Name: Ngô Hồng Quốc Bảo ID: B1809677

OPERATING SYSTEMS (CT104H)

ASSIGNMENT # 2

Due date: April 18

Please use English to answer the following questions.

Declaration of own work

I, ____*Ngô Hồng Quốc Bảo*___, certify that this assignment is my own work, is not copied from any other person's work.

VIRTUAL MACHINE (Chapter 16 – OS concepts 9th Edition)

1. What is a virtual machine (VM)?
2. What are the advantages and disadvantages of VM?
3. What are the types of VM and their implementation ?
4. What is the architecture of VMs?

CHAPTER 2: OS STRUCTURES

5. What are services provided by an OS?
6. What is the difference between “protection” and “security”
7. What are the types of user OS interface?
8. What is the purpose of system calls?
9. Describe methods for passing parameters to the OS
10. What are the advantages of the layered approach to system design? What are the disadvantages of the layered approach?

CHAPTER 4: CPU SCHEDULING

11. Use the **Round Robin (quantum = 2)** scheduling algorithm to answer questions:

- Draw Gantt charts that illustrate the execution of these processes

- What is the waiting time of each process?
- What is the response time of each process?
- What is the turnaround time of each process?

12. Use the **Priority preemtpive** scheduling algorithm to answer questions:

- Draw Gantt charts that illustrate the execution of these processes
- What is the waiting time of each process?
- What is the response time of each process?
- What is the turnaround time of each process?

13. Use the **FCFS, SJF non-preemptive, SJF preemptive, Priority non-preemptive, Priority preemptive, Round Robin (quantum = 3)** scheduling algorithms to answer questions:

- Draw Gantt charts that illustrate the execution of these processes
- What is the waiting time of each process for each of the scheduling algorithms? What is the average waiting time for each of the scheduling algorithms?
- What is the response time of each process for each of the scheduling algorithms? What is the average response time for each of the scheduling algorithms?

- What is the turnaround time of each process for each of the scheduling algorithms?

  What is the average turnaround time for each of the scheduling algorithms?

Process	CPU time	Arrival time	Priority
P1	5	5	2
P2	6	4	3
P3	7	3	1
P4	9	1	6
P5	2	2	5
P6	3	6	4

Task

1. A virtual machine is guest operating systems and applications run in an environment that appears to them to be native hardware and that behaves toward them as native hardware would but that also protects,manages, and limits them.

2.

a. Advantage:

- The host system is protected from the virtual machines, just as the virtual machines are protected from each other. A virus inside a guest operating system might damage that operating system but is unlikely to affect the host or the other guests.
- The ability to freeze, or suspend, a running virtual machine.
- A virtual machine system is a perfect vehicle for operating-system research and development. Operating systems are large and complex programs, and a change in one part may cause obscure bugs to appear in some other part. Therefore, changing an operating system is a difficult task. A virtual-machine system can eliminate much of this latter problem.

- Another advantage of virtual machines for developers is that multiple operating systems can run concurrently on the developer’s workstation. This virtualized workstation allows for rapid porting and testing of programs in varying environments.
- A major advantage of virtual machines in production data-center use is system consolidation, which involves taking two or more separate systems and running them in virtual machines on one system. Such physical-to-virtual conversions result in resource optimization, since many lightly used systems can be combined to create one more heavily used system.

b. Disadvantage:

- It can prevent resources due to isolation from the host system.
- Virtual machines are less efficient than real machines because they access the hardware indirectly. Running software on top of the host operating system means that it will have to request access to the hardware from the host. That will slow the usability.
- A virtual machine can be infected with the weaknesses of the host machine. As an example, process isolation is a feature usually employed by operating systems. However, there are bugs that violate it. A regular computer devoid of virtual machines would then only be affected. But, a computer with a number of virtual machines would then infect each of those “machines” as well.

3. Types of VM and their implementation:

a. **The virtual machine life cycle:** Whatever the hypervisor type, at the time a virtual machine is created, its creator gives the VMM certain parameters. These parameters usually include the number of CPUs, amount of memory, networking details, and storage details that the VMM will take into account when creating the guest.

b. **Type 0 Hypervisor:** Operating systems need do nothing special to take advantage of their features. The VMM itself is encoded in the firmware and loaded at boot time. In turn, it loads the guest images to run in each partition. The feature set of a type 0 hypervisor tends to be smaller than those of the other types because it is implemented in hardware.

c. **Type 1 Hypervisor:** Type 1 hypervisors run in kernel mode, taking advantage of hardware protection. Where the host CPU allows, they use multiple modes to give guest operating systems their own control and improved performance. They implement device drivers for the hardware they run on, because no other component could do so. Because they are operating systems, they must also provide CPU scheduling, memory management, I/O management, protection, and even security. Frequently, they provide APIs, but those APIs support applications in guests or external applications that supply features like backups, monitoring, and security.

