

Name : Omkar Arun Shinde
PRN No. : 122B1B258

Assignment No. 07

Problem Statement : Write a program to simulate memory allocation techniques: First Fit, Best Fit, Next Fit and Worst Fit.

Code :

```
#include<stdio.h>

#include<stdlib.h>

int M;

int N;

int Holes[10];

int Process[10];

void FirstFit() {

    int CopyHoles[10];

    int CopyProcess[10];

    for (int i = 0; i < M; i++) {

        CopyHoles[i] = Holes[i];

    }

    for (int i = 0; i < N; i++) {

        CopyProcess[i] = Process[i];

    }

    int index = 0;

    for(int i = 0; i < N; i++) {

        int found = 0;

        for(int j = 0; j < M; j++) {

            if(CopyHoles[j] >= CopyProcess[i]) {

                found = 1;

                index = j;
```

```

        break;
    }
}

if(found == 0) {
    printf("Process %d cannot be allocated\n", i+1);
    break;
}

else {
    printf("Process %d allocated to hole %d || Process Size = %d || Hole Size = %d ||
Updated Hole Size = %d\n", i+1, index+1, CopyProcess[i], CopyHoles[index],
CopyHoles[index] - CopyProcess[i]);
    CopyHoles[index] -= CopyProcess[i];
}
}
}

```

```

void BestFit() {
    int CopyHoles[10];
    int CopyProcess[10];
    for (int i = 0; i < M; i++) {
        CopyHoles[i] = Holes[i];
    }
    for (int i = 0; i < N; i++) {
        CopyProcess[i] = Process[i];
    }
    for(int i = 0; i < N; i++) {
        int index = -1;
        int small = 999;
        for(int j = 0; j < M; j++) {

```

```

        if(CopyHoles[j] >= CopyProcess[i] && CopyHoles[j] < small) {
            small = CopyHoles[j];
            index = j;
        }
    }
    if(index == -1) {
        printf("Process %d cannot be allocated\n", i+1);
        break;
    }
    else {
        printf("Process %d allocated to hole %d || Process Size = %d || Hole Size = %d ||  

Updated Hole Size = %d\n", i+1, index+1, CopyProcess[i], CopyHoles[index],  

CopyHoles[index] - CopyProcess[i]);
        CopyHoles[index] -= CopyProcess[i];
    }
}
}

```

```

void WorstFit() {
    int CopyHoles[10];
    int CopyProcess[10];
    for (int i = 0; i < M; i++) {
        CopyHoles[i] = Holes[i];
    }
    for (int i = 0; i < N; i++) {
        CopyProcess[i] = Process[i];
    }
    for(int i = 0; i < N; i++) {
        int index = -1;

```

```

int large = -999;
for(int j = 0; j < M; j++) {
    if(CopyHoles[j] >= CopyProcess[i] && CopyHoles[j] > large) {
        large = CopyHoles[j];
        index = j;
    }
}
if(index == -1) {
    printf("Process %d cannot be allocated\n", i+1);
    break;
}
else {
    printf("Process %d allocated to hole %d || Process Size = %d || Hole Size = %d ||
Updated Hole Size = %d\n", i+1, index+1, CopyProcess[i], CopyHoles[index],
CopyHoles[index] - CopyProcess[i]);

    CopyHoles[index] -= CopyProcess[i];
}
}
}

```

```

void NextFit() {
    int CopyHoles[10];
    int CopyProcess[10];
    for (int i = 0; i < M; i++) {
        CopyHoles[i] = Holes[i];
    }
    for (int i = 0; i < N; i++) {
        CopyProcess[i] = Process[i];
    }
}

```

```

int index = 0;
for(int i = 0; i < N; i++) {
    int found = 0;
    int count = 0;
    for(int j = index; count < M; j = (j + 1) % M) {
        if(CopyHoles[j] >= CopyProcess[i]) {
            index = j;
            found = 1;
            break;
        }
        count++;
    }
    if(found == 0) {
        printf("Process %d cannot be allocated\n", i+1);
        break;
    }
    else {
        printf("Process %d allocated to hole %d || Process Size = %d || Hole Size = %d || Updated Hole Size = %d\n", i+1, index+1, CopyProcess[i], CopyHoles[index], CopyHoles[index] - CopyProcess[i]);
        CopyHoles[index] -= CopyProcess[i];
    }
}

