Lab7

B06901160 電機三 翁挺瑋

1. (OS concepts) How similar or different between a mutex and a binary semaphore?

A :

| **BASIS FOR COMPARISON** | **SEMAPHORE** | **MUTEX** |
| --- | --- | --- |
| Basic | Semaphore is a signalling mechanism. | Mutex is a locking mechanism. |
| Existence | Semaphore is an integer variable. | Mutex is an object. |
| Function | Semaphore allow multiple program threads to access a finite instance of resources. | Mutex allow multiple program thread to access a single resource but not simultaneously. |
| Operation | Semaphore value is modified using wait() and signal() operation. | Mutex object is locked or unlocked by the process requesting or releasing the resource. |

1. How are differences between the IRQ (interrupt) context and the application (user) context?

A: Application context is the current state of process, application context can be go into the sleep, preemptable, It perform time consumable task, acquiring and releasing mutex.

Interrupt context is when the interrupt occurs state/priority goes to interrupt handler, and current process stops/saves until we complete interrupt, Interrupt context is not time consumable, non preemptable, It cannot go into the sleep.

1. What is IRQ safe? What is thread safe?

A: A piece of code is thread-safe if it functions correctly during simultaneous execution by multiple threads. In particular, it must satisfy the need for multiple threads to access the same shared data, and the need for a shared piece of data to be accessed by only one thread at any given time.

IRQ safe also means reentrancy. In computing, a computer program or subroutine is called reentrant if multiple invocations can safely run concurrently. The concept applies even on a single processor system, where a reentrant procedure can be interrupted in the middle of its execution and then safely be called again ("re-entered") before its previous invocations complete execution. The interruption could be caused by an internal action such as a jump or call, or by an external action such as an interrupt or signal.

1. What is a counting semaphore? What is the meaning of the counting semaphore value?

A: Conceptually, a semaphore is a nonnegative integer count. Semaphores are typically used to coordinate access to resources, with the semaphore count initialized to the number of free resources. Threads then atomically increment the count when resources are added and atomically decrement the count when resources are removed.When the semaphore count becomes zero, indicating that no more resources are present, threads trying to decrement the semaphore block wait until the count becomes greater than zero.