



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Assessment - IV, JUNE 2020

B.Tech, Winter-2019-2020

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE2002
COURSE NAME	WEB TECHNOLOGIES
SLOT	L-37+L-38
FACULTY	Prof. SUDHA S.

(a) Consider a memory hole of size 1kb initially. When a sequence of memory request arrives as following, illustrate the memory allocation by various approaches and calculate the total amount memory wasted by external fragmentation and internal fragmentation in each approach. **(Easy)**

a. First fit

CODE:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"18BIT0272 - PRIYAL BHARDWAJ\n";
    cout<<"Implement first fit file allocation scheme \n";
    int i,j,k,np,nf,f;
    cout<<"Enter the number of partition: ";
    cin>>np; int sp[np]; int flag[np];
    cout<<"Enter the size of each partition: ";
    for(i=0;i<np;i++)
        cin>>sp[i];flag[i]=0;
    cout<<"Enter the number of files: ";
    cin>>nf; int sf[nf];
    cout<<"Enter the size of each file:";
    for(i=0;i<nf;i++)
        cin>>sf[i];
    cout<<"Implementing first fit strategy\n";
    cout<<"File No. || File Size || Partition No. || Partition Size\n";
    for(i=0;i<nf;i++)
    {f=0;//
        for(j=0;j<np;j++)
        {
            if((flag[j]==0)&&(f==0))
            {
                if(sf[i]<sp[j])
                {
                    flag[j]=1;
                    cout<<"      "<<i<<"      ||      "<<sf[i]<<"      ||
"<<j<<"      ||      "<<sp[j]<<endl;
                    f=1;
                }
            }
        }
        if (f==0)
            cout<<"No partition is big enough to support the file: "<<i;
    }
}
```

OUTPUT:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\firstfit.exe"
18BIT0272 - PRIYAL BHARDWAJ
Implement first fit file allocation scheme
Enter the number of partition: 4
Enter the size of each partition: 1
6
5
8
Enter the number of files: 4
Enter the size of each file:3
6
4
5
Implementing first fit strategy
File No. || File Size || Partition No. || Partition Size
    0    ||     3     ||         1    ||         6
    1    ||     6     ||         3    ||         8
    2    ||     4     ||         2    ||         5
No partition is big enough to support the file: 3
Process returned 0 (0x0)   execution time : 9.373 s
```

b. Best fit

CODE:

