بسم الله الرحمن الرحيم

Palestine Polytechnic University

College of Computer Engineering and Information Technology

Operating system

Mai Hashlamoon

191075

Dr.Radwan Tahboub

Part 1:

First Come First Served Algorithm (FCFS):

is the simplest and non-preemptive scheduling algorithm. In First Come First Served (FCFS), the process is assigned to the CPU in the order of their arrival. A queue data structure is used to implement the FCFS scheduling algorithm. The process which is at the head of the ready queue is assigned to the CPU, when CPU is free. Then the process which is running is removed from the queue. When a new process enters into the ready queue, it is placed onto the tail of the ready queue.

Round Robin Scheduling Algorithm:

Round Robin (RR) Scheduling Algorithm is design for the time sharing system. This algorithm is the preemptive scheduling algorithm. In Round Robin Scheduling Algorithm a small unit of time called as time quantum or time slice for which the CPU is provided to each job. CPU is assigned to the each job for the duration equal to the time quantum in cyclic order. This time quantum, time slice or time interval is generally of the order of 10 to 100 milliseconds. Ready queue in the Round Robin Scheduling Algorithm is treated as the circular queue.

Shortest Job First (SJF) :

Shortest Job First (SJF) Scheduling Algorithm is based upon the burst time of the process. The processes are put into the ready queue based on their burst times. In this algorithm, the process with the least burst time is processed first. The burst time of only those processes is compared that are present or have arrived until that time. It is also non-preemptive in nature. Its preemptive version is called Shortest Remaining Time First (SRTF) algorithm.

The information in the Processes.txt

4096 (physical memory)
512 (page size , frame size)
10 (quantum)
1   (context switch)
0 3 10 8192
1 0 12 2048
2 1 3 512
3 5 21 4096
4 9 7 1024

Part 2:

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames. The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation. Paging is used for faster access to data, and it is a logical concept.

A page table is the data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses.

Physical Address identifies a physical location of required data in a memory. The user never directly deals with the physical address but can access by its corresponding logical address

Logical Address or Virtual Address (represented in bits): An address generated by the CPU

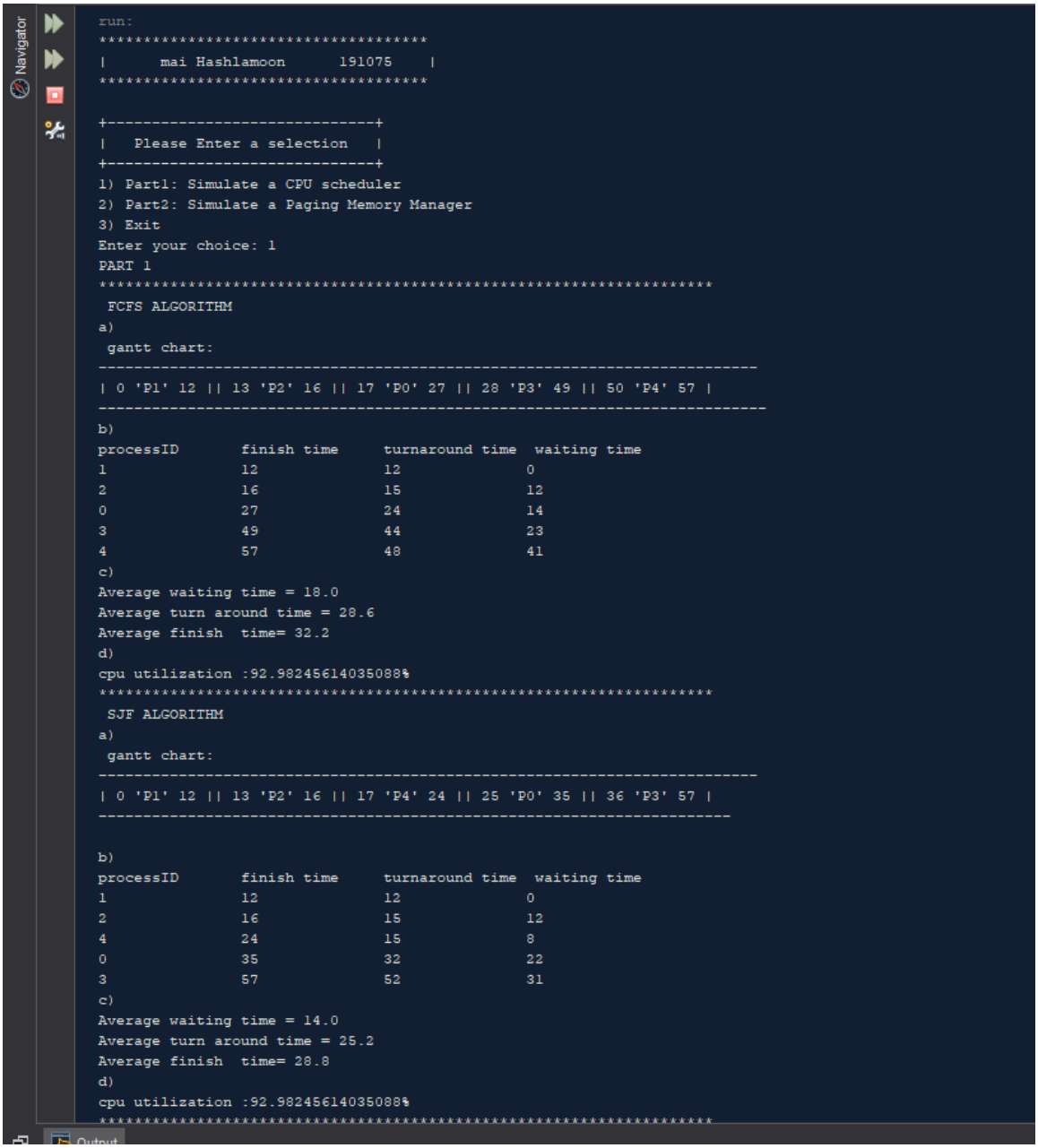
Physical Address (represented in bits): An address actually available on memory unit

