CHAPTER 1:

1. In what way is an operating system like a government?

It creates an environment within which other programs can do useful work.

Explanation:

This analogy is used to explain that an operating system creates an environment within which other programs can do useful work, similar to how a government creates an environment within which citizens can work, live, and interact.

1. Mainframe operating system are designed primarily to maximize resource utilization.

Explanation:

Mainframe operating systems are designed to efficiently allocate system resources, such as memory and CPU time, to maximize utilization of the system.

1. The most common secondary storage device is magnetic disk.

Explanation:

The most common secondary storage device is magnetic disk, which is used for storing large amounts of data at a relatively low cost.

1. which of the following would lead you to believe that given system is an SMP-type system? Each processor performs all tasks within the operating system.

Explanation:

The characteristic of an SMP-type system is that each processor is capable of performing all tasks within the operating system, meaning that they have equal access to all system resources.

1. A timer can be used to prevent a user program from never retuning control to the operating system.

Explanation:

A timer can be used to prevent a user program from monopolizing system resources by forcing it to return control to the operating system after a specified amount of time has elapsed.

1. embedded computers typically run on a real-time operating system.

Explanation:

Embedded computers typically require a real-time operating system to respond quickly and predictably to external events and to perform specific tasks with strict timing requirements.

1. Bluetooth and 802.11 devices use wireless technology to communicate over several feet, in essence creating a small-area network.

Explanation:

Bluetooth and 802.11 devices use wireless technology to communicate over short distances, creating a small-area network for devices to communicate with each other.

1. A cluster system gathers together multiple CPUs to accomplish computational work.

Explanation:

A cluster system is a group of multiple computers working together as a single system to accomplish computational work.

1. Which of the following is a property of peer-to-peer systems? Clients and servers are not distinguished from one another.

Explanation:

In a peer-to-peer system, clients and servers are not distinguished from one another, and all nodes have the same capabilities and responsibilities.

1. Two important design issues for cache memory are size and replacement policy.

Explanation:

Cache memory size and replacement policy are important design issues because they impact the efficiency and performance of the system.

1. what are some other terms for kernel mode?

All of the above (supervisor mode, system mode, privileged mode)

Explanation:

Kernel mode is also known as supervisor mode, system mode, or privileged mode, as it allows the operating system to have complete control over the system's resources.

1. which of the following statements concerning open source operating system is true? Source code is freely available.

Explanation:

The source code of open source operating systems is freely available to the public and can be modified and distributed without restriction.

1. Which of the following operating systems is not open source?

Windows

Explanation:

Windows is not an open source operating system, as its source code is proprietary and not freely available.

1. A file-server system provides a file-system interface which allows clients to create and modify files.

Explanation:

A file-server system provides a file-system interface that allows clients to create, read, update, and delete files stored on a remote server.

1. A distribution is a custom build of the linux operating system.

Explanation:

A Linux distribution is a custom build of the Linux operating system that includes a specific combination of software packages and configuration settings.

1. Middleware is a set of software frameworks that provide additional services to application developers.

Explanation:

Middleware is a set of software frameworks that provides additional services to application developers, such as communication protocols, database connectivity, and security services.

1. What Statement concerning privileged instructions is considered false?

Thy cannot be attempted from user mode

Explanation:

This statement is false, as privileged instructions can only be executed in kernel mode, not user mode.

1. Which of the following statements is false?

The difference in storage capacity between a mobile device and laptop is shrinking

Explanation:

This statement is false, as the difference in storage capacity between mobile devices and laptops continues to increase, with laptops generally having more storage capacity than mobile devices.

1. A(n) process is the unit of work in a system.

Explanation:

A process is the unit of work in a system, representing a program in execution and the associated resources it requires to run.

1. The two separated modes of operating in a system are user mode and kernel mode.

Explanation:

The two separated modes of operating in a system are user mode, which limits the access of the program to system resources, and kernel mode, which allows full access to system resources.

1. (FALSE) The operating system kernel consists of all system and application programs in a computer.

Explanation:

This statement is false, as the operating system kernel is a specific part of the operating system that manages system resources and provides services to applications.

1. (TRUE) Flash memory is slower that DRAM but needs no power to retain its contents.

Explanation:

This statement is true, as flash memory is slower than DRAM but does not require power to retain its contents.

1. (FALSE) A system call is triggered by hardware.

Explanation:

This statement is false, as a system call is a request made by a program to the operating system to perform a specific task, and is triggered by software, not hardware.

1. (FALSE) UNIX does not allow users to escalate privileges to gain extra permissions for a restricted activity.

