**Jinnah University for Women**

**Department of Computer Science and Software Engineering**

**Batch: BSSE 2021 (SEC-A)**

**Operating Systems - CSS 3102**

**Theory Assignment**

**Instructions:**

* The assignment must be submitted on JUW LMS. Email submission will not be accepted
* Each student should solve the assignment individually
* You are advised to go through the related topics before solving the assignment.
* Make your work clear and understandable
* Plagiarism may lead to marks deduction
* Use references where necessary

**Question # 1**

Reference Book: Operating System Concepts, 9th Edition, Silberschatz A., Peterson, J. L., & Galvin P. C.

1. What are opensource operating systems? How it differs from closed source approach? Discuss advantages and disadvantages of both. Give at least three examples of open source and closed source operating systems.

*Answer:*

*Open Source Operating System:*

Open source operating system is one in which source code is visible for all users or third party. It is editable and anyone has rights to modify the source code of operating system.

*How it Differ from Close Source Approach:*

Open Source OS is different from closed source OS as open source OS gives right to see and edit source code and closed source OS gives no permission to see and edit source code.

*Open Source OS:*

*Advantage:*

Generally open source OS is free to use.

It is flexible as they give the rights to edit the source code.

*Disadvantage:*

Generally open source OS are not user friendly.

There may be compatibility issues in open source OS as some applications cannot support both open and closed source OS.

*Closed Source OS:*

*Advantages:*

They are user friendly.

They provide more security.

*Disadvantages:*

They are expensive as we have to buy the license.

We have no rights to adjust it according to our need.

*Examples of Open Source OS:*

Linux

Open solaris

Free BSD

*Examples of Closed Source OS:*

Microsoft windows

IOS

Solaris unix

1. What is computing environment? Discuss different computing environments covered in book with examples for each.

*Answer:*

*Computing Environment:*

When a problem is solved by the computer, during that computer uses many devices, arranged in different ways and which work together to solve problems. This constitutes a computing environment where various number of computer devices arranged in different ways to solve different types of problems in different ways.

*Traditional Computing:*

Traditional Computing is a process of using physical data centers for various data assets. As a result, it also runs complete networking systems for day-to-day operations.

However, access to data, software, and storage is limited to users and devices. Hence, it only allows access to authorized devices that connect to the official network.

Therefore, it limits the users to only access the data from the system that stores it.

Moreover, users processes and systems processes provide services to a user. As a result, it manages the tasks frequently for optimization of the computer time.

**Example:** Windows is created, while a user is running another task on the computer. Therefore, the environment is allowing users to simultaneously execute different tasks.

*Mobile Computing:*

Mobile Computing refers to the type of environment that runs tasks on smartphones and tablets. Hence, it is computing on portable and lightweight devices.

Although, compared to other devices, mobile systems lack screen size, memory capacity, and other traditional functionalities. However, it does provide remote access to multiple services.

**Example:** Smartphones, tablets, Bluetooth devices, moreover, the two main operating systems that dominate this market are Apple iOS and Google Android.

*Client-Server Computing:*

Client-Server Computing is a type of environment that incorporates two machines. Therefore, it includes a client machine and a server machine. Sometimes, the same machine serves as the client and the server.

Subsequently, a client requests a resource or service and a server provides the same. Moreover, a server provides a resource or service to multiple clients simultaneously. Hence, the communication takes place using a computer network.

**Example:** email networking is example of client server architecture.

*Peer-To-Peer Computing:*

Peer-to-Peer is another type of distributed system in which clients and servers are not distinguished. All nodes of the system are considered as peers and each may act as either a client or a server, depending on whether it is requesting or providing service.

**Example:** Skype and torrent are the example of P2P computing.

*Virtualization:*

Virtualization is the creation of a virtual -- rather than actual -- version of something, such as an operating system (OS), a server, a storage device or network resources Virtualization is a technology that allows us to abstract the hardware of a single computer into several different execution environments, thereby creating the illusion that each separate environment is running on its own private computer. These environments can be viewed as different individual operating systems that may be running at the same time and may interact with each other. A user of a virtual machine can switch among the various operating systems in the same way a user can switch among the various processes running concurrently in a single operating system.

**Example:** Using VMware and virtual box we can create virtual systems.

*Cloud Computing:*

Cloud computing is a type of computing that delivers computing, storage, and even applications as a service across a network. In some ways, it’s a logical extension of virtualization, because it uses virtualization as a base for its functionality.

