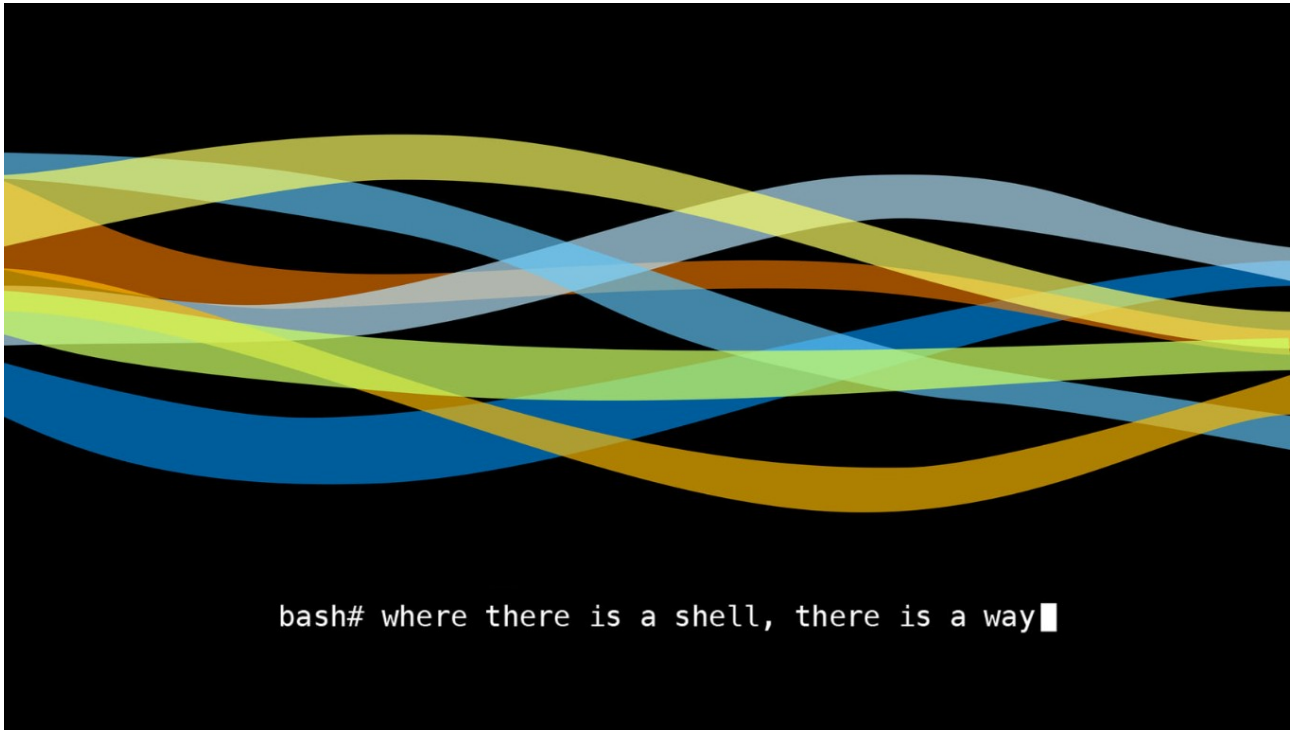
Operating Systems

Fall 2019 Solution



Saad Bin Khalid 20K-0161 7 March 2022

Q1: Answer the following Questions: [Marks=30]

i. What is the purpose of virtualization in operating systems?

Virtualization is a technology that allows us to run several different execution environments. These environments can be viewed as different individual operating systems, running on their own private computers. This technology allows the user to install multiple operating systems for exploration, testing, or to run non-native applications.

ii. What are system calls? Give at least two different examples of system call.

System calls are functions that provide an interface to the services made available by an operating system. They execute privileged code on behalf of the user. Following are the examples of system calls on a UNIX system:

- File management : read() write() close()
- Process management: fork() wait() abort()

iii. What are the advantages and disadvantages of using the microkernel approach of design of operating systems?

This approach structures the operating system by removing all nonessential components from the kernel and implementing them as user-level programs. The advantages are:

- Small size.
- Easier extendibility and portability.

The disadvantages are:

- Difficult inter-process communication because of message passing.
- Performance overhead because of increased intervention of kernel.

iv. What are CPU bound and I/O bound processes?

An I/O-bound process spends more of its time doing I/O than it spends doing computations. In contrast, a CPU-bound process generates I/O requests infrequently, using more of its time doing computations.

v. When does a process move from Running to Waiting state and from Running to Ready state?

A process switches from the running state to the waiting state as a result of some I/O request or some other events like an invocation of wait() for the of a child process.

A process switches from the running state to the ready state when an interrupt occurs or the time slice gets expired.

vi. What operations are performed during context switching of a process?

Context Switching requires performing a state save of the current process to its PCB and a state restore of a different process from its PCB.

vii. Differentiate between long term and short term schedulers?

Long term schedulers decide which processes to move from memory to ready queue. Whereas, short term schedulers are responsible form distributing CPU cycles to processes one by one.

viii. Differentiate between preemptive and non-preemptive scheduling algorithms?

Under non-preemptive scheduling, once the CPU has been allocated a process, the process keeps the CPU until it releases it either by terminating or by switching to the waiting state.

Under preemptive scheduling, the CPU has the control to stop the process anytime it wants and preempt it by some other process.

ix. Which CPU scheduling algorithm(s) may cause starvation of processes?

Shortest Job First (SJF) and Priority Scheduling algorithms may cause the starvation of processes whenever there is a large number of short or high priority processes respectively.

x. Differentiate between Ready queue and Disk queue.

As processes enter the system, they are put into a ready queue, where they are ready and waiting to execute on a CPU's core.

When a process requests for an IO operation of a disk, they are put into a disk queue, where they are ready and waiting to get IO operations.

Q2: Given the following piece of code: (6)

```
main(int argc, char ** argv)
{
        int child = fork();
        int c = 5;
        if(child == 0)
        {
                c += 5;
        }
        else
        {
                child = fork();
                c += 10;
                if(child)
                        c += 5;
        }
}
```

How many different copies of the variable c are there? What are their values?

1. c = 10
2. c = 15
3. c = 20

Q3 (a): Given the following processes with their next CPU burst and arrival time. Give Gantt chart using the following scheduling algorithms: (8)

i. Shortest-Job-First (Use preemptive scheme).

ii. Round-Robin (Time Quantum = 5ms)

Process	Next CPU burst	Arrival Time
P0	10	0
P1	15	2
P2	5	4
P3	4	6
P4	12	8

0 4 9 13 19 31 46

0 5 10 15 19 24 29 34 39 44 46

Q3 (b): Calculate the average waiting time and the average turnaround time for each of the scheduling algorithms mentioned in (a) above. (8)

Process	SJF		Round-Robin	
	Turn-around	**Waiting time**	**Turn-around**	**Waiting time**
P_0	19 – 0 = 19	19 – 10 = 9	29 – 0 = 29	29 – 10 = 19
P_1	46 – 2 = 44	44 – 15 = 29	44 – 2 = 42	42 – 15 = 27
P_2	9 – 4 = 5	5 – 5 = 0	15 – 4 = 11	11 – 5 = 6
P_3	13 – 6 = 7	7 – 4 = 3	19 – 6 = 13	13 – 4 = 9
P_4	31 – 8 = 23	23 – 12 = 11	46 – 8 = 38	38 – 12 = 26

Q3 (c): Using preemptive priority-based scheduling, give the Gantt chart and calculate the average waiting time for the following priorities where 1 is the highest priority: (8)

P0 = 5; P1 = 3; P2 = 6; P3 = 4; P4 = 1