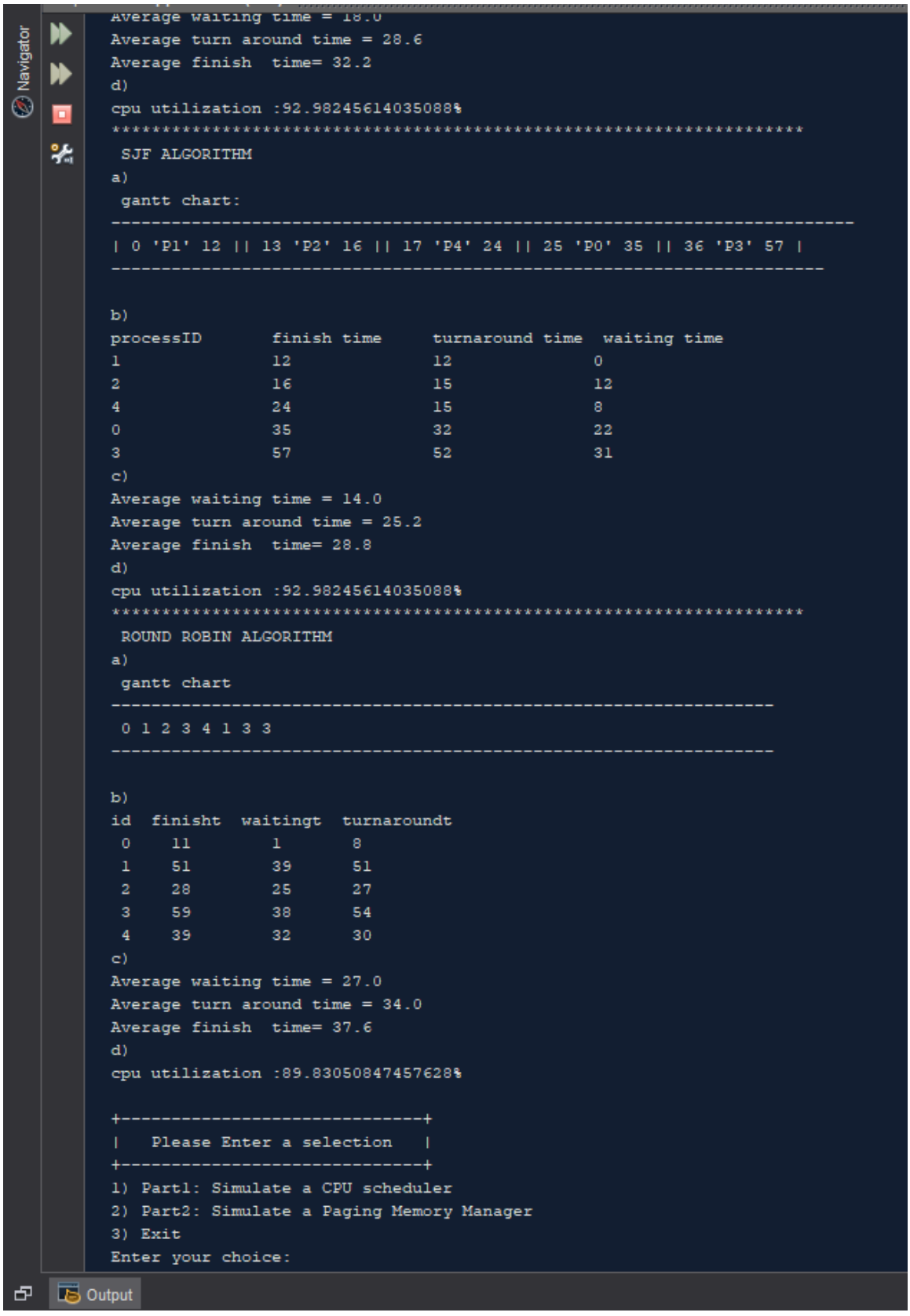
The Physical Address Space is conceptually divided into a number of fixed-size blocks, called frames.

