CSE 2005

OPERATING SYSTEMS

Assessment – 4

L7+L8 | PLBG17

WINTER SEMESTER 2020-21

by

SHARADINDU ADHIKARI

19BCE2105

Memory Management

Memory Management

(a) Consider a memory hole of size 1kb initially. When a sequence of memory request arrives as following, illustrate the memory allocation by various approaches and calculate the total amount memory wasted by external fragmentation and internal fragmentation in each approach.

i. First fit; ii. Best fit iii. Worst fit **(Easy)**

(b) Write a program to implement the page replacement algorithms.

i. FIFO ii. LRU iii. OPT **(Medium)**

(c) Write a program that implements the FIFO, LRU, and optimal pager replacement algorithms. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that demand paging is used. **(High)**

Solutions_MM

(a)

```
#include <iostream>
using namespace std;
int main()
{
    int c,i,j,k,n,l,m[10],p[10],po[20],flag,z,y,temp,temp1;
    cout<<"\t\t-----Welcome---"<<endl;
    cout<<"Enter memory total partitions:\t";
    cin>>n;
    cout<<"\nEnter memory size for\n";
    for(i=1;i<=n;i++)
    {
        cout<<"\npartition "<<i<<" :\t";
        cin>>m[i];
        po[i]=i;
    }
    cout<<"\nEnter total number of process:\t";
    cin>>j;
    cout<<"\nEnter memory size for\n";
    for(i=1;i<=j;i++)
    {
        cout<<"\nprocess "<<i<<" :\t";
        cin>>p[i];
    }
    cout<<"\n**Menu**\n1.first fit\n2.best fit\n3.worst fit\nenter choice:\t";
    cin>>c;
    switch(c)
    {
        case 1:
            for(i=1;i<=j;i++)
            {
                flag=1;
                for(k=1;k<=n;k++)
                {
                    if(p[i]<=m[k])
                    {
```

```
                        cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB
allocated at memory partition:\t"<<po[k];
                        m[k]=m[k]-p[i];
                        break;
                    }
                    else
                    {
                        flag++;
                    }
                }
                if(flag>n)
                {
                    cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB can't be
allocated";
                }
            }
            break;
        case 2:
            for(y=1;y<=n;y++)
            {
                for(z=y;z<=n;z++)
                {
                    if(m[y]>m[z])
                    {
                        temp=m[y];
                        m[y]=m[z];
                        m[z]=temp;
                        temp1=po[y];
                        po[y]=po[z];
                        po[z]=temp1;
                    }
                }
            }
            for(i=1;i<=j;i++)
            {
                flag=1;
                for(k=1;k<=n;k++)
                {
                    if(p[i]<=m[k])
                    {
                        cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB
allocated at memory partition:\t"<<po[k];
                        m[k]=m[k]-p[i];
                        break;
                    }
                    else
                    {
                        flag++;
                    }
                }
                if(flag>n)
                {
                    cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB can't be
allocated";
                }
            }
            break;
        case 3:
            for(y=1;y<=n;y++)
            {
                for(z=y;z<=n;z++)
                {
                    if(m[y]<m[z])
                    {
                        temp=m[y];
                        m[y]=m[z];
                        m[z]=temp;
                        temp1=po[y];
```

```
                        po[y]=po[z];
                        po[z]=temp1;
                    }
                }
            }
            for(i=1;i<=j;i++)
            {
                flag=1;
                for(k=1;k<=n;k++)
                {
                    if(p[i]<=m[k])
                    {
                        cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB
allocated at memory partition:\t"<<po[k];
                        m[k]=m[k]-p[i];
                        break;
                    }
                    else
                    {
                        flag++;
                    }
                }
                if(flag>n)
                {
                    cout<<"\nProcess "<<i<<" whose memory size is "<<p[i]<<"KB can't be
allocated";
                }
            }
            break;
    }
    return 0;
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:~$ cd /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ g++ 1a.cpp -o 1a
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./1a
            -----Welcome---
Enter memory total partitions:  4

Enter memory size for

partition 1 :   120

partition 2 :   1

partition 3 :   2

partition 4 :   3

Enter total number of process:  4

Enter memory size for

process 1 :     1

process 2 :     2

process 3 :     3

process 4 :     4

**Menu**
1.first fit
2.best fit
3.worst fit
enter choice:   1

Process 1 whose memory size is 1KB allocated at memory partition:       1
Process 2 whose memory size is 2KB allocated at memory partition:       1
Process 3 whose memory size is 3KB allocated at memory partition:       1
Process 4 whose memory size is 4KB allocated at memory partition:       1shara-d@Rohans-Workstation:/mnt/c/Users/shar
```

```
enter choice:   2

Process 1 whose memory size is 1KB allocated at memory partition:       2
Process 2 whose memory size is 2KB allocated at memory partition:       3
Process 3 whose memory size is 3KB allocated at memory partition:       4
Process 4 whose memory size is 4KB allocated at memory partition:       1shara-d@Rohans-Workstation:/mnt/c/Users/shar
enter choice:   3

Process 1 whose memory size is 1KB allocated at memory partition:       1
Process 2 whose memory size is 2KB allocated at memory partition:       1
Process 3 whose memory size is 3KB allocated at memory partition:       1
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

(b)

i. FIFO

