

4.1 Provide two programming examples in which multithreading does not provide better performance than a single-threaded solution.

ANS:

Any kind of sequential program is not a good candidate to be threaded. An example of this is a program that calculates an individual tax return. (2) Another example is a “shell” program such as the C-shell or Korn shell. Such a program must closely monitor its own working space such as open files, environment variables, and current working directory.

例如 Fibonacci

4.4 The program show in Figure 4.14 uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */
int main(int argc, char *argv[])
{
    int pid;
    pthread_t tid;
    pthread_attr_t attr;

    pid = fork();
    if (pid == 0){ /* child process*/
        pthread_attr_init(&attr);
        pthread_create(&tid,&attr,runner,NULL);
        pthread_join(tid,NULL);
        printf("CHILD: value = %d" , value); /* LINE C */
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d" , value); /* LINE P */
    }
}

void *runner(void *param) {
```

```
value = 5;
pthread_exit(0);
}
```

Figure 4.14 C program for Exercise 4.4

ANS:

Output at LINE C is 5. Output at LINE P is 0.

因為 Child 和 parent 的資料沒有共享。

4.6 What are two differences between user-level threads and kernel-level threads?

Under what circumstances is one type better than other?

ANS:

(1) User-level threads are unknown by the kernel, whereas the kernel is aware of kernel threads.

(2) On systems using either M: 1 or M: N mapping, user threads are scheduled by the thread library and the kernel schedules kernel threads.

(3) Kernel threads need not be associated with a process whereas every user thread belongs to a process. Kernel threads are generally more expensive to maintain than user threads as they must be represented with a kernel data structure.

請參考第八版課本答案 P157~159, 第七版 P.125~127