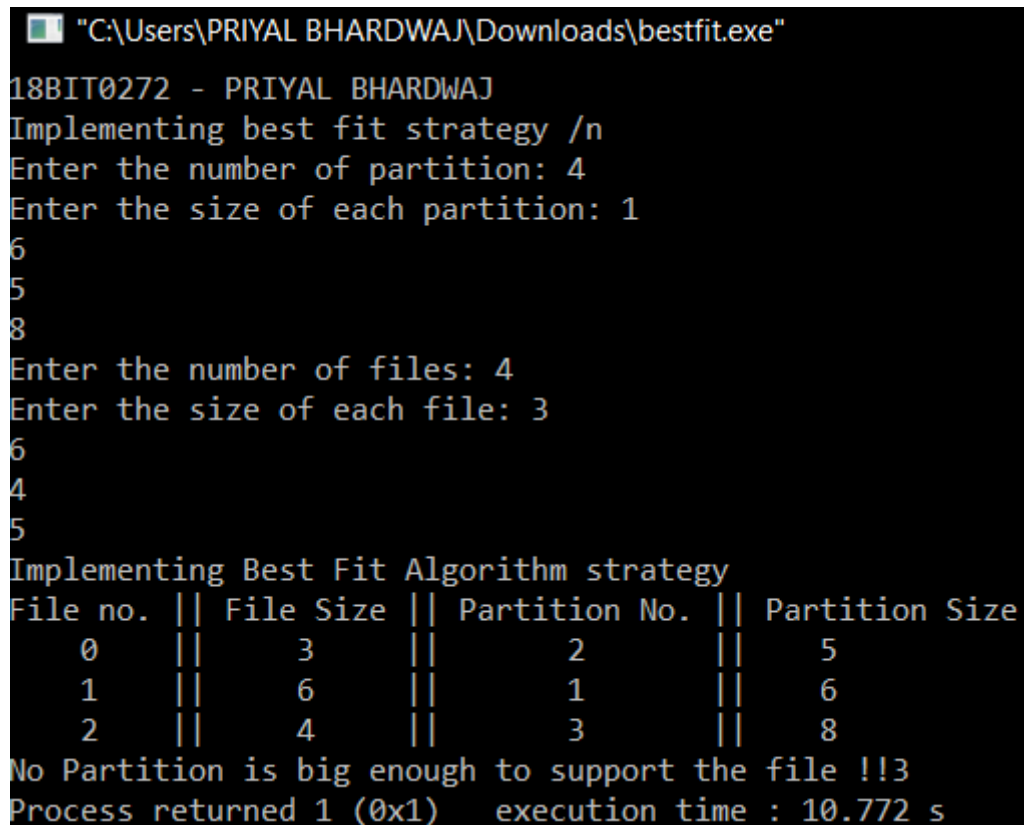
```
#include<iostream>
using namespace std;
int main()
{
    cout<<"18BIT0272 - PRIYAL BHARDWAJ\n";
    cout<<"Implementing best fit strategy /n";
    cout<<endl;
    int i,j,k,np,nf,f,b;
    cout<<"Enter the number of partition: ";
    cin>>np; int sp[np],flag[np];
    cout<<"Enter the size of each partition: ";
    for(i=0;i<np;i++)
    {
        cin>>sp[i];
        flag[i]=0;
    }
    cout<<"Enter the number of files: ";
    cin>>nf; int sf[nf];
    cout<<"Enter the size of each file: ";
    for(i=0;i<nf;i++)
        cin>>sf[i];
```

```

cout<<"Implementing Best Fit Algorithm strategy \n";
cout<<"File no. || File Size || Partition No. || Partition Size \n";
for(i=0;i<nf;i++)
{ f=-1;
  for(j=0;j<np;j++)
  {
    if(flag[j]==0)
    {
      if(sf[i]<=sp[j])
      {
        if(f==-1)
        {f=j;flag[j]=1;}
        else
        if(sp[j]<sp[f])
        {flag[f]=0;f=j;flag[j]=1;}
      }
    }
  }
  if(f==-1)
  cout<<"No Partition is big enough to support the file !!"<<i;
  else
  cout<<"      "<<i<<"      ||      "<<sf[i]<<"      ||      "<<f<<"
  ||      "<<sp[f]<<endl;
}
return 1;
}

```

OUTPUT:



```

"C:\Users\PRIYAL BHARDWAJ\Downloads\bestfit.exe"
18BIT0272 - PRIYAL BHARDWAJ
Implementing best fit strategy /n
Enter the number of partition: 4
Enter the size of each partition: 1
6
5
8
Enter the number of files: 4
Enter the size of each file: 3
6
4
5
Implementing Best Fit Algorithm strategy
File no. || File Size || Partition No. || Partition Size
0      ||      3      ||      2      ||      5
1      ||      6      ||      1      ||      6
2      ||      4      ||      3      ||      8
No Partition is big enough to support the file !!3
Process returned 1 (0x1)   execution time : 10.772 s

```

c. Worst fit

CODE:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"18BIT0272 - PRIYAL BHARDWAJ\n";
    cout<<"Implementing worst fit strategy";
    cout<<endl;
    int i,j,k,np,nf,f,b;
    cout<<"Enter the number of partition: ";
    cin>>np; int sp[np],flag[np];
    cout<<"Enter the size of each partition: ";
    for(i=0;i<np;i++)
        cin>>sp[i];flag[i]=0;
    cout<<"Enter the number of files: ";
    cin>>nf; int sf[nf];
    cout<<"Enter the size of each file : ";
    for(i=0;i<nf;i++)
        cin>>sf[i];
    cout<<"Implementing Worst Fit Algorithm strategy \n"<<endl;
    cout<<"File No. || File Size || Partition No. || Partition Size \n";
    for(i=0;i<nf;i++)
    { f=-1;
        for(j=0;j<np;j++)
        {
            if(flag[j]==0)
            {
                if(sf[i]<=sp[j])
                {
                    if(f==-1)
                    {f=j;flag[j]=1;}
                }
                else
                {
                    if(sp[j]>sp[f])
                    {flag[f]=0;f=j;flag[j]=1;}
                }
            }
        }
        if(f==-1)
            cout<<"No Partition is big enough to support the file !!";
        else
            cout<<"      "<<i<<"      ||      "<<sf[i]<<"      ||      "<<f<<"
            ||      "<<sp[f]<<endl;
    }

    return 1;
}
```

OUTPUT:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\worstfit.exe"
18BIT0272 - PRIYAL BHARDWAJ
Implementing worst fit strategy
Enter the number of partition: 4
Enter the size of each partition: 1
6
5
8
Enter the number of files: 4
Enter the size of each file : 3
6
4
5
Implementing Worst Fit Algorithm strategy

File No. || File Size || Partition No. || Partition Size
0      || 3      || 3      || 8
1      || 6      || 1      || 6
2      || 4      || 2      || 5
No Partition is big enough to support the file !!
Process returned 1 (0x1)   execution time : 10.399 s
```