Explanation:

This statement is false, as UNIX allows users to escalate privileges to gain extra permissions for restricted activities.

1. (TRUE) Processors for most mobile devices run at slower speed than a processor in a desktop PC.

Explanation:

This statement is true, as processors for mobile devices are designed for low power consumption and longer battery life, rather than high performance.

1. (TRUE) Interrupts may be triggered by either hardware or software.

Explanation:

Interrupts are signals sent by hardware or software to the processor to interrupt the current process and request attention for a particular task or event. Interrupts can be triggered by both hardware devices, such as keyboard or mouse, and software applications, such as an input/output operation.

1. (FALSE) a dual-core system requires each core has its own cache memory.

Explanation:

A dual-core system can have a shared cache memory or individual cache memory for each core. It depends on the design of the system and the processor.

1. (TRUE) Virtually all modern operating systems provide support for SMP.

Explanation:

SMP stands for Symmetric Multi-Processing, which is a computer architecture where multiple processors or cores share a common memory and perform computations in parallel. Most modern operating systems, including Windows, Linux, and macOS, support SMP to take advantage of multi-core processors and improve system performance.

1. (FALSE) All computer systems have some sort of user interaction.

Explanation:

Not all computer systems require user interaction. For example, some embedded systems are designed to perform a specific task without any human intervention.

1. (TRUE) Solid state disks are generally faster than magnetic disks.

Explanation:

Solid-state disks (SSDs) are storage devices that use flash memory to store data, while magnetic disks (hard disk drives) use spinning disks to read and write data. SSDs are generally faster than magnetic disks because they have no moving parts and can access data more quickly.

1. (FALSE) Solid State disks are considered volatile storage.

Explanation:

Solid-state disks are non-volatile storage devices, which means that they can retain data even when the power is turned off. Volatile storage, on the other hand, loses its data when the power is turned off, such as the memory in a computer's RAM.

1. (TRUE) There is no universally accepted definition of an operating system.

Explanation:

The definition of an operating system varies depending on the context and the author. However, most definitions agree that an operating system is a software program that manages the hardware and software resources of a computer and provides a common platform for other software applications to run on.

CHAPTER 2:

1. A command interpreter is an example of a systems program.

Explanation:

A command interpreter, also known as a shell, is a system program that provides an interface for users to interact with the operating system by entering commands. Examples of command interpreters include Bash, Zsh, and PowerShell.

1. If a program terminates abnormally, a dump of memory may be examined by a debugger to determine the cause of the problem.

Explanation:

If a program terminates abnormally, a dump of memory may be examined by a debugger to determine the cause of the problem. True. When a program terminates abnormally, it can leave behind information about what went wrong. A debugger can examine this information, including a dump of memory, to determine the cause of the problem.

1. A message-passing model is easier to implement than a shared memory model for interpreter communication.

Explanation:

A message-passing model is easier to implement than a shared memory model for interpreter communication. False. Both message-passing and shared memory models have their own advantages and disadvantages for inter-process communication. However, shared memory is generally considered to be more efficient for communication between processes that need to exchange large amounts of data.

1. Policy determines what will be done.

Explanation:

Policy determines what will be done. True. Policy refers to the rules and guidelines that govern how a system or organization operates. Policies can dictate what actions are allowed or not allowed, and who has the authority to perform those actions.

1. The major difficulty in designing a layered operating system approach is appropiatly defining the various layers.

Explanation:

The major difficulty in designing a layered operating system approach is appropriately defining the various layers. True. A layered operating system approach is a way of designing an operating system by dividing it into different layers, each of which provides a different level of abstraction. The main challenge is defining the different layers and their interfaces in a way that is clear and consistent.

1. A micro is a kernel that is stripped of all nonessential components.

Explanation:

A microkernel is a kernel that is stripped of all nonessential components. True. A microkernel is a type of kernel that is designed to be as small and simple as possible. It provides only the most essential services, such as basic process and memory management, and other services are implemented as separate user-space processes.

1. To the SYSGEN program of an operating system, the least useful piece of information is what applications to install.

Explanation:

To the SYSGEN program of an operating system, the least useful piece of information is what applications to install. False. The SYSGEN program is used to generate a customized operating system image, which includes only the components and features that are needed for a particular use case. Knowing what applications will be installed can be useful in determining which components and features are necessary for the operating system image.

1. A boot block typically only knows the location and length of the rest of the bootstrap program.

Explanation:

A boot block typically only knows the location and length of the rest of the bootstrap program. True. The boot block is the first sector of a storage device, such as a hard disk or floppy disk, and it contains the initial bootstrap program that is loaded into memory by the computer's firmware. The bootstrap program then loads the rest of the operating system into memory.

