

Algoritma Pergantian Halaman pada memori

Oleh : M. Dhais Firmansyah

NIM : 162410101076

Abstraksi

Sistem Operasi merupakan sebuah pengatur atau jembatan antar user dengan perangkat keras, tentu jika berbicara tentang sistem operasi akan ada banyak algoritma yang gunanya untuk manajemen sumber daya komputer agar komputer dapat berjalan sesuai dengan apa yang kita inginkan, untuk itu, ada beberapa pengendalian memori yang gunanya untuk mengatur page atau intruksi yang akan dilakukan sesuai dengan kelebihan dan kekurangannya masing masing,

Pembahasan

Dalam Sistem Operasi terjadi banyak sekali manajemen proses yang ada pada memori. Untuk itu ada beberapa pengalaman untuk memanajemen *virtual memory* agar memori yang digunakan sesuai dengan yang dibutuhkan, beberapa algoritma pergantian halaman pada memori menentukan bagaimana cara mereka untuk mengalokasikan memori yang digunakan untuk proses yang dikerjakan. Bahkan algoritma tersebut bisa mengambil tindakan atas proses yang overload dan menentukan hal apa yang harus dilakukan oleh sistem komputer itu. Berikut adalah contoh dari beberapa algoritma yang ada pada Pergantian halaman pada memori

1. Pergantian Halaman Optimal

Pada algoritma ini setiap page akan dilabeli setiap tahapan, ketika page tersebut dibutuhkan untuk di swap maka page tersebut akan diganti dengan string acuan, semisal: page 1 baru akan digunakan lagi pada 6 detik selanjutnya, maka page tersebut akan digunakan untuk page lain yang akan dikerjakan selama 0.6 detik selanjutnya, lalu ketika proses awal tadi sudah mendekati waktu pengerjaan, barulah page 1 akan dikerjakan ulang

Algoritma ini merupakan algoritma idealis, namun sesungguhnya tidak dapat di implementasikan pada sistem operasi di dunia nyata, karena sangat tidak memungkinkan suatu komputer menghitung berapa lama proses tersebut akan dikerjakan, kecuali ketika semua software yang akan dijalankan pada sistem telah dikenali sebelumnya dan menerima suatu analisis statik dari pola memori tersebut, Jadi suatu program atau software tidak akan berjalan relatif itu mengapa algoritma ini tidak memungkinkan, algoritma ini sangat baik untuk menguji kemampuan suatu Hardware (Benchmark) karena pada benchmark telah ditentukan *run time process* program yang akan diujikan, dan dapat mengukur nilai waktu pada program yang akan diuji.

2. Pergantian Halaman NRU

Pergantian Halaman Not Recently Used (NRU) merupakan sebuah algoritma yang menyimpan perubahan terhadap suatu page, algoritma ini bisa dikatakan sebagai bit checker terhadap page yang sedang dikerjakan apakah page tersebut referenced dan

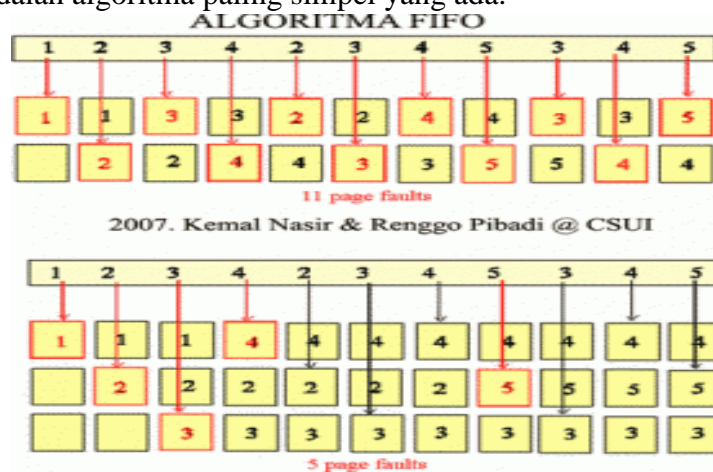
apakah page tersebut telah termodifikasi, sistem pengerjaan algoritma ini akan menelusuri nilai page terkecil yang akan dikerjakan terlebih dahulu.