**Example:** the Amazon Elastic Compute Cloud (ec2) facility has thousands of servers, millions of virtual machines, and petabytes of storage available for use by anyone on the Internet. Users pay per month based on how much of those resources they use

*Real-Time Embedded System:*

Embedded computers are the most prevalent form of computers in existence. These devices are found everywhere, from car engines and manufacturing robots to optical drives and microwave ovens. They tend to have very specific tasks. The systems they run on are usually primitive, and so the operating systems provide limited features. Usually, they have little or no user interface, preferring to spend their time monitoring and managing hardware devices.

**Example:** Digital camera, digital watches, automobile engines and robotic arms.

1. What is the purpose of bootstrap program for multiple operating systems installed on partitions on single system?

*Answer:*

When two operating system are installed on the computer system then how system knows which operating system is to boot? A boot loader that understand multiple file systems and multiple operating system can occupy the boot space. Once loaded, it can boot one of the operating systems available on the disk. The disk can have multiple partitions, each containing a different type of operating system. When a computer system turn on, a boot manager program displays a menu, allowing user to choose the operating system to use.

A bootstrap program is the first code that is executed when the computer system is started. The entire operating system depends on the bootstrap program to work correctly as it loads the operating system.

The booting procedure starts with the hardware procedures and then continues onto the software procedures that are stored in the main memory. The bootstrapping process involves self-tests, loading BIOS, configuration settings, hypervisor, operating system etc.

Without bootstrapping, the computer user would have to download all the software components, including the ones not frequently required. With bootstrapping, only those software components need to be downloaded that are legitimately required and all extraneous components are not required. This process frees up a lot of space in the memory and consequently saves a lot of time.

1. Root directory is one of the most significant directories in Unix/Linux directory structure. You are required to enlist three unique characteristics of Root directory which are not associated with other directories in Unix/Linux environment.

*Answer:*

In a [computer](https://en.wikipedia.org/wiki/Computing) [file system](https://en.wikipedia.org/wiki/File_system), and primarily used in the [Unix](https://en.wikipedia.org/wiki/Unix) and [Unix-like](https://en.wikipedia.org/wiki/Unix-like) [operating systems](https://en.wikipedia.org/wiki/Operating_system), the root directory is the first or top-most [directory](https://en.wikipedia.org/wiki/Directory_(computing)) in a hierarchy.

It can be likened to the trunk of a [tree](https://en.wikipedia.org/wiki/Tree_(data_structure)), as the starting point where all branches originate from. Only root user has write privilege under this directory.

The root file system is the file system contained on the same [disk partition](https://en.wikipedia.org/wiki/Disk_Partition_Recovery) on which the root directory is located; it is the file system on top of which all other file systems are [mounted](https://en.wikipedia.org/wiki/Mount_(computing)) as the system boots up

1. Contrast between Windows and Linux operating system environments.

*Answer:*

|  |  |
| --- | --- |
| LINUX OS | MICROSOFT OS |
| It is open-sourced and thus freely available and user have rights to see and modify source code. | It is not freely available and involved in a high-cost licensing approach, user have no rights to see source code. |
| It is very secured and less prone to any cyber threats | It is vulnerable to security threats and cyber crimes |
| It supports multitasking functionality | It is not a favorable option from the multi-user perspective. |
| It also provides administrative from the system admin support perspective. | It follows an easily customizable approach from the end-user perspective |
| It’s not much user-friendly | It is user-friendly and based on a graphical user interphase approach |
| It normally doesn’t have that much support from the driver creation and storage management perspective | System updates are easily installable compared to its competitors |

1. Answer the following practice exercises.

1.4 1.5 1.8 1.10 1.22 1.24

1.4 Keeping in mind the various definitions of operating system, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers

*Answer:*

Operating system is part which is continuously running while the computer system is on. Web browser and mail programs are the type of application program and there is no need to run these program all the time so there is no need to include these programs in operating system.

However if they are the part of operating system then there is benefit that we don’t need to install them separately as they are the parts of OS then when we install OS these programs install as well.

1.5 How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system?

*Answer:*

The distinction between kernel mode and user mode provides a rudimentary form of protection in the following manner. Certain instructions could be executed only when the CPU is in kernel mode. Similarly, hardware devices could be accessed only when the program is executing in kernel mode. Control over when interrupts could be enabled or disabled is also possible only when the CPU is in kernel mode. Consequently, the CPU has very limited capability when executing in user mode, thereby enforcing protection of critical resources.