1. System calls provide(s) an interface to the services provided by an operating system

Explanation:

System calls provide(s) an interface to the services provided by an operating system. True. System calls are a way for applications to request services from the operating system, such as file I/O, process management, and inter-process communication. System calls provide a standardized interface between applications and the operating system.

1. Security is not one of the major categories of system calls.

Explanation:

Security is not one of the major categories of system calls. False. Security is one of the major categories of system calls, along with process management, file I/O, inter-process communication, and network communication. Security-related system calls include those for authentication, access control, and encryption.

1. Modules allow operating system services to be loaded dynamically.

Explanation:

Modules allow operating system services to be loaded dynamically. True. Modules, also known as dynamically loadable kernel modules, are software components that can be loaded and unloaded from the operating system at runtime. They allow additional functionality to be added to the operating system without the need for a full system restart.

1. Microkernels use message passing for communication.

Explanation:

Microkernels use message passing for communication.

This statement is true. In a microkernel-based operating system, the kernel provides only the most basic services such as thread management, interprocess communication, and memory management. All other operating system services, such as device drivers and file systems, are implemented as separate user-space processes or servers. These servers communicate with each other and with the kernel via message passing, which can provide better isolation and fault tolerance compared to shared memory communication. An example of a microkernel-based operating system is QNX.

1. The Windows Create Process () system call creates a new process. What is the equivalent system call in UNIX: fork

Explanation:

The equivalent system call in UNIX for creating a new process is "fork." This system call creates a new process by duplicating the calling process. After forking, both the parent process and the child process continue executing at different points in the program.

1. The close () system call in UNIX is used to close a file. What is the equivalent system call in

Windows: CloseHandle ()

Explanation:

The equivalent system call in Windows for closing a file is "CloseHandle()." This system call releases a handle to an object, such as a file or a device driver, that was previously opened by the process.

1. The Windows Create File () system call is used to create a file. What is the equivalent system call in UNIX: open ()

Explanation:

The equivalent system call in UNIX for creating a file is "open()." This system call opens a file and returns a file descriptor that can be used in subsequent I/O operations on the file.

1. Android runs java programs in the Dalvik virtual machine.

Explanation:

This statement is no longer true. Prior to the release of Android 5.0, Android ran Java programs in the Dalvik virtual machine. However, starting with Android 5.0, Android switched to using the Android Runtime (ART) virtual machine.

1. IOS is a mobile operating system designed for the iPhone and iPad.

Explanation:

This statement is true. iOS is a mobile operating system developed by Apple Inc. for its mobile devices, including the iPhone and iPad.

1. The Standard C library provides a portion of the system call interface for UNIX and Linux.

Explanation:

This statement is true. The Standard C library provides a set of functions that can be used to perform various operations, such as I/O and string manipulation, in a C program. These functions are often implemented as wrappers around system calls.

1. Which of the following statements is incorrect?

Operating systems provide both command lines as well as graphical user interfaces.

Explanation:

This statement is incorrect. While most modern operating systems do provide both command line and graphical user interfaces, there are some operating systems that only provide one or the other. For example, some embedded operating systems only provide a command line interface.

1. Cache memory is/are not a technique for passing parameters from an application to a system call.

Explanation:

This statement is true. Cache memory is a type of high-speed memory that is used to temporarily store frequently accessed data. It is not used as a technique for passing parameters from an application to a system call.

1. (TRUE) KDE and GNOME desktops are available under open-source licenses.

Explanation:

This statement is true. Both KDE and GNOME are desktop environments for Unix-like systems, and they are available under open-source licenses.

1. (TRUE) Many operating systems merge I/O devices and files into combined file beacuase of the similarity of the system calls for each.

Explanation:

This statement is true. Many operating systems, including UNIX and Linux, treat I/O devices and files as if they were the same thing, and provide a uniform set of system calls for both.

1. (FALSE) An initial bootstrap program is the form of random-access memory (RAM).

Explanation:

This statement is false. An initial bootstrap program, also known as a bootloader, is a small program that runs when a computer is first powered on. Its job is to load the operating system into memory, typically from a storage device such as a hard drive or flash drive.

1. (FALSE) System calls can be run in either user mode or kernel mode.

Explanation:

This statement is actually false. System calls can only be executed in kernel mode, which is a privileged mode of the CPU. When a program makes a system call, it triggers a software interrupt that transfers control to the operating system's kernel, which runs in kernel mode. The kernel performs the requested service on behalf of the program, and then returns control to the program in user mode.

