

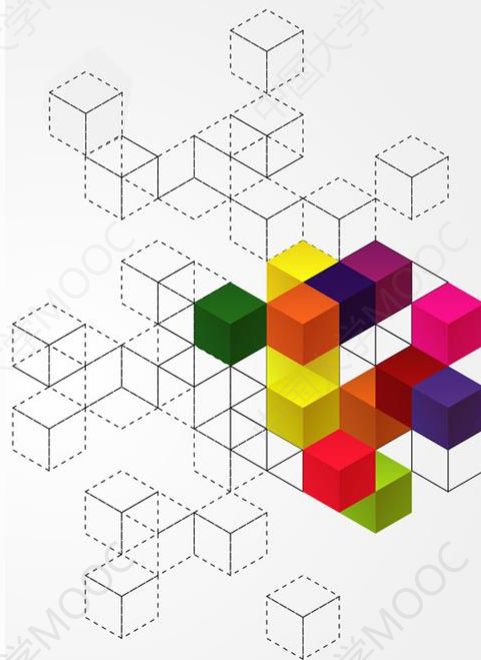
操作系统

Operating system

徐子川

大连理工大学

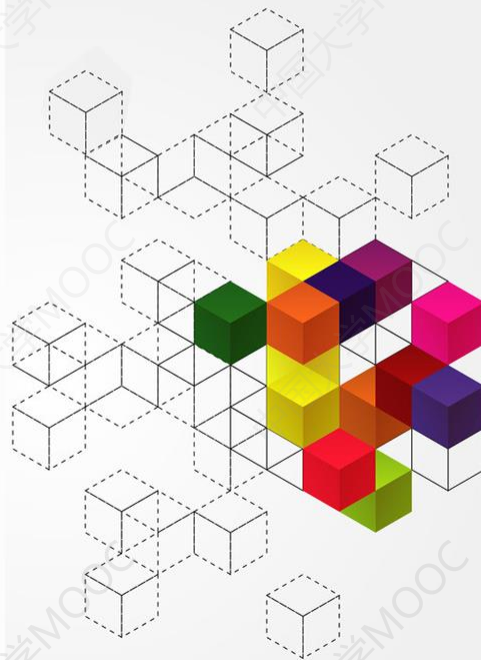
- 一、Windows线程编程API
- 二、Windows线程编程示例



一、Windows线程编程API

- 进程中的任意线程中都可以创建新线程
- 主线程在进程加载时自动创建
- 每个线程有自己的入口点函数
- 主线程的进入点函数

进入点	应用程序类型
WinMain	要求ANSI字符和字符串的GUI应用程序
wWinMain	要求Unicode字符和字符串的GUI应用程序
Main	要求ANSI字符或字符串的CLI应用程序
Wmain	要求Unicode字符和字符串的CLI应用程序

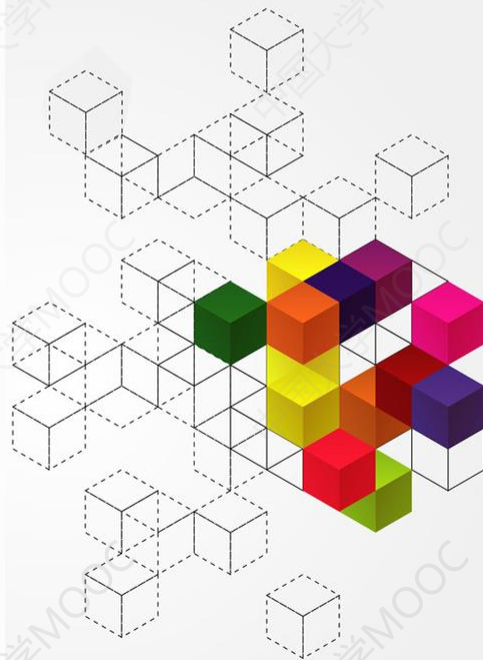


一、Windows线程编程API

- Windows线程入口点代码样式

```
DWORD WINAPI ThreadFunc(PVOID pvParam) {  
    DWORD dwResult = 0;  
    ...;  
    ...;  
    ...;  
    return dwResult;  
}
```

- 线程函数的返回值是线程的EXIT CODE
- 线程函数应尽可能使用函数传递的参数和局部变量

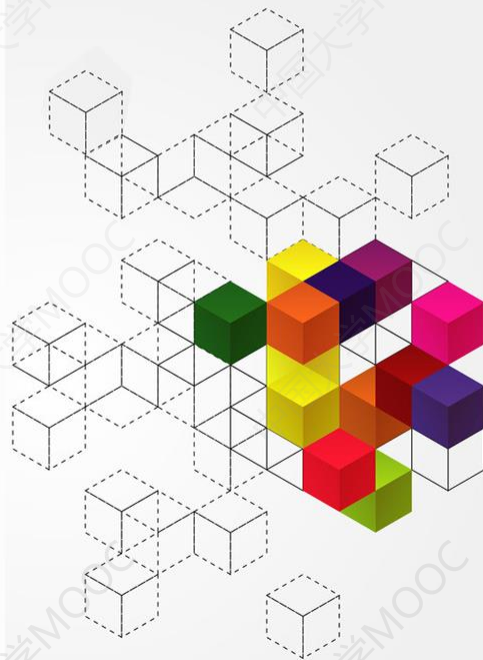


一、Windows线程编程API

- CreateThread: 创建线程的API函数

```
HANDLE CreateThread {  
    PSECURITY_ATTRIBUTES psa,  
    ...;  
    ...;  
    ...;  
    return dwResult;  
}
```

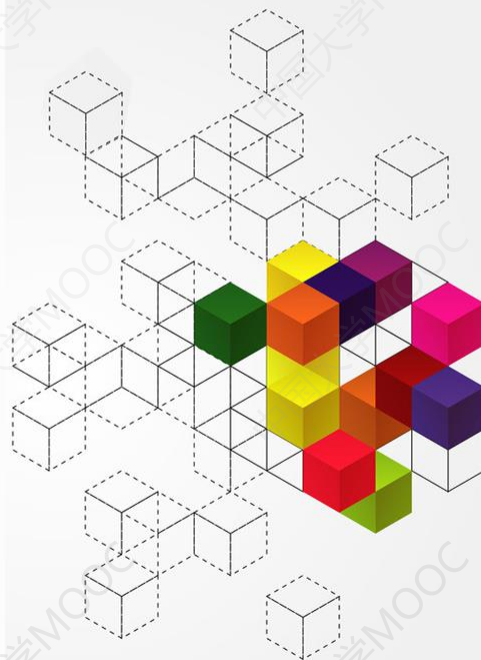
- 当调用CreateThread创建线程时，系统创建一个线程内核对象，该内核对象是用来管理进程的内核数据结构
- 在进程地址空间分配内存，供线程的堆栈使用



二、Windows线程编程示例

示例代码

```
#include <windows.h>
#include <iostream>
using namespace std;
DWORD WINAPI ThreadFunc(PVOID pvParam){
    cout << "create thread says 'Hello World!'" << endl;
    return 0;
}
int main(){
    HANDLE h = CreateThread(NULL,0,ThreadFunc,NULL,0,NULL);
    Sleep(100);
    cout << "Main thread" << endl;
    getchar();
    return 0;
}
```



本讲小结

- Windows线程编程API
- Windows线程编程示例