1.8 Some CPUs provide for more than two modes of operation. What are two possible uses of these multiple modes?

*Answer:*

Although most systems only distinguish between user and kernel modes, some CPUs have supported multiple modes. Multiple modes could be used to provide a finer-grained security policy. For example, rather than distinguishing between just user and kernel mode, you could distinguish between different types of user mode. Perhaps users belonging to the same group could execute each other’s code. The machine would go into a specified mode when one of these users was running code. When the machine was in this mode, a member of the group could run code belonging to anyone else in the group. Another possibility would be to provide different distinctions within kernel code. For example, a specific mode could allow USB device drivers to run. This would mean that USB devices could be serviced without having to switch to kernel mode, thereby essentially allowing USB device drivers to run in a quasi-user/kernel mode.

1.10 Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

*Answer:*

Caches are useful when two or more components need to exchange data, and the components perform transfers at differing speeds. Caches solve the transfer problem by providing a buffer of intermediate speed between the components

They speed up the processing as they contain the recent information if we need this information again then it can speedily be fetched trough cache.

The problem associated with cache is that if the information is changed in secondary memory then it must be updated in cache otherwise it will result in false computations.

Making cache as large as disk would be too costly, so that’s why we not make cache as large as disk and also cache is not replace with disk because cache generally contain temporary data and disk is used to store permanent data.

1.22 Many SMP systems have different levels of caches; one level is local to each processing core, and another level is shared among all processing cores. Why are caching systems designed this way?

*Answer:*

When the processor needs data that is not available in its cache, it can then access the cache shared by all of the processors. This is much more efficient each processor sharing the same cache. Why? Just imagine the scheduling overhead for each processor to access data.

1.24 Discuss, with examples, how the problem of maintaining coherence of cached data manifests itself in the following processing environments:

**a. Single-processor systems**

*Answer:*

In a computing environment where only one process executes at a time, this arrangement poses no difficulties, since an access to integer A will always be to the copy at the highest level of the hierarchy. However, in a multitasking environment, where the CPU is switched back and forth among various processes, extreme care must be taken to ensure that, if several processes wish to access A, then each of these processes will obtain the most recently updated value of A.

**b. Multiprocessor systems**

*Answer:*

The situation becomes more complicated in a multiprocessor environment where, in addition to maintaining internal registers, each of the CPUs also contains a local cache. In such an environment, a copy of A may exist simultaneously in several caches. Since the various CPUs can all execute in parallel, we must make sure that an update to the value of A in one cache is immediately reflected in all other caches where A resides. This situation is called cache coherency, and it is usually a hardware issue (handled below the operating-system level).

**c. Distributed systems**

*Answer:*

In a distributed environment, the situation becomes even more complex. In this environment, several copies (or replicas) of the same file can be kept on different computers. Since the various replicas may be accessed and updated concurrently, some distributed systems ensure that, when a replica is updated in one place, all other replicas are brought up to date as soon as possible. There are various ways to achieve this guarantee.

**Question # 2**

Discuss briefly.

1. The actions taken by a kernel to context-switch between processes

*Answer:*

Actions taken by a kernel to context-switch between processes are -

* The OS must save the PC and user stack pointer of the currently executing process, in response to a clock interrupt and transfers control to the kernel clock interrupt handler
* Saving the rest of the registers, as well as other machine state, such as the state of the floating point registers, in the process PCB is done by the clock interrupt handler.
* The scheduler to determine the next process to execute is invoked the OS.
* Then the state of the next process from its PCB is retrieved by OS and restores the registers. The restore operation takes the processor back to the state in which the previous process was previously interrupted, executing in user code with user-mode privileges.

1. The role of the init process on Linux systems in regard to process termination

*Answer:*

Init is the parent of all processes, executed by the kernel during the booting of a system. Its principle role is to create processes from a script stored in the file /etc/inittab. It usually has entries which cause init to spawn gettys on each line that users can log in.

Init is a daemon process that continues running until the system is shut down. It is the direct or indirect ancestor of all other processes and automatically adopts all orphaned processes. Init is started by the kernel during the booting process; a kernel panic will occur if the kernel is unable to start it.

If we can kill the init process the rest processes will become zombie process and system will stop functioning.

**Question # 3**

Gupta, Manjari, Lava Bhargava, and S. Indu. "Mapping techniques in multicore processors: current and future trends." *The Journal of Supercomputing* 77, no. 8 (2021): 9308-9363.

* Highlight the aim of this research.