In contrast, applications run in user mode, which is a non-privileged mode of the CPU. In user mode, an application can only access a limited set of resources and instructions. It cannot directly access hardware or privileged instructions. This is a key security feature of modern operating systems.

1. (TRUE) Application programmers typically use an API rather than a directory system calls.

Explanation:

Explanation: An API (Application Programming Interface) is a set of functions and procedures that provide access to the features or services of an operating system or a software library. Application programmers use APIs to access the functionality provided by an operating system, rather than directly calling system calls. This is because APIs are usually easier to use, provide better error handling and abstraction, and shield the programmer from the underlying system details.

Example: In Windows, the Win32 API provides a wide range of functions and procedures for interacting with the operating system, such as creating windows, handling messages, manipulating files, and networking.

1. (TRUE) In general, Windows system calls have longer, more descriptive names and UNIX system calls use shorter, less descriptive names.

Explanation:

Explanation: Windows system calls usually have long, descriptive names that reflect their purpose and functionality. This is because the Windows API is designed to be user-friendly and self-explanatory. On the other hand, UNIX system calls typically have shorter names that are often abbreviations of the function they perform. This is because UNIX is a more command-line oriented system, and its system calls are designed to be used with simple, terse commands.

Example: The Windows CreateProcess function is used to create a new process, while the equivalent UNIX system call is fork.

1. (TRUE) Mac OS X is a hybrid system consisting of both the mach microkernel and BSD UNIX.

Explanation:

Explanation: Mac OS X is built on a hybrid architecture that combines elements of both microkernel and monolithic kernel designs. Its kernel, called XNU (X is Not Unix), is based on the Mach microkernel, which provides basic services such as inter-process communication, memory management, and threading. However, many higher-level functions are implemented as part of the kernel or in kernel extensions, following the monolithic approach. In addition, Mac OS X includes a large set of UNIX utilities and APIs that are compatible with BSD UNIX.

Example: Mac OS X provides a set of command-line tools and utilities that are similar to those found in a typical UNIX system, such as ls, cp, mv, and grep.

1. (FALSE) IOS is open source, Android is closed source.

Explanation:

Explanation: iOS is a closed-source operating system developed by Apple for its mobile devices, including the iPhone and iPad. Its source code is not publicly available, and only authorized developers can build and distribute iOS apps. Android, on the other hand, is an open-source operating system based on the Linux kernel and developed by the Open Handset Alliance. Its source code is freely available, and anyone can download and modify it to create custom versions or distributions.

Example: The Android source code is hosted on the Android Open Source Project (AOSP) website, where developers can download and contribute to the codebase.

CHAPTER 3:

1)The stack of a process contains temporary data such as function parameters, return addresses, and local variables.

Explanation:

The stack of a process holds temporary data, including function parameters, return addresses, and local variables, while the process is executing. Example: When a function is called, its parameters are pushed onto the stack, and when the function returns, the return address is popped off the stack.

1. A process control block includes information on the process’s state.

Explanation:

A process control block (PCB) stores information about a process, including its state, priority, and memory allocation. Example: When a process is interrupted, the operating system saves its current state in the PCB before switching to another process.

1. The list of processes waiting for a particular I/O device is called a(n) device queue.

Explanation:

A device queue is a list of processes waiting for a specific I/O device to become available. Example: When a process requests data from a hard drive, it enters the device queue until the hard drive can fulfill the request.

1. The degree of multiprogramming refers to the number of processes in memory.

Explanation:

The degree of multiprogramming refers to the number of processes that are actively loaded into memory at any given time. Example: If the degree of multiprogramming is four, then there are four processes currently loaded into memory and able to execute.

1. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?

All of the above (The child process runs concurrently with the parent, The child process has a new program loaded into it,The child is a duplicate of the parent)

Explanation:

When a child process is created, all of the following are possibilities in terms of its execution or address space: the child process can run concurrently with the parent, the child process can have a new program loaded into it, or the child can be a duplicate of the parent process. Example: When a web server creates a new process to handle a client request, the child process will typically have a new program loaded into it.

1. A context switch saves the state of the currently running process and restores the state of the next process to run.

Explanation:

A context switch occurs when the operating system saves the state of the currently running process and restores the state of the next process to run. Example: When a process is interrupted by an I/O event, the operating system performs a context switch to allow another process to use the CPU.

1. A process may transition to the Ready state by which of the following actions?

All of the above (Completion of an I/O event, awaiting its turn on the CPU-Newly, admitted process)

Explanation:

A process can transition to the Ready state by completing an I/O event, awaiting its turn on the CPU, or being a newly admitted process. Example: When a process finishes reading data from a file, it enters the Ready state and waits for its turn to use the CPU.