```
#include<stdio.h>

int main()
{
    int reference_string[10], page_faults = 0, m, n, s, pages, frames;
    printf("\nEnter Total Number of Pages:\t");
    scanf("%d", &pages);
    printf("\nEnter values of Reference String:\n");
    for(m = 0; m < pages; m++)
    {
        printf("Value No. [%d]:\t", m + 1);
        scanf("%d", &reference_string[m]);
    }
    printf("\nEnter Total Number of Frames:\t");
    {
        scanf("%d", &frames);
    }
    int temp[frames];
    for(m = 0; m < frames; m++)
    {
        temp[m] = -1;
    }
    for(m = 0; m < pages; m++)
    {
        s = 0;
        for(n = 0; n < frames; n++)
        {
            if(reference_string[m] == temp[n])
            {
                s++;
                page_faults--;
            }
        }
        page_faults++;
        if((page_faults <= frames) && (s == 0))
        {
            temp[m] = reference_string[m];
        }
        else if(s == 0)
        {
            temp[(page_faults - 1) % frames] = reference_string[m];
        }
        printf("\n");
        for(n = 0; n < frames; n++)
        {
            printf("%d\t", temp[n]);
```

```
        }
    }
    printf("\nTotal Page Faults:\t%d\n", page_faults);
    return 0;
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc bfifo.c -o bfifo
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./bfifo

Enter Total Number of Pages:    5

Enter values of Reference String:
Value No. [1]:  4
Value No. [2]:  1
Value No. [3]:  2
Value No. [4]:  4
Value No. [5]:  5

Enter Total Number of Frames:   3

4       -1      -1
4       1       -1
4       1       2
4       1       2
5       1       2
Total Page Faults:      4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

ii. LRU

```
#include<stdio.h>

int main()
{
    int frames[10], temp[10], pages[10];
    int total_pages, m, n, position, k, l, total_frames;
    int a = 0, b = 0, page_fault = 0;
    printf("\nEnter Total Number of Frames:\t");
    scanf("%d", &total_frames);
    for(m = 0; m < total_frames; m++)
    {
        frames[m] = -1;
    }
    printf("Enter Total Number of Pages:\t");
    scanf("%d", &total_pages);
    printf("Enter Values for Reference String:\n");
    for(m = 0; m < total_pages; m++)
    {
        printf("Value No.[%d]:\t", m + 1);
        scanf("%d", &pages[m]);
    }
    for(n = 0; n < total_pages; n++)
    {
        a = 0, b = 0;
        for(m = 0; m < total_frames; m++)
```

```
        {
            if(frames[m] == pages[n])
            {
                a = 1;
                b = 1;
                break;
            }
        }
        if(a == 0)
        {
            for(m = 0; m < total_frames; m++)
            {
                if(frames[m] == -1)
                {
                    frames[m] = pages[n];
                    b = 1;
                    break;
                }
            }
        }
        if(b == 0)
        {
            for(m = 0; m < total_frames; m++)
            {
                temp[m] = 0;
            }
            for(k = n - 1, l = 1; l <= total_frames - 1; l++, k--)
            {
                for(m = 0; m < total_frames; m++)
                {
                    if(frames[m] == pages[k])
                    {
                        temp[m] = 1;
                    }
                }
            }
            for(m = 0; m < total_frames; m++)
            {
                if(temp[m] == 0)
                    position = m;
            }
            frames[position] = pages[n];
            page_fault++;
        }
        printf("\n");
        for(m = 0; m < total_frames; m++)
        {
            printf("%d\t", frames[m]);
        }
    }
    printf("\nTotal Number of Page Faults:\t%d\n", page_fault);
    return 0;
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc blru.c -o blru
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./blru

Enter Total Number of Frames:   4
Enter Total Number of Pages:    5
Enter Values for Reference String:
Value No.[1]:   5
Value No.[2]:   3
Value No.[3]:   1
Value No.[4]:   2
Value No.[5]:   4

5       -1      -1      -1
5       3       -1      -1
5       3       1       -1
5       3       1       2
4       3       1       2
Total Number of Page Faults:    1
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

iii. OPT