*Answer:*

Multicore systems are in demand due to their high performance thus making application mapping an important research area in this field. Breaking an application into multiple parallel tasks efficiently and task-core assignment decisions can drastically influence system performance. This has created an urgency to find potent mapping techniques which can handle the complexity of these systems. Task assignment methods are governed by the application model, user-requirements, and architecture model. This paper provides an overview and classification of mapping algorithms that would facilitate graphical interpretation of the known techniques. It details the mapping methodologies along with performance, energy consumption, communication cost, reliability, or thermal management on different target architectures. Upcoming trends and open research areas have also been discussed.

* Explain the comparison of multicore and multiprocessor with the example in detail.

*Answer:*

|  |  |
| --- | --- |
| Multicore | Multiprocessor |
| A single CPU or processor with two or more independent processing units called cores that are capable of reading and executing program instructions | A system with two or more CPUs allow simultaneous processing of programs |
| Example: Consider a dual-core system where two tasks need to be executed simultaneously like accessing word document and watching a video. Each task is executed by individual cores simultaneously. Although the tasks are executed independently they can communicate with each other as they share a global memory | Example: Consider a quad-processor system with P1, P2, P3 and P4 processors. In an uni-processor system, processes are executed one at a time and next process is picked when the ongoing process is complete. In the multiprocessor system, the  processes are executed parallelly. Thus, a quad processor system can execute 4 processes at a time |
| Executes single program faster | Executes multiple programs faster |
| Less reliable as compared to multiprocessor | More reliable. Failure of one CPU will not eﬀect others |
| Have less traﬃc. This is because cores are inte-grated into a single chip and thus require less time | Have more traﬃc |
| Cheaper (single CPU that does not require multiple CPU support system) | Expensive (Multiple separate CPUs that require a system that supports multiprocessors) |
| Used in domains like network, graphics, embedded and general purpose | Meets the performance requirements of multimedia, digital signal processing (DSP), telecommunications and network security domain applications |

* Explain the following terms:
  + **Simultaneous Multiprocessing**

*Answer:*

In symmetric multiprocessing, multiple processors share a common memory and operating system. All of these processors work in tandem to execute processes. The operating system treats all the processors equally, and no processor is reserved for special purposes.

* + **Multicore System**

*Answer:*

A processor that has more than one core is called Multicore Processor. These cores can individually read and execute program instructions, giving feel like computer system has several processors but in reality, they are cores and not processors. Multicore systems support [MultiThreading](https://www.geeksforgeeks.org/multithreading-in-operating-system/" \t "_blank) and [Parallel Computing.](https://www.geeksforgeeks.org/introduction-to-parallel-computing/) Multicore processors are widely used across many application domains, including general-purpose, embedded, network, digital signal processing (DSP), and graphics (GPU).

* + **Chip-level Multiprocessing**

*Answer:*

Chip multiprocessors, also known as [multi-core](https://en.wikipedia.org/wiki/Multi-core_(computing)) computing, involves more than one processor placed on a single chip and can be thought of the most extreme form of tightly coupled multiprocessing. Mainframe systems with multiple processors are often tightly coupled.

* + **System-on-Chip**

*Answer:*

A system on a chip is an [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit) that integrates most or all components of a computer or other [electronic system](https://en.wikipedia.org/wiki/Electronics). These components almost always include a CPU, [memory](https://en.wikipedia.org/wiki/Computer_memory) interfaces,onchip [input/output](https://en.wikipedia.org/wiki/Input/output) devices, [input/output](https://en.wikipedia.org/wiki/Input/output) interfaces, and [secondary storage](https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage) interfaces, often alongside other components such as [radio modems](https://en.wikipedia.org/wiki/Radio_modem) and a [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU) – all on a single [substrate](https://en.wikipedia.org/wiki/Wafer_(electronics)) or microchip. It may contain [digital](https://en.wikipedia.org/wiki/Digital_signal_(electronics)), [analog](https://en.wikipedia.org/wiki/Analog_signal), [mixed-signal](https://en.wikipedia.org/wiki/Mixed-signal_integrated_circuit), and often [radio frequency](https://en.wikipedia.org/wiki/Radio_frequency) [signal processing](https://en.wikipedia.org/wiki/Signal_processing) functions (otherwise it is considered only an application processor).

**Question # 4**

1. In some systems, it seems as there are multiple processors (actually there is only one) working at the same time satisfying varied needs of multiple concurrent users. Which technique (s) is/are used to hide the reality of uniprocessor?

*Answer:*

The technique of multicore is using to hide the reality of uniprocessor. Multicore systems support multithreading and parallel computing.

1. Assume there is a multiprocessing system and different caches use a shared memory for local and global variables. Which process/ technique will ensure both caches to have only the updated value?

*Answer:*

The technique of cache coherency is used when different caches used shared memory then it is possible to have multiple copies of shared data, cache coherency assure that a change in any copy will must update the other copies of the same data.

