Threadpool實作:

1. 在server初始化的時候建立threadpool，先create固定數量的thread。
2. Thread 第一件要做的事情就是去搶奪 pool 的 lock，當搶到 lock 的 Thread 發現沒有工作可以做的時候， 就會執行 pthread\_cond\_wait 來等待通知。這時候 pool->lock 會被 Unlock，因此這時候其他 Thread 也可以進來這個區域。 所以在完全沒有工作的情況下，所有的 Thread 都會在這邊 Waiting。
3. 當 Thread 被透過 pthread\_cond\_signal 喚醒的時候，該 Thread 就會重新取得 pool->lock。 這時他就可以安心的取出 queue 中的 task，等待取出完畢之後，再 unlock 讓其他被喚醒的 Thread 也可以去取得 Task。
4. 之後就是執行 task 中的 handle\_request 做該做的工作。
5. 當server select的時候，若有thread以經在執行同個conn\_fd的request則先略過。
6. 當server 有新的request時，若沒有閒置的thread，沒有則回傳BUSY。

使用process代替thread:

用fork()來創造child process 來做handle\_request()，parent process 只要負責select client，讓child process 來執行不同的request

Pocess與thread的比較:

1. thread分享同樣的memory和resource，而process只能透過shared memory和message passing的方法來分享資料
2. thread比process輕巧，創建和context switch thread比創建一個process來的快速
3. create a process is expensive in terms of time and memory.
4. The schedule among threads can be conducted by main thread for both user level thread and kernel level thread.