Version B

90.5

CSCE 5640.003: Operating System Design

Midterm Exam, October 24, Fall 2022

Name: Yasoda Devi Devarapalli

Euid: yd0151

This closed-book 120-minutes exam is worth 100 points. Please make sure that you understand each question before answering. If you feel there is any ambiguity in the question, please make and state a reasonable assumption to resolve the ambiguity.

Fill in the blank. (1 point each)

1. In many-to-one multithreading model, many user-level threads mapped to single kernel thread.
2. system calls refers to the special operation which allows a user program to request an operating system service.
3. Application Programming Interface (API) specifies a set of functions that are available to an application programmer, including the parameters that are passed to each function and the return values the programmer can expect.
4. In thread cancellation, cancellation points are associated with deferred cancellation.
5. Round Robin scheduling algorithm preempts the currently running process after its time quantum has expired.
6. A(n) dispatcher gives control of the CPU to the process selected by the scheduler.
7. Throughput is the number of processes that are completed per time unit.
8. A(n) Vizualization is a technology that can allow an OS to run on non-native hardware
9. A device controller informs CPU that it has finished its operation by causing a(n) Interrupt.
10. In a single-threaded process, a(n) Program counter specifies location of next instruction to execute.

True or False.

For each of the problems below, indicate whether it is true or false. Justify your answer in either case. (2 points each)

11. **Most modern operating systems combines different operating system structures or models.**

True. Most modern operating system combines differen OS structures or models for better performence, efficency, convinience. Depending on requirement we can switch between the differen Operating systems.

2

12. **Application Programming Interface (API) and System Calls are same thing.**

False. API is a set of functions that are available for application programmer. where as system calls are special programs that allow user program to request operating system services. So both are not same as API is function with input and output expected. and system calls are internal to system.

2

13. **Policy and Mechanism are the same thing.**

False. Policy determines what will be done. where as mechanizam determines how it is done. ~~and what need to be~~

2

14. **Shared memory is typically faster than message passing.** True.

Shared memory uses/requires one system call for establishing a sha memory segment. But message passing impliments in system calls an where kernal bail intervision. For message passing send or receive manage need system calls but for shared memory only required for estab

2

15. **Preemptive scheduling can result in race conditions when data are shared among several processes.** True

In this when ever a higher priority proces enter or arrives the prexnt proces in preempt for it. The data of that thread is ~~not~~ updated/stored for use. As if there are several proceses to run increase in preempling ~~raises~~ the race conditions

2

Version: B

Multiple Choice Questions. (1 point each)

16. Which of the following command is correct to generate main.out executable (using g++) from two object files myThread.o and main.o, where myThread.o contains some functions that uses POSIX Pthreads functions, such as pthread_create(), and main.o contains the main function of C++ program?
    a. g++ main.o myThread.o -lpthread
    b. g++ -o main.out main.o myThread.o
    c. g++ -o main.out main.o -lpthread
    d. g++ -o main.out main.o myThread.o -lpthread

17. Policy _____.
    a. determines how to do something
    b. determines what will be done
    c. is not likely to change over time

18. A _____ is another term for command interpreter.
    a. shell
    b. shell script
    c. gesture

19. A process control block _____.
    a. contains a process' states
    b. stores the executable code the process will run
    c. determines which process is to be run next

20. Which of the following forms of storage has the largest capacity?
    a. registers
    b. hard-disk drives
    c. nonvolatile memory

21. A stack uses the _____ principle for organizing data while a queue uses the _____ principle.
    a. FIFO, LIFO
    b. LIFO, FIFO
    c. left child, right child

22. Which of the following is not considered a form of a system service?
    a. Application programs
    b. Programming language support
    c. Background services

23. What best describes the return value from fork() for the child process?
    a. Value < 0
    b. Value of 0
    c. Value > 0

24. In POSIX Pthreads, what does the third parameter passed to the pthread_create() function specify?
    a. The parameter being passed to the thread being created.
    b. The thread identifier
    c. The name of the function that will be run by the thread being created.

25. What best describes what occurs when a child process calls the exec() system call?
    a. The parent must wait for the child to terminate.
    b. The child is a duplicate of the parent.
    c. The child process has a new program loaded onto it

26. What is the relationship between threads and processes?
    a. A process consists of one or more threads.
    b. A thread consists of one or more processes.
    c. A thread is a separate entity than a process.