```c
#include<stdio.h>

int main()
{
    int reference_string[25], frames[25], interval[25];
    int pages, total_frames, page_faults = 0;
    int m, n, temp, flag, found;
    int position, maximum_interval, previous_frame = -1;
    printf("\nEnter Total Number of Pages:\t");
    scanf("%d", &pages);
    printf("\nEnter Values of Reference String\n");
    for(m = 0; m < pages; m++)
    {
        printf("Value No.[%d]:\t", m + 1);
        scanf("%d", &reference_string[m]);
    }
    printf("\nEnter Total Number of Frames:\t");
    scanf("%d", &total_frames);
    for(m = 0; m < total_frames; m++)
    {
        frames[m] = -1;
    }
    for(m = 0; m < pages; m++)
    {
        flag = 0;
        for(n = 0; n < total_frames; n++)
        {
            if(frames[n] == reference_string[m])
            {
                flag = 1;
                printf("\t");
                break;
            }
        }
```

```
        if(flag == 0)
        {
            if (previous_frame == total_frames - 1)
            {
                for(n = 0; n < total_frames; n++)
                {
                    for(temp = m + 1; temp < pages; temp++)
                    {
                        interval[n] = 0;
                        if (frames[n] == reference_string[temp])
                        {
                            interval[n] = temp - m;
                            break;
                        }
                    }
                }
                found = 0;
                for(n = 0; n < total_frames; n++)
                {
                    if(interval[n] == 0)
                    {
                        position = n;
                        found = 1;
                        break;
                    }
                }
            }
            else
            {
                position = ++previous_frame;
                found = 1;
            }
            if(found == 0)
            {
                maximum_interval = interval[0];
                position = 0;
                for(n = 1; n < total_frames; n++)
                {
                    if(maximum_interval < interval[n])
                    {
                        maximum_interval = interval[n];
                        position = n;
                    }
                }
            }
            frames[position] = reference_string[m];
            printf("FAULT\t");
            page_faults++;
        }
        for(n = 0; n < total_frames; n++)
        {
            if(frames[n] != -1)
            {
                printf("%d\t", frames[n]);
            }
        }
        printf("\n");
    }
    printf("\nTotal Number of Page Faults:\t%d\n", page_faults);
    return 0;
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc bopt.c -o bopt
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./bopt

Enter Total Number of Pages:    5

Enter Values of Reference String
Value No.[1]:   5
Value No.[2]:   3
Value No.[3]:   1
Value No.[4]:   2
Value No.[5]:   4

Enter Total Number of Frames:   4
FAULT   5
FAULT   5       3
FAULT   5       3       1
FAULT   5       3       1       2
FAULT   4       3       1       2

Total Number of Page Faults:    5
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

