*Answer all questions in complete sentences explaining your answer thoroughly in your own words. Short answers or incomplete sentences will be give zero to partial credit. Please be detailed with your answers.*

*Name: Choi Tim Antony Yung*

[1] (25 pts) List three major components of any operating system and briefly describe their

role of each.

Interface: provide a way for user to interact with the system.

Kernel: the heart of the operating system that handles process and resources.

System applications: applications/utilities that was built into the operating system that are not part of the kernel.

[2] (25 pts) What is a likely sequence of events that must happen when you turn on a personal

computer (ie laptop or dekstop)? Hint: I’m looking for roughly eight steps.

The computer powers on; BIOS initialize CPU and RAM and test the RAM; BIOS boot the computer off one of the supported boot devices; BIOS locate the boot loader; Boot loader locates the kernel; Kernel is loaded into memory and started; Kernel initializes the hardware; Root file system for the OS is mounted; GUI is brought up.

[3] (25 pts) Many round-robin schedulers use a fixed size quantum. Provide an argument in favor of and against a small quantum. Thoroughly explain your answer.

An advantage of a smaller quantum is that process that have a larger burst time sees a decrease in turnaround time as it get allocated time slices more frequently.

The disadvantage of a small quantum is that there are more switching between processes as quantum get smaller and more context switching means more overhead.

Also, smaller processes sees their wait time and turnaround time increase as they may otherwise finish in one time slice where the quantum size bigger.

[4] (25 pts) There are two major design types for operating system kernels: monolithic kernels and microkernels. Which of these types better satisfies the following requirements? If both are equally as good write both. Justify your answers.

(a) Flexibility and ease of adding new hardware components.

Monolithic kernels need to be recompiled in order to add new driver and that is not needed in microkernels

(b) Security and reliability.

Monolithic kernels are more secure and reliable as everything is tightly coupled.

(c) Convenient access to operating system data structures.

Since the microkernel need to communicate with the kernel processes to handle the requests there are overhead associate with it comparing to monolithic kernel

[5] (25 pts) Discuss the purpose of a boot loader as it relates to modern operating systems,

including the differences between the boot loader in embedded systems vs desktop/laptop

class machines.

The purpose of the boot loader is to load the operating system into memory and run it.

Since there are no BIOS in embedded system, the boot loader will handle some tasks that BIOS and OS would otherwise do e.g. tests hardware and initialize hardware.

[6] (25 pts) Consider the following processes arriving at the ready queue and dispatched to

a single core CPU using priority based scheduling where jobs in the ready queue for a continuous 7 ms are increased in priority by 1:

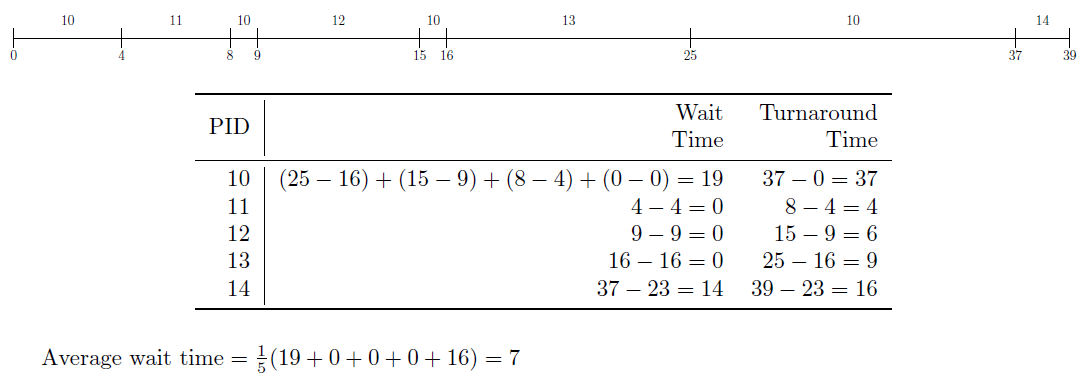
|  |  |  |  |
| --- | --- | --- | --- |
| PID | Arrival Time | Burst (ms) | Priority |
| 10 | 0 | 18 | 10 |
| 11 | 4 | 4 | 2 |
| 12 | 9 | 6 | 7 |
| 13 | 16 | 9 | 3 |
| 14 | 23 | 2 | 8 |

Answer the following:

(a) Draw a Gantt chart showing the arrival time and run times for each process.

(b) Calculate the wait time for each process and average waiting time.

(c) Calculate the turnaround time for each process



[7] (25 pts) Answer the following questions as they relate to deadlock:

(a) List and explain the four conditions required for deadlock.

Mutual exclusion: only one thread can share a non-sharable resource

Hold and Wait: thread is holding resources and waiting

No Preemption: Resources can not be preempted

Circular Wait: Threads wait in a circular way for the resources

(b) Does a cyclic dependency always lead to deadlock? Why or why not?

[8] (25 pts) In terms of scheduling and dispatching of processes:

(a) What is the purpose of a context switch and what role does the PCB play in a context switch?

Context switching prepare the resources to be at a certain state for the execution of next task. PCB record the information needed to restore the resource state before process continue to run.

(b) Why is it important to limit the amount of time processes spend in the ready queue?

Processes have resources allocated until it terminates. In the ready queue processes are not executing or waiting. Reducing the amount of time processes are in the ready queue allow for more resources to be available.

(c) Does the process wait for I/O to complete while it is in the ready queue? Explain.

[9] (25 pts) Discuss the purpose of hardware based cache on modern CPUs and the advantages

they provide to the overall system. Be sure to discuss why the underlying advantages exist

and the problem(s) the CPU cache attempts to solve

CPU need to access data stored in memory but it usually takes long time to do so due to the physical location and speed of memory which make I/O wait a bottleneck. By using hardware based cache it allows a CPU to store data frequently used at a small memory physically much closer to the CPU to limit the amount of time it takes to be transferred to the CPU and limit the cost associated with a fast memory.

[10] (25 pts) Explain CPU stall and how it affects the overall performance of a system. What method(s) are available to limit or prevent CPU stall?

CPU stall occur when the CPU wait for data needed for execution. As the data was not loaded into cache the CPU will need to fetch it. Processor affinity allows processes to be ran on the same processor such that it does not need to again fetch data to cache when switching processor.