Green University of Bangladesh
Department of Computer Science and Engineering(CSE)
Faculty of Sciences and Engineering
Semester: (Spring , Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #05

Course Title: Operating System Lab

Course Code: CSE - 310 Section: 213_D5

Lab Experiment Name: Contiguous Memory Allocation Techniques.

Student Details

Name	ID
MD Dulal Hossain	213902116

Lab Date	:	**22 - 05 - 2024**
Submission Date	:	**29 - 05 - 2024**
Course Teacher's Name	:	**Md. Solaiman Mia**

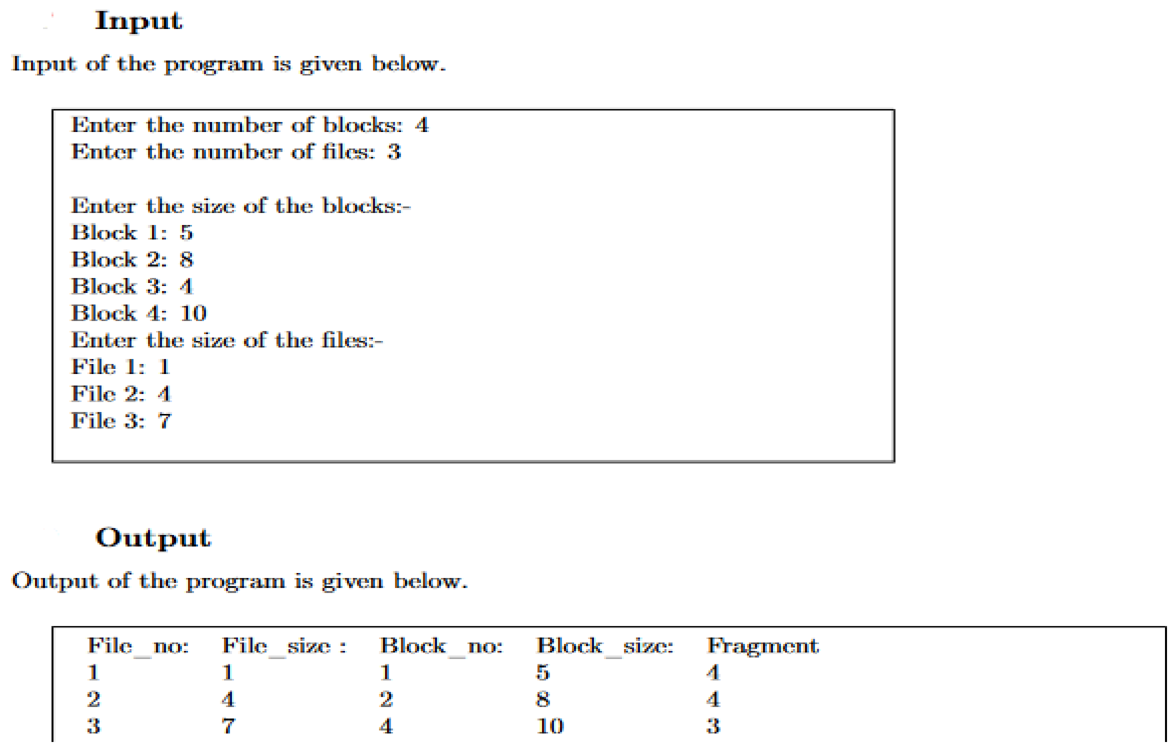
[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status
Marks: …………………………………… **Signature:.....................**
Comments:... **Date:..............................**

Task :

Title :

Implement first fit contiguous memory allocation algorithm.



Input

Input of the program is given below.

```
Enter the number of blocks: 4
Enter the number of files: 3

Enter the size of the blocks:-
Block 1: 5
Block 2: 8
Block 3: 4
Block 4: 10
Enter the size of the files:-
File 1: 1
File 2: 4
File 3: 7
```

Output

Output of the program is given below.

```
File_no:   File_size :   Block_no:   Block_size:   Fragment
1          1             1           5             4
2          4             2           8             4
3          7             4           10            3
```

Algorithms :

1. Initialize arrays frag, b, f, bf, and ff. Set variables nb, nf, allocated_files, and total_internal_frag to 0.
2. Prompt the user to enter the number of blocks (nb).
3. Prompt the user to enter the number of files (nf).
4. Prompt the user to enter the sizes of the blocks and store them in array b.
5. Prompt the user to enter the sizes of the files and store them in array f.
6. For each file:
   a. For each block:
   i. Check if the block is unallocated (bf[j] -ne 1).
   ii. Calculate the remaining space if the file is placed in this block (temp = b[j] - f[i]).
   iii. If the block can accommodate the file (temp >= 0): Assign the block to the file (ff[i] = j). Record the internal fragmentation (frag[i] = temp). Mark the block as allocated (bf[j] = 1). Increment allocated_files. Add temp to total_internal_frag. Break the inner loop to move to the next file.
   b. If file is not allocated after checking all blocks, mark it as not allocated (ff[i] = -1 and frag[i] = -1).
7. Print the allocation results with headers: "File_no", "File_size", "Block_no", "Block_size", "Internal_Fragment".
8. For each file, print its allocation status and internal fragmentation if allocated, otherwise print "Not Allocated".
9. Print the total number of files allocated.
10. Print the total internal fragmentation.
11. Print the header "Unused blocks:".
12. For each block, if it is unallocated (bf[i] -ne 1), print the block number and size.