The Logical address Space is also splitted into fixed-size blocks, called pages.

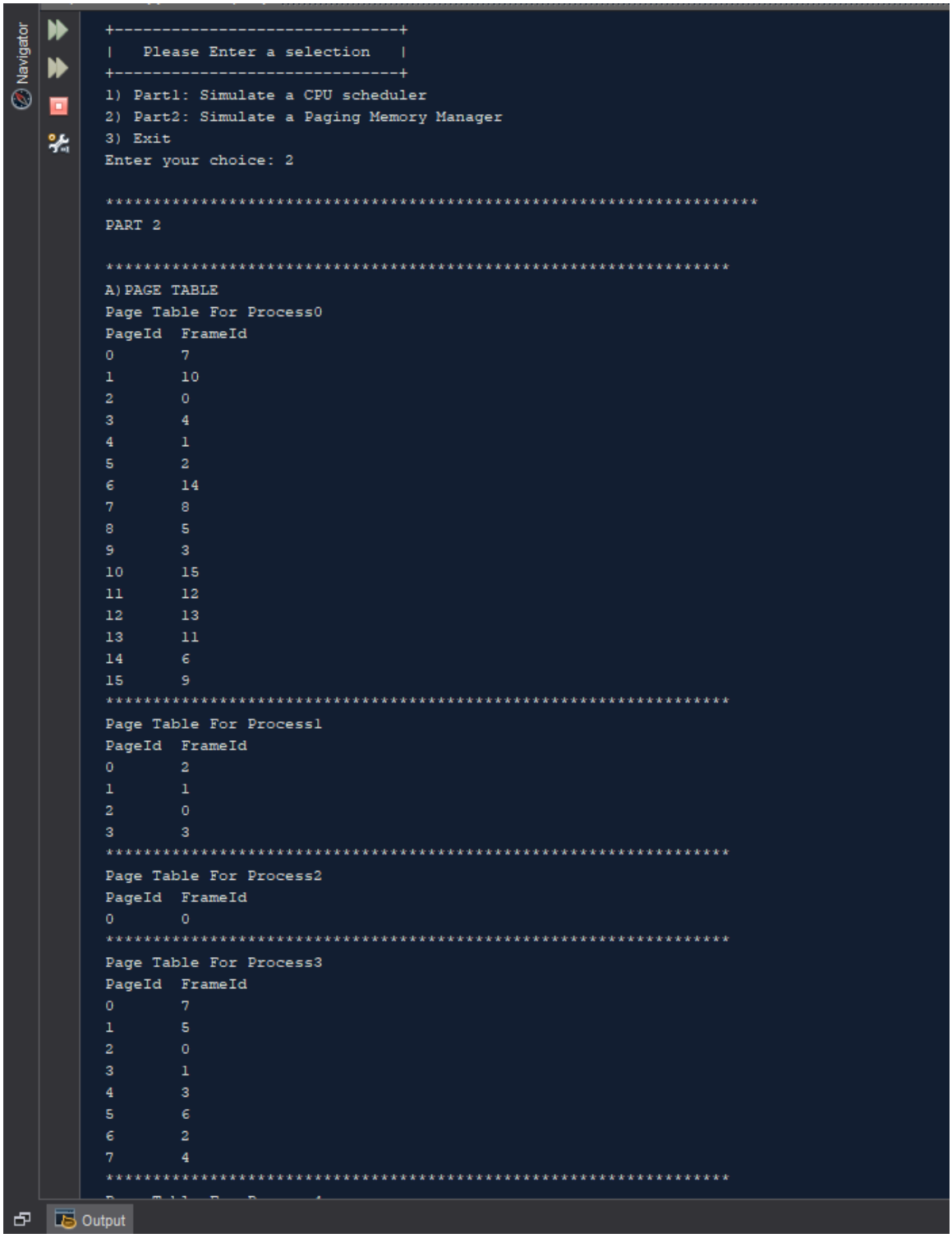
Page size = frame size

The output:


Navigator
run:
****
| mai Hashlamoon 191075 |
****
+----------+
| Please Enter a selection |
+----------+
1) Part1: Simulate a CPU scheduler
2) Part2: Simulate a Paging Memory Manager
3) Exit
Enter your choice: 1
PART 1
****
FCFS ALGORITHM
a)
gantt chart:
| 0 'P1' 12 || 13 'P2' 16 || 17 'P0' 27 || 28 'P3' 49 || 50 'P4' 57 |
b)
processID finish time turnaround time waiting time
1 12 12 0
2 16 15 12
0 27 24 14
3 49 44 23
4 57 48 41
c)
Average waiting time = 18.0
Average turn around time = 28.6
Average finish time= 32.2
d)
cpu utilization :92.982456140350885
****
SJF ALGORITHM
a)
gantt chart:
| 0 'P1' 12 || 13 'P2' 16 || 17 'P4' 24 || 25 'P0' 35 || 36 'P3' 57 |
b)
processID finish time turnaround time waiting time
1 12 12 0
2 16 15 12
4 24 15 8
0 35 32 22
3 57 52 31
c)
Average waiting time = 14.0
Average turn around time = 25.2
Average finish time= 28.8
d)
cpu utilization :92.982456140350885
****
Output

