Rupesh Dharme
31124

Assignment 06

Memory management - partitioning

problem statement:

Write a c++/Java/Python program to simulate memory placement strategies- first fit, next fit, best fit and worst fit.

learning objectives:

1. To learn and understand memory placement strategies
2. To implement first, next, best and worst fit.

Outcomes:

Student will be able to upderstand and implement memory placement strategies.

S/W H/W requirements:

Windows 10, python 3.9, vscode Editor
4GB RAM 512GB SSD

Theory:

To optimize the use of CPU maximum number of programs need to be braught into memory. This is done by memory management system. It manages and coordinates the allocation of space to process.

Teacher's Signature _____

31124

Page No.
Date: / /

Types of memory allocation:

1. non contiguous:
It allocates storage to processes in various pretions. Due to which the distrubution is scattered.

2. contiguous:
In this type of allocation a single contiguous part of memory is allocated to a process.
- Fixed sized partitioning
- Dynamic partioning.

Memory management strategies

1. First fit: The first location having more than required space is allocated.
2. Next fit: The next loction having more than required space is allocated.
3. Best fit: The partition that will create minimum internal fragment is allocated.
4. Worst fit: The partition that will create maximum internal fragment is allocated.

Teacher's Signature

31124

Algorithms

1. First fit:
   i. loop through available partition
   ii. check program size ≤ partition
   if true : allocate and stop
   if False : continue

2. Next Fit:
   1. set counter = 0
   2. loop through while incrementing counter
   3. if program size ≤ partition
   allocate and stop
   else : continue

3. best fit:
   i. loop through available parts
   ii. find part where (part - prog) is min
   iii. allocate this location