(c)

```c
#include <stdio.h>

/* c program which implement FIFO, LRU, and optimal page replacement algorithms */

int n, pg[30], fr[10];
void fifo();
void optimal();
void lru();
void main()
{
    int i, ch;
    printf("\nEnter total number of pages:");
    scanf("%d", &n);
    printf("\nEnter page references:");
    for (i = 0; i<n; i++)           //accepting sequence
        scanf("%d", &pg[i]);
    do
    {
        printf("\n\tMENU\n");
        printf("\n1)FIFO");
        printf("\n2)OPTIMAL");
        printf("\n3)LRU");
        printf("\n4)Exit");
        printf("\nEnter your choice:");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: fifo();
                break;
            case 2: optimal();
                break;
            case 3: lru();
```

```
                break;
            }
        } while (ch != 4);
        getchar();
    }
    void fifo()
    {
        int i, f, r, s, count, flag, num, psize;
        f = 0;
        r = 0;
        s = 0;
        flag = 0;
        count = 0;

        printf("\nEnter size of page frame:");
        scanf("%d", &psize);
        for (i = 0; i<psize; i++)
        {
            fr[i] = -1;
        }
        while (s<n)
        {
            flag = 0;
            num = pg[s];

            //check wether the page is already exist

            for (i = 0; i<psize; i++)
            {
                if (num == fr[i])
                {
                    s++;
                    flag = 1;
                    break;
                }
            }

            //if page is not already exist
            if (flag == 0)
            {
                if (r<psize)
                {
                    fr[r] = pg[s];
                    r++;
                    s++;
                    count++;
                }
                else
                {
                    if (f<psize)
                    {
                        fr[f] = pg[s];
                        s++;
                        f++;
                        count++;
                    }
                    else
                        f = 0;
                }
            }

            //print the page frame

            printf("\n");
```

```
        for (i = 0; i<psize; i++)
        {
            printf("%d\t", fr[i]);
        }
    }
    printf("\nPage Faults=%d", count);
    getchar();
}
void optimal()
{
    int count[10], i, j, k, l, m, p, r, fault, fSize, flag, temp, max, tempflag = 0;
    fault = 0;
    k = 0;
    printf("\nEnter frame size:");
    scanf("%d", &fSize);

    //initilizing frames array

    for (i = 0; i<fSize; i++)
    {
        count[i] = 0;
        fr[i] = -1;
    }
    for (i = 0; i<n; i++)
    {
        flag = 0;
        temp = pg[i];

        //check wether the page is already exist

        for (j = 0; j<fSize; j++)
        {
            if (temp == fr[j])
            {
                flag = 1;
                break;
            }
        }

        //if the page is not already exist and frame has empty slot

        if ((flag == 0) && (k<fSize))
        {
            fault++;
            fr[k] = temp;
            k++;

            //printf("\n Test 0");
        }

            //if the page is not already exist and frame is full

        else if ((flag == 0) && (k == fSize))
        {
            fault++;
            for (l = 0; l<fSize; l++)
            {
                count[l] = 0;
            }

            //apply optimal replacemnt strategy

            for (m = 0; m<fSize; m++)
```

```
            {
                tempflag = 0;
                for (r = i + 1; r<n; r++)
                {
                    if (fr[m] == pg[r])
                    {
                        if (count[m] == 0)
                            count[m] = r;
                        tempflag = 1;
                    }
                }
                if (tempflag != 1)
                {
                    count[m] = n + 1;
                }
            }

            //find optimal page to replace

            p = 0;
            max = count[0];
            for (l = 0; l<fSize; l++)
            {
                if (count[l]>max)
                {
                    max = count[l];
                    p = l;
                }
            }
            fr[p] = temp;
        }

        //print the page frame

        printf("\n");
        for (l = 0; l<fSize; l++)
        {
            printf("%d\t", fr[l]);
        }
    }
    printf("\nTotal number of faults=%d", fault);
    getchar();
}
void lru()
{
    int count[10], i, j, k, fault, f, flag, temp, current, c, dist, least, m, cnt, p,
x;
    fault = 0;
    dist = 0;
    k = 0;
    printf("\nEnter frame size:");
    scanf("%d", &f);

    //initilizing distance and frame array

    for (i = 0; i<f; i++)
    {
        count[i] = 0;//helps to know recently used page
        fr[i] = -1;
    }
    for (i = 0; i<n; i++)
    {
        flag = 0;
        temp = pg[i];
```

```
        //check wether the page is already exist or not

        for (j = 0; j<f; j++)
        {
            if (temp == fr[j])
            {
                flag = 1;
                count[j] = i;
                break;
            }
        }

        //if the page is not already exist and frame has empty slot

        if ((flag == 0) && (k<f))
        {