1. In a(n) zero capacity temporary queue, the sender must always block until the recipient receives the message.

Explanation:

In a zero capacity temporary queue, the sender must always block until the recipient receives the message. Example: When a process sends a message using a zero capacity temporary queue, it blocks until the recipient has successfully received the message.

1. A blocking send () and blocking receive() is known as a(n) rendezvous.

Explanation:

A blocking send() and blocking receive() is known as a rendezvous. Example: When two processes need to synchronize their execution, they may use a rendezvous to ensure that one process blocks until the other process is ready to receive the message.

1. Which of the following is true in a Mach operating system? All messages have the same priority.

Explanation:

In a Mach operating system, all messages have the same priority. Example: When multiple processes send messages to a receiving process, the Mach operating system treats all of the messages as having equal priority.

1. When communicating with sockets, a client process initiates a request for a connection and is assigned a port by the host computer. Which of the following would be a valid port assignment for the host computer? 1625

Explanation:

A valid port assignment for a host computer communicating with a client process could be 1625. Example: When a web server communicates with a client's browser, it assigns a unique port number to the browser so that it can receive data.

1. A(n) anonymous pipe allows several unrelated processes to use the pipe for communication.

Explanation:

An anonymous pipe allows several unrelated processes to use the pipe for communication. Example: When a program needs to share data between two unrelated processes, it can use an anonymous pipe to establish a communication channel.

1. Which of the following statements is true?

Shared memory is typically faster than message passing.

Explanation:

Shared memory is typically faster than message passing. Example: When two processes need to share data frequently and quickly, using shared memory may be faster than using message passing.

1. Imagine that a host with IP address 150.55.66.77 wishes to download a file from the web server at IP address 202.28.15.123. Select a valid socket pair for a connection between this pair of hosts.

150.55.66.77:2000 and 202.28.15.123:80

Explanation:

A valid socket pair for a connection between a host with IP address 150.55.66.77 and a web server at IP address 202.28.15.123 could be 150.55.66.77:2000 and 202.28.15.123:80. Example: When a web browser requests a webpage from a web server, it establishes a socket connection using a unique IP address and port number.

1. Child processes inherit UNIX ordinary pipes from their parent process because: A pipe is treated as a file descriptor and child processes inherit open file descriptors from their parents.

Explanation:

UNIX ordinary pipes are a form of interprocess communication, and a pipe is treated as a file descriptor. When a process forks, the child process inherits all open file descriptors from its parent, including any pipes that were created by the parent.

1. Which of the following statements is true? Reading and writing to ordinary pipes on both UNIX and Windows systems can be performed like ordinary file I/O.

Explanation:

This statement is true. Both UNIX and Windows treat pipes as file-like objects, so reading and writing to them can be performed using ordinary file I/O functions.

1. Which of the following is not a process type in the Chrome browser? Sandbox

Explanation:

The Chrome browser does have a sandbox process type, so this statement is false.

1. The foreground application is the application appearing on the display screen of a mobile device.

Explanation:

This statement is true. The foreground application is the one that is currently being displayed on the screen and receiving user input.

1. A process that has terminated, but whose parent has not yet called wait (), is known as a zombie process.

Explanation:

This statement is true. A zombie process is a process that has terminated but whose exit status has not yet been collected by its parent process.

1. The init process is assigned as the parent to orphan processes.

Explanation:

This statement is true. When a process terminates, its child processes become orphaned and are assigned to the init process as their new parent.

1. (TRUE) All processes in UNIX first translate to a zombie process upon termination.

Explanation:

This statement is false. When a process terminates, it first becomes a "terminated" process, and then its parent process can collect its exit status and it becomes a "zombie" process

1. (FALSE) The difference between a program and a process is that a program is an active entity while a process is a passive entity.

Explanation:

This statement is false. A program is a passive entity, while a process is an active entity that results from executing a program.

1. (FALSE) The exec() system call creates a new process.

Explanation:

This statement is false. The exec() system call replaces the current process image with a new process image.

1. (FALSE) All access to POSIX shared memory requires a system call.

Explanation:

This statement is false. Once shared memory has been created, processes can access it using regular memory access operations, without the need for a system call.

1. (TRUE) Local Procedure Calls in Windows XP are similar to Remote Procedure Calls.

Explanation:

This statement is true. Local Procedure Calls (LPCs) in Windows XP are similar to Remote Procedure Calls (RPCs) in that they allow processes to communicate with each other by calling functions that execute in the context of another process.