27. Thread-local storage is _____.
    a. another term for local variables
    b. data that has been modified by a thread, but not yet updated to all other threads belonging to the same process
    c. data that is unique to each thread

28. A(n) _____ allows several unrelated processes to use the pipe for communication.
    a. anonymous pipe
    b. named pipe
    c. ordinary pipe

29. Which of the following belongs to the multicore programming challenges?
    a. Identifying tasks
    b. Balance
    c. Data splitting
    d. Data dependency
    e. Testing and debugging
    f. All of the other answers

30. The _____ multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads.
    a. many-to-one
    b. one-to-one
    c. many-to-many

31. In Pthreads and Windows threads any data declared _____ are shared by all threads belonging to the same process.
    a. locally
    b. globally
    c. using shared memory

32. Which of the following statements best describes the role of the dispatcher?
    a. The dispatcher is involved with making scheduling decisions.
    b. The dispatcher gives control of the CPU to the process selected by the scheduler.
    c. The dispatcher is not involved during a context switch.

33. Which of the following is true of multilevel queue scheduling?
    a. Processes can move among queues.
    b. Each queue is assigned a specific scheduling algorithm.
    c. Threads in one queue cannot have absolute priority over threads in lower-priority queues.

34. The two general approaches to load balancing are _____ and _____.
    a. coarse grained, fine grained
    b. soft real-time, hard real-time
    c. push migration, pull migration

35. Which of the following is true of nonpreemptive scheduling?
    a. Nonpreemptive scheduling requires a timer.
    b. A process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.
    c. Nonpreemptive scheduling incurs a cost associated with access to shared data.

Short Answer. Briefly define the following terms. **(5 points each)**

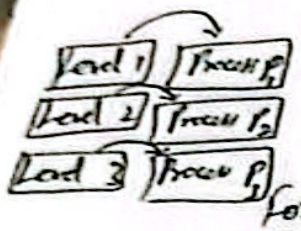
36. Message passing

The communication between the sender() and receiver() is known as message passing technique. The messages are sent to the CPU by shared bus memory with the help of operating system. The messages are sent through hardware I/O inputs.

37. Many-to-one multithreading model

The many-to-one multithreading model multiplexes many user-level threads to a one number of kernel threads

38. Multilevel queue scheduling



It is one of the CPU scheduling algorithms. This type of queue scheduling consists of multi-levels queue while the processes are in the queue. for an instance, if there are 3 levels ie, level 1, level 2, level 3, the process is pushed to P_1, P_2, P_3. All the processes are executed parallely in the system.

4

Listing.

39. List five (5) types of CPU scheduling algorithms. (5 points)

1) First Come First Serve Scheduling algorithm.
2) Shortest Job First scheduling algorithm
3) ~~Non-preemptive Scheduling algorithm.~~
3) Round Robin Scheduling algorithm.
4) ~~Multi level queue scheduling~~. Priority scheduling. Algorithm.
5) Non-preemptive Scheduling algorithm.

4

40. List five (5) different states a process may be in. (5 points)

1) New - The process has been created.
2) Running - Instructions are to be executed.
3) Waiting - I/O signals are standing in the queue to be processed.
4) Ready - All I/O signals are ready to be loaded into the queue.
5) Termination - The process is to be aborted or stopped.

5

13

Numerical/Code Answer.

41. **Using the program shown in the figure below, what will be the output at LINE X and LINE Y. (5 points)**

