

OS-Practical (Demo).

Name :- TUSHAR KUMAR Course :- BSC.IT

Uni. Roll No :- 2023109 Sem/sec :- 2A

Std. ID :- 20051120 Sub. Name :- OS-Practical

Date :- 26 Aug 2021 Sub. Code :-

Solⁿ → ★ Algorithm

1. Start
2. At first get the no. of processes and blocks.
3. Allocate the process by if (size of block \geq size of the process) then allocate the process else move to the next block.
4. Now display the process with block and allocate the respective process.
5. STOP

★ Codings

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int bsize[10], psize[10], bno, pno, flags[10], i, j,  
        allocation[10];
```

```
    for (i=0 ; i<10 ; i++)
```

```
    { flags[i]=0;
```

```
      allocation[i]=-1;  
    }
```



```

printf("Enter no. of blocks : ");
scanf("%d", &bno);
printf("\n Enter size of each block : ");
for(i=0; i<bno; i++)
    scanf("%d", &bsize[i]);

```

```

printf("\n Enter no. of processes : ");
scanf("%d", &pno);
printf("\n Enter size of each process : ");
for(i=0; i<pno; i++)
    scanf("%d", &psize[i]);
for(i=0; i<pno; i++)
    for(j=0; j<bno; j++)
        if(flags[j]==0 && bsize[j]>=psize[i])
        {
            allocation[j]=i;
            flags[j]=1;
            break;
        }

```

```

printf("\n Block no.  It size  It It process no.
        It It size");

```

```

for(i=0; i<bno; i++)
{
    printf("\n %d It It %d It It", i+1, bsize[i]);
    if(flags[i]==1)
        printf("%d It It It %d", allocation[i]+1,
            psize[allocation[i]]);
}

```

```
else  
    printf("Not Allocated");  
}
```


Enter no. of blocks: 3

Enter size of each block: 9

10

46

Enter no. of processes: 3

Enter size of each process: 5

26

10

Block no.	size	process no.	size
1	9	1	5
2	10	3	10
3	46	2	26

②.

Solⁿ 2 → Coding.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int arrival_time[10], burst_time[10], temp[10];
```

```
    int i, smallest, count=0, time, limit;
```

```
    double wait_time=0, turnaround_time=0, end;
```

```
    float average_waiting_time, average_turnaround_time;
```

```
    printf("\n Enter the Total Number of Processes : \t");
```

```
    scanf("%d", &limit);
```

```
    printf("\n Enter Details of %d Processes\n", limit);
```

```
    for (i=0; i < limit; i++)
```

```
    {
```

```
        printf("\n Enter Arrival Time : \t");
```

```
        scanf("%d", &arrival_time[i]);
```

```
        printf("Enter Burst Time : \t");
```

```
        scanf("%d", &burst_time[i]);
```

```
        temp[i] = burst_time[i];
```

```
    }
```



```
burst-time [9] = 9999;  
for (time = 0; count != limit; time++)  
{  
    smallest = 9;  
    for (i = 0; i < limit; i++)  
    {  
        if (arrival-time[i] <= time &&  
            burst-time[i] < burst-time[smallest]  
            && burst-time[i] > 0)  
        {  
            smallest = i;  
        }  
    }  
    burst-time[smallest]--;  
    if (burst-time[smallest] == 0)  
    {  
        count++;  
        end = time + 1;  
        wait-time = wait-time + end - arrival-  
            time[smallest] - temp[smallest];  
        turnaround-time = turnaround-time + end  
            - arrival-time[smallest];  
    }  
}
```

average-waiting-time = wait-time / limit;


```
average_turnaround_time = turnaround_time / limit;  
printf("In In Average Waiting Time : %lf\n",  
       average_waiting_time);  
printf("Average Turnaround Time : %lf\n",  
       average_turnaround_time);  
return 0;  
}
```

Mustafa

Enter the Total Number of Processes: 5

Enter Details of 5 Processesn

Enter Arrival Time: 0

Enter Burst Time: 2

Enter Arrival Time: 0

Enter Burst Time: 1

Enter Arrival Time: 0

Enter Burst Time: 4

Enter Arrival Time: 0

Enter Burst Time: 3

Enter Arrival Time: 0

Enter Burst Time: 2

Average Waiting Time: 3.400000

Average Turnaround Time: 5.800000

...Program finished with exit code 0

Press ENTER to exit console.