OS Lab ASSIGNMENT-10

Name: Ashmit Thawait

Roll No: 102203790

Group: CO-17

Q. Write a program using C/C++/Java to simulate the first fit, best fit and worst fit memory allocation strategy. Assume memory chunk and initial requirement for memory block from your side.

Example Program Flow

1. Initialize the program with an initial memory chunk of a specified size.

2. Display the menu for the user to choose an allocation strategy (First Fit, Best Fit, Worst Fit) or to exit.

3. Prompt the user to request a memory block allocation, specifying the size.

4. Allocate memory based on the chosen strategy.

5. Display the updated state of the memory chunk.

6. Repeat steps 3 to 5 until the user chooses to exit.

Code:

```
#include <iostream>
#include <cstring>
using namespace std;

void firstFit(int blockSize[], int m, int processSize[], int n)
{
int allocation[n];
memset(allocation, -1, sizeof(allocation));
for (int i = 0; i < n; i++)
{
for (int j = 0; j < m; j++)
{
if (blockSize[j] >= processSize[i])
{
allocation[i] = j;
blockSize[j] -= processSize[i]; break;
}
}
}
```

```
cout << "\nProcess No.\tProcess Size\tBlock no.\n";
for (int i = 0; i < n; i++)
{
cout << " " << i + 1 << "\t\t" << processSize[i] << "\t\t";

if (allocation[i] != -1)
cout << allocation[i] + 1;
else
cout << "Not Allocated"; cout << endl;
}
}

void bestFit(int blockSize[], int m, int processSize[], int n)
{
int allocation[n];
for (int i = 0; i < n; i++)
allocation[i] = -1;

for (int i = 0; i < n; i++)
{
int bestIdx = -1;

for (int j = 0; j < m; j++)
{
if (blockSize[j] >= processSize[i])
{
if (bestIdx == -1)
bestIdx = j;
else if (blockSize[bestIdx] > blockSize[j])
bestIdx = j;
}
}

if (bestIdx != -1)
{
allocation[i] = bestIdx;
blockSize[bestIdx] -= processSize[i];
}
}
cout << "\nProcess No.\tProcess Size\tBlock no.\n";
for (int i = 0; i < n; i++)
{
cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
if (allocation[i] != -1)
cout << allocation[i] + 1;
else
cout << "Not Allocated"; cout << endl;
}
}
```

```
void worstFit(int blockSize[], int m, int processSize[], int n)
{
int allocation[n];
memset(allocation, -1, sizeof(allocation));

for (int i=0; i<n; i++)
{
int wstIdx = -1;
for (int j=0; j<m; j++)
{
if (blockSize[j] >= processSize[i])
{
if (wstIdx == -1)
wstIdx = j;
else if (blockSize[wstIdx] < blockSize[j])
wstIdx = j;
}
}

if (wstIdx != -1)
{
allocation[i] = wstIdx;
blockSize[wstIdx] -= processSize[i];
}
}

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

for (int i = 0; i < n; i++)
{
cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
if (allocation[i] != -1)
cout << allocation[i] + 1;
else
cout << "Not Allocated"; cout << endl;
}
}

int main()
{
int choice;
int processes;
int blockSize[] = {100, 500, 200, 300, 600};

while (true)
{
cout<<"1. First Fit \n";
cout<<"2. Best Fit \n";
cout<<"3. Worst Fit \n";
cout<<"4. Exit \n";
cout<<"Select option: \n"; cin>>choice;
```

```
if (choice == 4) break;

cout << "No of processes ?\n";
cin >> processes;

int processSize[processes];
int m = sizeof(blockSize) / sizeof(blockSize[0]);
int n = sizeof(processSize) / sizeof(processSize[0]);

cout<<"Enter process sizes: \n";

for (int i = 0; i < processes; i++)
{
cout << "Process " << i+1 << endl;
cin >> processSize[i];
}

if (choice==1) firstFit(blockSize, m, processSize, n);

else if (choice ==2) bestFit(blockSize, m , processSize, n);

else if (choice == 3) worstFit(blockSize, m , processSize, n);

else cout << "Invalid Option!" << endl;
}
return 0;
}
```

OUTPUT:

First Fit –

```
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ g++ assign10.cpp
ashmit@ashmit-ubuntu:~/Desktop/ashmit$ ./a.out
1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Select option:
1
No of processes ?
3
Enter process sizes:
Process 1
120
Process 2
100
Process 3
650

Process No.     Process Size    Block no.
 1              120             2
 2              100             1
 3              650             Not Allocated
```

Best fit –

```
1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Select option:
2
No of processes ?
3
Enter process sizes:
Process 1
120
Process 2
100
Process 3
650

Process No.     Process Size    Block no.
 1              120             3
 2              100             4
 3              650             Not Allocated
```

Worst Fit –

```
1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Select option:
3
No of processes ?
3
Enter process sizes:
Process 1
120
Process 2
100
Process 3
650

Process No.     Process Size    Block no.
 1              120             5
 2              100             5
 3              650             Not Allocated
1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Select option:
4
ashmit@ashmit-ubuntu:~/Desktop/ashmit$
```