

EECS3221 exam review

Jerry Wu

2023/08/15

Multiple choice

1. 1. If the time quantum get too large, Round Robin scheduling degenerates to...?
 - A) SJF
 - B) **FCFS**
 - C) Shortest-remaining-time-first
 - D) Multilevel queue

Reasoning

When the time quantum is large, round-robin scheduling can start to exhibit behavior similar to first-come-first-served (FCFS) scheduling. This happens because as the time quantum increases, the overhead of process switching becomes less significant compared to the actual execution time of processes. As a result, processes that get scheduled first will be able to complete their execution before their time quantum expires, and new processes won't get a chance to run until the currently running process completes. This makes the behavior resemble FCFS scheduling, where the process that arrives first gets to complete its execution before the next process gets a chance to run.

2. Computing systems need cache because
 - A) **accessing main memory is slow and cache speeds it up.**
 - B) register access is slow and cache speeds it up.
 - C) main memory is expensive and cache offsets the cost.
 - D) All of the above.

Reasoning

A computer's CPU cache is a small, high-speed memory component located on the processor chip that stores frequently accessed instructions and data from main memory. It acts as a buffer between the much slower main memory and the faster CPU, reducing the time it takes for the processor to access frequently needed information. By storing copies of frequently used data and instructions in the cache, the CPU can access them quickly, improving overall system performance and reducing latency in data retrieval operations.

3. If the base register is loaded with value 12000 and limit register is loaded with value 1000, which of the following memory address access will not result in a trap to the operating system?
- A) **12500**
 - B) 13200
 - C) 13346
 - D) 15200

Reasoning

In order for a memory access to not result in a trap, the register value must reside between the base value and the base value + the limit value, i.e.

$$base \leq n \leq base + limit$$

4. Suppose a program is operating with execution-time binding and the physical address generated is 400. The relocation register is set to 200. What is the corresponding logical address?
- A) 199
 - B) 201
 - C) **200**
 - D) 300

Reasoning

$$\textit{logical} = \textit{physical} - \textit{relocation}$$

5. Which of the following is a function that can be provided by Pthreads API for constructing a multithreaded program?
- A) `pthread_start`
 - B) `pthread_stop`
 - C) **`pthread_join`**
 - D) all of the above

Reasoning

Just memorize the API

6. In . . . , the process accesses the resources shared with others.
- A) entry section
 - B) **critical section**
 - C) exit section
 - D) remainder section

Reasoning

The critical section refers to a specific portion of a program or code where multiple concurrent processes or threads can potentially access shared resources or variables.

Short answers

7. Describe the difference between binary and counting semaphores.

Answer

Binary and counting semaphores are synchronization primitives used in concurrent programming. A binary semaphore can have only two values, 0 or 1, and is typically used for mutual exclusion, allowing one process/thread at a time to access a shared resource or critical section. It ensures that only a single process enters the critical section, preventing concurrent access. In contrast, a counting semaphore can have a range of non-negative integer values and is employed for scenarios where multiple processes need controlled access to a finite number of resources. It tracks the availability of resources and allows a specified number of processes to enter a critical section simultaneously while respecting the available resource count.

8. Describe the difference between named and unnamed semaphores.

Answer

Named and unnamed semaphores are synchronization mechanisms used in concurrent programming. An unnamed semaphore, often referred to as an anonymous semaphore, is typically limited to a single process or thread and is used for local synchronization within a program. It is not easily shareable across different processes. On the other hand, a named semaphore is associated with a unique name within the system and can be accessed and shared among multiple processes, enabling inter-process synchronization. Named semaphores provide a means for coordination and communication between unrelated processes by allowing them to synchronize their activities based on the same named semaphore object, facilitating a higher level of concurrency and collaboration in distributed systems.

9. What is the advantage of dynamic linking?

Answer

Multiple answers:

- * **Reduced Memory Usage:** Dynamic linking allows multiple programs to share a single copy of a library, reducing memory consumption as compared to static linking where each program has its own copy of the library.
- * **Easier Updates:** When a shared library is updated, all programs using that library automatically benefit from the update without needing to recompile them, simplifying maintenance and reducing deployment time.
- * **Faster Program Startup:** Dynamic linking defers the actual loading of library code until runtime, which can result in faster program startup times as only the necessary parts of the library are loaded when they are needed.
- * **Smaller Executable Size:** Dynamic linking can result in smaller executable files because the code of shared libraries is not duplicated in every program using them.
- * **Flexibility:** Users can update or replace shared libraries independently without affecting the entire application, offering more flexibility in managing software components.
- * **Version Control:** Dynamic linking enables versioning of shared libraries, allowing different programs to use different versions of the library based on compatibility requirements.
- * **Hot Patching:** Some systems allow dynamic updates to shared libraries without restarting the programs using them, facilitating "hot patching" to fix bugs or apply security patches without interruption.
- * **Language Interoperability:** Dynamic linking supports mixed-language programming by enabling different languages to share and interact with the same library code.
- * **Efficient Disk Usage:** When multiple programs use the same shared libraries, disk space is conserved since the library code is stored in a single location rather than being duplicated for each program.
- * **Smoother Software Distribution:** Dynamic linking simplifies software distribution by allowing libraries to be distributed separately,

reducing the size of installer packages and enabling more efficient updates.

These advantages collectively contribute to more efficient memory usage, easier maintenance, faster execution, and improved software distribution in dynamic linking scenarios.

10. Describe the different types of thread cancellation modes (ex. deferred, asynchronous, and off).

Answer

* **Deferred (or Default) Cancellation Mode:**

- Thread remains cancellable until it explicitly checks for cancellation points, like system calls or library functions that could block the thread.
- Cancellation request is deferred until the thread reaches a cancellation point.
- Provides controlled cancellation, allowing the thread to perform cleanup operations before termination.

* **Asynchronous Cancellation Mode:**

- The thread can be canceled at any time, regardless of whether it's at a cancellation point.
- Cancellation takes place immediately after the cancel request is issued, without waiting for the thread to reach a cancellation point.
- Cleanup functions registered with the thread may not have an opportunity to execute fully, potentially leading to resource leaks.

* **Disabled (or Off) Cancellation Mode:**

- Cancellation is disabled for the thread.
- A thread in this mode ignores any cancellation requests, and cancellation points are effectively ignored.
- Useful when critical sections need to be protected from cancellation.

Each of these thread cancellation modes has its own use cases and implications in terms of control, responsiveness, and resource management within a multi-threaded environment.

11. Describe the benefits of hierarchical paging tables and hash tables, compared to traditional paging.

Answer

Hierarchical paging tables and hash tables offer benefits compared to traditional paging by providing more efficient and flexible memory management. Hierarchical paging tables organize the virtual address space into a multi-level structure, reducing memory overhead by allocating memory for page tables only as needed. This helps manage large address spaces more efficiently. Hash tables, on the other hand, enable faster page table lookups by directly mapping virtual pages to physical frames, avoiding the need for traversing multiple levels of page tables. Hash tables can provide constant-time access for page table entries, improving memory access times and reducing overhead. These approaches enhance memory utilization, reduce memory overhead, and potentially lead to faster memory lookups, all contributing to improved system performance in modern memory management.