**ASSIGNMENT 1**



**Spring 2023**

**Operating Systems**

**CSE-204**

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Dr. Durr-e-Nayab**

April 20, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Question: (Topic: OS types) [CLO-1]**

**1. Differentiate between multiprogramming systems and multiprocessing systems.**

**Ans:**

**Multiprogramming Systems:**

These computer systems support the simultaneous operation of multiple programmes. The CPU shifts between processes quickly, creating the appearance that they are all active at once. This method, also referred to as time-sharing, enables numerous users to access a computer system at the same time.

**Multiprocessing Systems:**

These are computer systems with multiple CPUs. (Central Processing Unit). Each CPU in a multiprocessing system can run another process at the same time. This greatly improve the system's performance, particularly for tasks that can be broken up into smaller units and carried out concurrently.

| **Multiprocessing** | **Multiprogramming** |
| --- | --- |
| The number of CPU is more than one. | The number of CPUs is one. |
| It takes less time for job processing. | It takes more time to process the jobs. |
| In this, more than one process can be executed at a time. | In this, one process can be executed at a time. |
| It is not economical. | It is economical. |
| The number of users is can be one or more than one. | The number of users is one at a time. |
| Throughput is maximum. | Throughput is less. |
| Its efficiency is maximum. | Its efficiency is Less |

**2. Differentiate between a multicore, multi-kernel, and multiple processor system. Discuss in terms of efficiency.**

**Ans:**

**Multicore Systems:**

In these computer systems, a single physical chip comprises multiple functioning cores (CPUs). The ability of each core to carry out independent instructions enables the system to carry out many tasks or threads concurrently. In general, multicore systems outperform single-core systems in terms of efficiency because they can run multiple operations at once without the need for additional hardware.

**Multi-kernel Systems:**

These are computer systems that have multiple operating system kernels running on the same hardware. Each kernel can run its own set of tasks and handle its own memory and resources. Multi-kernel systems are more efficient than traditional single-kernel systems because they allow for better resource allocation and can avoid bottlenecks caused by a single kernel.

**Multiple-Processor Systems:**

These are computer systems with numerous physical CPUs that can cooperate to carry out tasks in parallel are known as multiple processor systems. For large-scale tasks involving a lot of processing capacity, multiple processor systems may be more effective than single-processor systems. To properly utilize them, however, they may also be a challenge to handle and call for specialized software.

**Question: (Topic: OS modes) [CLO-1]**

**1. Some computer systems do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is and that it is not possible.**

**Ans:**

An operating system for a machine of this type would always need control. This could be done by two methods:

A) Software interpretation of all user programs. The software interpreter would provide, in software, what the hardware does not provide.

B) Requirement that all programs be written in high-level languages so that all object code is compiler produced. The compiler would generate the protection checks that the hardware is missing.

**2. What is the purpose of system calls, and how do system calls relate to the OS and to the concept of dual-mode (kernel-mode and user-mode) operation?**

**Ans:**

**System Call:**

A system call is a request made by a user-level program or process to the operating system's kernel to perform a privileged operation on its behalf. Some examples of privileged operations include accessing hardware resources like disks, network cards, or printers, allocating memory, and creating new processes.

**Relation to OS and Dual-Mode:**

The OS operates in two distinct modes: kernel mode and user mode. Kernel mode is the privileged mode where the OS kernel runs with unrestricted access to all system resources, while user mode is the non-privileged mode where user-level programs run with restricted access to system resources.

When a user-level program makes a system call, it switches the processor from user mode to kernel mode, and the operating system's kernel takes over the execution of the requested operation on behalf of the user-level program. Once the operation is complete, the processor is switched back to user mode, and the user-level program continues its execution.

**3. To a programmer, a system call looks just like a function call. Explain the difference in the underlying implementation. How does OS pass arguments to system calls?**

**Ans:**

A system call involves a costly context switch from user mode to kernel mode. In contrast, a regular function call stays in user mode and operates within the program's own memory space. It doesn't involve a context switch and can be executed much faster than a system call.

The OS passes arguments to system calls by copying them from the user's memory space to the kernel's memory space. This is necessary because the kernel has its own separate memory space, and it cannot directly access the memory of user programs. The OS must ensure that the arguments are valid and do not contain any malicious code before executing the system call.

**4. Discuss which of the following instructions should be privileged and why?**

**Ans:**

**I. Set the value of the timer:**

This instruction should be privileged because allowing user programs to set the timer could lead to issues such as denial-of-service attacks, where a program sets the timer to an extremely short value to prevent other programs from running.

**III. Clear memory:**

This instruction should be privileged because allowing user programs to clear memory could cause programs or data to be lost or corrupted.

**IV. Issue a trap instruction:**

This instruction should be privileged because it allows user programs to enter kernel mode, which can only be allowed by the operating system.

**V. Turn off interrupts:**

This instruction should be privileged because it affects the operation of the entire system, not just the user program executing it.

**VI. Modify entries in the device status table:**

This instruction should be privileged because it allows direct access to hardware resources, which can only be controlled by the kernel.

**VII. Switch from user to kernel mode:**

This instruction should be privileged because it allows user programs to access privileged resources that can compromise the security and stability of the system.

**VIII. Access I/O device:**

This instruction should be privileged because it allows direct access to hardware resources, which can only be controlled by the kernel.

### **References and Links :**

[4] A. Author(s), "Title of webpage," Name of Website, Date of Publication/Update, https://www.geeksforgeeks.org/difference-between-multiprocessing-and-multiprogramming/