            fault++;
            fr[k] = temp;
            count[k] = i;
            k++;
        }

            //if the page is not already exist and frame is full

        else if ((flag == 0) && (k == f))
        {
            fault++;

            //find the least recenlty used page

            least = count[0];
            for (m = 0; m<f; m++)
            {
                if (count[m]<least)
                {
                    least = count[m];
                    p = m;
                }
            }
            fr[p] = temp;
            count[p] = i;
            p = 0;
        }

        //print the page frame

        printf("\n");
        for (x = 0; x<f; x++)
        {
            printf("%d\t", fr[x]);
        }
    }
    printf("\nTotal number of faults=%d", fault);
    getchar();
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc cmm.c -o cmm
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./cmm

Enter total number of pages:20

Enter page references:1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3

        MENU

1)FIFO
2)OPTIMAL
3)LRU
4)Exit
Enter your choice:1

Enter size of page frame:3

1       -1      -1
1       2       -1
1       2       3
1       2       3
1       2       3
5       2       3
5       2       3
5       1       3
5       1       6
5       1       6
2       1       6
2       5       6
2       5       6
2       5       3
2       5       3
1       5       3
1       5       3
1       6       3
1       6       3
1       6       2
1       6       2
4       6       2
4       3       2
Page Faults=14
        MENU

1)FIFO
2)OPTIMAL
3)LRU
4)Exit
Enter your choice:2

Enter frame size:3

1       -1      -1
1       2       -1
1       2       3
1       2       3
1       2       3
1       2       5
1       2       5
1       2       5
6       2       5
6       2       5
6       2       5
6       2       5
6       2       3
6       1       3
6       1       3
6       1       3
6       1       3
2       1       3
4       1       3
4       1       3
Total number of faults=9
        MENU
```

```
1)FIFO
2)OPTIMAL
3)LRU
4)Exit
Enter your choice:3

Enter frame size:3

1       -1      -1
1       2       -1
1       2       3
1       2       3
1       2       3
1       2       5
1       2       5
1       2       5
1       2       6
1       2       6
5       2       6
5       2       6
5       3       6
1       3       6
1       3       6
1       3       6
1       3       6
1       2       6
1       2       4
3       2       4
Total number of faults=11
        MENU

1)FIFO
2)OPTIMAL
3)LRU
4)Exit
Enter your choice:4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

File System & Disk Management

File system and Disk Management

(a) Implement the following Disk scheduling algorithms:

i. SSTF ii. SCAN iii. C-SCAN iv. FCFS **(Medium)**

(b) Consider a file of size 1 MB. The size of a disk block is 512Bytes. Assume any number of available free blocks in the disk contiguously or non-contiguously. Implement the following algorithms to perform file allocation. Determine the efficiency of each file allocation strategies.

i. Sequential
ii. Indexed
iii. Linked **(High)**

Solutions_FsDm

(a)

i. SSTF

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int i,j,k,n,m,sum=0,x,y,h;
    cout<<"Enter the size of disk\n";
    cin>>m;
    cout<<"Enter number of requests\n";
    cin>>n;
    cout<<"Enter the requests\n";
    //creating two arrays, array a will store the input
    //I/O requests and array b will store the output
    vector <int> a(n),b;
    //creating a map to store the count of each element
    //in the array a.
    map <int,int> mp;
    for(i=0;i<n;i++){
        cin>>a[i];
        mp[a[i]]++;
    }
    for(i=0;i<n;i++){
        if(a[i]>m){
            cout<<"Error, Unknown position "<<a[i]<<"\n";
            return 0;
        }
    }
    cout<<"Enter the head position\n";
    cin>>h;
    int temp=h;
    int ele;
    b.push_back(h);
    int count=0;
    while(count<n){
        //initially taking diff to be very large.
        int diff=999999;
        //traversing in map to find the least difference
```

```
        for(auto q:mp){
            if(abs(q.first-temp)<diff){
                ele=q.first;
                diff=abs(q.first-temp);
            }
        }
        //deleting the element that has the least
        //difference from the map
        mp[ele]--;
        if(mp[ele]==0){
            mp.erase(ele);
        }
        //adding that element to our output array.
        b.push_back(ele);
        temp=ele;
        count++;
    }
    //printing the output array
    cout<<b[0];
    temp=b[0];
    for(i=1;i<b.size();i++){
        cout<<" -> "<<b[i];
        sum+=abs(b[i]-temp);
        temp=b[i];
    }
    cout<<'\n';
    cout<<"Total head movements = "<< sum<<'\n';
    cout<<"Average head movement = "<<(float)sum/n<<'\n';
    return 0;
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ g++ 2asstf.cpp -o 2asstf
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2asstf
Enter the size of disk
199
Enter number of requests
8
Enter the requests
98 183 37 122 14 124 65 67
Enter the head position
53
53 -> 65 -> 67 -> 37 -> 14 -> 98 -> 122 -> 124 -> 183
Total head movements = 236
Average head movement = 29.5
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

ii. SCAN

