

Title: Contiguous memory allocation: next fit and first fit

Objectives:

- To learn the algorithm of the Next fit
- To learn the algorithm of the first fit

Background:

First Fit:

In the first fit, a partition is allocated which is first sufficient from the top of the Main Memory.

Next Fit:

The next fit is a modified version of the 'first fit'. It begins as the first fit to find a free partition but when called next time it starts searching from where it left off, not from the beginning. This policy makes use of a roving pointer. The pointer moves along the memory chain to search for the next fit.

Lab Activities:

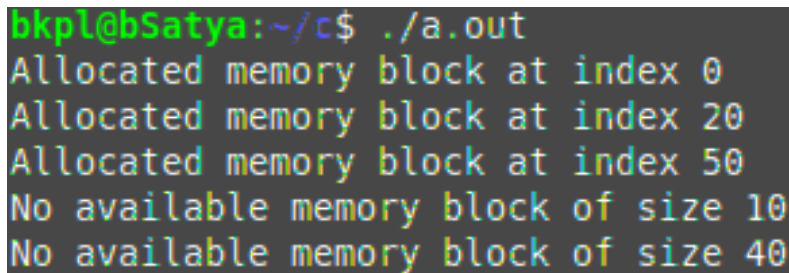
1. Write a program to demonstrate the Next fit algorithm.

```
#include <stdio.h>
#define MAX_MEMORY 100 // total memory size
#define MAX_PROCESSES 10 // maximum number of processes
int memory[MAX_MEMORY]; // memory array
int pointer = 0; // pointer to last allocated block
void next_fit(int size) {
    int found = 0;
    for (int i = pointer; i < MAX_MEMORY; i++) {
        if (memory[i] == 0) {
            found++;
            if (found == size) {
                pointer = i;
                for (int j = i-size+1; j <= i; j++) {
                    memory[j] = 1;
                }
                printf("Allocated memory block at index %d\n", i-size+1); return;
            }
        } else {
            found = 0;
        }
    }
    for (int i = 0; i < pointer; i++) {
```

```

if (memory[i] == 0) {
    found++;
    if (found == size) {
        pointer = i;
        for (int j = i-size+1; j <= i; j++) {
            memory[j] = 1;
        }
        printf("Allocated memory block at index %d\n", i-size+1); return;
    }
} else {
    found = 0;
}
}
printf("No available memory block of size %d\n", size); }
int main() {
    next_fit(20);
    next_fit(30);
    next_fit(50);
    next_fit(10);
    next_fit(40);
    return 0;
}
}

```



```

bkpl@bSatya:~/c$ ./a.out
Allocated memory block at index 0
Allocated memory block at index 20
Allocated memory block at index 50
No available memory block of size 10
No available memory block of size 40

```

2. Write a program to demonstrate the first fit algorithm.

```

#include<stdio.h>
#include<stdlib.h>
void main(){
    int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j;
    for(i = 0; i < 10; i++)
    {
        flags[i] = 0;
        allocation[i] = -1;
    }
    printf("Enter no. of blocks: ");
    scanf("%d", &bno);
    printf("\nEnter size of each block: ");
    for(i = 0; i < bno; i++)
        scanf("%d", &bsize[i]);
    printf("\nEnter no. of processes: ");
    scanf("%d", &pno);

```

```

printf("\nEnter size of each process: ");
for(i = 0; i < pno; i++)
scanf("%d", &psize[i]);
for(i = 0; i < pno; i++) //allocation as per first fit
for(j = 0; j < bno; j++)
if(flags[j] == 0 && bsize[j] >= psize[i])
{
allocation[j] = i;
flags[j] = 1;
break;
}
//display allocation details
printf("\nBlock no.\tsize\tprocess no.\tsize");
for(i = 0; i < bno; i++)
{
printf("\n%d\t\t%d\t\t", i+1, bsize[i]);
if(flags[i] == 1)
printf("%d\t\t\t%d", allocation[i]+1, psize[allocation[i]]);
else
printf("Not allocated");
} }

```

```

a.out hello.c p.c
bkpl@bSatya:~/c$ ./a.out
Enter no. of blocks: 5

Enter size of each block: 100
60
40
50
20

Enter no. of processes: 5

Enter size of each process: 20
50
35
25
30

Block no.      size      process no.      size
1             100         1                20
2              60         2                50
3              40         3                35
4              50         4                25
5              20         Not allocated
bkpl@bSatya:~/c$

```

Conclusion:

In the last lab session, we learned about the algorithm of next fit and algorithm of first fit.