4. Worst fit:
   i. loop through available parts.
   ii. find the biggest part
   iii. allocate this part.
   iv. end.

Teacher's Signature

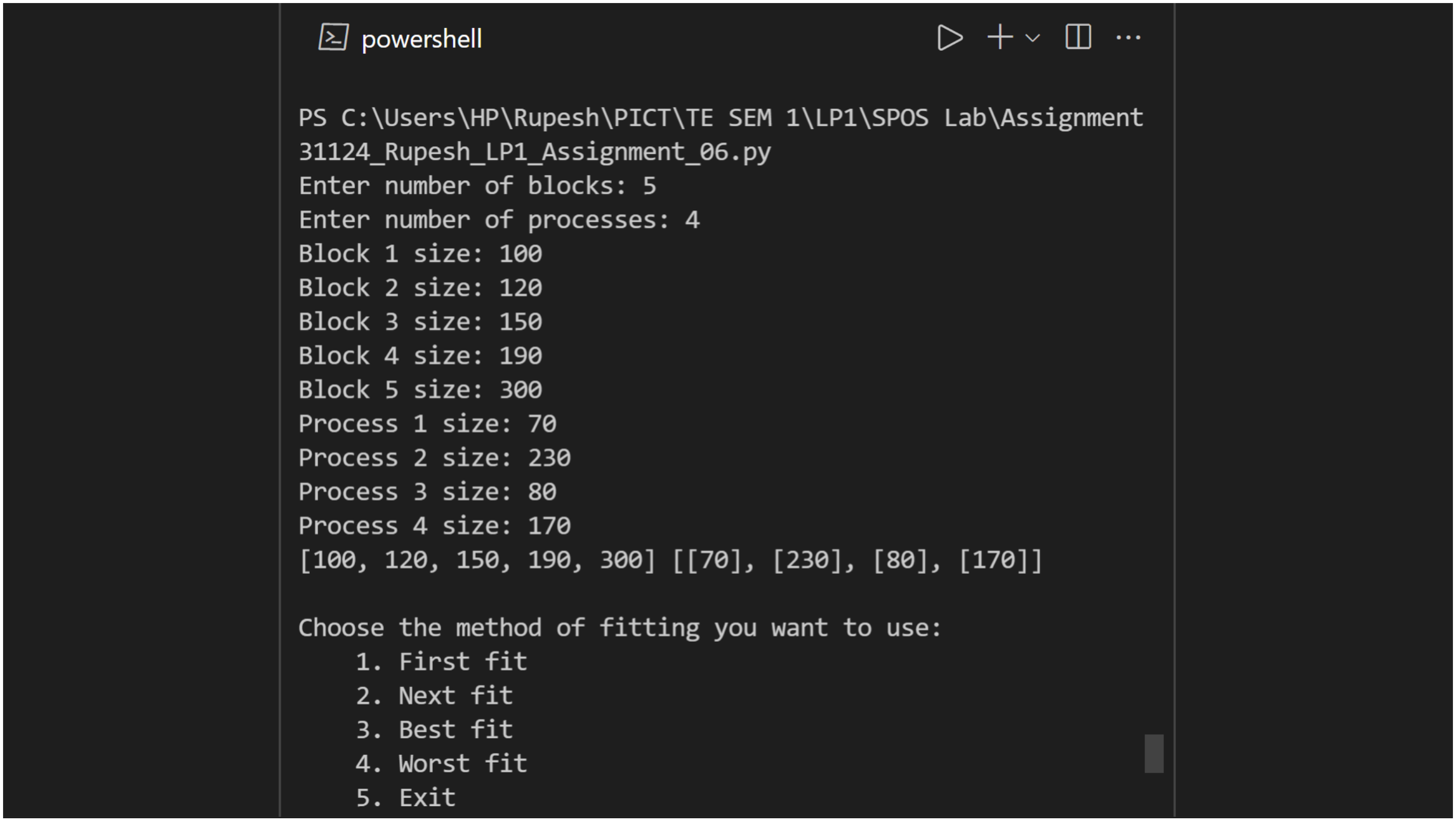
31124

Test cases:
blocks : 10 20 30 40 50
processes : 4 15 25 75 98

case	Exp o/p			actual o/p	result
	proc	block	Fragment		
1	0	4	10		
	1	15	20	same as exp	pass
	2	25	30		
	3	75	wait	-	
	4	98	wait	-	
2	0	4	46		
	1	15	31	same as exp	pass
	2	25	15		
	3	75	wait	-	
	4	98	wait	-	
3	0	4	6		
	1	15	5	same as exp	pass
	2	25	5		
	3	75	wait	-	
	4	98	wait	-	

Conclusion: Memory management strategies understood and implemented using python programming.

Teacher's Signature


powershell
PS C:\Users\HP\Rupesh\PICT\TE SEM 1\LP1\SPOS Lab\Assignment
31124_Rupesh_LP1_Assignment_06.py
Enter number of blocks: 5
Enter number of processes: 4
Block 1 size: 100
Block 2 size: 120
Block 3 size: 150
Block 4 size: 190
Block 5 size: 300
Process 1 size: 70
Process 2 size: 230
Process 3 size: 80
Process 4 size: 170
[100, 120, 150, 190, 300] [[70], [230], [80], [170]]
Choose the method of fitting you want to use:
1. First fit
2. Next fit
3. Best fit
4. Worst fit
5. Exit

powershell

```
Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
1

Process  Process time   block        fragmentation
0         70             1            100
1         230            5            300
2         80             2            120
3         170            4            190

Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
```

powershell

```
Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
2

Process  Process time   block        fragmentation
0         70             Waiting...    -
1         230            Waiting...    -
2         80             Waiting...    -
3         170            Waiting...    -

Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
3
```

powershell

```
Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
3

Process  Process time   block        fragmentation
0         70             1            30
1         230            5            70
2         80             2            40
3         170            4            20

Choose the method of fitting you want to use:
    1. First fit
    3. Best fit
    4. Worst fit
    5. Exit
4
```

powershell

```
Choose the method of fitting you want to use:
    1. First fit
    3. Best fit
    4. Worst fit
    5. Exit
4

Process  Process time   block        fragmentation
0         70             5            230
1         230            5            0
2         80             4            110
3         170            Waiting...   -

Choose the method of fitting you want to use:
    1. First fit
    2. Next fit
    3. Best fit
    4. Worst fit
    5. Exit
5
Thank you
```