(b) Write a program to implement the page replacement algorithms.

a. FIFO

b. LRU

c. OPT (**Medium**)

CODE:

```
#include<stdio.h>

int n,nf;
int in[100];
int p[50];
int hit=0;
int i,j,k;
int pgfaultcnt=0;

void getData()
{
printf("\nEnter length of page reference sequence:");
scanf("%d",&n);
printf("\nEnter the page reference sequence:");
for(i=0; i<n; i++)
    scanf("%d",&in[i]);
printf("\nEnter no of frames:");
```

```

scanf("%d",&nf);
}

void initialize()
{
pgfaultcnt=0;
for(i=0; i<nf; i++)
p[i]=9999;
}
int isHit(int data)
{ hit=0;
for(j=0; j<nf; j++)
{
if(p[j]==data)
{
hit=1;
break;
}
}
return hit;
}
int getHitIndex(int data)
{ int hitind;
for(k=0; k<nf; k++)
{
if(p[k]==data)
{
hitind=k;
break;
}
}
return hitind;
}

void dispPages()
{
for (k=0; k<nf; k++)
{
if(p[k]!=9999)
printf(" %d",p[k]);
}
}

void dispPgFaultCnt()
{
printf("\nTotal no of page faults:%d",pgfaultcnt);
}

void fifo()
{

```

```

initialize();
for(i=0; i<n; i++)
{
printf("\nFor %d :",in[i]);
if(isHit(in[i])==0)
{
for(k=0; k<nf-1; k++)
p[k]=p[k+1];
p[k]=in[i];
pgfaultcnt++;
dispPages();
}
else
printf("No page fault");
}
dispPgFaultCnt();
}

```

```

void optimal()
{
initialize();
int near[50];

for(i=0; i<n; i++)
{
printf("\nFor %d :",in[i]);
if(isHit(in[i])==0)
{
for(j=0; j<nf; j++)
{
int pg=p[j];
int found=0;
for(k=i; k<n; k++)
{
if(pg==in[k])
{
near[j]=k;
found=1;
break;
}
else
found=0;
}
if(!found)
near[j]=9999;
}
int max=-9999;
int repindex;

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

```



```

{
if(near[j]>max)
{
max=near[j];
repindex=j;
}
}
p[repindex]=in[i];
pgfaultcnt++;
dispPages();
}
else
printf("No page fault");
}
dispPgFaultCnt();
}

void lru()
{
initialize();
int least[50];

for(i=0; i<n; i++)
{
printf("\nFor %d :",in[i]);
if(isHit(in[i])==0)
{

for(j=0; j<nf; j++)
{
int pg=p[j];
int found=0;

for(k=i-1; k>=0; k--)
{
if(pg==in[k])
{
least[j]=k;
found=1;
break;
}
}
else
found=0;
}
if(!found)
least[j]=-9999;
}
int min=9999;
int repindex;

```

```

for(j=0; j<nf; j++)
{
if(least[j]<min)
{
min=least[j];
repindex=j;
}
}
p[repindex]=in[i];
pgfaultcnt++;
dispPages();
}
else
printf("No page fault!");
}
dispPgFaultCnt();
}

int main()
{
printf("18BIT0272 - PRIYAL BHARDWAJ\n");
int choice;
getData();

while(1)
{
printf("\nPage Replacement
Algorithms\n1.FIFO\n2.Optimal\n3.LRU\n4.Exit\nEnter your choice: ");
scanf("%d",&choice);

switch(choice)
{
case 1:
fifo();
break;
case 2:
optimal();
break;
case 3:
lru();
break;
default:
return 0;
break;
}
}
}

```

OUTPUT:

FIFO:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\replAlgo.exe"
18BIT0272 - PRIYAL BHARDWAJ

