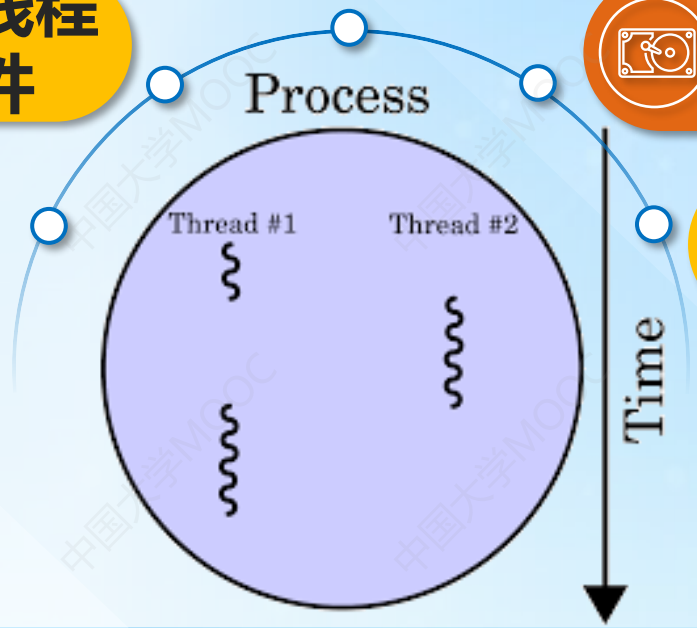
操作系统及Linux内核

西安邮电大学

线程及其模型

为什么引入线程?

- 提高应用程序的性能
- 同一进程内的线程共享内存和文件
- 使多处理机效率更高
- 占用较少的系统资源
- 改善程序的结构

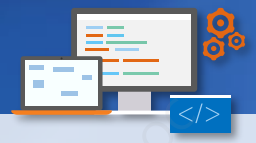


什么是线程?

