

Experiment No. – 07 (Memory Allocation)
Objective:

Implement memory allocation for six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)?

And print the output in following manner for each process under every scheme:

Program:

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int i, j, flag, x, y, min, alloc[100], k = 0, index, bestIdx, size1[100], size2[100];
    int size[6] = {300, 600, 350, 200, 750, 125};
    int process[5] = {115, 500, 358, 200, 375};
    for (i = 0; i < 6; i++)
    {
        size1[i] = size[i];
        size2[i] = size[i];
    }
    printf("-----First Fit Allocation-----\n\n");
    for (j = 0; j < 5; j++)
    {
        flag = 0;
        for (i = 0; i < 6; i++)
        {
            if (size[i] >= process[j])
            {
                flag = 1;
                x = size[i];
                size[i] = size[i] - process[j];
                y = size[i];
                break;
            }
        }
        if (flag == 1)
        {
            printf("Process of size %d is allocated in the memory block of %d and new hole is created of size %d\n", process[j], x, y);
        }
        else
        {
            printf("Process of size %d is not allocated because sufficient contiguous memory hole is not available to load the process.\n", process[j]);
        }
    }
    printf("\n-----Best Fit Allocation-----\n\n");
    for (j = 0; j < 5; j++)
    {
        index = -1;
        for (i = 0; i < 6; i++)
        {
```

```

        if (size1[i] >= process[j])
        {
            if (index == -1)
                index = i;
            else if (size1[index] > size1[i])
                index = i;
        }
    }
    if (index != -1)
    {
        x = size1[index];
        size1[index] -= process[j];
        y = size1[index];
        printf("Process of size %d is allocated in the memory block of %d and new hole is created of size %d\n", process[j], x, y);
    }
    else
        printf("Process of size %d is not allocated because sufficient contiguous memory hole is not available to load the process.\n", process[j]);
}
printf("\n-----For worst fit allocation-----\n\n");
for (j = 0; j < 5; j++)
{
    index = -1;
    for (i = 0; i < 6; i++)
    {
        if (size2[i] >= process[j])
        {
            if (index == -1)
                index = i;
            else if (size2[index] < size2[i])
                index = i;
        }
    }
    if (index != -1)
    {
        x = size2[index];
        size2[index] -= process[j];
        y = size2[index];
        printf("Process of size %d is allocated in the memory block of %d and new hole is created of size %d\n", process[j], x, y);
    }
    else
        printf("Process of size %d is not allocated because sufficient contiguous memory hole is not available to load the process.\n", process[j]);
}
}

```

Input/Output:

```
"E:\OneDrive - ABES\0.Semester\7.Operating Systems Lab (KCS451)\Source Code\Experiment 7 - Memory Allocation Scheme.exe"
-----First Fit Allocation-----
Process of size 115 is allocated in the memory block of 300 and new hole is created of size 185
Process of size 500 is allocated in the memory block of 600 and new hole is created of size 100
Process of size 358 is allocated in the memory block of 750 and new hole is created of size 392
Process of size 200 is allocated in the memory block of 350 and new hole is created of size 150
Process of size 375 is allocated in the memory block of 392 and new hole is created of size 17

-----Best Fit Allocation-----
Process of size 115 is allocated in the memory block of 125 and new hole is created of size 10
Process of size 500 is allocated in the memory block of 600 and new hole is created of size 100
Process of size 358 is allocated in the memory block of 750 and new hole is created of size 392
Process of size 200 is allocated in the memory block of 200 and new hole is created of size 0
Process of size 375 is allocated in the memory block of 392 and new hole is created of size 17

-----For worst fit allocation-----
Process of size 115 is allocated in the memory block of 750 and new hole is created of size 635
Process of size 500 is allocated in the memory block of 635 and new hole is created of size 135
Process of size 358 is allocated in the memory block of 600 and new hole is created of size 242
Process of size 200 is allocated in the memory block of 350 and new hole is created of size 150
Process of size 375 is not allocated because sufficient contiguous memory hole is not available to load the process.

Process returned 0 (0x0)   execution time : 0.014 s
Press any key to continue.
```