1. (TRUE) For a single-processor system, there will never be more than one process in the Running state.

Explanation:

In a single-processor system, only one process can execute at a time, so there can never be more than one process in the Running state.

1. (FALSE) Shared memory is a more appropriate IPC mechanism than message passing for distributed systems.

Explanation:

False. Shared memory is not the most appropriate IPC mechanism for distributed systems since distributed systems typically have a physical distance between the communicating processes, and therefore, require message passing for communication.

1. (TRUE) Ordinary pipes in UNIX require a parent-child relationship between the communicating processes.

Explanation:

True. Ordinary pipes in UNIX can only be used for communication between processes that have a common ancestor or a parent-child relationship.

1. (TRUE) Ordinary pipes in Windows require a parent-child relationship between the communicating processes.

Explanation:

True. Like UNIX, ordinary pipes in Windows are unidirectional and can only be used for communication between processes that have a common ancestor.

1. (TRUE) Using a section object to pass messages over a connection port avoids data copying.

Explanation:

True. In Windows, section objects can be used to map a section of shared memory into the address space of a process, which can then be used to send messages between processes without copying the data.

1. (TRUE) A socket is identified by an IP address concatenated with a port number.

Explanation:

True. A socket is a combination of an IP address and a port number that uniquely identifies a process.

1. (FALSE) Sockets are considered a high-level communications scheme.

Explanation:

False. Sockets are considered a low-level communications scheme since they provide a direct interface to the network layer.

1. (TRUE) The Mach operating system treats system calls with message passing.

Explanation:

True. In the Mach operating system, system calls are treated as messages that are sent from a user-level process to the kernel.

1. (TRUE) Named pipes continue to exist in the system after the creating process has terminated.

Explanation:

True. Unlike ordinary pipes, named pipes in Windows can be used for communication between unrelated processes and continue to exist even after the creating process has terminated.

1. (FALSE) A new browser process is created by the Chrome browser for every new website that is visited.

Explanation:

False. The Chrome browser creates a new process for every tab that is opened, not for every website.

1. (TRUE) The iOS mobile operating system only supports a limited form of multitasking.

Explanation:

True. The iOS mobile operating system only allows certain types of applications, such as media players and VoIP apps, to run in the background while the user interacts with other applications.

CHAPTER 4:

1)Green threads is a thread library for Solaris that maps many user-level threads to one kernel thread.

Explanation:

Green threads is a thread library for Solaris that maps many user-level threads to one kernel thread. This is true. Green threads are a user-level thread library that is used in Solaris. It maps multiple user-level threads to one kernel thread. This approach can be more efficient in terms of memory usage and context switching, but it also has some limitations.

1. Pthreads refers to a specification for thread behavior.

Explanation:

Pthreads refers to a specification for thread behavior. This is true. Pthreads is a POSIX standard for threads. It defines the behavior of threads and provides an API for creating and managing threads.

1. The many-to-many model multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads.

Explanation:

The many-to-many model multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads. This is true. In the many-to-many model, multiple user-level threads are multiplexed to a smaller or equal number of kernel threads. This model provides a balance between the one-to-one and many-to-one models, allowing more concurrency and avoiding the limitations of the many-to-one model.

1. Cancellation points are associated with synchronous cancellation.

Explanation:

Cancellation points are associated with synchronous cancellation. This is false. Cancellation points are associated with asynchronous cancellation. When a thread is canceled, it needs to perform some cleanup operations before it can terminate. Cancellation points are predefined points in the code where the thread can be canceled and perform the necessary cleanup.

1. Which of the following would be an acceptable signal handling scheme for a multithreaded program?

All of the above (Deliver the signal to the thread to which the signal applies, Deliver the signal to every thread in the process, Deliver the signal to only certain threads in the process)

Explanation:

Which of the following would be an acceptable signal handling scheme for a multithreaded program? All of the above (Deliver the signal to the thread to which the signal applies, Deliver the signal to every thread in the process, Deliver the signal to only certain threads in the process). This is true. All of the above are acceptable signal handling schemes for a multithreaded program, depending on the specific requirements of the program.

1. Signals can be emulated in windows through asynchronous procedure calls.

Explanation:

Signals can be emulated in windows through asynchronous procedure calls. This is true. In Windows, signals can be emulated using asynchronous procedure calls (APCs). APCs are a mechanism for asynchronous execution of code in the context of a thread. They can be used to simulate the behavior of Unix signals.

