Tutorial Letter 201/2/2018
Solutions to assignment no.1

Operating Systems And Architecture

COS3721

Semesters 2

School of Computing

This tutorial letter contains important information
about your module.

BARCODE

Assignment 1 – Semester 2

Due Date:	27 August 2018
Submission Procedure:	**Written/Typed. Submission online via myUnisa in a PDF format;**
Chapters in SGG:	Chapters 1 – 6 and 18 (section 18.1 and 18.2)
Marking scheme	
Question 1 [10 marks]	1.1 (4), 1.2(3), and 1.3(3)
Question 2 [06 marks]	2.1 (4) and 2.2(2)
Question 3 [10 marks]	3.1(2) and 3.2(8)
Question 4 [06 marks]	4.1(3) and 4.2(3)
Question 5 [06 marks]	5.1(4) and 5.2 (2)
Question 6 [12 marks]	[6.1 or 6.3](4), and 6.2(8)

Question 1 – Based on Chapter 1 of SGG

1.1 What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

Answer:

An interrupt is a hardware-generated change of flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

1.2 Describe some of the challenges of designing operating systems for mobile devices compared with designing operating systems for traditional PCs.

Answer:

The greatest challenges in designing mobile operating systems include:

- Less storage capacity means the operating system must manage memory carefully.
- The operating system must also manage power consumption carefully.
- Less processing power plus fewer processors mean the operating system must carefully apportion processors to applications.

1.3 Identify several advantages and several disadvantages of open-source operating systems. Include the types of people who would find each aspect to be an advantage or a disadvantage.

Answer: [3marks for 3 correct answers including advantages and/or disadvantages]

Open source operating systems have the advantages of having many people working on them, many people debugging them, ease of access and distribution, and rapid update cycles. Further, for students and programmers, there is certainly an advantage to being able to view and modify the source code. Typically open source operating systems are free for some forms of use, usually just requiring payment for support services. Commercial operating system companies usually do not like the competition that open source
operating systems bring because these features are difficult to compete against. Some open source operating systems do not offer paid support programs. Some companies avoid open source projects because they need paid support, so that they have some entity to hold accountable if there is a problem or they need help fixing an issue. Finally, some complain that a lack of discipline in the coding of open source operating systems means that backward compatibility is lacking making upgrades difficult, and that the frequent release cycle exacerbates these issues by forcing users to upgrade frequently.

Question 2 - Based on Chapter 2 of SGG

2.1 What are the two models of inter process communication? What are the strengths and weaknesses of the two approaches?

Answer:

The two models of interprocess communication are message-passing model and the shared-memory model. Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. It is also easier to implement than is shared memory for intercomputer communication. Shared memory allows maximum speed and convenience of communication, since it can be done at memory transfer speeds when it takes place within a computer. However, this method compromises on protection and synchronization between the processes sharing memory

2.2 Explain why Java programs running on Android systems do not use the standard Java API and virtual machine.

Answer:

It is because the standard API and virtual machine are designed for desktop and server systems, not mobile devices. Google developed a separate API and virtual machine for mobile devices.

Question 3 - Based on Chapter 3 of SGG

3.1 Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes.

Answer:

Simple communication works well with ordinary pipes. For example, assume we have a process that counts characters in a file. An ordinary pipe can be used where the producer writes the file to the pipe and the consumer reads the files and counts the number of characters in the file. Next, for an example where named pipes are more suitable, consider the situation where several processes may write messages to a log. When processes wish to write a message to the log, they write it to the named pipe. A server reads the messages from the named pipe and writes them to the log file

3.2 What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.
a. Synchronous and asynchronous communication
b. Automatic and explicit buffering
c. Send by copy and send by reference
d. Fixed-sized and variable-sized messages

Answer:

a. **Synchronous and asynchronous communication**—A benefit of synchronous communication is that it allows a rendezvous between the sender and receiver. A disadvantage of a blocking send is that a rendezvous may not be required and the message could be delivered asynchronously. As a result, message-passing systems often provide both forms of synchronization.
b. **Automatic and explicit buffering**—Automatic buffering provides a queue with indefinite length, thus ensuring the sender will never have to block while waiting to copy a message. There are no specifications on how automatic buffering will be provided; one scheme may reserve sufficiently large memory where much of the memory is wasted. Explicit buffering specifies how large the buffer is. In this situation, the sender may be blocked while waiting for available space in the queue. However, it is less likely that memory will be wasted with explicit buffering.
c. **Send by copy and send by reference**—Send by copy does not allow the receiver to alter the state of the parameter; send by reference does allow it. A benefit of send by reference is that it allows the programmer to write a distributed version of a centralized application. Java's RMI provides both; however, passing a parameter by reference requires declaring the parameter as a remote object as well.
d. **Fixed-sized and variable-sized messages**—The implications of this are mostly related to buffering issues; with fixed-size messages, a buffer with a specific size can hold a known number of messages. The number of variable-sized messages that can be held by such a buffer is unknown. Consider how Windows 2000 handles this situation: with fixed-sized messages (anything < 256 bytes), the messages are copied from the address space of the sender to the address space of the receiving process. Larger messages (i.e. variable-sized messages) use shared memory to pass the message.