```
#include <stdio.h>
void main()
{
    int i,j,n,h,temp=0,dEnd=199,hPos,sum=0,count=1;
    int rq[100],sq[100];

    printf("\nEnter No. of Processes: ");
    scanf("%d",&n);
```

```
printf("\nEnter Head value: ");
scanf("%d",&h);

//Enter value into Request Queue

printf("\nEnter elements into Request Queue");
for(i=0;i<n;i++)
{
    scanf(" %d",&rq[i]);
}
rq[i]=h;
rq[i+1]=0;

//Scheduling - SCAN

//Sort

for(i=0;i<n;i++)
{
    for(j=0;j<n-1;j++)
    {
        if(rq[j]>rq[j+1])
        {
            temp=rq[j];
            rq[j]=rq[j+1];
            rq[j+1]=temp;
        }
    }
}

//Position
for(i=0;i<n;i++)
{
    if(rq[i]>h)
    {
        hPos=i-1;
        break;
    }
}

//Schedule

sq[0]=h;
printf("\nScheduling\n");
if(h<(dEnd-h))
{
    for(i=hPos;i>=0;i--)
    {
        sq[count]=rq[i];
        count++;
        printf("\t%d ",rq[i]);
    }
    for(i=hPos+1;i<n;i++)
    {
        sq[count]=rq[i];
        count++;
        printf("\t%d ",rq[i]);
    }
}
else
{
    for(i=hPos+1;i<n;i++)
    {
        sq[count]=rq[i];
```

```
            count++;
            printf("\t%d ",rq[i]);
        }
        for(i=hPos;i>=0;i--)
        {
            sq[count]=rq[i];
            count++;
            printf("\t%d ",rq[i]);
        }
    }

    printf("\n Head Movements: ");
    for(i=0;i<n;i++)
    {
        if(sq[i]>sq[i+1])
        {
            sum+=(sq[i]-sq[i+1]);
        }
        else
        {
            sum+=(sq[i+1]-sq[i]);
        }
    }
    printf(" %d \n",sum);
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc 2ascan.c -o 2ascan
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2ascan

Enter No. of Processes: 5

Enter Head value: 7

Enter elements into Request Queue4
7
9
2
31

Scheduling
        7       4       2       9       31
 Head Movements:  34
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

iii. C-SCAN

```
#include <stdio.h>
void main()
{
    int i,j,n,h,temp=0,dEnd=199,hPos,sum=0,count=1;
    int rq[100],sq[100];

    printf("\nEnter No. of Processes: ");
    scanf("%d",&n);

    printf("\nEnter Head value: ");
    scanf("%d",&h);

    //Enter value into Request Queue

    printf("\nEnter elements into Request Queue");
```

```
for(i=0;i<n;i++)
{
    scanf(" %d",&rq[i]);
}
rq[i]=h;
rq[i+1]=0;

//Scheduling - CSCAN

//Sort

for(i=0;i<n;i++)
{
    for(j=0;j<n-1;j++)
    {
        if(rq[j]>rq[j+1])
        {
            temp=rq[j];
            rq[j]=rq[j+1];
            rq[j+1]=temp;
        }
    }
}

//Position
for(i=0;i<n;i++)
{
    if(rq[i]>h)
    {
        hPos=i-1;
        break;
    }
}

//Schedule

sq[0]=h;
printf("\nScheduling\n");
if(h<(dEnd-h))
{
    for(i=hPos;i>=0;i--)
    {
        sq[count]=rq[i];
        count++;
        printf("\t%d ",rq[i]);
    }
    for(i=n-1;i>hPos;i--)
    {
        sq[count]=rq[i];
        count++;
        printf("\t%d ",rq[i]);
    }
}
else
{
    for(i=hPos+1;i<n;i--)
    {
        sq[count]=rq[i];
        count++;
        printf("\t%d ",rq[i]);
    }
    for(i=0;i>=hPos;i++)
    {
        sq[count]=rq[i];
```

```
            count++;
            printf("\t%d ",rq[i]);
        }
    }

    printf("\n Head Movements: ");
    for(i=0;i<n;i++)
    {
        if(sq[i]>sq[i+1])
        {
            sum+=(sq[i]-sq[i+1]);
        }
        else
        {
            sum+=(sq[i+1]-sq[i]);
        }
    }
    printf(" %d \n",sum);
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc 2aCscan.c -o 2aCscan
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2aCscan

Enter No. of Processes: 6

Enter Head value: 8

Enter elements into Request Queue9
2
4
5
1
8

Scheduling
        8       5       4       2       1       9
 Head Movements:  15
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

iv. FCFS