线程（Thread）

是OS调度的最小单位，被包含在进程之中，一条线程指的是进程内一个执行单元。

- 轻量实体
- 调度的基本单位
- 可并发执行
- 共享进程资源



回顾：进程的两个基本属性

进程

既是一个拥有资源的独立单位，它可独立分配虚地址空间、主存和其它系统资源；

又是一个可独立调度和分派的基本单位。

进程和线程

代码

数据

进程空间

打开文件

寄存器

栈

线程1

寄存器

栈

线程2

寄存器

栈

线程3

寄存器

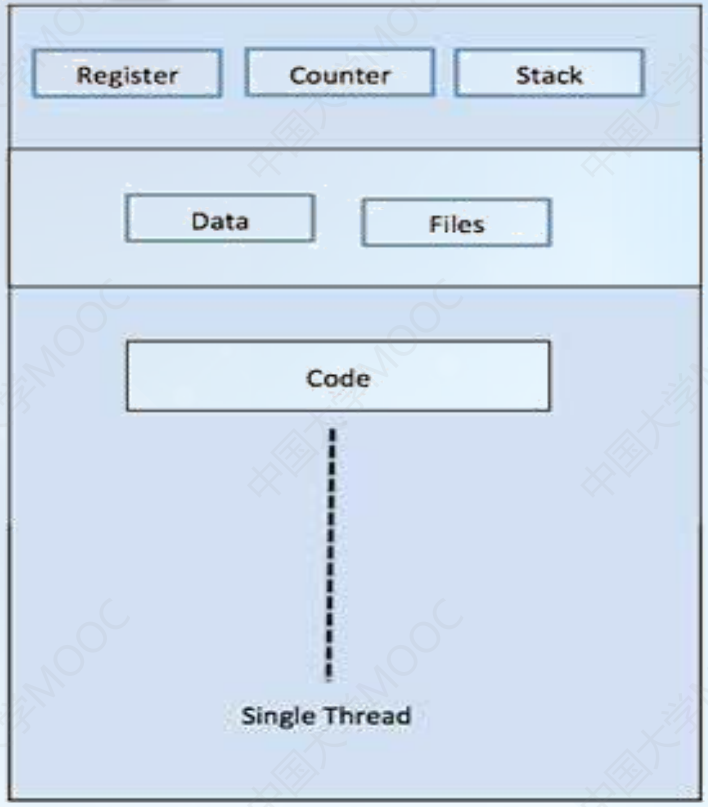
栈

线程4

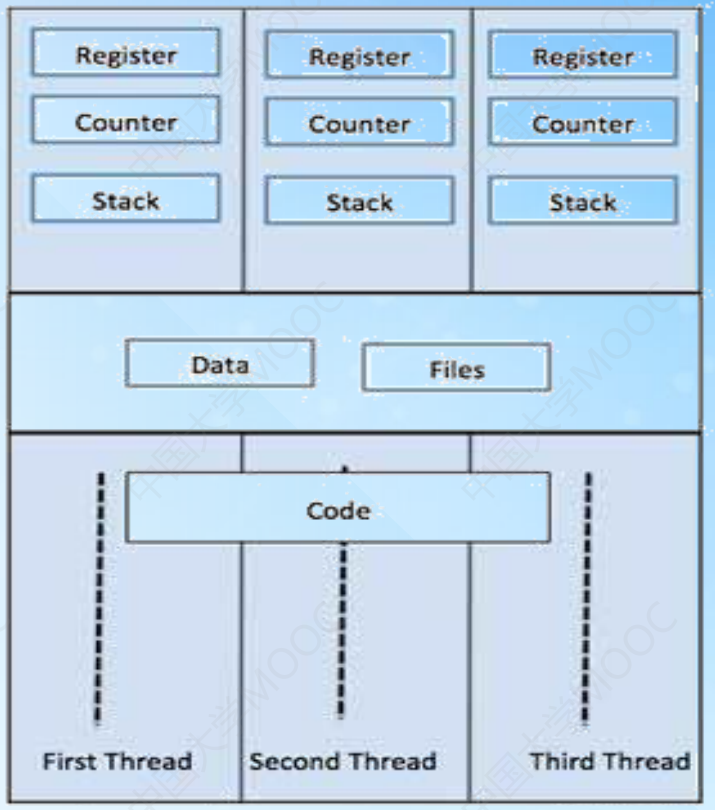
进程和线程的不同

	进程	线程
组成	程序+数据+PCB	函数+数据+TCB
基本单位	资源分配和独立调度的单位	仅仅是独立调度的单位
并发性	多个进程可并发执行，并发度低	多个线程可并发执行，并发度高
共享资源	每个进程拥有自己的资源	线程共享其所在进程的资源

单线程和多线程比较



Single Process P with single thread



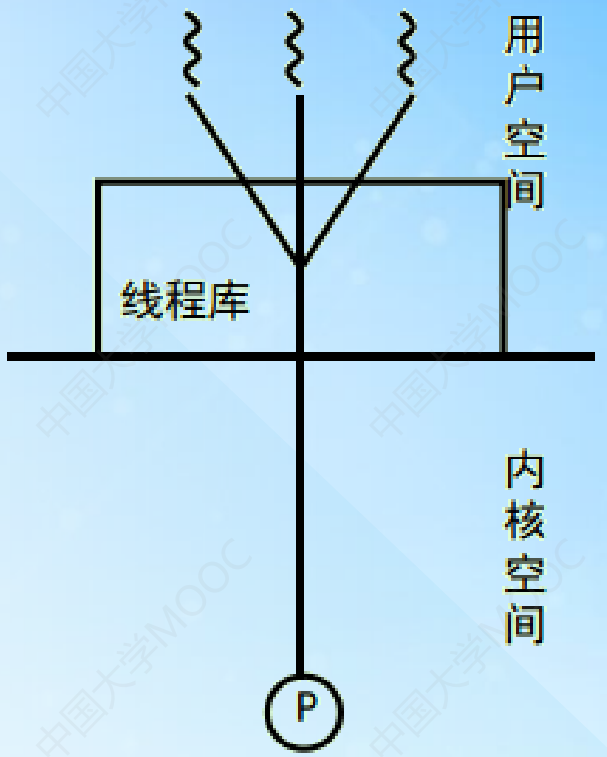
Single Process P with three threads

线程的实现

1. 用户级线程 — User-Level Threads（ULT）
2. 内核级线程 — Kernel Supported Threads（KST）
3. 两者结合的方法 — 既支持ULT，也支持KST