d. **Type 2 Hypervisor:** This type of VMM is simply another process run and managed by the host, and even the host does not know virtualization is happening within the VMM.

e. **Paravirtualization:** paravirtualization takes a different tack than the other types of virtualization. Rather than try to trick a guest operating system into believing it has a system to itself, paravirtualization presents the guest with a system that is similar but not identical to the guest's preferred system. The guest must be modified to run on the paravirtualized virtual hardware. The gain for this extra work is more efficient use of resources and a smaller virtualization layer .

f. **Programming-Environment Virtualization:** another kind of virtualization, based on a different execution model, is the virtualization of programming environments. A programming language is designed to run within a custom-built virtualized environment.

g. **Emulation**: is useful when the host system has one system architecture and the guest system was compiled for a different architecture.

h. **Application Containment:** is implemented for the applications that are all compiled for the same operating system (therefore, we do not need a complete virtualization).

4. A virtual machine can support individual processes or a complete system depending on the abstraction level where virtualization occurs. Some VMs support flexible hardware usage and software isolation, while others translate from one instruction set to another. Virtualizing a system or component -such as a processor, memory, or an I/O device - at a given abstraction level maps its interface and visible resources onto the interface and resources of an underlying, possibly different, real system. Consequently, the real system appears as a different virtual system or even as multiple virtual systems. Interjecting virtualizing software between abstraction layers near the HW/SW interface forms a virtual machine that allows otherwise incompatible subsystems to work together. Further, replication by virtualization enables more flexible and efficient and efficient use of hardware resources.

5. Services provide by an OS:
    - User Interface.
    - Program execution.
    - I/O operations.
    - File-system manipulation.
    - Communications.
    - Error detection.
    - Resource allocation.
    - Accounting.
    - Protection and Security.

6. Differences between “protection” and “security”:

	Protection	**Security**
Definition	A method used in operating systems that manages threats within the system to maintain the proper functioning of the system.	A method used in operating systems that handles the threats from outside of the system to maintain the proper functioning of the system.
Main Focus	Focuses on internal threats of the system.	Focuses on external threats to the system.
Functionality	Provides a mechanism for controlling the access to programs, processes, and user resources.	Provides a mechanism to safeguard the system resources and user resources from external users.
Policy	Protection policy specifies whether a user can access a specific resource. The owner of the resource performs this function when creating it.	Security policy specifies whether a person can become a user of the system. It is performed by the system administrator.
Mechanism	Protection involves mechanisms such as setting or changing protection information of a resource and checking whether that resource is accessible by a user.	Security involves mechanisms such as adding, deleting users, verifying whether a specific user is authorized, using anti-malware software, etc.

7. Types of user interface:

- Command interpreters.
- Graphical user interfaces.
- Choice of interface.

8. System call provide an interface to the services made available by an operating system.

9. Three general methods are used to pass parameters to the operating system.

- Passing the parameters in registers.
- When there are more parameters than registers, parameters can be stored in a block and the block address can be passed as a parameter to a register.
- Parameters can also be pushed on or popped off the stack by the operating system.

10. What are the advantages of the layered approach to system design? What are the disadvantages of the layered approach?

❖ Advantage: The main advantage of the layered approach is simplicity of construction and debugging. The layers are selected so that each uses functions (operations) and services of only lower-level layers. This approach simplifies debugging and system verification.

❖ Disadvantage:

➢ The major difficulty with the layered approach involves appropriately defining the various layers because a layer can use only lower-level layers.

➢ They tend to be less efficient than other types.

11. Round Robin (quantum = 2) scheduling algorithm:

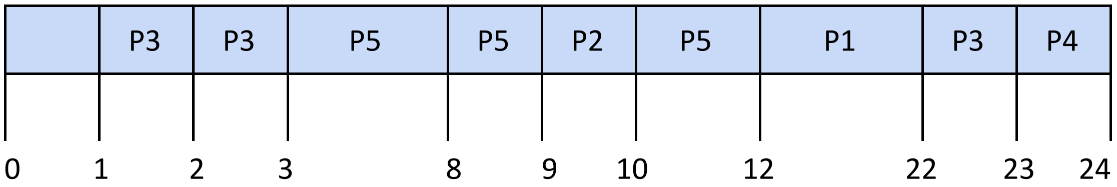
	P3	P4	P5	P3	P5	P1	P2	P5	P1	P5	P1	P1	P1

0 1 3 4 6 7 9 11 12 14 16 18 20 22 24

Process	**Waiting Time**	**Response Time**	**Turnaround Time**

P1	(9 - 8)+(14 - 11)+(18 - 16)=6	9 - 8 = 1	24 - 8 = 16
P2	11 - 9 = 2	11 - 9 = 2	12 - 9 = 3
P3	(1 - 1)+(6 - 3) = 3	1 - 1 = 0	7 - 1 = 6
P4	3 - 2 = 1	3 - 2 = 1	4 - 2 = 2
P5	(4 - 3)+(7 - 6)+(12 - 9)+(16 - 14) = 7	4 - 3 = 1	18 - 3 = 15