```
#include<stdio.h>

void main()
{
    int h,i,rq[100],sum=0,n,j;

    printf("\n Enter the length: ");
    scanf("%d",&n);
    printf("\n Enter the Head Value: ");
    scanf("%d",&h);

    //Input into Request Queue

    printf("\n Enter the Request Queue ");
    for(i=1;i<n+1;i++)
    {
        scanf("%d",&rq[i]);
    }
```

```
    //Scheduling - FCFS

    rq[0]=h;
    for(j=0;j<n;j++)
    {
        if(rq[j]>rq[j+1])
        {
            sum=(sum+(rq[j]-rq[j+1]));
        }
        else
        {
            sum=(sum+(rq[j+1]-rq[j]));
        }
    }
    printf("\n Total Head movements are %d \n",sum);
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ gcc 2afcfs.c -o 2afcfs
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2afcfs

 Enter the length: 3

 Enter the Head Value: 4

 Enter the Request Queue 2
3
6

 Total Head movements are 6
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

(b)

i. Sequential

```
#include<stdio.h>
#include<conio.h>
main()
{
    int n,i,j,b[20],sb[20],t[20],x,c[20][20];
    clrscr();
    printf("Enter no.of files:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter no. of blocks occupied by file%d",i+1);
        scanf("%d",&b[i]);
        printf("Enter the starting block of file%d",i+1);
        scanf("%d",&sb[i]);
        t[i]=sb[i];
        for(j=0;j<b[i];j++)
            c[i][j]=sb[i]++;
    }
    printf("Filename\tStart block\tlength\n");
    for(i=0;i<n;i++)
```

```
        printf("%d\t %d \t%d\n",i+1,t[i],b[i]);
    printf("blocks occupiedare:");
    for(i=0;i<n;i++)
    { printf("fileno%d",i+1);
        for(j=0;j<b[i];j++)
            printf("\t%d",c[i][j]);
        printf("\n");
    }
    getch();
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ g++ 2bseq.cpp -o 2bseq
2bseq.cpp:3:6: warning: ISO C++ forbids declaration of 'main' with no type [-Wreturn-type]
    3 | main()
      |      ^
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2bseq
Enter no.of files:1
Enter no. of blocks occupied by file110
Enter the starting block of file12
Filename        Start block     length
1       2       10
blocks occupiedare:fileno1      2       3       4       5       6       7       8       9       10      11
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2bseq
Enter no.of files:2
Enter no. of blocks occupied by file14
Enter the starting block of file12
Enter no. of blocks occupied by file210
Enter the starting block of file25
Filename        Start block     length
1       2       4
2       5       10
blocks occupiedare:fileno1      2       3       4       5
fileno2 5       6       7       8       9       10      11      12      13      14
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

ii. Indexed

