SP MP4 report

資工三 B03902125 林映廷

thread pool implementation：

在 server 端，先 create 指定的 threads 數量，並讓 threads 的 handle_requests_loop()函數進入 busy loop 的狀態。等到 add_request()函數將 request 加入 queue 裡時，會 call pthread_cond_signal()函數通知 available threads，threads 再 call get_request()函數，將 request 從 queue 裡拿出來。但由於要讓 request queue 在沒有 available threads 時的 requests 數量為零，此時，不會 call add_request()函數，而是重新要求 client 之後再 send request。

how to use process instead of thread to handle multiple clients：

在 server 端，一遇到 request，就先 call fork()，產生 child process 去處理 request。等到 child process 已經把 request 處理完了，call exit()。parent process 會 call wait()，等到收到 child process 的 termination status。

但如果是用 vfork()，會先讓 child process 先執行完，才輪到 parent process。因此，如果這個時候有新的 request 進來，而且 child process 也還沒處理完 request，parent process 無法指派 request 給另外一個 child process 處理。而且由於 child process 還沒執行完，很有可能 call recv_message()時，收到不該收的東西。

compare throughput between using processes and using threads：

using processes：會多開一個 memory space，也就是 process 會有自己的 memory space，並沒有 share 同一塊 memory。而且 delay time 也比較大。但如果 child process crash 並不會影響到 parent process 的執行。

using threads：每個 thread 有自己的 stack frame，但 text segment 和 data segment，在每個 thread 之間是 shared。而且 delay time 比較小。但如果其中有一個 thread crash，整個 process 也會跟著 crash。