Experiment No. 8

Title: Implementation and Performance Evaluation of Dynamic Partitioning Placement Algorithms (First Fit & Best Fit)

Aim: Write a program to demonstrate the concept of dynamic partitioning placement algorithms i.e. Best Fit, First Fit, Worst-Fit etc.

Theory:

There are various memory management schemes in operating system like first fit, best fit and worst fit.

First Fit:

What is First Fit Memory Management Scheme?

In this scheme we check the blocks in a sequential manner which means we pick the first process then compare its size with first block size if it is less than size of block it is allocated otherwise, we move to second block and so on.

When first process is allocated, we move on to the next process until all processes are allocated.

First Fit Algorithm

1. Get no .of Processes and no. of blocks.
2. After that get the size of each block and process requests.

   Now allocate processes
   if(block size >= process size)
   //allocate the process
   Else
   //move on to next block
3. Display the processes with the blocks that are allocated to a respective process.
4. Stop.

Best Fit:

What is Best Fit Memory Management Scheme?

Best fit uses the best memory block based on the Process memory request. In best fit implementation the algorithm first selects the smallest block which can adequately fulfil the memory request by the respective process.

Because of this memory is utilized optimally but as it compares the blocks with the requested memory size it increases the time requirement and hence slower than other methods. It suffers from Internal Fragmentation which simply means that the memory block size is greater than the memory requested by the process, then the free space gets wasted.

Once we encounter a process that requests a memory which is higher than block size, we stop the algorithm.

Best Fit Algorithm

1. Get no .of Processes and no. of blocks.
2. After that get the size of each block and process requests.
3. Then select the best memory block that can be allocated using the above definition.
4. Displaytheprocesseswiththeblocksthatareallocatedtoarespectiveprocess.
5. ValueofFragmentationisoptionaltodisplaytokeeptrackofwastedmemory.
6. Stop.

Program:

First Fit

```
#include<stdio.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++)                    //allocationasperfirst
        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\t\tprocess no.\t\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("Notallocated");

                }
        }
```

Best Fit

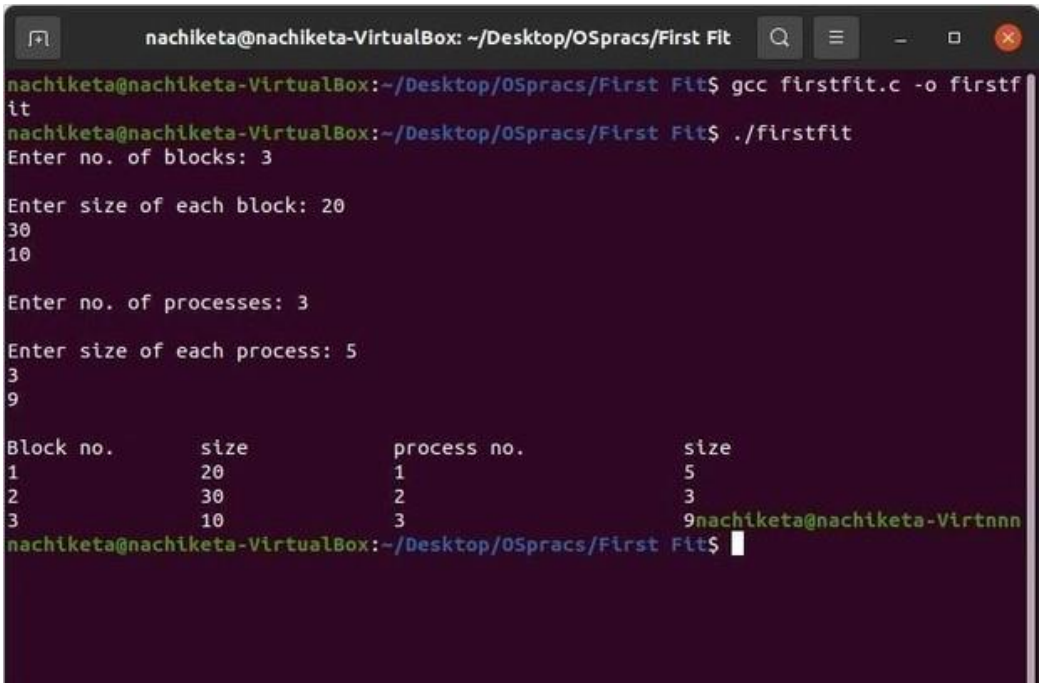
```
#include<stdio.h>
void main()
{ int
        fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999
        ; static int barray[20],parray[20];

        printf("\n\t\t\tMemory Management Scheme -
        Best Fit"); printf("\nEnter the number of blocks:");
        scanf("%d",&nb);
        printf("Enter the number of processes:");
        scanf("%d",&np); printf("\nEnter the size
        of the blocks:-\n");
        for(i=1;i<=nb;i++)
    {
                printf("Block no.%d:",i);
```