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#define SIZE 5
int nums[SIZE] = {1,2,3,4,5};
                  0 1 2 3 4
int main() {
   int i;
   pid t pid;
   pid = fork();
   if (pid == 0) {
        for (i = 1; i <= SIZE; i++) {
            nums[i-1] *= -i;
            printf("CHILD: %d ",nums[i-1]); /* LINE X */
        }
   }
   else if (pid > 0) {
        wait(NULL); Parent waits unlill child terminates.
        for (i = 0; i < SIZE; i++)
            printf("PARENT: %d ",nums[i]); /* LINE Y */
   }
   return 0;
} /* main */
```

2.5

CHILD: -1✓ → PARENT: -1✗
CHILD: -4✓ → PARENT: -4✗
CHILD: -9✓ → PARENT: -9✗
CHILD: -16✓ → PARENT: -16✗
CHILD: -25✓ → PARENT: -25✗

nums is a global variable ~~changes~~ and parent process is waiting for the child process to complete execution. The changes done by the child are reflected in parent as well

2.5

Version B

42. Consider the following set of processes, with the length of the CPU burst time given in milliseconds: **(30 points)**

Process	Burst Time	Priority
P1	10	8
P2	20	3
P3	30	4
P4	25	4
P5	5	5

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

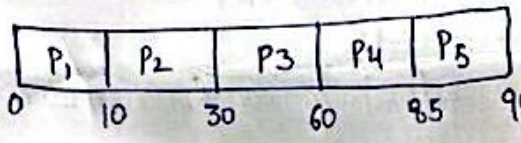
a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2). **(20 points)**
quantum = 10

b. Which of the algorithms results in the minimum average waiting time (over all processes)? **(10 points)**

(a) arrival time is 0.

FCFS:- First come First serve:-

Gantt Chart



20

wait times :- From ganttchart

$P_1 \rightarrow 0$

$P_2 \rightarrow 10$

$P_3 \rightarrow 30$

$P_4 \rightarrow 60$

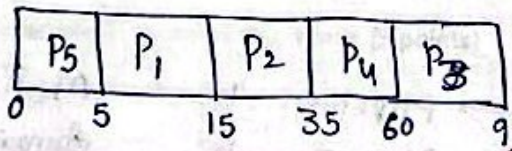
$P_5 \rightarrow 85$

Avgerage waiting time $= \frac{0+10+30+60+85}{5}$

$= 185/5 = 37$ milliseconds

Shortest Job First :-

Gantt Chart



wait times from Chart :-

$P_1 \rightarrow 5$ $P_2 \rightarrow 15$

$P_3 \rightarrow 60$ $P_4 \rightarrow 35$

$P_5 \rightarrow 0$

Avg. waiting time $= \frac{5+15+60+35+0}{5} = \frac{115}{5}$

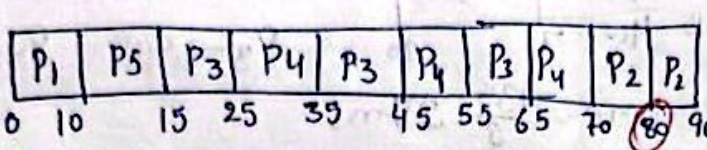
Avg. waiting time = 23 ms

(20)

Round Robin:- quantum = 10 with (non preemptive priority)

Priority order from high to low is P1, P5, P3, P4, P2

Gantt Chart



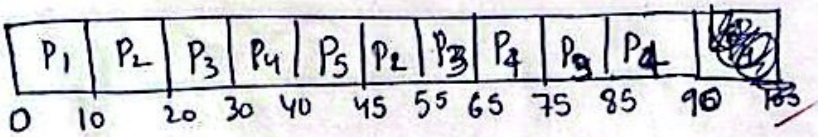
waiting time :- $P_1 \rightarrow 0$ $P_2 \rightarrow 70$ $P_3 \rightarrow (55-20) = 35$

$P_4 \rightarrow (65-20) = 45$ $P_5 \rightarrow 10$

Avg. wait time = $\frac{0+70+35+45+10}{5} = \frac{160}{5} = 32$ ms

Round Robin:- quantum = 10 without priority

Gantt Chart



waiting time:- $P_1 \rightarrow 0$ $P_2 \rightarrow (45-10) = 35$

$P_3 \rightarrow (75-20) = 55$ $P_4 \rightarrow (85-20) = 65$

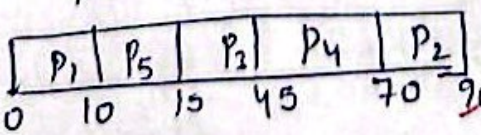
$P_5 \rightarrow 40$

Avg. waiting time = $\frac{0+35+55+65+40}{5} = \frac{195}{5} = 39$ ms

(b) The shortest job first algorithm result in minimum avg. time then FCFS, RR, non preemptive priority i.e 23ms.

(a) (a) non preemptive priority with P_3 first

Gantt Chart



wait time:-

$P_1 \rightarrow 10$ $P_2 \rightarrow 70$ $P_3 \rightarrow 15$

$P_4 \rightarrow 45$ $P_5 \rightarrow 10$

Avg. wait time = $\frac{150}{5} = 30$ ms

Version: B

88

CSCE 5640.001: Operating System Design

Midterm Exam, October 22, Fall 2022

Name: Himaja Kutikuppala

Euid: 11600609

This closed-book 120-minutes exam is worth 100 points. Please make sure that you understand each question before answering. If you feel there is any ambiguity in the question, please make and state a reasonable assumption to resolve the ambiguity.

Fill in the blank. (1 point each)

1. A(n) linker file combines relocatable object files into a single executable file.
2. Throughput is the number of processes that are completed per time unit.
3. Virtual Machine is a technology that abstracts the hardware of an actual computer into several different execution environments.
4. A(n) process is an instance of a program in execution.
5. A(n) Bootstrap program starts the operating system.
6. Application Binary Interface is architecture equivalent of API.
7. In message passing technique for IPC, a blocking send() and blocking receive() is known as a(n) Rendezvous.