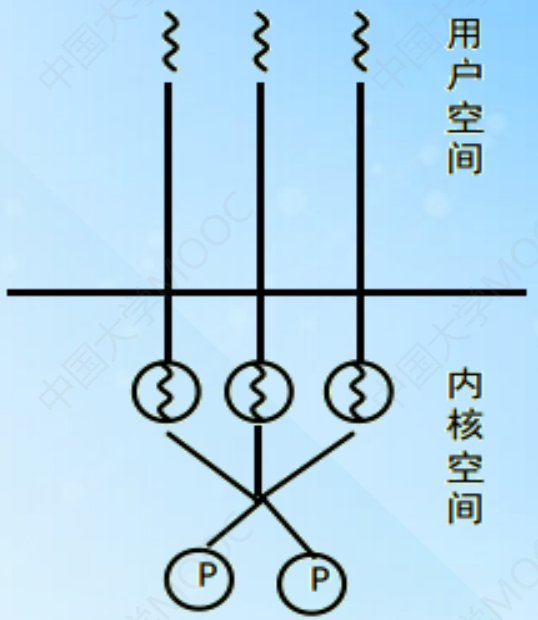
用户级线程

- 仅存在于用户空间中
- 由应用程序通过线程库完成所有线程的管理
- 内核不知道线程的存在，线程切换不需要内核特权
- 内核管理含线程的进程的活动，但不管理线程
- 当用户级线程调用系统调用时，整个进程阻塞

