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.