课堂练习题

(1)操作系统创建一个新进程所执行的步骤是什么？

(2)模式切换和进程切换有什么区别？

(3)当一个用户级线程执行系统调用时，不仅这个线程被阻塞，而且进程中所有线程都被阻塞，为什么？

(4)给出操作系统进行进程管理时的五种主要活动，并简单描述为什么需要它们；

课堂练习题

操作系统创建一个新进程所执行的步骤是什么？

1. Assign a unique process identifier to the new process.
2. Allocate space for the process.
3. Initialize the process control block.
4. Set the appropriate linkages.
5. Create or expand other data structures.

课堂练习题

模式切换和进程切换有什么区别？

- A mode switch may occur without changing the state of the process that is currently in the Running state. A process switch involves taking the currently executing process out of the Running state in favor of another process. The process switch involves saving more state information.

课堂练习题

当一个用户级线程执行系统调用时，不仅这个线程被阻塞，而且进程中所有线程都被阻塞，为什么？

- Because, with ULTs, the thread structure of a process is not visible to the operating system, which only schedules on the basis of processes.

课堂练习题

- 给出操作系统进行进程管理时的五种主要活动，并简单描述为什么需要它们；

➢ **Creation and deletion of both user and system processes.**

The processes in the system can execute concurrently for information sharing, computation speedup, modularity, and convenience. Concurrent execution requires a mechanism for process creation and deletion. The required resources are given to the process when it is created, or allocated to it while it is running. When the process terminates, the OS needs to reclaim any reusable resources.

➢ **Suspension and resumption of processes.**

In process scheduling, the OS needs to change the process's state to waiting or ready state when it is waiting for some resources. When the required resources are available, OS needs to change its state to running state to resume its execution.

课堂练习题

- 给出操作系统进行进程管理时的五种主要活动，并简单描述为什么需要它们；

- **Provision of mechanism for process synchronization.**
  Cooperating processes may share data. Concurrent access to shared data may result in data inconsistency. OS has to provide mechanisms for processes synchronization to ensure the orderly execution of cooperating processes, so that data consistency is maintained.
- **Provision of mechanism for process communication.**
  The processes executing under the OS may be either independent processes or cooperating processes. Cooperating processes must have the means to communicate with each other.
- **Provision of mechanisms for deadlock handling.**
  In a multiprogramming environment, several processes may compete for a finite number of resources. If a deadlock occurs, all waiting processes will never change their waiting state to running state again, resources are wasted and jobs will never be completed.