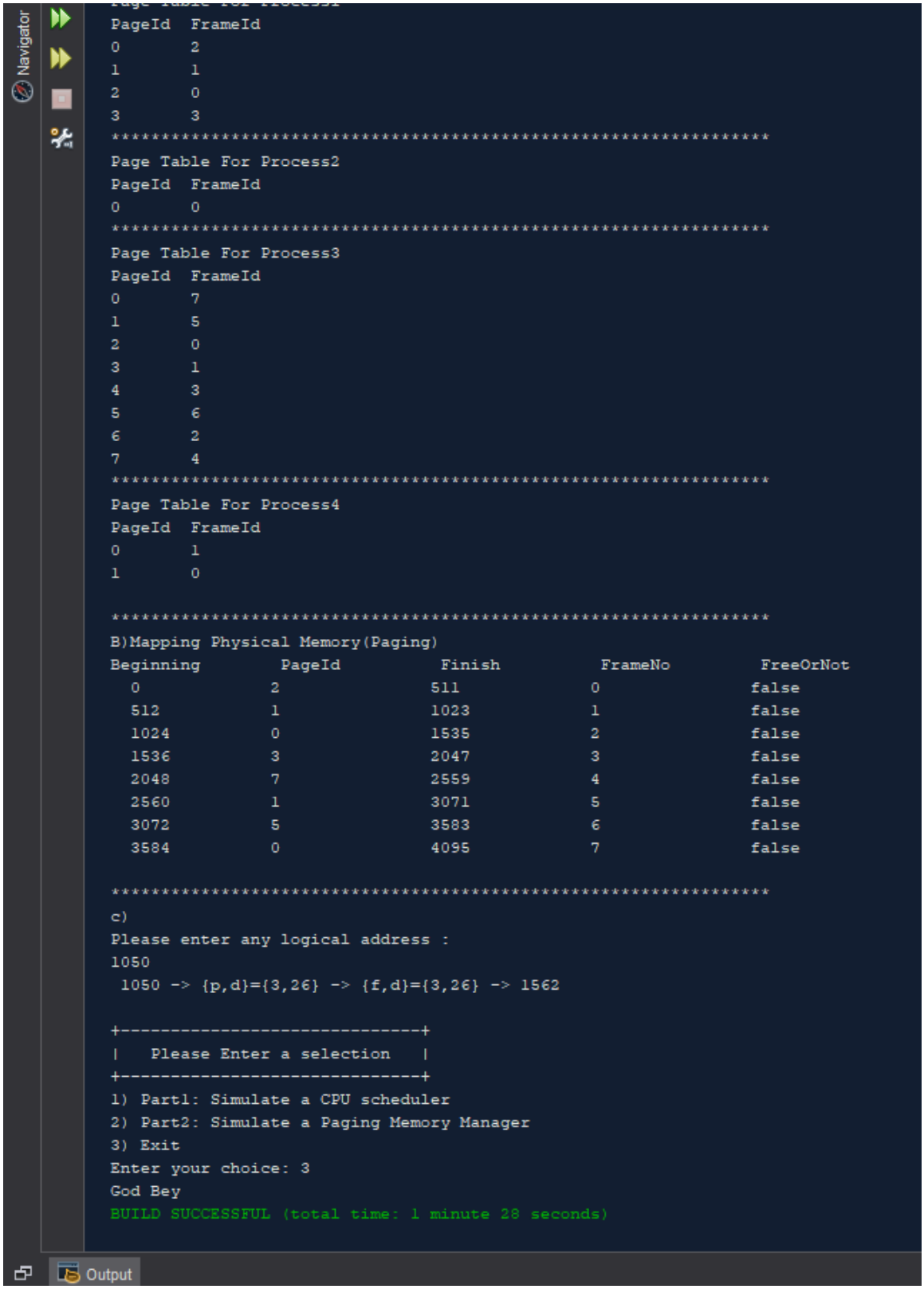

Navigator
Average waiting time = 18.0
Average turn around time = 28.6
Average finish time= 32.2
d)
cpu utilization :92.98245614035088%
****
 SJF ALGORITHM
a)
 gantt chart:

| 0 'P1' 12 || 13 'P2' 16 || 17 'P4' 24 || 25 'P0' 35 || 36 'P3' 57 |
----
b)
processID   finish time   turnaround time waiting time
1   12   12   0
2   16   15   12
4   24   15   8
0   35   32   22
3   57   52   31
c)
Average waiting time = 14.0
Average turn around time = 25.2
Average finish time= 28.8
d)
cpu utilization :92.98245614035088%
****
 ROUND ROBIN ALGORITHM
a)
 gantt chart
----
 0 1 2 3 4 1 3 3
----
b)
id finisht waitingt turnaroundt
 0   11   1   8
 1   51   39   51
 2   28   25   27
 3   59   38   54
 4   39   32   30
c)
Average waiting time = 27.0
Average turn around time = 34.0
Average finish time= 37.6
d)
cpu utilization :89.830508474576 28%
+------------------------------+
| Please Enter a selection |
+------------------------------+
1) Part1: Simulate a CPU scheduler
2) Part2: Simulate a Paging Memory Manager
3) Exit
Enter your choice:
Output