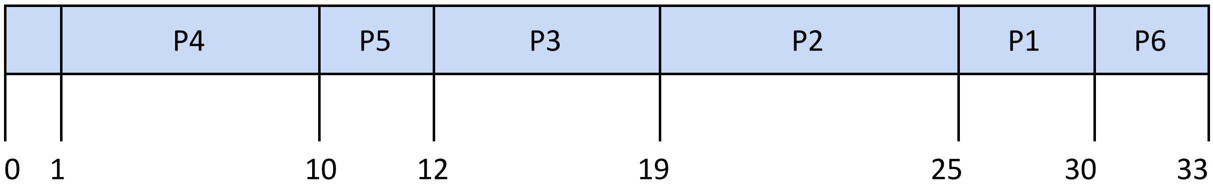
12. Priority preemptive scheduling algorithm:



Process	Waiting Time	Response Time	Turnaround Time
P1	12 - 8 = 4	12- 8 = 4	22 - 8 = 14
P2	9 - 9 = 0	9 - 9 = 0	10 - 9 = 1
P3	(1 - 1) + (22 - 3) = 19	1 - 1 = 0	23 - 1 = 22
P4	23 - 2 = 21	23 - 2 = 21	24 - 2 = 22
P5	(3 - 3) + (10 - 9) = 1	3 - 3 = 0	12 - 3 = 9

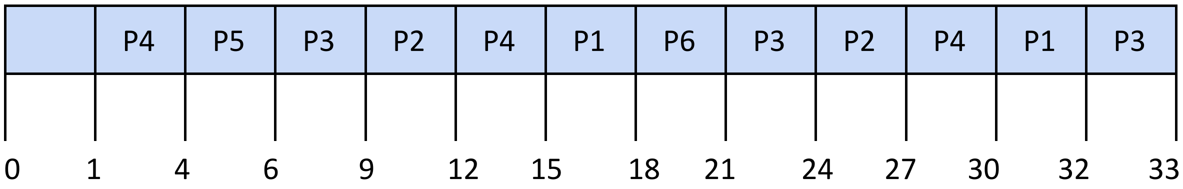
13.

a. **FCFS** scheduling algorithm:



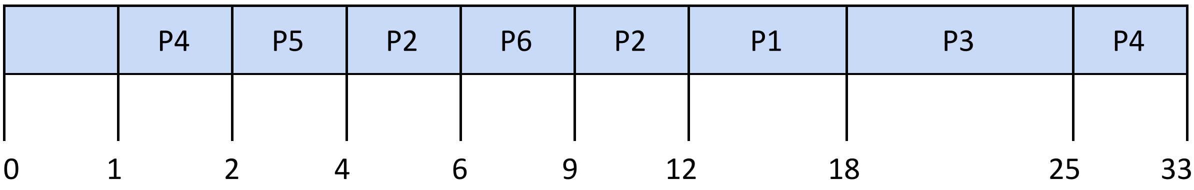
Process	Waiting Time	Response Time	Turnaround Time
P1	25 - 5 = 30	25 - 5 = 30	30 - 5 = 25
P2	19 - 4 = 15	19 - 4 = 15	25 - 4 = 21
P3	12 - 3 = 9	12 - 3 = 9	19 - 3 = 16
P4	1 - 1 = 0	1 - 1 = 0	10 - 1 = 9
P5	10 - 2 = 8	10 - 2 = 8	12 - 2 = 10
P6	30 - 6 = 24	30 - 6 = 24	33 - 6 = 27
Average	12.67	12.67	18

b. **Round Robin (quantum = 3)** scheduling algorithm:



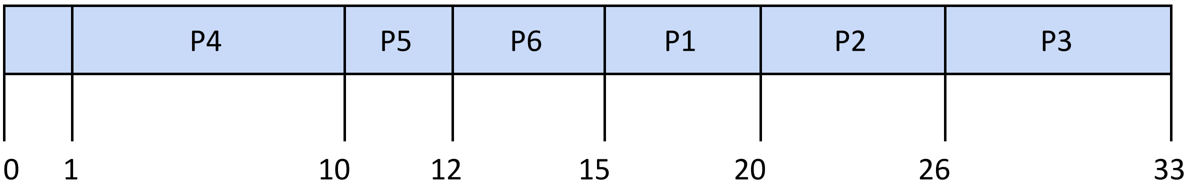
Process	Waiting Time	Response Time	Turnaround Time
P1	(15 - 5)+(30 - 18) = 22	15 - 5 = 10	32 - 5 = 27
P2	(9 - 4)+(24 - 12) = 17	9 - 4 = 5	27 - 4 = 23
P3	(6 - 3)+(21 - 9)+(32 - 24) = 23	6 - 3 = 3	33 - 3 = 30
P4	(1 - 1)+(12 - 4)+(27 - 15) = 20	1 - 1 = 0	30 - 1 = 29
P5	4 - 2 = 2	4 - 2 = 2	6 - 2 = 4
P6	18 - 6 = 12	18 - 6 = 12	21 - 6 = 15
Average	16	5.33	21.33

c. **SJF preemptive** scheduling algorithm:



Process	Waiting Time	Response Time	Turnaround Time
P1	13 - 5 = 18	13 - 5 = 18	18 - 5 = 13
P2	(4 - 4) + (9 - 6) = 3	4 - 4 = 0	13 - 4 = 9
P3	18 - 3 = 15	18 - 3 = 15	25 - 3 = 22
P4	(1 - 1) + (25 - 2) = 23	1 - 1 = 0	33 - 1 = 32
P5	2 - 2 = 8	2 - 2 = 8	4 - 2 = 2
P6	6 - 6 = 0	6 - 6 = 0	9 - 6 = 3
Average	9.5	3.83	13.5

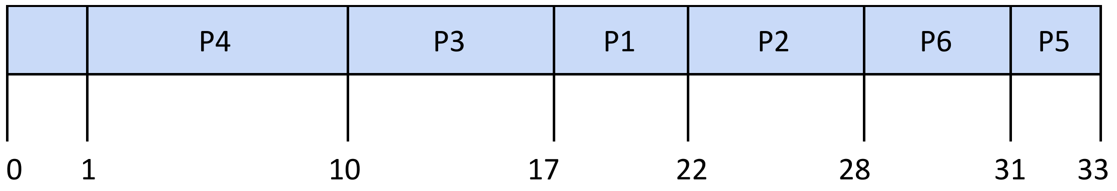
d. **SJF non-preemptive** scheduling algorithm:



Process	Waiting Time	Response Time	Turnaround Time
P1	15 - 5 = 10	15 - 5 = 10	20 - 5 = 15

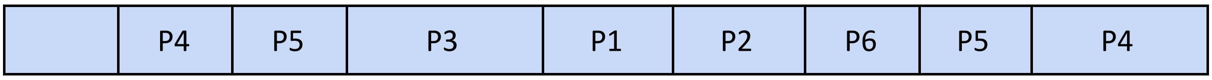
P2	20 - 4 = 16	20 - 4 = 16	26 - 4 = 22
P3	26 - 3 = 23	26 - 3 = 23	33 - 3 = 30
P4	1 - 1 = 0	1 - 1 = 0	10 - 1 = 9
P5	10 - 2 = 8	10 - 2 = 8	12 - 2= 10
P6	12 - 6 = 6	12 - 6 = 6	15 - 6 = 9
Average	10.6	10.6	15.82

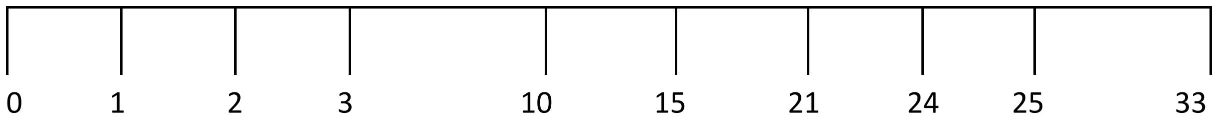
e. **Priority non-preemptive** scheduling algorithm:



Process	Waiting Time	Response Time	Turnaround Time
P1	17 - 5 = 12	17 - 5 = 12	22 - 5 = 17
P2	22 - 4 = 18	22 - 4 = 18	28 - 4 = 24
P3	10 - 3 = 7	10 - 3 = 7	17 - 3 = 14
P4	1 - 1 = 0	1 - 1 = 0	10 - 1 = 9
P5	31 - 2 = 29	31 - 2 = 29	33 - 2 = 31
P6	28 - 6 = 22	28 - 6 = 22	31 - 6 = 25
Average	14.67	14.67	20

f. **Priority preemptive** scheduling algorithm:





Process	Waiting Time	Response Time	Turnaround Time
P1	10 - 5 = 5	10 - 5 = 5	15 - 5 =10
P2	15 - 4 = 11	15 - 4 = 11	21 - 4 = 17
P3	3 - 3 = 0	3 - 3 = 0	10 - 3 = 7
P4	(1 - 1) + (25 - 2) = 23	1 - 1 = 0	33 - 1 = 32
P5	(2 - 2) + (24 - 3) = 21	2 - 2 = 0	25 - 2 = 23
P6	21 - 6 = 15	21 - 6 = 15	24 - 6 = 18
Average	12.5	5.17	17.83