1. Thread-local storage is data that is unique to each thread.

Explanation:

Thread-local storage is data that is unique to each thread. This is true. Thread-local storage (TLS) is a mechanism for storing data that is unique to each thread. TLS allows each thread to have its own private copy of a variable, which can be useful for avoiding synchronization issues.

1. LWP is placed between user and kernel threads.

Explanation:

LWP is placed between user and kernel threads. This is true. In Solaris, the Lightweight Process (LWP) is a layer between user-level threads and kernel-level threads. The LWP provides a more lightweight thread abstraction than a traditional kernel thread.

1. Windows uses the one-to-one model.

Explanation:

Windows uses the one-to-one model. This is true. In Windows, the one-to-one model is used, where each user-level thread is mapped to a kernel-level thread. This model provides good performance and flexibility, but can be more resource-intensive than other models.

1. n multithreaded programs, the kernel informs an application about certain events using a procedure known as a(n) upcall.

Explanation:

In multithreaded programs, the kernel informs an application about certain events using a procedure known as a(n) upcall. This is true. In multithreaded programs, the kernel can use an upcall to notify the application about certain events, such as the availability of I/O or the completion of a system call. An upcall is a procedure that is called in the context of the application's thread.

1. Ensuring there is a sufficient number of cores is not considered a challenge when designing applications for multicore systems.

Explanation:

Ensuring there is a sufficient number of cores is not considered a challenge when designing applications for multicore systems. This is false. Ensuring that there is a sufficient number of cores is a major challenge when designing applications for multicore systems. If the application does not scale well with the number of cores, adding more cores will not improve its performance.

1. A thread library provides an API for creating and managing threads.

Explanation:

A thread library provides an API for creating and managing threads. For example, the POSIX threads (pthreads) library is commonly used in UNIX-like systems.

1. The many-to-many model multiplexes many user-level threads to a smaller or equal number of kernel threads.

Explanation:

The many-to-many model multiplexes many user-level threads to a smaller or equal number of kernel threads. This model provides better performance and scalability than the other models.

1. The many-to-one model maps many-user-level threads to one kernel thread.

Explanation:

The many-to-one model maps many user-level threads to one kernel thread. This model is less efficient than the other models because it doesn't utilize all available cores.

1. The one-to-one model maps each user-level thread to one kernel thread.

Explanation:

The one-to-one model maps each user-level thread to one kernel thread. This model is more efficient than the many-to-one model because it allows better utilization of available cores.

1. The two-level model allows a user-level thread to be bound to one kernel thread.

Explanation:

The two-level model allows a user-level thread to be bound to one kernel thread, which can then be executed on a core. This model combines the benefits of both the many-to-one and one-to-one models.

1. The most common technique for writing multithreaded Java program is implementing the runnable interface and defining its run () method.

Explanation:

The most common technique for writing multithreaded Java programs is implementing the Runnable interface and defining its run() method.

1. In threads, a parent uses the pthread\_join () function to wait. For its child thread to complete. What is the equivalent function in Win32?

WaitForSingleObject ()

Explanation:

In threads, a parent uses the pthread\_join() function to wait for its child thread to complete. The equivalent function in Win32 is WaitForSingleObject().

1. Which of the following statements regarding threads is false? Sharing is automatically provided in Java threads.

Explanation:

The statement "Sharing is automatically provided in Java threads" is false. Sharing must be explicitly managed through synchronization mechanisms in Java.

1. A thread pool uses an existing thread – rather than creating a new one – to complete a task.

Explanation:

A thread pool uses an existing thread, rather than creating a new one, to complete a task. This allows for better utilization of available threads and avoids the overhead of creating and destroying threads.

1. According to Amdahl’s Law, what is the speedup gain for an application that is 60% parallel and we run it on a machine with 4 processing cores? 1.43

Explanation:

According to Amdahl’s Law, the speedup gain for an application that is 60% parallel and run on a machine with 4 processing cores is 1.43. Amdahl's Law provides an estimate of the maximum speedup that can be achieved by parallelizing a program.

1. task parallelism involves distributing tasks across multiple computing cores.

Explanation:

Task parallelism involves distributing tasks across multiple computing cores. This is useful when the tasks are independent and do not require shared data.

1. Amdahl’s Law is a formula that identifies potential performance gains from adding additional computing cores to an application that has a parallel and serial component.

Explanation:

Amdahl’s Law is a formula that identifies potential performance gains from adding additional computing cores to an application that has both a parallel and serial component. It helps to identify the maximum speedup that can be achieved by parallelizing a program.

1. When OpenMP encounters the #pragma omp parallel directive, it creates as many threads as many threads as there are processing cores.

