Conceptual Questions

1. **pthread\_create** creates a new thread

**pthread\_join** waits for the given thread to finish before continuing

**pthread\_exit** terminates the calling thread

2. Threads share the global data and shared memory, unlike processes. Threads also have their own local data which is not shared between threads.

3. Unlike multiprocessing, multithreading does not need to interact with the operating system. Unlike processes, threads can share the same data. Processes can operate independently, threads cannot.

Multithreading has the advantage over multiprocessing in being less resource intensive. Multithreading is at a disadvantage in terms of security between threads. One thread may overwrite the data of another thread.

4. Mutual exclusion is the principle that only one thread can execute the critical section of its code at one time. The critical section is a part of the code that needs to access and modify shared data.

5. **pthread\_mutex\_lock(mutex)** acquires a lock on the specified mutex variable

**pthread\_mutex\_trylock(mutex)** attempts to lock the mutex, returns an error code if it is already locked

**pthread\_mutex\_unlock(mutex)** unlocks a mutex if it is owned by the calling thread. Returns an error if the mutex is already unlocked or owned by another thread