```
#include<stdio.h>
#include<conio.h>
main()
{
    int n,m[20],i,j,ib[20],b[20][20];
    clrscr();
    printf("Enter no. of files:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    { printf("Enter index block :",i+1);
        scanf("%d",&ib[i]);
        printf("Enter blocks occupied by file%d:",i+1);
        scanf("%d",&m[i]);
        printf("enter blocks of file%d:",i+1);
        for(j=0;j<m[i];j++)
            scanf("%d",&b[i][j]);
    } printf("\nFile\t index\tlength\n");
    for(i=0;i<n;i++)
        printf("%d\t%d\t%d\n",i+1,ib[i],m[i]);
    printf("blocks occupiedare:");
    for(i=0;i<n;i++)
    { printf("fileno%d",i+1);
        for(j=0;j<m[i];j++)
```

```
            printf("\t%d--->%d\n",ib[i],b[i][j]);
        printf("\n");
    }
    getch();
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ g++ 2bindex.cpp -o 2bindex
2bindex.cpp:3:6: warning: ISO C++ forbids declaration of ‘main’ with no type [-Wreturn-type]
    3 | main()
      |      ^
2bindex.cpp: In function ‘int main()’:
2bindex.cpp:10:14: warning: too many arguments for format [-Wformat-extra-args]
   10 |     { printf("Enter index block :",i+1);
      |              ^~~~~~~~~~~~~~~~~~~~~~~~~~
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2bindex
Enter no. of files:1
Enter index block :2
Enter blocks occupied by file1:9
enter blocks of file1:10
4
3
2
1
2
3
4
5

File     index  length
1       2       9
blocks occupiedare:fileno1      2--->10
        2--->4
        2--->3
        2--->2
        2--->1
        2--->2
        2--->3
        2--->4
        2--->5

shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2bindex
Enter no. of files:2
Enter index block :3
Enter blocks occupied by file1:4
enter blocks of file1:9
4
6
7
Enter index block :5
Enter blocks occupied by file2:2
enter blocks of file2:10
8

File     index  length
1       3       4
2       5       2
blocks occupiedare:fileno1      3--->9
        3--->4
        3--->6
        3--->7

fileno2 5--->10
        5--->8

shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```

iii. Linked

```
#include<stdio.h>
#include<conio.h>
struct file
{
    char fname[10];
    int start,size,block[10];
}f[10];
main()
{
    int i,j,n;
    clrscr();
    printf("Enter no. of files:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter file name:");
        scanf("%s",&f[i].fname);
        printf("Enter starting block:");
        scanf("%d",&f[i].start);
        f[i].block[0]=f[i].start;
        printf("Enter no.of blocks:");
        scanf("%d",&f[i].size);
        printf("Enter block numbers:");
        for(j=1;j<=f[i].size;j++)
        {
            scanf("%d",&f[i].block[j]);
        }
    }
    printf("File\tstart\tsize\tblock\n");
    for(i=0;i<n;i++)
    {
        printf("%s\t%d\t%d\t",f[i].fname,f[i].start,f[i].size);
        for(j=0;j<f[i].size;j++)
            printf("%d--->",f[i].block[j]);
        printf("%d",f[i].block[j]);
        printf("\n");
    }
    getch();
}
```

```
shara-d@Rohans-Workstation: /mnt/c/Users/shara/OS/Lab4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ g++ 2blinked.cpp -o 2blinked
2blinked.cpp:8:6: warning: ISO C++ forbids declaration of ‘main’ with no type [-Wreturn-type]
    8 | main()
      |      ^
2blinked.cpp: In function ‘int main()’:
2blinked.cpp:17:17: warning: format ‘%s’ expects argument of type ‘char*’, but argument 2 has type ‘char (*)[10]’ [-W
format=]
   17 |         scanf("%s",&f[i].fname);
      |                ~^  ~~~~~~~~~~~
      |                 |  |
      |                 |  char (*)[10]
      |                char*
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$ ./2blinked
Enter no. of files:2
Enter file name:sharadindu
Enter starting block:20
Enter no.of blocks:10
Enter block numbers:4
12
15
45
32
25
23
33
41
22
Enter file name:sharadindu1
Enter starting block:12
Enter no.of blocks:9
Enter block numbers:12
2
2
31
33
1
2
3
4
File    start   size    block
sharadindu      20      10      20--->4--->12--->15--->45--->32--->25--->23--->33--->41--->1918986355
sharadindu1     12      9       12--->12--->2--->2--->31--->33--->1--->2--->3--->4
shara-d@Rohans-Workstation:/mnt/c/Users/shara/OS/Lab4$
```