

Chương 2 Quản lý tiến trình

2. Luồng

2.3. Cài đặt luồng với Windows

- Giới thiệu
- Mô hình đa luồng
- Cài đặt luồng với Windows
- Vấn đề đa luồng

Chương 2 Quản lí tiến trình

2. Luồng

2.3. Cài đặt luồng với Windows

Một số hàm với luồng trong WIN32 API

- HANDLE CreateThread(. . .);
 - LPSECURITY_ATTRIBUTES lpThreadAttributes,
⇒Trở tới cấu trúc an ninh: thẻ trả về có thể được kế thừa?
 - DWORD dwStackSize,
⇒Kích thước ban đầu của stack cho luồng mới
 - LPTHREAD_START_ROUTINE lpStartAddress,
⇒Trở tới hàm được thực hiện bởi luồng mới
 - LPVOID lpParameter,
⇒Trở tới các biến được gửi tới luồng mới (tham số của hàm)
 - DWORD dwCreationFlags,
⇒Phương pháp tạo luồng
 - CREATE_SUSPENDED : Luồng ở trạng thái tạm ngưng
 - 0: Luồng được thực hiện ngay lập tức
 - LPDWORD lpThreadId
⇒Biến ghi nhận định danh luồng mới
- Kết quả trả về: Thẻ của luồng mới hoặc giá trị NULL nếu không tạo được luồng

Chương 2 Quản lý tiến trình

2. Luồng

2.3. Cài đặt luồng với Windows

Ví dụ

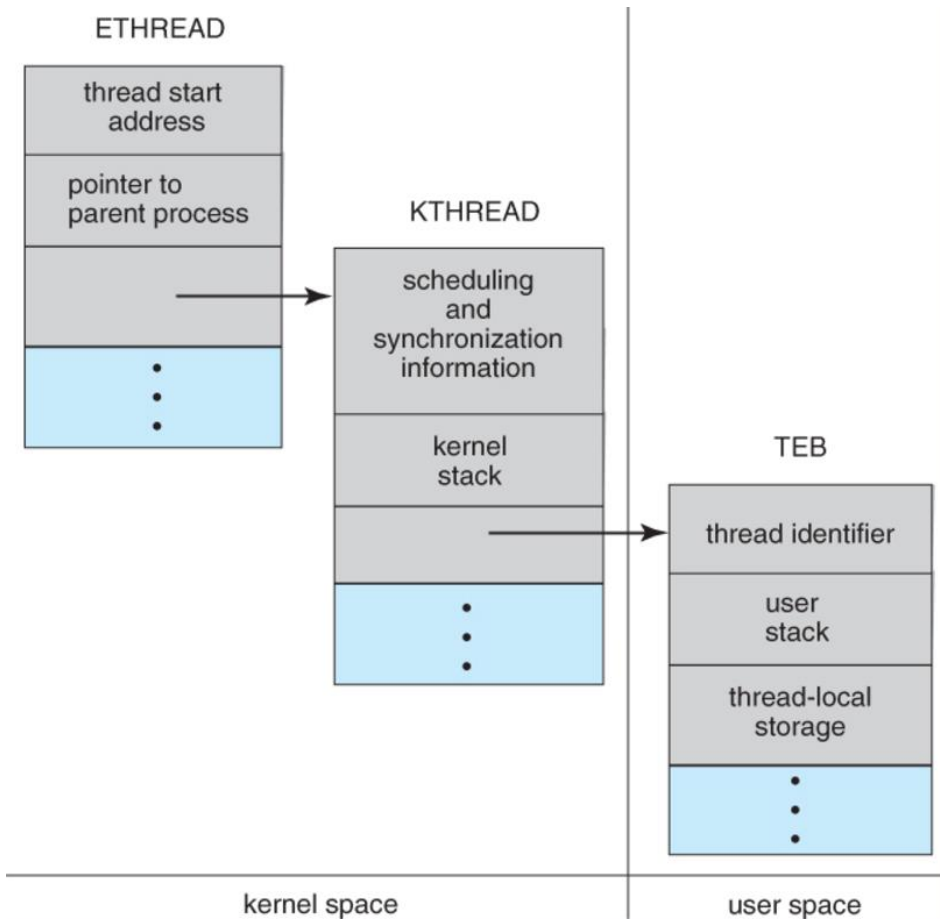
```
#include <windows.h>
#include <stdio.h>
void Routine(int *n){
printf("My argument is %d\n", *n);
}
int main(){
    int i, P[5]; DWORD Id;
    HANDLE hHandles[5];
    for (i=0;i < 5;i++) {
        P[i] = i;
        hHandles[i] = CreateThread(NULL,0,
            (LPTHREAD_START_ROUTINE)Routine,&P[i],0,&Id);
        printf("Thread %d was created\n",Id);
    }
    for (i=0;i < 5;i++) WaitForSingleObject(hHandles[i],INFINITE);
    return 0;
```

