IMPLEMENTATION OF ADDITIONAL-REFERNCE-BIT ALGORITHM IN C++

OPERATING SYSTEMS ASSIGNMENT



Vaishakh B N 1RV09CS114

Under the Guidance Of Ms Sandhya (Assistant Professor, Dept Of Computer Science RVCE)

INTRODUCTION

In a computer operating system that uses paging for virtual memory management, **page replacement algorithms** decide which memory pages to page out (swap out, write to disk) when a page of memory needs to be allocated. Paging happens when a page fault occurs and a free page cannot be used to satisfy the allocation, either because there are none, or because the number of free pages is lower than some threshold.

When the page that was selected for replacement and paged out is referenced again it has to be paged in (read in from disk), and this involves waiting for I/O completion. This determines the *quality* of the page replacement algorithm: the less time waiting for page-ins, the better the algorithm. A page replacement algorithm looks at the limited information about accesses to the pages provided by hardware, and tries to guess which pages should be replaced to minimize the total number of page misses, while balancing this with the costs (primary storage and processor time) of the algorithm itself.

**The Optimal Page Replacement Algorithm**

The best possible page replacement algorithm is easy to describe but impossible to implement. It goes like this. At the moment that a page fault occurs, some set of pages is in memory. One of these pages will be referenced on every next

instruction (the page containing that instruction). Other pages may not be referenced until 10, 100, or perhaps 1000 instructions later. Each page can be labeled with the number of instructions that will be executed before that page is first referenced.

The optimal page algorithm simply says that the page with the highest label should be removed. If one page will not be used for 8 million instructions and another page will not be used for 6 million instructions, removing the former

pushes the page fault that will fetch it back as far into the future as possible. Computers, like people, try to put off unpleasant events for as long as they can. The only problem with this algorithm is that it is unrealizable. At the time of

the page fault, the operating system has no way of knowing when each of the pages will be referenced next. (We saw a similar situation earlier with the shortest job first scheduling algorithm—how can the system tell which job is shortest?)

Still, by running a program on a simulator and keeping track of all page references, it is possible to implement optimal page replacement on the *second* run by using the page reference information collected during the *first* run.

In this way it is possible to compare the performance of realizable algorithms with the best possible one. If an operating system achieves a performance of, say, only 1 percent worse than the optimal algorithm, effort spent in looking for a better algorithm will yield at most a 1 percent improvement. To avoid any possible confusion, it should be made clear that this log of page references refers only to the one program just measured and then with only one

specific input. The page replacement algorithm derived from it is thus specfic to that one program and input data.

Additional-Reference-Bits Algorithm

We can gain additional ordering information by recording the reference bits at regular intervals. We can keep an 8-bit byte for each page in a table in memory. At regular intervals (say, every 100 milliseconds), a timer interrupt transfers control to the operating system. The operating system shifts the reference bit

for each page into the high-order bit of its 8-bit byte, shifting the other bits right 1 bit, discarding the low-order bit. These &bit shift registers contain the history of page use for the last eight time periods. If the shift register contains 00000000, then the page has not been used for eight time periods; a page that is used at

least once each period would have a shift register value of 11111111. A page with a history register value of 11000100 has been used more recently than has one with 01110111. If we interpret these 8-bit bytes as unsigned integers, the page with the lowest number is the LRU page, and it can be

replaced. Notice that the numbers are not guaranteed to be unique, however. We can either replace (swap out) all pages with the smallest value, or use a FIFO selection among them.

The number of bits of history can be varied, of course, and would be selected (depending on the hardware available) to make the updating as fast as possible. In the extreme case, the number can be reduced to zero, leaving only the reference bit itself. This algorithm is called the second-chance page replacement

algorithm.

