

Name: Gunjan Dinesh

Roll no: 20052082

Subject: Operating System Practical

Subject Code: UPB-202

Gunjan
27/08/2021

Q1) C Code for worst fit Memory Management Scheme.

Source Code

```
#include <stdio.h>
int main()
{
    printf("\n\t\t\t\t\tMemory Management" " - Worst fit");
    int i, j, nblocks, nfiles, temp, top = 0;
    int frag[10], blocks[10], files[10];
    static int block-arr[10], file-arr[10];
    printf("\nEnter the Total Number of Blocks: ");
    scanf("%d", &nblocks);
    printf("\nEnter the Total Number of files: ");
    scanf("%d", &nfiles);
    printf("\nEnter the Total Size of the Blocks: \n");
    for (i = 0; i < nblocks; i++)
    {
        printf("Block No. %d: ", i + 1);
        scanf("%d", &blocks[i]);
    }
    printf("\nEnter the size of the files: \n");
    for (i = 0; i < nfiles; i++)
    {
        printf("File No. %d: ", i + 1);
        scanf("%d", &files[i]);
    }
    for (i = 0; i < nfiles; i++)
    {
        for (j = 0; j < nblocks; j++)
```

Name: Gunjan Dimer
Roll no: 20052092
Subject: Operating System Practical
Subject Code: PSI-202

Gunjan
27/08/2021

```
{
if (block_arr[j] != 1)
{
temp = blocks[j] - files[i];
if (temp >= 0)
{
if (top < temp)
{
file_arr[i] = j;
top = temp;
}
}
}
flag[i] = top;
block_arr[file_arr[i]] = 1;
top = 0;
}
}
printf ("|files Number|file size|""Block Number|Block
Size|fragment");
for (i = 0; i < nfiles; i++)
{
printf ("|n%d|t|t%d|t|t%d|t|t%d|t|t%d", i, files[i],
file_arr[i], blocks[file_arr[i]], flag[i]);
}
printf ("|n");
return 0;
}
```

Memory Management - Worst Fit

Enter the Total Number of Blocks: 3

Enter the Total Number of Files: 2

Enter the Size of the Blocks:

Block No.1: 5

Block No.2: 2

Block No.3: 7

Enter the Size of the Files:

File No.1: 1

File No.2: 4

File Number	File Size	Block Number	Block Size	Fragment
0	1	2	7	6
1	4	0	5	0

...Program finished with exit code 0

Press ENTER to exit console.

Name: Gunjan Divedi

Roll no: 20052092

Univ. Roll no: 2023055

Subject: Operating System Practical

Subject Code: PBI-202

Gunjan
27/08/2021

Q2) C program for SCAN Disk scheduling Algorithm.

Source Code

```
#include <stdio.h>
int absolutevalue (int);
void main()
{
    int queue[25], n, headposition, i, j, k, seek = 0, maxrange,
    difference, temp, queue1[20], queue2[20], temp1 = 0,
    temp2 = 0; float averageSeekTime;
    printf("Enter the maximum range of Disk: ");
    scanf("%d", &maxrange);
    printf("Enter the number of queue requests: ");
    scanf("%d", &n);
    printf("Enter the initial head position: ");
    scanf("%d", &headposition);
    printf("Enter the disk positions to be read (queue):");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &temp);
        if (temp > headposition)
        {
            queue1[temp1] = temp;
            temp1++;
        }
        else
        {
            queue2[temp2] = temp;
            temp2++;
        }
    }
}
```

```
for (i = 0; i < temp1 - i; i++)
```

```
{  
    for (j = i + 1; j < temp1; j++)
```

```
{  
    if (queue1[i] > queue1[j])
```

```
{  
        temp = queue1[i];
```

```
        queue1[i] = queue1[j];
```

```
        queue1[j] = temp;
```

```
    }
```

```
}
```

```
}
```

```
for (i = 0; i < temp2 - 1; i++)
```

```
{  
    for (j = i + 1; j < temp2; j++)
```

```
{  
    if (queue2[i] < queue2[j])
```

```
{  
        temp = queue2[i];
```

```
        queue2[i] = queue2[j];
```

```
        queue2[j] = temp;
```

```
    }
```

```
}
```

```
}
```

```
for (i = 1, j = 0; j < temp1; i++, j++)
```

```
{  
    queue[i] = queue[j];
```

```
}
```

```
queue[i] = maxrange;
```

```
for (i = temp1 + 2, j = 0; j < temp2; i++, j++)
```


Name: Gunjan Dimer

Roll no: 20052092

Subject: Operating System Practical

Subject Code: PSI-202

Univ. Roll no: 2023055

Gunjan
07/08/2021

```
{
    queue[i] = queue[j];
}
queue[i] = 0;
queue[0] = headposition;
for(j=0; j<=n; j++)
{
    difference = absoluteValue(queue[j+1] - queue[j]);
    seek = seek + difference;
    printf("Disk head moves from position %d to %d\n",
           queue[j], queue[j+1], difference);
}
averageSeekTime = seek / (float)n;
printf("Total Seek Time = %d\n", seek);
printf("Average Seek Time = %f\n", averageSeekTime);
}

int absoluteValue(int x)
{
    if(x > 0)
    {
        return x;
    }
    else
    {
        return x * -1;
    }
}
```

```
Enter the maximum range of Disk: 99
Enter the number of queue requests: 7
Enter the initial head position: 24
Enter the disk positions to be read(queue): 12
26
24
4
42
8
50
Disk head moves from position 24 to 26 with Seek 2
Disk head moves from position 26 to 42 with Seek 16
Disk head moves from position 42 to 50 with Seek 8
Disk head moves from position 50 to 99 with Seek 49
Disk head moves from position 99 to 24 with Seek 75
Disk head moves from position 24 to 12 with Seek 12
Disk head moves from position 12 to 8 with Seek 4
Disk head moves from position 8 to 4 with Seek 4
Total Seek Time= 170
Average Seek Time= 24.285715

...Program finished with exit code 0
Press ENTER to exit console.
```