R	M	Kelas	Keterangan
0	0	0	not referenced, not modified
0	1	1	not referenced, modified
1	0	2	referenced, not modified
1	1	3	referenced, modified

Gambar diatas menunjukkan nilai dari bit yang tersedia pada NRU, setiap mode kelas terbagi menjadi 4 bit, yang dimulai dari 0 (0,1,2,3). Tetapi nilai-nilai tersebut terlalu rancu untuk dijadikan nilai referensi dari suatu page. Page yang tidak memungkinkan adalah ketika page belum termodifikasi dan tidak tereferensi, hal ini hanya terjadi ketika bit pada kelas 3 telah direferensikan oleh *timer interrupt*, jadi semua nilai bitnya adalah 0, maka algoritma ini akan memanggil atau menjalankan page secara random karena semua nilai page nya sama jadi keluar dari 4 nilai dari bit-bit tersebut.

3. Pergantian Halaman FIFO

Pada algoritma pergantian halaman FIFO atau bisa disebut sebagai First In First Out (yang pertama masuk maka yang pertama keluar), algoritma ini adalah penerapan dari algoritma suatu antrian, jika terdapat beberapa page dalam komputer yang harus dikerjakan maka page yang pertama kali masuklah yang akan pertama kali dikerjakan, dan tentunya page tersebut juga akan menjadi page yang pertama kali keluar, Algoritma ini adalah algoritma paling simpel yang ada.



Pada penerapannya di keadaan nyata pada komputer, pergantian halaman FIFO mengalami permasalahan yaitu tidak dapat menentukan suatu skala prioritas dari page-page yang akan dikerjakan. Jadi page yang akan dikerjakan hanya dikerjakan secara berurutan saja sehingga ketika ada page yang seharusnya dikerjakan terlebih dahulu karena keadaan mendesak maka sistem akan men-delay page dan mengaturnya tetap sesuai antrian.

4. Pergantian Halaman Modifikasi FIFO (Second Chance)

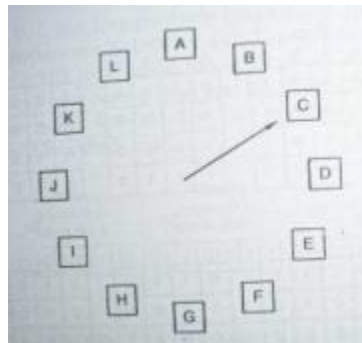
Second Chance atau bisa disebut sebagai hasil modifikasi dari algoritma FIFO yang tentunya mengalami perubahan yang lebih baik. Algoritma Second Chance melakukan pengurutan page sama seperti FIFO tapi algoritma ini melakukan pengecekan terhadap page yang akan dieksekusi dengan melakukan pembacaan terhadap page dan apakah page yang akan dieksekusi nilai bitnya telah diatur, jika tidak ada pengaturan nilai bit maka nilai bit yang direferensikan akan dihapus dan menaruh page tersebut ke belakang antrian, hal ini berulang terus menerus hingga menjadikan algoritma ini menjadi antrian melingkar, jika semua nilai bit yang direferensikan telah diatur pada pergantian halaman kedua dalam daftar pertama, maka page tersebut akan ditukar. Tapi jika semua page memiliki nilai referensi bit yang kosong maka algoritma Second Chance hanyalah akan menjadi algoritma FIFO.



Contoh gambar algoritma page replacement SCP

5. Pergantian Halaman Clock

Algoritma pergantian halaman clock adalah algoritma yang menyerupai algoritma FIFO dan lebih efisien daripada algoritma Second Chance, cara kerjanya juga sama dengan Second Chance yaitu dengan membuat baris lingkaran page seperti pada jarum jam, dan mengerjakan sesuai dengan urutan jarum jam, bedanya algoritma ini tidak selalu menukar page ke bagian belakang daftar halaman yang akan dikerjakan.



Cara pengecekan yang terjadi pada algoritma ini adalah ketika terjadi kesalahan maka halaman yang ditunjuk akan diperiksa, jika setelah diperiksa dan nilai bit referensi adalah 0 maka page akan diganti dengan page baru, serta pengerjaan dilanjutkan ke halaman selanjutnya, jika nilai bit referensi adalah 1 maka nilai bit R tersebut direset dan diganti menjadi 0, begitu seterusnya hingga menemui nilai bit R 0.