8. In Priority. scheduling, the next CPU burst is predicted with an exponential average of the measured lengths of previous CPU bursts.
9. The Many-to-many multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads.
10. In thread cancellation, cancellation points are associated with deferred. cancellation.

True or False.

For each of the problems below, indicate whether it is true or false. Justify your answer in either case. (2 points each)

11. **Privileged instructions can be executed in both user mode and kernel mode.**

False.

All instructions can't be executed in both user mode and kernel mode because the programs loaded into the queues.

12. **Concurrency and Parallelism is the same thing.**

False.

Both Concurrency and Parallelism are two different things. As, the process can parallelly be executed simultaneously in parallelism., where in terms of Concurrency, it only concentrate on single program to be executed.

13. **In process scheduling, there is one Ready queue and one Wait queue only.**

True.

While the process is in running state, it is ready to load I/O signals into queue. where the instructions are executed. Whereas, in wait queue all the I/O signals are wait for the signal or can be terminated based on the program.

14. **Shared memory is typically faster than message passing.**

True.

shared memory has the highest power while the memory is stored in RAM (Read & Random Access Memory).

15. **Shortest-Job-First (SJF) is a type of a priority scheduling algorithm.**

True. False.

(As, it prioritizes the job and the burst time is calculated as per the priority number scheduled)

The processors are called into the queue based on the burst time.

Multiple Choice Questions. (1 point each)

16. A shell is another term for command interpreter.
    (a) shell
    b. shell script
    c. gesture

17. What statement concerning privileged instructions is considered false? C
    a. Unless used appropriately, privileged instructions may cause harm to the system.
    b. Privileged instructions can only be executed in kernel mode.
    (c) Privileged instructions can be executed in both user mode and kernel mode.

18. Which of the following is true of clustered systems? a
    (a) They can provide high-availability service.
    b. Because they are typically connected across a computer network, they cannot meet high-performance computing needs.
    c. Data sharing is not possible on clustered systems as the cluster consists of separate computer systems

19. A bootstrap program a.
    (a) starts the operating system
    b. consists of the entire operating system
    c. cannot be used for diagnosing system issues

20. Which of the following statements concerning open-source operating systems is true? a
    (a) Source code is made available.
    b. Open-source operating systems are always more secure than commercial, closed systems.
    c. All open-source operating systems share the same set of goals.

21. Which of the following operating system services is related to terminating a program? b
    a. I/O operations
    (b) Program execution
    c. Communications
    d. Error detection

22. Which of the following command is correct to generate main.out executable (using g++) from two object files myThread.o and main.o, where myThread.o contains some functions that uses POSIX Pthreads functions, such as pthread_create(), and main.o contains the main function of C++ program? a
    (a) g++ -o main.out main.o myThread.o -lpthread
    b. g++ main.o myThread.o -lpthread
    c. g++ -o main.out main.o myThread.o
    d. g++ -o main.out main.o -lpthread

23. Which of the following is not a technique that allows a program to be run by different operating systems? C
    a. The program is written in an interpreted language such as Python.
    b. The program is written in a language that includes a virtual machine as part of the application's runtime such as Java.
    c. The program has been compiled into a binary executable file.

24. The convention in modern operating systems and compilers is that the d of a process contains temporary data such as function parameters, return addresses, and local variables.