SOURCE CODE:

#include<iostream>

#include<math.h>

int replace(int sum[],int n)

{

int least,j;

least=sum[0],j=0;

for(int i=1;i<n;i++)

{

if(sum[i]<least){

least=sum[i];

j=i; }

}

return j;

}

using namespace std;

main()

{

int i,pages,sum1,j;

cout<<"Enter the number of pages\n";

cin>>pages;

if(pages>100)

{

cout<<"The maximum no of pages is 100\n";

return 0;

}

cout<<"No Of Reference bits used is 8\n";

int matrix[100][8];

int sum[100];

//int \*matrix,\*sum;

//matrix=new int[pages][8];

//sum=new int[pages];

for(i=0;i<pages;i++)

{

for(j=7;j>=0;j--)

{

input: cout<<"Is the "<<i+1<<"page used during"<<8-j<<" time?(1=used 0=not used)\n";

cin>>matrix[i][j];

if(matrix[i][j]>1)

{

cout<<"Enter value 1 or 0\n";

goto input;

}

}

}

for(i=0;i<pages;i++)

{

sum1=0;

for(j=0;j<8;j++)

{

sum1=sum1+(pow(10,j)\*matrix[i][j]);

}

sum[i]=sum1;

}

cout<<"The Reference Bits for all pages are shown below:\n";

for(i=0;i<pages;i++)

cout<<sum[i]<<" ";

int x= replace(sum,pages);

cout<<endl<<" Page "<<x+1<<"is removed\n";

}

OUTPUT SCREENSHOTS:

vaishakh@vaishakh-Dell-System-XPS-L502X:~$ ./a.out

Enter the number of pages

2

No Of Reference bits used is 8

s the 1page used during1 time?(1=used 0=not used)

1

s the 1page used during2 time?(1=used 0=not used)

0

s the 1page used during3 time?(1=used 0=not used)

1

s the 1page used during4 time?(1=used 0=not used)

0

s the 1page used during5 time?(1=used 0=not used)

1

s the 1page used during6 time?(1=used 0=not used)

0

s the 1page used during7 time?(1=used 0=not used)

1

s the 1page used during8 time?(1=used 0=not used)

0

s the 2page used during1 time?(1=used 0=not used)

1

s the 2page used during2 time?(1=used 0=not used)

1

s the 2page used during3 time?(1=used 0=not used)

1

s the 2page used during4 time?(1=used 0=not used)

1

s the 2page used during5 time?(1=used 0=not used)

1

s the 2page used during6 time?(1=used 0=not used)

1

s the 2page used during7 time?(1=used 0=not used)

1

s the 2page used during8 time?(1=used 0=not used)

1

The Reference Bits for all pages are shown below:

10101010 11111111

Page 1is removed

2ND INSTANCE:

Enter the number of pages : 3

No Of Reference bits used is 8

s the 1page used during1 time?(1=used 0=not used)

1

s the 1page used during2 time?(1=used 0=not used)

1

s the 1page used during3 time?(1=used 0=not used)

1

s the 1page used during4 time?(1=used 0=not used)

1

s the 1page used during5 time?(1=used 0=not used)

1

s the 1page used during6 time?(1=used 0=not used)

1

s the 1page used during7 time?(1=used 0=not used)

1

s the 1page used during8 time?(1=used 0=not used)

1

s the 2page used during1 time?(1=used 0=not used)

0

s the 2page used during2 time?(1=used 0=not used)

0

s the 2page used during3 time?(1=used 0=not used)

0

s the 2page used during4 time?(1=used 0=not used)

0

s the 2page used during5 time?(1=used 0=not used)

0

s the 2page used during6 time?(1=used 0=not used)

0

s the 2page used during7 time?(1=used 0=not used)

0

s the 2page used during8 time?(1=used 0=not used)

0

s the 3page used during1 time?(1=used 0=not used)

0

s the 3page used during2 time?(1=used 0=not used)

0

s the 3page used during3 time?(1=used 0=not used)

0

s the 3page used during4 time?(1=used 0=not used)

0

s the 3page used during5 time?(1=used 0=not used)

1

s the 3page used during6 time?(1=used 0=not used)

1

s the 3page used during7 time?(1=used 0=not used)

1

s the 3page used during8 time?(1=used 0=not used)

1

The Reference Bits for all pages are shown below:

11111111 0 1111

Page 2is removed