Navigator
Output
+------------------------------+
| Please Enter a selection |
+------------------------------+
1) Part1: Simulate a CPU scheduler
2) Part2: Simulate a Paging Memory Manager
3) Exit
Enter your choice: 2
****
PART 2
****
A) PAGE TABLE
Page Table For Process0
PageId FrameId
0   7
1   10
2   0
3   4
4   1
5   2
6   14
7   8
8   5
9   3
10   15
11   12
12   13
13   11
14   6
15   9
****
Page Table For Process1
PageId FrameId
0   2
1   1
2   0
3   3
****
Page Table For Process2
PageId FrameId
0   0
****
Page Table For Process3
PageId FrameId
0   7
1   5
2   0
3   1
4   3
5   6
6   2
7   4
****

```
6       14
7       8
8       5
9       3
10      15
11      12
12      13
13      11
14      6
15      9
**************************************************************************
Page Table For Process1
PageId  FrameId
0       2
1       1
2       0
3       3
**************************************************************************
Page Table For Process2
PageId  FrameId
0       0
**************************************************************************
Page Table For Process3
PageId  FrameId
0       7
1       5
2       0
3       1
4       3
5       6
6       2
7       4
**************************************************************************
Page Table For Process4
PageId  FrameId
0       1
1       0

**************************************************************************
B)Mapping Physical Memory(Paging)
Beginning        PageId          Finish          FrameNo         FreeOrNot
  0             2               511             0               false
  512           1               1023            1               false
  1024          0               1535            2               false
  1536          3               2047            3               false
  2048          7               2559            4               false
  2560          1               3071            5               false
  3072          5               3583            6               false
  3584          0               4095            7               false

**************************************************************************
c)
Please enter any logical address :
```

```
14        6
15        9
*************************************************************************
Page Table For Process1
PageId   FrameId
0        2
1        1
2        0
3        3
*************************************************************************
Page Table For Process2
PageId   FrameId
0        0
*************************************************************************
Page Table For Process3
PageId   FrameId
0        7
1        5
2        0
3        1
4        3
5        6
6        2
7        4
*************************************************************************
Page Table For Process4
PageId   FrameId
0        1
1        0

*************************************************************************
B)Mapping Physical Memory(Paging)
Beginning         PageId          Finish          FrameNo          FreeOrNot
  0               2               511             0                false
  512             1               1023            1                false
  1024            0               1535            2                false
  1536            3               2047            3                false
  2048            7               2559            4                false
  2560            1               3071            5                false
  3072            5               3583            6                false
  3584            0               4095            7                false

*************************************************************************
c)
Please enter any logical address :
1050
 1050 -> {p,d}={3,26} -> {f,d}={3,26} -> 1562

+------------------------------+
|   Please Enter a selection   |
+------------------------------+
1) Part1: Simulate a CPU scheduler
2) Part2: Simulate a Paging Memory Manager
3) Exit
Enter your choice:
```



```
PageId  FrameId
0       2
1       1
2       0
3       3
************************************************************************
Page Table For Process2
PageId  FrameId
0       0
************************************************************************
Page Table For Process3
PageId  FrameId
0       7
1       5
2       0
3       1
4       3
5       6
6       2
7       4
************************************************************************
Page Table For Process4
PageId  FrameId
0       1
1       0

************************************************************************
B)Mapping Physical Memory(Paging)
Beginning        PageId          Finish          FrameNo         FreeOrNot
  0              2               511             0               false
  512            1               1023            1               false
  1024           0               1535            2               false
  1536           3               2047            3               false
  2048           7               2559            4               false
  2560           1               3071            5               false
  3072           5               3583            6               false
  3584           0               4095            7               false

************************************************************************
c)
Please enter any logical address :
1050
 1050 -> {p,d}={3,26} -> {f,d}={3,26} -> 1562

+-----------------------------+
|   Please Enter a selection  |
+-----------------------------+
1) Part1: Simulate a CPU scheduler
2) Part2: Simulate a Paging Memory Manager
3) Exit
Enter your choice: 3
God Bey
BUILD SUCCESSFUL (total time: 1 minute 28 seconds)
```

This Project has been discussed with:

Mohanad fatafta

Malak naji

Marwa qafisheh

Yasmeen dawod

Amina khalawe