2CS5

Group2

Rishi Malik: 102017096

Naman Khurana: 102017099

Amol Sharma: 102017106

**UCS303 – Operating Systems**

**Lab 9**

Write a C / C++ program to simulate the following **contiguous memory allocation** techniques:

a) Worst-fit

b) Best-fit

c) First-fit

1. **Worst-Fit**

#include<bits/stdc++.h>

using namespace std;

void worstFit(int partition\_size[], int m, int file\_size[],int n)

{

    int allocated[n];

    for(int i=0;i<n;i++){

        allocated[i]=-1;

    }

    for (int i=0; i<n; i++)

    {

        int index = -1;

        for (int j=0; j<m; j++)

        {

            if (partition\_size[j] >= file\_size[i])

            {

                if (index == -1)

                    index = j;

                else if (partition\_size[index] < partition\_size[j])

                    index = j;

            }

        }

        if (index != -1)

        {

            allocated[i] = index;

            partition\_size[index] -= file\_size[i];

        }

    }

    cout << "\nProcess id.\tProcess Size\tBlock no.\n";

    for (int i = 0; i < n; i++)

    {

        cout << " " << i+1 << "\t\t" << file\_size[i] << "\t\t";

        if (allocated[i] != -1)

            cout << allocated[i] + 1;

        else

            cout << "Not Allocated";

        cout << endl;

    }

}

int main()

{

    int m,n;

    cout<<"Enter number of partitions and number of files"<<endl;

    cin>>n>>m;

    int partition\_size[n];

    int file\_size[m];

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<n;i++){

        cin>>partition\_size[i];

    }

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<m;i++){

        cin>>file\_size[i];

    }

    worstFit(partition\_size, n, file\_size, m);

    return 0 ;

}

**2)Best Fit**

#include<bits/stdc++.h>

using namespace std;

void bestFit(int file\_size[], int m, int partition\_size[], int n)

{

    int allocated[n];

    for(int i=0;i<n;i++){

        allocated[i]=-1;

    }

    for (int i=0; i<n; i++)

    {

        int index = -1;

        for (int j=0; j<m; j++)

        {

            if (file\_size[j] >= partition\_size[i])

            {

                if (index == -1)

                    index = j;

                else if (file\_size[index] > file\_size[j])

                    index = j;

            }

        }

        if (index != -1)

        {

            allocated[i] = index;

            file\_size[index] -= partition\_size[i];

        }

    }

    cout << "\nProcess No.\tProcess Size\tBlock no.\n";

    for (int i = 0; i < n; i++)

    {

        cout << " " << i+1 << "\t\t" << partition\_size[i] << "\t\t";

        if (allocated[i] != -1)

            cout << allocated[i] + 1;

        else

            cout << "Not Allocated";

        cout << endl;

    }

}

int main()

{

    int m,n;

    cout<<"Enter number of partitions and number of files"<<endl;

    cin>>n>>m;

    int partition\_size[n];

    int file\_size[m];

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<n;i++){

    cin>>partition\_size[i];

    }

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<m;i++){

        cin>>file\_size[i];

    }

    bestFit(partition\_size, n, file\_size, m);

    return 0 ;

}

**3) First-Fit**

#include<bits/stdc++.h>

using namespace std;

void firstFit(int partition\_size[], int m,int file\_size[], int n)

{

    int allocated[n];

    for(int i=0;i<n;i++){

        allocated[i]=-1;

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < m; j++)

        {

            if (partition\_size[j] >= file\_size[i])

            {

                allocated[i] = j;

                partition\_size[j] -= file\_size[i];

                break;

            }

        }

    }

    cout << "\nProcess No.\tProcess Size\tBlock no.\n";

    for (int i = 0; i < n; i++)

    {

        cout << " " << i+1 << "\t\t"<< file\_size[i] << "\t\t";

        if (allocated[i] != -1)

            cout << allocated[i] + 1;

        else

            cout << "Not Allocated";

        cout << endl;

    }

}

int main()

{

    int m,n;

    cout<<"Enter number of partitions and number of files"<<endl;

    cin>>n>>m;

    int partition\_size[n];

    int file\_size[m];

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<n;i++){

    cin>>partition\_size[i];

    }

    cout<<"Enter all the partition sizes"<<endl;

    for(int i=0;i<m;i++){

        cin>>file\_size[i];

    }

    firstFit(partition\_size, n, file\_size, m);

    return 0 ;

}