Explanation:

When OpenMP encounters the #pragma omp parallel directive, it creates as many threads as there are processing cores. OpenMP is a popular API for writing parallel programs in C, C++, and Fortran.

1. Grand Central Dispatch handles blocks by placing them on a dispatch queue.

Explanation:

Grand Central Dispatch handles blocks by placing them on a dispatch queue. This allows the blocks to be executed concurrently and asynchronously, which can improve performance.

1. (TRUE) A traditional (or heavyweight) process has a single thread of control.

Explanation:

(TRUE) A traditional (or heavyweight) process has a single thread of control.

1. (FALSE) A thread is composed of a thread ID, program counter, register set, and heap.

Explanation:

(FALSE) A Pthread is composed of a thread ID, program counter, register set, and heap. This describes a thread's execution context.

1. (TRUE) Virtually all contemporary operating systems support kernel threads.

Explanation:

(TRUE) Virtually all contemporary operating systems support kernel threads.

1. (FALSE) Linux distinguishes between processes and threads.

Explanation:

(FALSE) Linux distinguishes between processes and threads. Like most modern operating systems, Linux treats threads as lightweight processes.

1. (FALSE) In Java, data shared between threads is simply declared globally.

Explanation:

(FALSE) In Java, data shared between threads is simply declared globally. To manage data sharing between threads, Java provides synchronization mechanisms such as locks, semaphores, and monitors.

1. (TRUE) Each thread has its own register set and stack.

Explanation:

(TRUE) Each thread has its own register set and stack. This allows each thread to have its own set of local variables and function parameters.

1. (TRUE) Deferred cancellation is preferred over asynchronous cancellation.

Explanation:

(TRUE) Deferred cancellation is preferred over asynchronous cancellation. Deferred cancellation allows a thread to clean up its resources before terminating, while asynchronous cancellation may terminate a thread immediately, potentially leaving resources in an inconsistent state.

1. (FALSE) The single benefit of a thread pool is to control the number of threads.

Explanation:

This statement is false. In addition to controlling the number of threads, thread pools can improve performance by reducing the overhead of creating and destroying threads for short-lived tasks. By reusing existing threads, a thread pool can reduce the amount of time spent on thread management and increase overall efficiency.

1. (TRUE) it is possible to create a thread library without any kernel-level support.

Explanation:

This statement is true. It is possible to create a thread library entirely in user space, without relying on any kernel-level support. However, this approach may not be as efficient as using kernel-level threads.

1. (TRUE) It is possible to have concurrency without parallelism.

Explanation:

This statement is true. Concurrency refers to the ability of a system to support multiple tasks that are making progress simultaneously. Parallelism, on the other hand, involves using multiple processing units to execute tasks simultaneously. It is possible to have concurrency without parallelism if a system has only a single processing unit, but can switch between tasks quickly enough to give the illusion of concurrency.

1. (TRUE) Amdhal’s Law describes performance gains for applications with both a serial and parallel component.

Explanation:

This statement is true. Amdahl's Law is a formula that calculates the theoretical maximum speedup that can be achieved by parallelizing a computation. It takes into account both the proportion of the computation that can be parallelized (the "parallel fraction") and the amount of time spent on the non-parallelizable portion of the computation (the "serial fraction").

1. (TRUE) OpenMP only works for C, C++, and Fortran programs.

Explanation:

This statement is true. OpenMP (Open Multi-Processing) is an API that allows developers to write parallel programs in C, C++, and Fortran. It is designed to work with shared memory architectures and provides a set of compiler directives and library routines for creating parallel programs.

1. (FALSE) Grand Central Dispatch requires multiple threads.

Explanation:

This statement is false. Grand Central Dispatch (GCD) is a technology developed by Apple for creating concurrent programs in macOS, iOS, and other Apple operating systems. GCD uses a task-based model, where tasks are blocks of code that can be executed concurrently. GCD can work with a single thread or with multiple threads.

1. (TRUE) The trend in developing parallel applications is to use implicit threading.

Explanation:

This statement is true. Implicit threading involves using high-level programming models or libraries to abstract away the details of thread management, allowing developers to focus on the parallel aspects of their application. Examples of implicit threading models include OpenMP and Intel's Threading Building Blocks (TBB).

1. (FALSE) Task parallelism distributes threads and data across multiple computing cores.

Explanation:

This statement is false. Task parallelism involves breaking up a larger task into smaller subtasks that can be executed in parallel. The subtasks may or may not be distributed across multiple computing cores, depending on the details of the implementation. The distribution of threads and data is typically handled by the underlying parallel programming model or library.