6. Pergantian Halaman LRU

Least Recently Used Page Replacement Algorithm

Reference String

18
faults

0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3
0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3
0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3

Reference String

18
faults

0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3	
0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3	
0	0	0	1	1	2	3	6	0	0	1	2	4	7	0	0	1	1	2	5	8	0	0			

Reference String

18
faults

0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3	
0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3	
0	0	0	1	1	2	3	6	0	0	1	2	4	7	0	0	1	1	2	5	8	0	0			
0	0	0	1	1	2	3	6	0	0	1	2	4	7	7	0	0	1	2	5	8	8				

Reference String

15
faults

0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3
0	3	1	3	2	3	6	0	4	1	4	2	4	7	0	5	1	5	2	5	8	0	3	1	3
0	0	0	1	1	2	3	6	0	0	1	1	2	4	7	0	0	1	1	2	5	8	0	0	0
0	0	0	1	1	2	3	6	0	0	1	2	4	7	7	0	0	1	2	5	8				
0	0	0	1	1	2	3	6	0	1	2	4	4	7	7	0	1	2	5						

Salah satu bagian dari LRU yang memiliki cost tertinggi adalah linked list, cara kerjanya adalah dengan menyimpan detail dari pengerjaan, seperti dibagian belakang daftar pengerjaan adalah page yang terakhir kali dieksekusi dan dibagian depan adalah page yang paling sering dilakukan, hal ini menyebabkan item item yang tersimpan di list akan selalu berubah yang setiap waktunya juga menghabiskan konsumsi dari proses dan memori.

7. Pergantian Halaman NFU

Algoritma NFU alias Not Frequently Used merupakan algoritma yang memiliki sebuah program counter, atau alat untuk menghitung page dalam daftar, pada awalnya semua nilai bit pada daftar page akan bernilai 0, namun seiring berjalannya interval waktu nilai bit akan berubah tergantung seberapa sering page tersebut digunakan. Contoh ketika setiap page memiliki 8 bit untuk counter dengan isi 00000000, maka

ketika page tertentu digunakan maka isi dari bit akan berubah dan di geser tergantung dari skala prioritas page atau di counter sendiri akan membaca nilai bit terbesar.

Kekurangan pada algoritma ini adalah hanya meninjau dari satu hal saja yakni seberapa sering digunakannya page tersebut dan tidak meninjau dari hal rentang waktu atau seberapa lama pengerjaan page yang paling sering digunakan, ini tentu akan mengakibatkan page yang lain akan menjadi minoritas dan bahkan mungkin tidak akan pernah dikerjakan karena page yang paling sering digunakan akan meminta untuk selalu dikerjakan tanpa batasan waktu, yang menjadikan looping tanpa henti.

8. Pergantian Halaman Aging

Algoritma aging adalah merupakan sebuah modifikasi algoritma dari NFU, cara kerja hampir sama dengan algoritma NFU, hanya saja algoritma aging dibuat untuk agar dapat memahami waktu pengerjaan dari page yang ada pada daftar. Cara kerja dari algoritma ini adalah dengan menggeser bit counter ke kanan (dibagi 2) dan menambahkan bit referenced di paling kiri. Sebagai contoh semisal sebuah page di akses pada satu clock pertama dan tidak diakses selama dua clock, lalu diakses lagi selama dua kali clock dan selanjutnya tidak diakses lagi pada satu clock maka akan menjadi 1,0,0,1,1,0.... Dan nilai counter akan menjadi 10000000, 01000000, 00100000, 10010000, 11001000, 01100100, ...

Algoritma ini menjamin bahwa halaman yang paling baru diakses, walaupun tidak sering diakses, memiliki prioritas lebih tinggi dibanding halaman yang sering diakses sebelumnya. Yang perlu diketahui, aging dapat menyimpan informasi pengaksesan halaman sampai 16 atau 32 interval sebelumnya. Hal ini sangat membantu dalam memutuskan halaman mana yang akan diganti. Dengan demikian, aging menawarkan kinerja yang mendekati optimal dengan harga yang cukup rendah.

9. Pergantian Halaman WorkingSet

Algoritma WorkingSet adalah merupakan algoritma yang digunakan untuk mengeksekusi page dalam beberapa interval waktu atau dengan penjadwalan yang lebih teratur, dari beberapa sumber algoritma WorkingSet adalah merupakan algoritma yang tidak masuk akal, karena algoritma ini membutuhkan banyak sekali data penunjang untuk eksekusi page yang akan dilakukan. WorkingSet sendiri memiliki data penunjang $w=(k,t)$ dimana k merupakan referensi bit yang berisi ambil intruksi, ambil data atau menyimpan data pada bit referensi, dan t adalah interval waktu. Untuk mendapatkan referensi page agar dapat menjalankan algoritma ini Sistem Operasi perlu untuk menyimpan banyak sekali info-info dari page yang ada di list, maka dari itu Algoritma ini termasuk algoritma yang memiliki cost sangat tinggi. Ketika page fault terjadi, page yang diluar working set dapat segera dihapus dari memori dan algoritma akan melakukan pengecekan keseluruhan page.

