Lab Experiments (Group Activity -1)

A) Contiguous Allocation Techniques

Implementation of Contiguous allocation techniques:

1. Worst-Fit
2. Best-Fit
3. First-Fit

IMPLEMENTATION DETAILS:

(A) INPUT/s: (Take the inputs from the user:)

a) List number of blocks and processes
b) Free space list of blocks or block size
c) Size of Processes

(B) STEPS TO PERFORM:

Before performing storage allocation, we need to perform the following common steps:

I. During program execution, we need an illusion that we have a large disk space for allocation strategies.

II. Take a big array of structure (representing storage space in terms of sectors). This space is equal to the disk size available for storage.

III. Next step is to see free space list. The free space list is not entirely free. Some portion of this list is occupied by some OS or user files/data.

IV. The free-space is to be maintained in a suitable manner as per allocation strategy.

V. Upon the above formed disk space and free space list, Symbolic file name and size will be inputted by the user and accordingly the assumed disk space as well as free space list will be updated.

VI. Then implement the above mentioned three contiguous allocation techniques. Also, the free space list is updated from the free blocks left out after performing allocation.

a) **Worst-fit**: In worst fit technique largest available block/partition which will hold the page is selected. Blocks are sorted according to their size in descending order.
b) **Best-fit**: Best-fit is one of the optimal technique in which page is stored in the block/partition which is large enough to hold it. Blocks are sorted according to their size in ascending order.
c) **First-fit**: In first-fit technique page is stored in the block which is encountered first that is big enough to hold it.

VII. Also, the free space list is updated from the free blocks left out after performing allocation.

(C) OUTPUT/s:

Processes and files allocated to free blocks. List of processes and files which are not allocated memory. The remaining free space list left out after performing allocation.

B) External and Internal Fragmentation

Calculation of external and internal fragmentation.

IMPLEMENTATION DETAILS:

INPUT/s:

(i) Free space list of blocks from system
(ii) List processes and files from the system

STEPS TO PERFORM:

1. Completing above experiment, you end up getting list of allotted files, remaining part of allotted block and blocks which cannot be allotted.

2. After implementing each allocation algorithm, list the amount of free space blocks left out after performing allocation.

3. When a block which is not at all allocated to any process or file, it adds to external fragmentation.

4. When a file or process is allocated the free block and still some part of it is left unused, we count such unused portion into internal fragmentation.

OUTPUT/s:

Processes and files allocated to free blocks. From the list of unused blocks, we determine the count of total internal and external fragmentation.