Chương 2 Quản lý tiến trình

2. Luồng

2.3. Cài đặt luồng với Windows

Luồng trong Windows XP



Thread bao gồm

- Thread ID
- Registers
- user stack dùng trong user mode, kernel stack dùng trong kernel mode.
- Vùng lưu trữ riêng được dùng bởi các thư viện khi thực hiện (run-time) và thư viện liên kết động

executive thread block

Kernel thread block

Thread environment block

Chương 2 Quản lý tiến trình

2. Luồng

2.4. Vấn đề đa luồng

- Giới thiệu
- Mô hình đa luồng
- Cài đặt luồng với Windows
- Vấn đề đa luồng

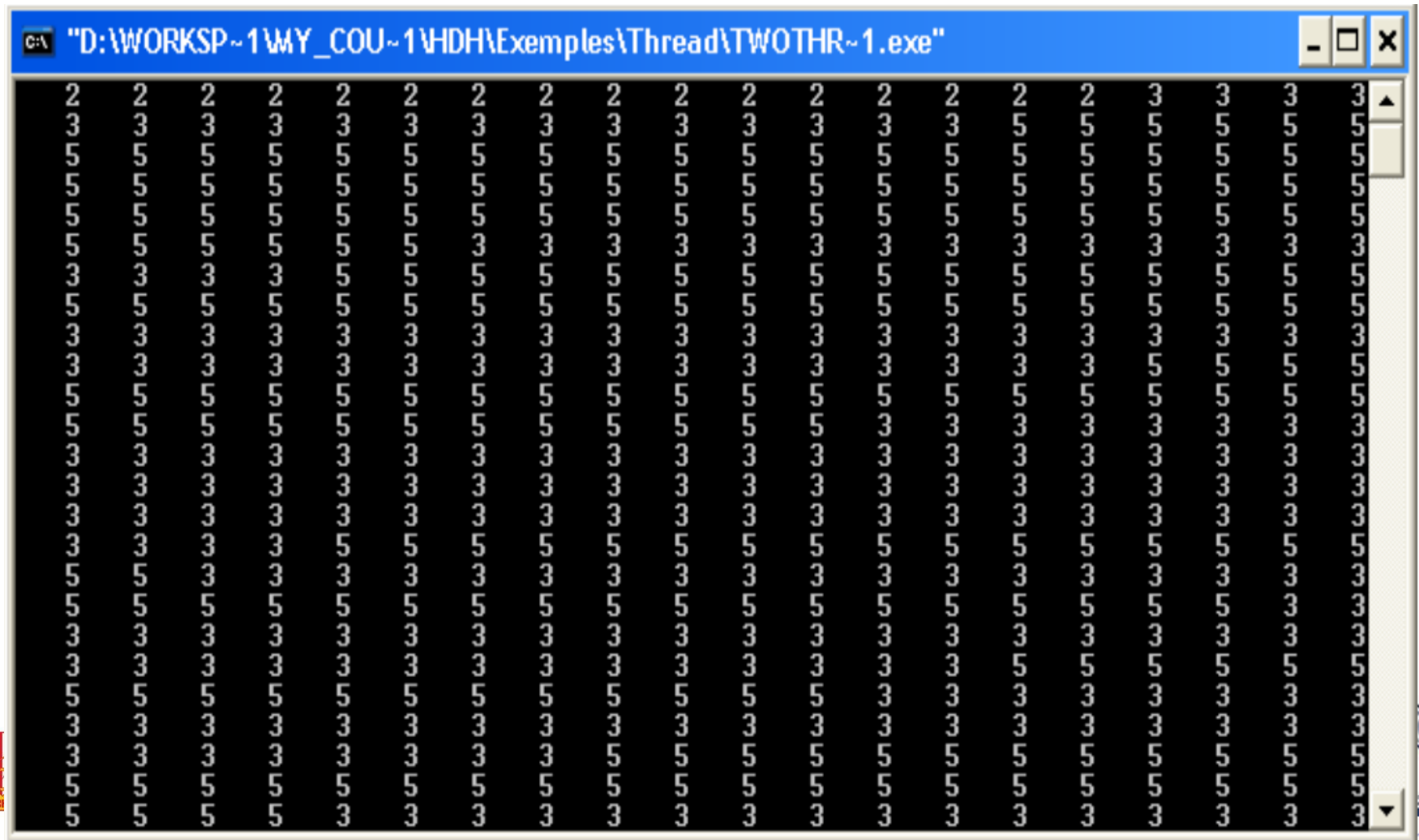
Chương 2 Quản lý tiến trình
2. Luồng
2.4. Vấn đề đa luồng

Ví dụ

```
#include <windows.h>
#include <stdio.h>
int x = 0, y = 1;
void T1(){
    while(1){ x = y + 1; printf("%4d", x); }
}
void T2(){
    while(1){ y = 2; y = y * 2; }
}
int main(){
    HANDLE h1, h2; DWORD Id;
    h1=CreateThread(NULL,0,(LPTHREAD_START_ROUTINE)T1,NULL,0,&Id);
    h2=CreateThread(NULL,0,(LPTHREAD_START_ROUTINE)T2,NULL,0,&Id);
    WaitForSingleObject(h1,INFINITE);
    WaitForSingleObject(h2,INFINITE);
    return 0;
```

2.4. Vấn đề đa luồng

Kết quả thực hiện



Chương 2 Quản lí tiến trình
2. Luồng
2.4. Vấn đề đa luồng

Giải thích

Shared int $y = 1$	
Thread T_1	Thread T_2
$x \leftarrow y + 1$	$y \leftarrow 2$
	$y \leftarrow y * 2$
$x = ?$	

Thread T1

$x = y + 1$ \longleftrightarrow

$x = y + 1$ \longleftrightarrow

$x = y + 1$ \longleftrightarrow

Thread T2

$\longleftrightarrow y = 2$

$\longleftrightarrow y = y * 2$

Kết quả thực hiện các luồng song song phụ thuộc trật tự truy nhập biến dùng chung giữa chúng

