**NAME: DHANRAJ SUBHASH KORE**

**DIV: TY B** **BATCH: B3**

**ROLL NO: 60**

**SUB: SS LAB ASSIGNMENT**

**Placement Strategies**

Implement Following placement algorithms :

1. First Fit

2. Next Fit

3. Best Fit

4. Worst Fit

**CODE:**

#include <bits/stdc++.h>

using namespace std;

void Nextfit()

{

    int m,n,i=0;

    cout<<"Enter no. of blocks: ";

cin>>m;

int blockSize[m];

cout<<"\nEnter size of each block: ";

for(i = 0; i < m; i++)

cin>>blockSize[i];

cout<<"\nEnter no. of processes: ";

cin>>n;

    int processSize[n];

cout<<"\nEnter size of each process: ";

for(i = 0; i < n; i++)

    cin>>processSize[i];

    // Stores block id of the block allocated to a

    // process

    int allocation[n], j = 0;

    // Initially no block is assigned to any process

    memset(allocation, -1, sizeof(allocation));

    // pick each process and find suitable blocks

    // according to its size ad assign to it

    for (int i = 0; i < n; i++) {

        // Do not start from beginning

        while (j < m) {

            if (blockSize[j] >= processSize[i]) {

                // allocate block j to p[i] process

                allocation[i] = j;

                // Reduce available memory in this block.

                blockSize[j] -= processSize[i];

                break;

            }

        // mod m will help in traversing the blocks from

        // starting block after we reach the end.

        j = (j + 1) % m;

    }

}

        cout<<"\n\n\t-------------------------------Next Fit--------------------------------------";

    //display allocation details

    cout << "\n\n\t\tBlock No. \tBlock Size\tProcess No.\tProcess Size\t Allocated Block \n\n";

for(i = 0; i < n; i++)

{

cout <<"\t \t " << i + 1 <<"\t\t   " << blockSize[i]<<"\t\t\t" << i + 1 << "\t\t" << processSize[i]

    << "\t\t";

    if (allocation[i] != -1)

        cout << allocation[i] + 1;

    else

        cout << "Not Allocated";

    cout << "\n\n";

        }

                cout<<"\t------------------------------------------------------------------------------------\n\n\n";

}

// Function to allocate memory to blocks as per worst fit

// algorithm

void Worstfit()

{

    int m,n,i=0;

    cout<<"Enter no. of blocks: ";

cin>>m;

int blockSize[m];

cout<<"\nEnter size of each block: ";

for(i = 0; i < m; i++)

cin>>blockSize[i];

cout<<"\nEnter no. of processes: ";

cin>>n;

    int processSize[n];

cout<<"\nEnter size of each process: ";

for(i = 0; i < n; i++)

    cin>>processSize[i];

    // Stores block id of the block allocated to a

    // process

    int allocation[n];

    // Initially no block is assigned to any process

    memset(allocation, -1, sizeof(allocation));

    // pick each process and find suitable blocks

    // according to its size ad assign to it

 for (int i=0; i<n; i++)

 {

 // Find the best fit block for current process

 int largestblock = -1;

 for (int j=0; j<m; j++)

 {

 if (blockSize[j] >= processSize[i])

 {

 if (largestblock == -1)

 largestblock = j;

 else if (blockSize[largestblock] < blockSize[j])

 largestblock = j;

 }

 }

 // If we could find a block for current process

 if (largestblock != -1)

 {

 // allocate block j to p[i] process

 allocation[i] = largestblock;

 // Reduce available memory in this block.

 blockSize[largestblock] -= processSize[i];

 }

 }

     cout<<"\n\n\t-----------------------------------Worst Fit------------------------------------------\n\n";

   //display allocation details

    cout << "\n\n\t\tBlock No. \tBlock Size\tProcess No.\tProcess Size\t Allocated Block \n\n";

for(i = 0; i < n; i++)

{

cout <<"\t\t  " << i + 1 <<"\t\t   " << blockSize[i]<<"\t\t\t" << i + 1 << "\t\t" << processSize[i]