Enter length of page reference sequence:7

Enter the page reference sequence:1
6
5
3
4
6
3

Enter no of frames:4

Page Replacement Algorithms
1.FIFO
2.Optimal
3.LRU
4.Exit
Enter your choice: 1

For 1 : 1
For 6 : 1 6
For 5 : 1 6 5
For 3 : 1 6 5 3
For 4 : 6 5 3 4
For 6 :No page fault
For 3 :No page fault
Total no of page faults:5
```

LRU:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\replAlgo.exe"
18BIT0272 - PRIYAL BHARDWAJ

Enter length of page reference sequence:7

Enter the page reference sequence:1
6
5
3
4
6
3

Enter no of frames:4

Page Replacement Algorithms
1.FIFO
2.Optimal
3.LRU
4.Exit
Enter your choice: 3

For 1 : 1
For 6 : 1 6
For 5 : 1 6 5
For 3 : 1 6 5 3
For 4 : 4 6 5 3
For 6 :No page fault!
For 3 :No page fault!
Total no of page faults:5
```

OPTIMAL:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\replAlgo.exe"
18BIT0272 - PRIYAL BHARDWAJ

Enter length of page reference sequence:7

Enter the page reference sequence:1
6
5
3
4
6
3

Enter no of frames:4

Page Replacement Algorithms
1.FIFO
2.Optimal
3.LRU
4.Exit
Enter your choice: 2

For 1 : 1
For 6 : 6
For 5 : 6 5
For 3 : 6 3
For 4 : 6 3 4
For 6 :No page fault
For 3 :No page fault
Total no of page faults:5
```

(c) Write a program that implements the FIFO, LRU, and optimal pager replacement algorithms. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that demand paging is used. **(High)**

CODE:

```
#include<stdio.h>
void FIFO();
void LRU();
void OPTIMAL();
int main()
{
printf("18BIT0272 - PRIYAL BHARDWAJ");
int ch;
do {
printf("\n\n\t1.FIFO\n\t2.LRU\n\t3.Optimal\n\t4.Exit\n\tEnter
Choice: ");
scanf("%d",&ch);
switch(ch)
{
case 1:
FIFO();
break;
case 2:
LRU();
break;
case 3:
OPTIMAL();
break;
}
}while(ch!=4);
}
//A. FIFO
void FIFO()
{
int frame[3]={-1,-1,-1},ref[20],cnt=0,i,j,no,flag;
float ratio,hitcnt=0.00;
printf("\n\tEnter length of reference string: ");
scanf("%d",&no); printf("\n\tEnter reference String with giving
space ....\n\t");
for(i=0;i<no;i++)
scanf("%d",&ref[i]);
//printf("\n\tExecution is started here .....");
for(i=0;i<no;i++)
{
flag=0;
for(j=0;j<3;j++)
if(frame[j]==ref[i])
{
printf("\n\tPage Hit ");
hitcnt++;
flag=1;
break;
}
}
```

```

if(flag==0)
{
printf("\n\tPage Miss");
printf("\tBefore :\t");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
frame[cnt]=ref[i];
cnt++;
if(cnt>=3)
cnt=0;
printf("\tAfter :\t");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
}
}
ratio=hitcnt/no;
printf("\n\n\tHit ratio = %f ",ratio);
}
//B. LRU
void LRU()
{
int frame[3]={-1,-1,-1},used[3]={-1,-1,-1},cnt=0,ref[20],i,j,flag,no,index,value;
float ratio,hitcnt=0;
printf("\n\tEnter length of reference string : ");
scanf("%d",&no);
printf("\n\tEnter reference String with giving space \n\t");
for(i=0;i<no;i++)
scanf("%d",&ref[i]);
//printf("\n\tExecution is started here ");
for(i=0;i<no;i++)
{
flag=0;
for(j=0;j<3;j++)
if(frame[j]==ref[i])
{
printf("\n\tPage Hit ");
hitcnt++;
flag=1;
used[j]=cnt;
break;
}
if(flag==0)
{
printf("\n\tPage Miss");
printf("\tBefore :");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
//selection of victim for replacement
index=0;

```