Source Code in Hand Written :

```
#!/bin/bash
declare -a frag
declare -a b
declare -a f
declare -a bf
declare -a ff
nb=0
nf=0
allocated_files =0
total_internal_frag=0
echo -e "\t\t first fit "
read -p " Enter the number of blocks: " nb
read -p " Enter the number of files: " nf
echo -e "\n Enter size of the blocks:- "
 for (( i=1; i<= nb; i++ )) ; do
   read -p "Block$i: " b[$i]
  done
echo -e " Enter size of the filess:- "
   for (( i=1 ; i<= nf ; i++ )) ; do
   Read -p " File $i: " f [$i]
 done
  for (( i=1; i<=nf; i++)) ;do
     for ((J=1 ; J<=nb; j++)) ; do
           if [[ ${bf[$J]} -ne 1]]; then
             temp= $((b[$J] - f[$i] ))
           if [[temp -ge 0]]; then
             ff [$i]= $J
             frag [$i]= $temp
            bf[${ff [$i]}] =1
     allocated=file = $((alocated_files+1))
```

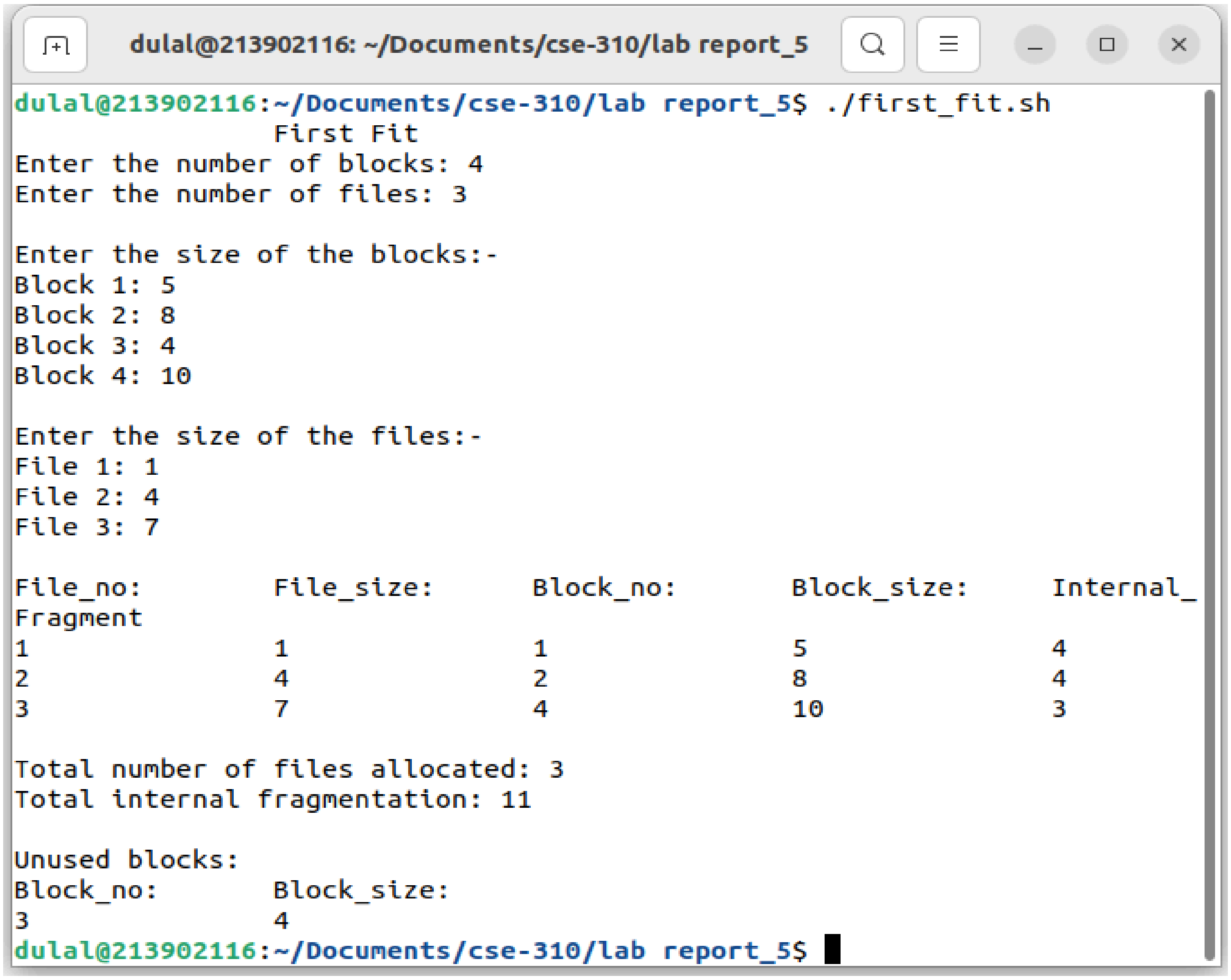
```
    total-internal-frag= $((total-internal-flag +flag[$i]))
     break.
   fi
  fi
done
      if [[$J -gt $nb]]; then
         ff[$i] = -1
         frag[$i]=-1
       fi

done.
echo-e "\n file no: \tfile size: \t Block_no: \tBlock_size
            \tInternal_fragment"
   for ((i=1; i<=nf; i++)); do
    if [[$ {ff[$i]} -ne -1 ]]; then
echo -e "$i \t\t ${f[$i]} \t \t $ {ff[$i]} \t \t${b[${ff[$i]}]}
         \t\t ${frag[$i]}"
else
   echo -e "$i\t\t ${f[$i]} \t\t Not Allocated"
   fi
 done
 echo -e "\nTotal number of files allocated: $ allocated_f"
echo-e"Total internal fragmantation: $ total_internal_frag"
echo-e "\n Unused blocks:"
echo- e  " Block_no: \tBlock_size: "
   for ((i=1; i<= nb; i++)); do
     if [[$ {bf[$i]} -ne 1 ]]; then
      echo-e" $i \t\t $ {b[$i]}"
     fi
 done
```