    << "\t\t";

 if (allocation[i] != -1)

 cout << allocation[i] + 1;

 else

 cout << "Not Allocated";

 cout<<"\n\n";

 }

    cout<<"\t------------------------------------------------------------------------------------------\n\n\n";

}

void Firstfit(){

int blocksize[10], prs[10], n, p, flags[10], allocation[10], i, j;

for(i = 0; i < 10; i++)

{

flags[i] = 0;

allocation[i] = -1;

}

cout<<"Enter no. of blocks: ";

cin>>n;

cout<<"\nEnter size of each block: ";

for(i = 0; i < n; i++)

cin>>blocksize[i];

cout<<"\nEnter no. of processes: ";

cin>>p;

cout<<"\nEnter size of each process: ";

for(i = 0; i < p; i++)

cin>>prs[i];

for(i = 0; i < p; i++) //allocation as per first fit

for(j = 0; j < n; j++)

if(flags[j] == 0 && blocksize[j] >= prs[i])

{

allocation[j] = i;

flags[j] = 1;

break;

}

    cout<<"\n\n\t-----------------------------------First Fit--------------------------------------------";

//display allocation details

    cout << "\n\n\t\tBlock No. \tBlock Size\tProcess No.\tProcess Size\t Allocated Block \n\n";

for(i = 0; i < n; i++)

{

cout <<"\t\t  " << i + 1 <<"\t\t   " << blocksize[i]<<"\t\t\t" << i + 1 << "\t\t" << prs[i]

    << "\t\t";

    if(flags[i] == 1)

cout<<allocation[i]+1;

else

cout<<"Not allocated";

cout<<"\n\n";

}

     cout<<"\t------------------------------------------------------------------------------------\n\n\n";

}

void Bestfit(){

    int fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999;

    static int barray[20],parray[20];

cout<<"\n\n\t-----------------------------------Best Fit--------------------------------------------";

    cout<<"\nEnter the number of blocks : ";

    cin>>nb;

    cout<<"Enter the number of processes : ";

    cin>>np;

    cout<<"\nEnter the size of the blocks:-\n ";

for(i=1;i<=nb;i++)

{

cout<<"Block no. "<<i<<" : ";

cin>>b[i];

}

cout<<"\nEnter the size of the processes :-\n";

for(i=1;i<=np;i++)

{

cout<<"Process no."<<i<<" : ";

cin>>p[i];

}

for(i=1;i<=np;i++)

{

for(j=1;j<=nb;j++)

{

if(barray[j]!=1)

{

temp=b[j]-p[i];

if(temp>=0)

if(lowest>temp)

{

parray[i]=j;

lowest=temp;

}

}

}

fragment[i]=lowest;

barray[parray[i]]=1;

lowest=10000;

}

cout<<"\n\n\t\tProcess\_no\tProcess\_size\tBlock\_no\tBlock\_size\tFragment";

for(i=1;i<=np && parray[i]!=0;i++)

cout<<"\n\n\t\t"<<i<<"\t\t"<<p[i]<<"\t\t"<<parray[i]<<"\t\t"<<b[parray[i]]<<"\t\t"<<fragment[i]<<"\n\n";

     cout<<"\t------------------------------------------------------------------------------------\n\n\n";

}

int main(){

    int ch,choice;

    do{

    cout<<"\n\n\t-------------------------------------------------------------------------";

    cout<<"\n\n\t\t1. First Fit \t2. Next Fit \t3. Best Fit \t4. Worst Fit\n\n";

    cout<<"\t-------------------------------------------------------------------------\n\n";

    cout<<"Enter your choice : ";

    cin>>ch;

    cout<<"\n";

    switch(ch){

        case 1: Firstfit();

            break;

        case 2: Nextfit();

            break;

        case 3: Bestfit();

            break;

        case 4: Worstfit();

            break;

        default : cout<<"Please Enter valid choice!\n";

        }

        cout<<"Do you want to Continue ? 1 for YES & 0 for NO : ";

        cin>>choice;

    }

    while(choice);

    return 0;

}

**O/P:**