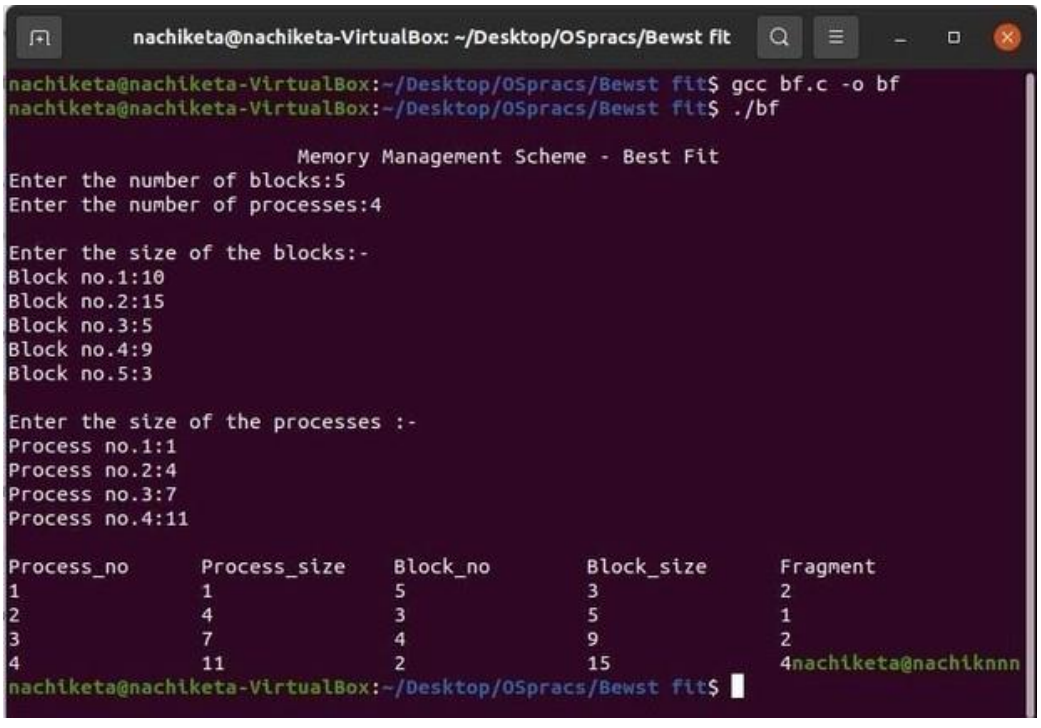
```
        scanf("%d",&b[i]);
    } printf("\nEnter the size of the processes :-\n");
        for(i=1;i<=np;i++)
    { printf("Process no.%d:",i);
        scanf("%d",&p[i]);
    } for(i=1;i<=np;i++)
        { for(j=1;j<=nb;j++)
                { if(barray[j]!=1)
                        { temp=b[j]-p[i];
                                if(temp>=0)
                                if(lowest>temp)
                                        { parray[i]=j
                                                ;
                                                lowest=te
                                                mp
                                                ;
                                        }
                        }
                }
                fragment[i]=lowest;
                barray[parray[i]]=1;
                lowest=10000;
        }
        printf("\nProcess_no\tProcess_size\tBlock_no\tBlock_size\tFragment");
        for(i=1;i<=np && parray[i]!=0;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,p[i],parray[i],b[parray[i]],fragment[i
                                                                        ]);
}
```

Output:

First Fit



Best Fit



Outcome:

Name : GURAV PRATHAMESH MANOHAR Roll No. :36

Implement various Memory Management techniques and evaluate their performance.