|  |  |  |  |
| --- | --- | --- | --- |
| Instructor | ***Luke Papademas*** | Due Date | **7/14** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 8 in your Computer Architecture and Organization textbook.

**Part 1 Glossary Terms - System Software**

Define, in detail, each of these glossary terms from the realm of computer architecture and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes

or course textbook(s) as references or the Internet by visiting Web sites such as:

[**http://www.ask.com**](http://www.ask.com) or [**http://www.webopedia.com**](http://www.webopedia.com/)

**(a) 3 - Tiered Architecture**

|  |
| --- |
| The three tiered architecture is an architecture that consists of a presentation layer, logic layer, and data layer. It is the most commonly used tiered architecture. Data is moved from the presentation layer to the logic layer and data layer, and vice versa. |

**(b) Parse Tree**

|  |
| --- |
| A parse tree is a data structure that is created during syntax analysis for the language that the programmer is using. It involves parsing the token stream of the programming language to produce the statements being parsed. |

**(c) Resident Monitor**

|  |
| --- |
| The resident monitor allowed programs to be processed without human interaction. They were the predecessor to modern operating systems. |

**(d) Tightly Coupled Multiprocessors**

|  |
| --- |
| Tightly coupled multiprocessors share a single centralized memory that requires the operating system to synchronize processes carefully. This is typically used for multiprocessors consisting of 16 or fewer processors. |

**(e) Transaction Processing Monitor**

|  |
| --- |
| A transaction processing monitor is designed to manage resources across various users and processes to ensure each process completes successfully. |

**Part 2 Exercises - System Software**

For each of the following, enter True or False.

\_\_T\_\_\_ **(1)** Spooling is the simplest form of multiprogramming.

\_\_F\_\_\_ **(2)** A virtual machine is the real hardware of the real computer that is controlling the program.

\_\_F\_\_\_ **(3)** Absolute code is non - executable binary code that must always be loaded at a particular location in memory.

\_\_F\_\_\_ **(4)** Interpreted languages also have a many - to - many relationship between the source code statements and executable machine instructions.

\_\_T\_\_\_ **(5)** In Java, deallocation of heap memory is referred to as garbage collection, which is done by the JVM automatically.

\_\_T\_\_\_ **(6)** To speed up the performance of the java software, Java’s Just - In - Time ( JIT ) compiler is used.

\_\_T\_\_\_ **(7)** The goal of database management systems is to provide timely and easy access to a large volume of data efficiently.

\_\_T\_\_\_ **(8)** Real - time systems are used for process control in manufacturing plants, assembly lines, robotics and complex physical systems.

\_\_T\_\_\_ **(9)** Java programs are stored in an intermediate code called bytecode.

\_\_F\_\_\_ **(10)** A timesharing system does not permit the systems to be accessed by multiple concurrent users.

**Part 3 Exercises - System Software**

**(1)** **( Database Processing )**

Answer the following with respect to database processing:

(a) What is a race condition? Give an example.

(b) How can race conditions be prevented?

(c) What are the risks in race condition prevention?

|  |
| --- |
| 1. **A race condition occurs when multiple process are competing for the same resource, and essentially are trying to finish before the other. An example of a race condition would be where two database jobs are attempting to update the same table without appropriately locking the table. Race conditions can result in different outputs than expected.** 2. **A race condition can be prevented through appropriate locking strategies to prevent access to the same record while another process is modifying it.** 3. **A risk of race condition prevention would be that processes could end up in high wait states while waiting for other processes to release exclusive locks on records.** |

**(2)** **( Deadlocks )**

We said that the risk of deadlock is always present anytime a system resource is locked.

Describe how a deadlock can occur such as in the situation given below.

Suppose T1 has X locked, and T2 has Y locked.

Now suppose T1 needs Y and T2 needs X .

|  |
| --- |
| **A deadlock can occur when T1 and T2 are updating separate data blocks, but either T1 or T2 finishes before the other and then requires access to the other resource but T1 or T2 has not finished and committed the update, and therefore has not release its lock. This can happen in a database when a process runs a transaction but does not commit or rollback the transaction, and another process attempts to update the same row.** |

**(3)** **( Operating Systems )**

What do you feel are the limitations of a computer that has no operating system? How

would a user load and execute a program?

|  |
| --- |
| **A limitation of a computer with no operating system may be that it is more difficult to run multiple tasks at once. A user would potentially need to program instructions for each task that they want to run. To run a program, they could load a program from an external source such as a disk and code instructions for the computer to execute the program.** |

**(4)** **( Compilers )**

Match the analyzer relative to compilers:

\_\_C\_\_\_ Semantic analyzer (a) the phase of a compiler that would give you a syntax error

\_\_B\_\_\_ Lexical analyzer (b) the phase that complains about undefined variables

\_\_A\_\_\_ Syntax analysis (c) the compiler phase that would emit an error message if you try to add an integer to a character string

**(5)** **( Java Virtual Machine )**

Why is the execution environment of a Java class called a virtual machine? How does this virtual machine compare to a real machine running code written in C ?

|  |
| --- |
| **The execution environment of a Java class is called a virtual machine because it has private memory areas addressable only by processes running inside the Java Virtual Machine, and it has its own ISA. It compares similarly to a machine running code written in C because it includes a native method area that is a workspace for objects external to Java.** |

**Part 4 Exercises - System Software**

Write a complete answer for each of these.

**(1) ( Mobile Operating Systems )**

What do you feel are the limitations of a mobile telephone that has no operating system? How would a user load and execute an application program?

|  |
| --- |
| **The limitations of a mobile telephone that has no operating system may include a requirement that instructions be coded for each component in the device, which would be time consuming and difficult to maintain. To run a program, a user would need to develop instructions for the hardware to interpret the program and then load the program from an external source.** |

**(2) ( Multiprogramming, Multiprocessing and Multithreading )**

What is the difference between multiprogramming and multiprocessing?

Multiprogramming and multithreading?

|  |
| --- |
| **Multiprogramming is the concurrent execution of multiple programs within one CPU. Multiprocessing is the use of multiple CPUs working in parallel. Multithreading is the process of subdividing multiple processes into different threads to increase concurrency.** |

**(3) ( Subsystems )**

Under what circumstances is it desirable to collect groups of processes and programs into

subsystems running on a large computer? What advantages would there be to creating

logical partitions on this system?

|  |
| --- |
| **It would be desirable to collect groups of processes and programs into subsystems to allow an administrator to start and stop these subsystems individually without affecting other subsystems in the system. The advantages of logical partitions include increased security by preventing access to files stored on a different partition and by allowing sandbox environments that allow user training on new programs without affecting other partitions.** |

**(4) ( Dynamic Linking )**

Discuss the advantages and disadvantages of dynamic linking.

|  |
| --- |
| **Dynamic linking enables disk space saving because they do not require multiple copies of the same code, a change to one module does not require all other modules to be relinked, and they allow for third parties to create common libraries. Disadvantages of dynamic linking include load time startup delays and that the programmer has no direct control over the contents of the dynamic link library routine.** |

**(5) ( Assembly Language )**

Why should assembly language be avoided for general application development? Under

what circumstances is assembly language preferred or required?

|  |
| --- |
| **Assembly language should be avoided for general application development because it is difficult to program in, it is error prone, and it is difficult to maintain. Assembly language is preferred when specialized hardware is being used and it is also extremely fast.** |