

OS_110_CH4

Thread Introduction

- 又稱 **lightweight process**
- All threads belonging to the same process share
 - code section
 - data section
 - OS resources
- Each thread has own
 - thread ID
 - program counter
 - register set
 - stack
- Example : web browser, web server, RPC server

Benefits

- Responsiveness : 反應好
- Resource sharing : 有共同的位址空間
- Utilization of MP arch : 可以多執行緒
- Economy : cost 比 process 還要高

Multithreaded Programming

- advantage
 - efficient use of multiple cores
 - improved concurrency
- challenge

- Dividing activities : 拆計算
- Data splitting : 拆資料
- Data dependency : 資料同步
- Balance : 執行時間的平均
- Testing and debugging

User threads vs. Kernel Threads

User threads

- POSIX Pthreads, Win32 threads...
- By user-level **threads library**
- 比較快創建，因為不會碰到 system call
- 如果 kernel 是單線程，就算 user 創幾個都沒用

Kernel threads

- Windows 2000, Linux...
- By **kernel**
- 會相對比較慢
- 獨立執行

Multithreading Models

Multithreading Models

- Many-to-One

- One-to-One
- Many-to-Many

Mant-to-One

- Mapping to single kernel thread
- 並非真正的多線程，因為不會創建其他 kernel 線程，無法做平行運算
- 創建線程相對快
- 會有 block 的問題

One-to-One

- 可以同時
- 會比較慢
- User 創建的線程有數量限制

Many-to-Many

- 沒有數量限制
- 會更慢

Threaded Case Study

What is Pthread?

- POSIX (Portable Operating System Interface)
- Pthread is the **implementation** of **POSIX standard** for thread
- `pthread_create(thread_id, attr, routine, &arg);`
- `pthread_join(thread_id, *status);` // 可以接受回傳

```
pthread_create(thread, NULL, func, &arg);  
// &arg -> void pointer  
pthread_join(thread, *status);
```

language-c

```
func(&arg){  
    ...  
    return(*status);  
}
```

language-c

- pthread_detach(thread_id); // 不用回傳

Linux Threads

- Linux 不支援多線程，是對於 OS，不是對於 programmer
- 可以用 fork 創建一個 process，複製 parent process 相連的資料
- clone 也可以創建一個 process，可以 share 記憶體

Threading Issue

fork() and exec()

- 有些系統兩種 fork() 的方式都支援
 - 直接複製一整個 process
 - 只複製該線程
- execp works the same; replace the entire process

Thread Cancellation

- Target thread : a thread that is to be cancelled
- **Asynchronous cancellation** : 執行到一個階段做 clean up
- **Deferred cancellation** : 等到一個 check point 做 clean up

Signal Handling

- 被特定事件產生
- is delivered to a process
- handled

Thread Pools

- 創建一個數量的 threads，然後可以重複使用這些 threads

- 處理一個服務速度會變得更快，因為不用多時間創建

- 可以控制資源使用量