma yang berfungsi untuk membuang proses yang paling lama tersimpan pada memori atau paling jarang digunakan dari sekumpulan proses-proses yang ada.

Kriteria :

Fleksibilitas dan Adaptasi, di katakan memiliki kriteria tersebut karena, algoritma ini dapat diadaptasi untuk berbagai situasi dan bekerja dengan baik dalam banyak skenario aplikasi.

Contoh :

Dalam sistem pengelolaan file, algoritma LRU dapat digunakan untuk mempertahankan cache file. File yang tidak sering diakses akan dikeluarkan dari cache untuk memberikan ruang bagi file-file yang lebih baru atau lebih sering diakses.

3. Second Change (Clock)

Algoritma yang berfungsi untuk penggantian halaman yang dilakukan berdasarkan konsep jam (Clock). Dalam algoritma ini, setiap halaman memiliki sebuah bit referensi yang menunjukkan apakah halaman itu telah diakses sejak terakhir kali dicek.

Kriteria :

Efisiensi Implementasi, di katakana memiliki kriteria tersebut karena, algoritma ini relatif mudah diimplementasikan dengan menggunakan konsep jam untuk menyimpan urutan halaman.

Contoh :

Suatu sistem operasi menggunakan algoritma Second Chance untuk mengelola cache penyimpanan cepat (fast storage cache) di dalamnya. Cache ini digunakan untuk menyimpan sejumlah kecil halaman data yang paling sering diakses oleh program-program yang berjalan.

4. Enhanced Second Change (Clock)

Algoritma varian dari algoritma Second Chance yang ditingkatkan dengan menambahkan tingkat prioritas. Ini memungkinkan algoritma untuk lebih efisien mengelola cache dan memberikan prioritas lebih tinggi pada halaman-halaman yang lebih penting.

Kriteria :

Prioritas Tinggi, di katakana memiliki kriteria tersebut karena, algoritma ini memberikan prioritas lebih tinggi pada halaman-halaman yang dianggap lebih penting. Prioritas ini dapat ditentukan oleh faktor-faktor seperti frekuensi akses atau bobot.

Contoh :

Prioritas Berdasarkan Frekuensi Akses:

Halaman file yang sering diakses diberi prioritas lebih tinggi. Sebagai contoh, file yang baru saja diunduh oleh pengguna akan mendapatkan prioritas tinggi karena kemungkinan besar akan sering diakses dalam waktu dekat.

5. Counting

Algoritma pengurutan yang berbasis pada pemahaman distribusi frekuensi elemen-elemen dalam kumpulan data.

Kriteria :

Frekuensi Distribusi Elemen, di katakan memiliki kriteria tersebut karena, memerlukan pemahaman atau perhitungan distribusi frekuensi elemen-elemen yang akan diurutkan.

Contoh :

Data Awal:

[2, 5, 1, 0, 3, 2, 4, 1, 0, 5, 2, 3, 1, 4, 4, 0]

Hitung Frekuensi:

Hitung frekuensi masing-masing nilai dalam kumpulan data. Misalnya, ada 3 buah nilai 0, 3 buah nilai 1, dan seterusnya.

Distribusi Frekuensi:

Distribusikan kembali nilai-nilai berdasarkan frekuensinya. Misalnya, posisikan 3 nilai 0 di awal, 3 nilai 1 selanjutnya, dan seterusnya.

Data Terurut:

Hasilnya setelah proses algoritma Counting adalah kumpulan data yang terurut: [0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5]

6. Page Buffering

Algoritma Page Buffering berfokus pada manajemen buffer atau penyimpanan sementara untuk halaman-halaman data yang dibaca atau ditulis dalam suatu sistem, seperti sistem manajemen basis data atau sistem operasi.

Kriteria :

Efficient I/O Handling, di katakan memiliki kriteria tersebut karena algoritma Page Buffering dirancang untuk menangani operasi input/output (I/O) dengan efisien. Hal ini mencakup strategi menyimpan halaman-halaman yang sering diakses dalam buffer untuk mengurangi keterlambatan I/O.

Contoh :

Buffer Management:

Algoritma Page Buffering akan menyimpan sejumlah halaman dari tabel karyawan dalam buffer. Jika seorang manajer sering mengakses data karyawan tertentu, halaman tersebut akan tetap di dalam buffer untuk menghindari membaca dari penyimpanan utama berulang kali.

Referensi :

<http://www.ocw.upj.ac.id/files/Slide-IFA214-IFA214-Slide-13-14.pdf>

<https://staffnew.uny.ac.id/upload/198812242014042002/pendidikan/PPT%20Algoritma%20dan%20Struktur%20Data%202014.pdf>

<https://cgeduntuksemua.blogspot.com/2012/05/contoh-soal-dan-jawaban-algoritma.html?m=1>

<https://www.scaler.com/topics/lru-page-replacement-algorithm/>

<https://journal.unsika.ac.id/index.php/syntax/article/download/3823/2375/9839>

<https://www.geeksforgeeks.org/second-chance-or-clock-page-replacement-policy/>