    a. Text section
    b. Data section
    c. Heap
    d. Stack

25. A process may transition to the Ready state by which of the following actions? b
    a. While the process is waiting for completion of an I/O event.
    b. The process timed out.
    c. The process is dispatched by the scheduler to run on a CPU core.

26. C saves the state of the currently running process and restores the state of the next process to run.
    a. The CPU scheduler
    b. Swapping
    c. A context switch

27. A loadable kernel model (LKM) is a kernel a.
    a. containing many a set of core components and load additional service in kernel mode either in boot time or run time.
    b. containing many a set of core components and load additional service in user mode either in boot time or run time.
    c. has very-limited or no structure at all.

28. A UNIX/Linux process that calls exit() to terminate, but whose parent has not yet called wait(), is known as a(n) a process.
    a. zombie
    b. orphan
    c. terminated

29. Message passing with direct communication requires b.
    a. either the send() message or receive() message functions to specify a process as a parameter to the function.
    b. both the send() message and receive() message functions to specify a process as a parameter to the function.
    c. a separate mailbox.

Version B

30. Grand Central Dispatch handles blocks by a
   a. placing them on a dispatch queue.
   b. creating a new thread.
   c. placing them on a dispatch stack.

31. In C scheduling, the next CPU burst is predicted with an exponential average of the measured lengths of previous CPU bursts.
   a. Multilevel queue
   b. FCFS
   c. SJF

32. In Little's formula, λ, represents the a.
   a. average waiting time in the queue
   b. average arrival rate for new processes in the queue
   c. average queue length

33. A(n) b allows several unrelated processes to use the pipe for communication.
   a. anonymous pipe
   b. named pipe
   c. ordinary pipe

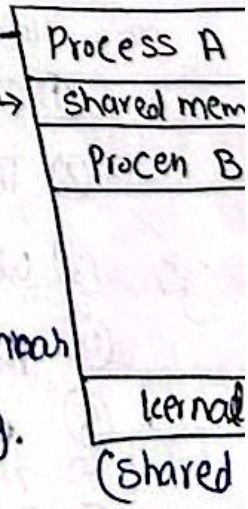
34. Which of the following is not shared by threads? b
   a. Code
   b. Stack
   c. Data

35. Which of the following scheduling algorithms could result in starvation? d
   a. First-come, first-served
   b. Shortest job first
   c. Round robin
   d. Priority

Short Answer.

Briefly define the following terms. **(5 points each)**

36. Shared memory : In this, the process share a segment of memory for data or information exchange.
→ It require kernal only to establish the shared memory segment.
→ It is faster than menage paning because of less kernel intervition.
→ The sharing is done in producer and consumer manner with bounded or unbounded sending and receiving.



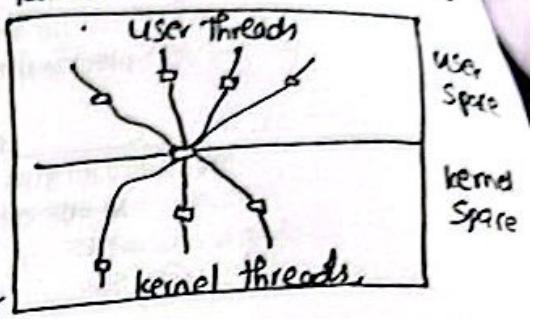
Version B

37. Many-to-many multithreading model

→ Many user threads are mapped to many kernel threads. i.e many user threads are assigned to equal or less number of kernel threads

→ A blocking system call thread will not effect the system and can be assigned to another

→ Operating system has sufficient kernel threads for execution. More useful in multi core processor system.



38. Multilevel Feedback Queue Scheduling

The threads are switching/running among queues

The parameters for this are :- ① number of ques
② scheduling alogorithm for each queue
③ scheduling among the squeue.

with priority scheduling, each priority has seperate queues.
scheduling done higher to low priorities

Listing.

39. List five (5) different parts/sections of a process in memory. (5 points)

Process status
Program Counter
Process Id
Stack Information
resource allocation infonation (Register information)
Instructions of process

40. List five (5) types of CPU scheduling criteria. (5 points)

① CPU utilization
② Throughput
③ waiting time
④ Turnaround time
⑤ Response time

For good scheduling max untilization and throughput and min wait, Turnaround and response time.

Version: B

Numerical/Code Questions.

41. **The program shown in the figure below uses the Pthreads API. What would be the output from the program at LINE A and LINE B? Justify your answer briefly. (5 points)**