Question 4 - Based on Chapter 4 of SGG

4.1 Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single processor system? Explain.

Answer:

A multithreaded system comprising of multiple user-level threads cannot make use of the different processors in a multiprocessor system simultaneously. The operating system sees only a single process and will not schedule the different threads of the process on separate processors. Consequently, there is no performance benefit associated with executing multiple user-level threads on a multiprocessor system.

4.2 In Chapter 3, we discussed Google's Chrome browser and its practice of opening each new website in a separate process. Would the same benefits have been achieved if instead Chrome had been designed to open each new website in a separate thread? Explain.

Answer:[2marks for No and 1mark for sound justification]

No. The primary reason for opening each website in a separate process is that if a web application in one website crashes, only that renderer process is affected, and the browser process, as well as other renderer processes, are unaffected. Because multiple threads all belong to the same process, any thread that crashes would affect the entire process.

Question 5 - Based on Chapter 5 of SGG

5.1 Describe two kernel data structures in which race conditions are possible. Be sure to include a description of how a race condition can occur.

Answer:

There are many answers to this question. Some kernel data structures include a process id (pid) management system, kernel process table, and scheduling queues. With a pid management system, it is possible two processes may be created at the same time and there is a race condition assigning each process a unique pid. The same type of race condition can occur in the kernel process table: two processes are created at the same time and there is a race assigning them a location in the kernel process table. With scheduling queues, it is possible one process has been waiting for IO which is now available. Another process is being context-switched out. These two processes are being moved to the Runnable queue at the same time. Hence there is a race condition in the Runnable queue.

5.2 Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs.

Answer:

If a user-level program is given the ability to disable interrupts, then it can disable the timer interrupt and prevent context switching from taking place, thereby allowing it to use the processor without letting other processes execute.

Question 6 - Based on Chapter 6 of SGG

6.1 Discuss how the following pairs of scheduling criteria conflict in certain settings.

a. CPU utilization and response time
b. Average turnaround time and maximum waiting time
c. I/O device utilization and CPU utilization

Answer:[4marks for two correct answers]

a. **CPU utilization and response time**: CPU utilization is increased if the overheads associated with context switching is minimized. The context switching overheads could be lowered by performing context switches infrequently. This could, however, result in increasing the response time for processes.
b. **Average turnaround time and maximum waiting time**: Average turnaround time is minimized by executing the shortest tasks first. Such a scheduling policy could, however, starve long-running tasks and thereby increase their waiting time.
c. **I/O device utilization and CPU utilization**: CPU utilization is maximized by running long-running CPU-bound tasks without performing context switches. I/O device utilization is maximized by scheduling I/O-bound jobs as soon as they become ready to run, thereby incurring the overheads of context switches.

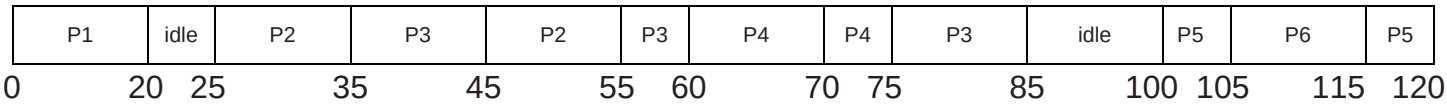
6.2 The following processes are being scheduled using a preemptive, round robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an ***idle task*** (which consumes no CPU resources and is identified as *Pidle*). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

Thread	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

a. Show the scheduling order of the processes using a Gantt chart.

Answer: [4 marks]

Note: pre-emption is performed at the time a higher prioritized process becomes ready for execution.



Note:

The concepts of Turnaround time and waiting time are explained in SGG, 8th edition, on page 187. Practical formulae, generated from the explanation given in the textbook were included in the tutorial letter 102 available for download on the myUnisa website. Those formulas are:

Turnaround time = Time of completion - Time of submission

∑ Waiting time = Turnaround time − ∑ CPU burst time

Applying these formulae yields the following results, presented in a tabular form:

b. What is the turnaround time for each process?

Answer: [2 marks]

Process	**Turnaround**
P1	(20 - 0) = 20
P2	(55 - 25) = 30
P3	(85 - 30) = 55
P4	(75 - 60) = 15
P5	(120 - 100) = 20
P6	(115 - 105) = 10

c. What is the waiting time for each process?

Answer: [2 marks]

Process	Waiting time
P1	(20 – 20) = 0
P2	(30 - 25) = 5
P3	(55 - 25) = 30
P4	(15 15) = 0
P5	(20 - 10) = 10
P6	(10 - 10) = 0

d. What is the CPU utilization rate?

Answer: [Not marked]

6.3 Which of the following scheduling algorithms could result in starvation?

a. First-come, first-served
b. Shortest job first
c. Round robin
d. Priority

Answer: [1mark for each correct answer]

No.	ALGORITHM	COULD RESULT IN STARVATION
a.	First-Come, First-Seved	No
b.	Shortest job first	Yes
c.	Round robin	No
d.	Priority	Yes