Source Code in write :

```
#!/bin/bash
declare -a frag
declare -a b
declare -a f
declare -a bf
declare -a ff
nb=0
nf=0
allocated_files=0
total_internal_frag=0
echo -e "\t\tFirst Fit"
read -p "Enter the number of blocks: " nb
read -p "Enter the number of files: " nf
echo -e "\nEnter the size of the blocks:-"
for ((i = 1; i <= nb; i++)); do
   read -p "Block $i: " b[$i]
done
echo -e "\nEnter the size of the files:-"
for ((i = 1; i <= nf; i++)); do
   read -p "File $i: " f[$i]
done
for ((i = 1; i <= nf; i++)); do
   for ((j = 1; j <= nb; j++)); do
      if [[ ${bf[$j]} -ne 1 ]]; then
         temp=$((b[$j] - f[$i]))
         if [[ $temp -ge 0 ]]; then
            ff[$i]=$j
            frag[$i]=$temp
            bf[${ff[$i]}]=1
            allocated_files=$((allocated_files + 1))
            total_internal_frag=$((total_internal_frag + frag[$i]))
            break
         fi
      fi
   done
   if [[ $j -gt $nb ]]; then
      ff[$i]=-1
      frag[$i]=-1
   fi
done
echo -e "\nFile_no:\tFile_size:\tBlock_no:\tBlock_size:\tInternal_Fragment"
for ((i = 1; i <= nf; i++)); do
```

```
    if [[ ${ff[$i]} -ne -1 ]]; then
        echo -e "$i\t\t${f[$i]}\t\t${ff[$i]}\t\t${b[${ff[$i]}]}\t\t${frag[$i]}"
    else
        echo -e "$i\t\t${f[$i]}\t\tNot Allocated"
    fi
done
echo -e "\nTotal number of files allocated: $allocated_files"
echo -e "Total internal fragmentation: $total_internal_frag"
echo -e "\nUnused blocks:"
echo -e "Block_no:\tBlock_size:"
for ((i = 1; i <= nb; i++)); do
    if [[ ${bf[$i]} -ne 1 ]]; then
        echo -e "$i\t\t${b[$i]}"
    fi
done
```

Output :



```
dulal@213902116:~/Documents/cse-310/lab report_5$ ./first_fit.sh
                First Fit
Enter the number of blocks: 4
Enter the number of files: 3

Enter the size of the blocks:-
Block 1: 5
Block 2: 8
Block 3: 4
Block 4: 10

Enter the size of the files:-
File 1: 1
File 2: 4
File 3: 7

File_no:        File_size:      Block_no:       Block_size:     Internal_
Fragment
1               1               1               5               4
2               4               2               8               4
3               7               4               10              3

Total number of files allocated: 3
Total internal fragmentation: 11

Unused blocks:
Block_no:       Block_size:
3               4
dulal@213902116:~/Documents/cse-310/lab report_5$ 
```

Figure 5.1 : Output in show Successfully.

Explain Output :

Blocks : Block number (0-based index) where the File is allocated (if allocated).
Files : File number (1-based index).
Int. Frag. : Internal fragmentation (unused space within the allocated block).

For example:
Block 1 size : 5
Block 2 size : 8
Block 3 size : 4
Block 4 size : 10
File 1 with size 1 is allocated to Block 1 with size 1; internal fragmentation is 4.
File 2 with size 4 is allocated to Block 2 with size 4; internal fragmentation is 4.
File 3 with size 7 is allocated to Block 3 with size 7; internal fragmentation is 3.

Total Internal Fragmentation : 4 + 4 + 3 = 11

Unused Blocks:

Block_no	Block_size
3	4