```

```

int main() {
    int choice;

    printf("Enter the No. of Holes(Max = 10): ");
    scanf("%d", &M);

```

```

printf("Enter the No. of Processes(Max = 10): ");
scanf("%d", &N);
printf("Enter the Hole Size one by one:\n");
for(int i = 0; i < M; i++) {
    scanf("%d", &Holes[i]);
}
printf("Enter the Process Size one by one:\n");
for(int i = 0; i < N; i++) {
    scanf("%d", &Process[i]);
}
do {
    printf("\n*****Menu*****\n");
    printf("1. First Fit\n2. Best Fit\n3. Worst Fit\n4. Next Fit\n5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    switch(choice) {
        case 1: {
            printf("\nFirst Fit Allocation\n");
            FirstFit();
            printf("-----\n");
            break; }
        case 2: {
            printf("\nBest Fit Allocation\n");
            BestFit();
            printf("-----\n");
            break; }
        case 3: {
            printf("\nWorst Fit Allocation\n");

```

```
    WorstFit();  
    printf("-----\n");  
    break; }  
    case 4: {  
        printf("\nNext Fit Allocation\n");  
        NextFit();  
        printf("-----\n");  
        break; }  
    case 5: {  
        printf("Exiting...\n");  
        break; }  
    default: {  
        printf("Invalid choice. Please try again.\n");  
        break; }  
    }  
} while(choice != 5);  
return 0;  
}
```

Output :

```
ies Terminal Mar 28 4:30 PM
pccoe@pccoe: ~/122B1B258

(base) pccoe@pccoe:~/122B1B258$ gcc OSL7.c
(base) pccoe@pccoe:~/122B1B258$ ./a.out
Enter the No. of Holes(Max = 10): 5
Enter the No. of Processes(Max = 10): 4
Enter the Hole Size one by one:
100 500 200 300 600
Enter the Process Size one by one:
212 417 112 426

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 1

First Fit Allocation
Process 1 allocated to hole 2 || Process Size = 212 || Hole Size = 500 || Updated Hole Size = 288
Process 2 allocated to hole 5 || Process Size = 417 || Hole Size = 600 || Updated Hole Size = 183
Process 3 allocated to hole 2 || Process Size = 112 || Hole Size = 288 || Updated Hole Size = 176
Process 4 cannot be allocated
-----

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 2

Best Fit Allocation
Process 1 allocated to hole 4 || Process Size = 212 || Hole Size = 300 || Updated Hole Size = 88
Process 2 allocated to hole 2 || Process Size = 417 || Hole Size = 500 || Updated Hole Size = 83
Process 3 allocated to hole 3 || Process Size = 112 || Hole Size = 200 || Updated Hole Size = 88
Process 4 allocated to hole 5 || Process Size = 426 || Hole Size = 600 || Updated Hole Size = 174
-----

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 3
```



```
ies Terminal ▾ Mar 28 4:30 PM
pccoe@pccoe: ~/122B1B258

Process 1 allocated to hole 4 || Process Size = 212 || Hole Size = 300 || Updated Hole Size = 88
Process 2 allocated to hole 2 || Process Size = 417 || Hole Size = 500 || Updated Hole Size = 83
Process 3 allocated to hole 3 || Process Size = 112 || Hole Size = 200 || Updated Hole Size = 88
Process 4 allocated to hole 5 || Process Size = 426 || Hole Size = 600 || Updated Hole Size = 174
-----

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 3

Worst Fit Allocation
Process 1 allocated to hole 5 || Process Size = 212 || Hole Size = 600 || Updated Hole Size = 388
Process 2 allocated to hole 2 || Process Size = 417 || Hole Size = 500 || Updated Hole Size = 83
Process 3 allocated to hole 5 || Process Size = 112 || Hole Size = 388 || Updated Hole Size = 276
Process 4 cannot be allocated
-----

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 4

Next Fit Allocation
Process 1 allocated to hole 2 || Process Size = 212 || Hole Size = 500 || Updated Hole Size = 288
Process 2 allocated to hole 5 || Process Size = 417 || Hole Size = 600 || Updated Hole Size = 183
Process 3 allocated to hole 5 || Process Size = 112 || Hole Size = 183 || Updated Hole Size = 71
Process 4 cannot be allocated
-----

*****Menu*****
1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit
5. Exit
Enter your choice: 5
Exiting...
(base) pccoe@pccoe:~/122B1B258$
```