```
#include <pthread.h>
#include <stdio.h>
#include <unistd.h>
int value = 5;
void *runner(void *param) {  /* the thread */
   value = value * 5;
   pthread_exit(0);
}
int main(int argc, char *argv[]) {
   pid_t pid;
   pthread_t tid;
   pthread_attr_t attr;

   pid = fork();
   if (pid == 0) { /* child process */
      pthread_attr_init(&attr);
      pthread_create(&tid, &attr, runner, NULL);
      pthread_join(tid, NULL);
      printf("CHILD: value = %d",value); /* LINE A */
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE B */
   }
} /* main */
```

Output :

Line A : 10

The child node have a value of 10 because, the child node has a thread joining, when an argument is passed to run a function, it returns the value of 10.

Line B : The parent node has a value of 5 because, the value variable is initialized globally & thread joining will not be called globally.

Version: B

42. **Consider the following set of processes, with the length of the CPU burst time given in milliseconds: (30 points)**

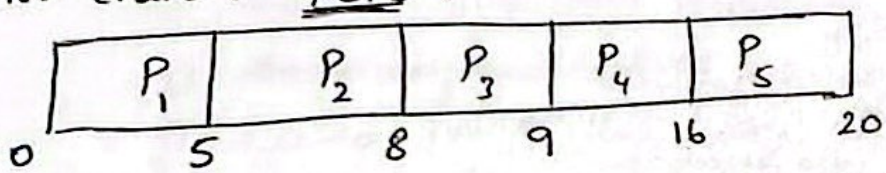
Process	Burst Time	Priority
P1	5	4
P2	3	1
P3	1	2
P4	7	2
P5	4	3

The processes are assumed to have arrived in the order *P1*, *P2*, *P3*, *P4*, *P5*, all at time 0.

a. **Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2). (20 points)**

b. **Which of the algorithms results in the minimum average waiting time (over all processes)? (10 points)**

a) Gantt chart : FCFS :

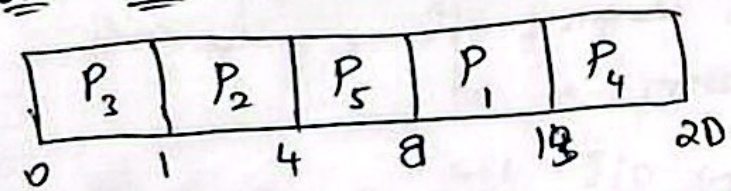


Average waiting time = $\frac{P_1 + P_2 + P_3 + P_4 + P_5}{5}$

$= \frac{0+5+8+9+16}{5} = \frac{38}{5}$

$= 7.6$ m/s

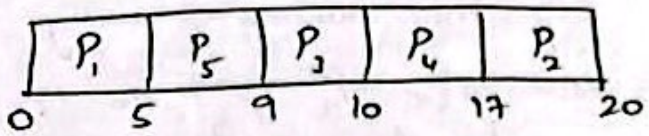
Shortest Job First :



Average waiting time = $\frac{0+1+4+8+13}{5}$

$= \frac{26}{5} = 5.2$ m/s

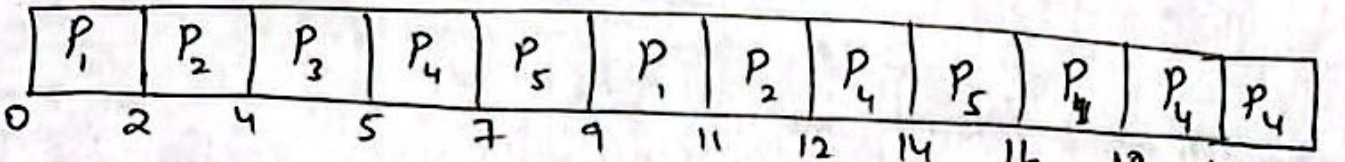
Non-preemptive priority:



Average waiting time : $\frac{0+5+9+10+17}{5}$

$= \frac{41}{5} = 8.2$ ms

Round Robin :



Average waiting time = $\frac{6+9+4+13+12}{5}$

$= \frac{44}{5} = 8.8$ ms

Process	Exit time	Wait time
P_1	11	6
P_2	12	9
P_3	5	4
P_4	20	13
P_5	16	12