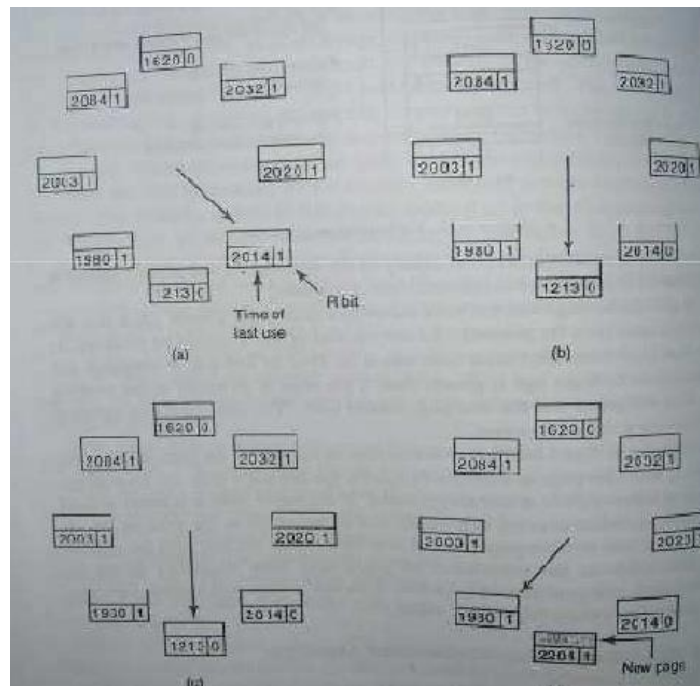
10. Pergantian Halaman WsClock

Algoritma WsClock merupakan gabungan dari algoritma WorkingSet dan clock, karena pada WorkingSet tidaklah praktis untuk pengerjaan ketika terjadi page fault maka dibentuklah gabungan antara dua algoritma ini, pada nyatanya algoritma ini memiliki suatu performa yang baik dan akhirnya banyak digunakan.

Cara kerja algoritma ini adalah pendistribusian page dengan bentuk sirkular menyerupai clock dan menyimpan informasi page dengan acuan R bit dan *time of last used*.

Langkah-langkah :

1. Periksa R pada page yg ditunjuk oleh jarum
2. Bila R=1, Set R menjadi 0 (khusus utk page pertama yg ditunjuk saja), gerakkan jarum ke arah berikutnya
3. Bila R=0, lihat nilai $age > T$ (hapus) atau $age \leq T$ (write isi page to disk clean page)
4. Bila semua page sudah dikunjungi dan kembali ke titik awal, bagaimana membedakannya?
5. Sedikitnya satu 'write isi page to disk' telah dilakukan (ambil clean page yg pertama dijumpai untuk di remove)
6. 'write isi page to disk' belum pernah dilakukan, ini artinya semua R=1 (tetapi page 1 skrng R=0). Ambil isi page 1 utk disalin ke disk page 1 dihapus.



Penutup

Kesimpulan

Untuk menjalankan tugasnya Sistem Operasi memiliki banyak sekali algoritma yang menjadi acuan untuk mengambil langkah yang akan dilakukan. Dan setiap algoritma memiliki kelebihan dan kekurangannya masing-masing. Tergantung dimana dan kapan algoritma tersebut akan digunakan dan dengan kondisi yang tepat

Referensi

<http://2010194-if-unsika.blogspot.co.id/2013/02/algoritma-penggantian-page.html>

<http://2009061-if-unsika.blogspot.co.id/2012/01/algoritma-penggantian-page.html>

http://2010046-if-unsika.blogspot.co.id/2013/01/algoritma-algoritma-penggantian-page_13.html

https://en.wikipedia.org/wiki/Page_replacement_algorithm

https://id.wikipedia.org/wiki/Algoritma_penggantian_halaman

<https://sites.google.com/a/student.unsika.ac.id/riadhi-sigaluhpakuan/home-1/algoritmapenggantianclockpage>

<https://reiyza.wordpress.com/2012/07/05/macam-macam-algoritma-page-replacement/>

<https://bedagres.wordpress.com/2012/07/08/algoritma-page-replace/>