```

value=used[0];
if(cnt!=0)
{
for(j=0;j<3;j++)
if(value>used[j]&&value!=used[j])
{
index=j;
value=used[j];
}
}
//printf("\tVictim is %d ",index);
frame[index]=ref[i];
used[index]=cnt;
printf("\tAfter :");
for(j=0;j<3;j++)
printf("%d",frame[j]);
}
cnt++;
}
ratio=hitcnt/no;
printf("\n\n\tHit ratio = %f ",ratio);
}
void OPTIMAL()
{
int
frame[3]={-1,-1,-1},used[3]={-1,-1,-
1},cnt=0,ref[20],i,j,flag,no,val1,val2,val3,index;
float ratio,hitcnt=0;
printf("\n\tEnter length of reference string : ");
scanf("%d",&no);
printf("\n\tEnter reference String with giving space \n\t");
for(i=0;i<no;i++)
scanf("%d",&ref[i]);
//printf("\n\tExecution is started here ");
for(i=0;i<no;i++)
{
flag=0;
for(j=0;j<3;j++)
if(frame[j]==ref[i])
{
flag=1;
printf("\n\tPage Hit");
hitcnt++;
break;
}
if(flag==0)
{
printf("\n\tPage Miss");
if(cnt<3)
{

```



```

frame[cnt]=ref[i];
printf("\tStatus :");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
cnt++;
}
else
{
printf("\tBefore :");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
//selection of victim
val1=frame[0];
flag=0;
for(j=i;j<no;j++)
if(ref[j]==val1)
{
val1=j;
flag=1;
break;
}
if(flag==0)
val1=no;
val2=frame[1];
flag=0;
for(j=i;j<no;j++)
if(ref[j]==val2)
{
val2=j;
flag=1;
break;
}
if(flag==0)
val2=no;
val3=frame[2];
flag=0;
for(j=i;j<no;j++)
if(ref[j]==val3)
{
val3=j;
flag=1;
break;
}
if(flag==0)
val3=no;
if(val1<val2)
if(val3<val2)
index=1;
else
index=2;

```

```

else if(val3<val1)
index=0;
else
index=2;
frame[index]=ref[i];
printf("\tAfter :");
for(j=0;j<3;j++)
printf(" %d",frame[j]);
}
}
}
ratio=hitcnt/no;
printf("\n\n\tHit ratio = %f ",ratio);
}

```

OUTPUT:

FIFO:

```

"C:\Users\PRIYAL BHARDWAJ\Downloads\DemandPage.exe"
18BIT0272 - PRIYAL BHARDWAJ

1.FIFO
2.LRU
3.Optimal
4.Exit
Enter Choice: 1

Enter length of reference string: 8

Enter reference String with giving space ....
6 9 5 6 2 5 1 3

Page Miss      Before :      -1 -1 -1      After :  6 -1 -1
Page Miss      Before :      6 -1 -1      After :  6 9 -1
Page Miss      Before :      6 9 -1 After :  6 9 5
Page Hit
Page Miss      Before :      6 9 5  After :  2 9 5
Page Hit
Page Miss      Before :      2 9 5  After :  2 1 5
Page Miss      Before :      2 1 5  After :  2 1 3

Hit ratio = 0.250000

```

LRU:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\DemandPage.exe"
18BIT0272 - PRIYAL BHARDWAJ

1.FIFO
2.LRU
3.Optimal
4.Exit
Enter Choice: 2

Enter length of reference string : 8

Enter reference String with giving space
6 9 5 6 2 5 1 3

Page Miss      Before : -1 -1 -1      After :6-1-1
Page Miss      Before : 6 -1 -1      After :69-1
Page Miss      Before : 6 9 -1 After :695
Page Hit
Page Miss      Before : 6 9 5  After :625
Page Hit
Page Miss      Before : 6 2 5  After :125
Page Miss      Before : 1 2 5  After :135

Hit ratio = 0.250000
```

OPTIMAL:

```
"C:\Users\PRIYAL BHARDWAJ\Downloads\DemandPage.exe"
18BIT0272 - PRIYAL BHARDWAJ

1.FIFO
2.LRU
3.Optimal
4.Exit
Enter Choice: 3

Enter length of reference string : 8

Enter reference String with giving space
6 9 5 6 2 5 1 3

Page Miss      Status : 6 -1 -1
Page Miss      Status : 6 9 -1
Page Miss      Status : 6 9 5
Page Hit
Page Miss      Before : 6 9 5   After : 2 9 5
Page Hit
Page Miss      Before : 2 9 5   After : 2 9 1
Page Miss      Before : 2 9 1   After : 2 9 3

Hit ratio = 0.250000
```
