Chapter 4 Threads - 2

The OS kernel controls the following properties/tasks for individual user-level threads:

- [ ] Thread creation and termination
- [ ] Thread scheduling on the CPU
- [x] Opening and closing files called from the thread
- [ ] Storing the thread's hardware state on a context switch

A program cannot realize concurrent thread-level execution on multicore machines with "kernel-level" threads.

- ( ) True
- (•) False

Clear selection

Kernel-level threads have the following properties and benefits over use-level threads:

- [x] threads can concurrently execute (if multiple CPU cores are available)
- [ ] each thread can open thread-specific files
- [x] required to support the one-to-one multithreading model
- [x] requires support from the OS kernel

On Linux, calling "fork()" from a thread will only duplicate a single flow of control in the child process

- (●) True
- ( ) False

Clear selection

On linux, with "deferred" cancellation a thread is required to terminate itself.

- (●) True
- ( ) False

Clear selection

Synchronous signals are always only delivered to the thread to which the signal applies

- (●) True
- ( ) False

Clear selection

Submit