1. There are two systems with the details given below. Which of the two can be considered more efficient and why?

System 1:

RAM 8 GB, HDD 500 GB

Support of virtual memory: yes

Processor speed: 1.9 GHz

System 2:

RAM 2 GB, HDD 80 GB, SSD 32 GB

Support of virtual memory on SSD

Processor speed:1.9GHz

*Answer:*

System1 is considered more efficient because it has 8 GB RAM along with 500GB HDD, however system2 has extra support of SSD but it has RAM of only 2GB and there is no benefit to increase in secondary storage with 2GB RAM.

1. On a single user system, a student is working on a text document and entering input through keyboard. The system is not much responsive and key strokes on keyboard are not echoed on screen timely. However, the screen shows input given by user if he waits for few seconds. Which mechanism is working to hold the user input for speed mismatch?

*Answer:*

The main memory has an area called buffer that is used to store or hold the data temporarily that is being transmitted either between two devices or between a device or an application. Buffering is an act of storing data temporarily in the buffer. It helps in matching the speed of the data stream between the sender and the receiver. If the speed of the sender’s transmission is slower than the receiver, then a buffer is created in the main memory of the receiver, and it accumulates the bytes received from the sender and vice versa.

**Question # 5**

Discuss following kernel data structures.

1. Lists, stacks and queues

*Answer:*

**Lists:**

Lists are the most fundamental data structures in computer science. Whereas each item in an array can be accessed directly, the items in a list must be accessed in a particular order. That is, a list represents a collection of data values as a sequence. The most common method for implementing this structure is a linked list

**Stacks:**

A stack is a sequentially ordered data structure that uses the last in, first out (LIFO) principle for adding and removing items, meaning that the last item placed onto a stack is the first item removed. The operations for inserting and removing items from a stack are known as push and pop

**Queues**:

A queue, in contrast, is a sequentially ordered data structure that uses the first in, first out (FIFO) principle: items are removed from a queue in the order in which they were inserted. There are many everyday examples of queues, including shoppers waiting in a checkout line at a store and cars waiting in line at a traffic signal. Queues are also quite common in operating systems—jobs that are sent to a printer are typically printed in the order in which they were submitted

1. Trees

*Answer:*

A tree is a data structure that can be used to represent data hierarchically. Data values in a tree structure are linked through parent–child relationships. In a general tree, a parent may have an unlimited number of children. In a binary tree, a parent may have at most two children, which we term the left child and the right child. A binary search tree additionally requires an ordering between the parent’s two children in which left child <= right child. When we search for an item in a binary search tree, the worst-case performance is O(n) (consider how this can occur). To remedy this situation, we can use an algorithm to create a balanced binary search tree. Here, a tree containing n items has at most lg n levels, thus ensuring worst-case performance of O(lg n). That Linux uses a balanced binary search tree (known as a red-black tree) as part its CPU-scheduling algorithm.

1. Hash functions and maps

*Answer:*

Hashing is the process of generating a value from a text or a list of numbers using a mathematical function known as a [hash function](https://www.geeksforgeeks.org/what-are-hash-functions-and-how-to-choose-a-good-hash-function/).

A **Hash Function**is a function that converts a given numeric or alphanumeric key to a small practical integer value. The mapped integer value is used as an index in the hash table. In simple terms, a hash function **maps** a significant number or string to a small integer that can be used as the **index**in the hash table.

1. Bitmaps

*Answer:*

A bitmap is a string of n binary digits that can be used to represent the status of n items. For example, suppose we have several resources, and the availability of each resource is indicated by the value of a binary digit: 0 means that the resource is available, while 1 indicates that it is unavailable (or vice-versa). The value of the ith position in the bitmap is associated with the ith resource.

References: <https://www.hitechwhizz.com/2021/05/6-advantages-and-disadvantages-drawbacks-benefits-of-open-source-software.html>

<https://brandongaille.com/19-pros-and-cons-of-closed-source-software/>

<https://www.hitechnectar.com/blogs/learn-more-about-computing-environment-and-its-different-types/>

<https://www.tutorialspoint.com/>

<https://www.geeksforgeeks.org/>

<https://en.wikipedia.org/>

<https://www.os-book.com/OS8/os8c/practice-exer-dir/1-web.pdf>

<https://medium.com/@alexchambers/operating-systems-midterm-review-70b7be321593>

<https://www.compuhoy.com/what-is-the-role-of-the-init-process-on-unix-and-linux-systems-in-regard-to-process-termination/>