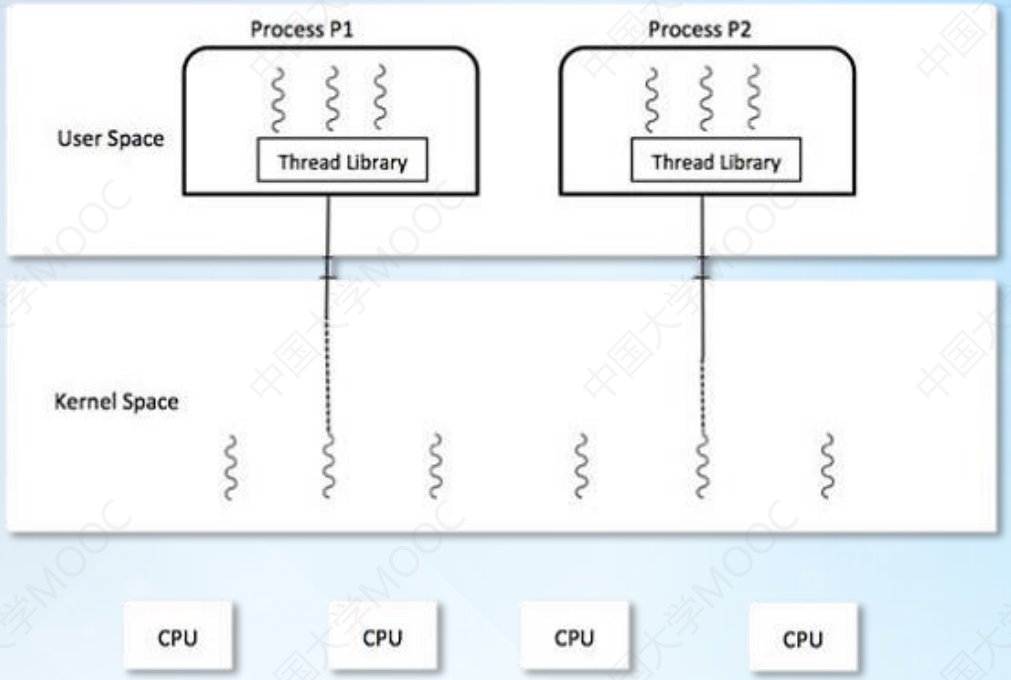


内核级线程

- 所有线程管理由内核完成
- 内核维护进程和线程的上下文
- 线程之间的切换需要内核支持
- 以线程为基础进行调度

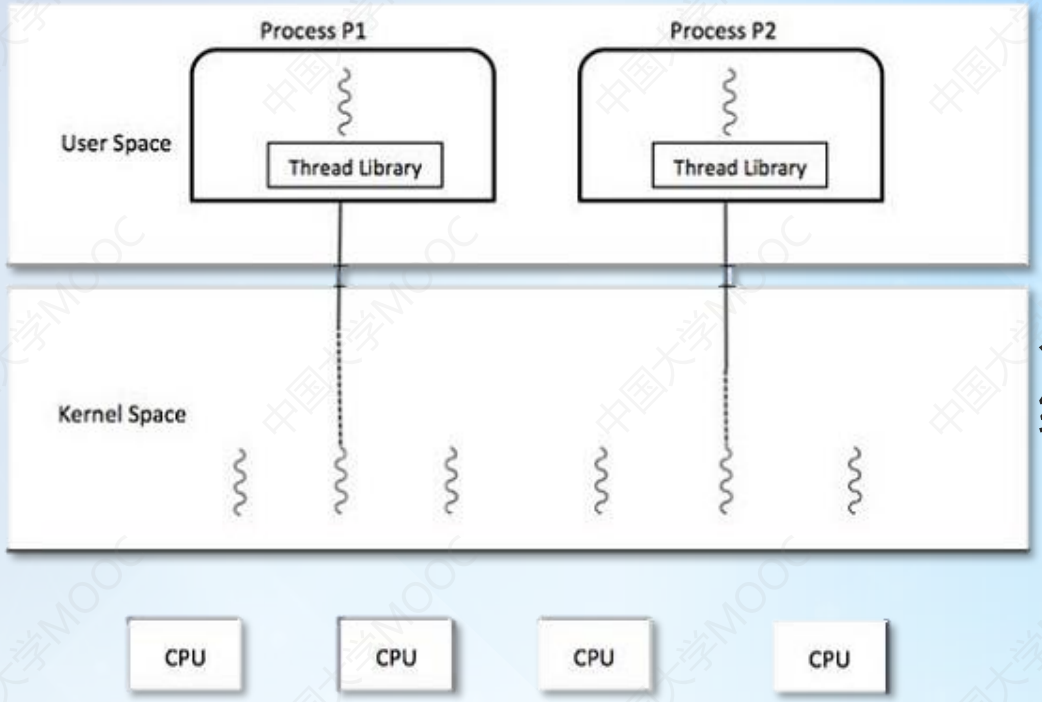


线程模型--多对一关系



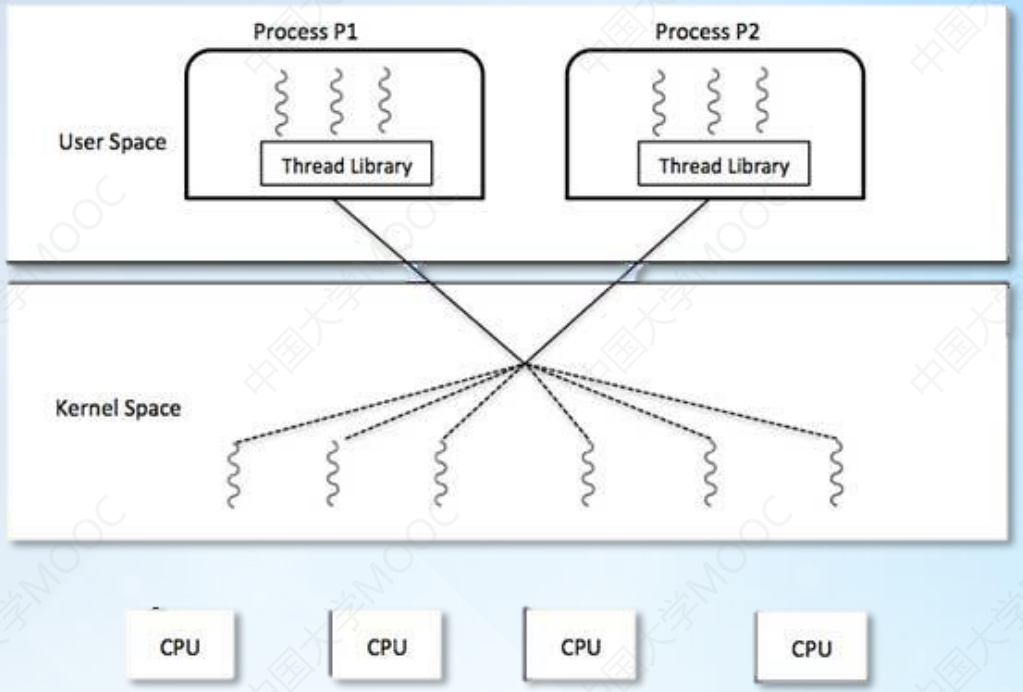
多对一模型将多个用户级线程映射到一个内核级线程。

线程模型--一对一关系



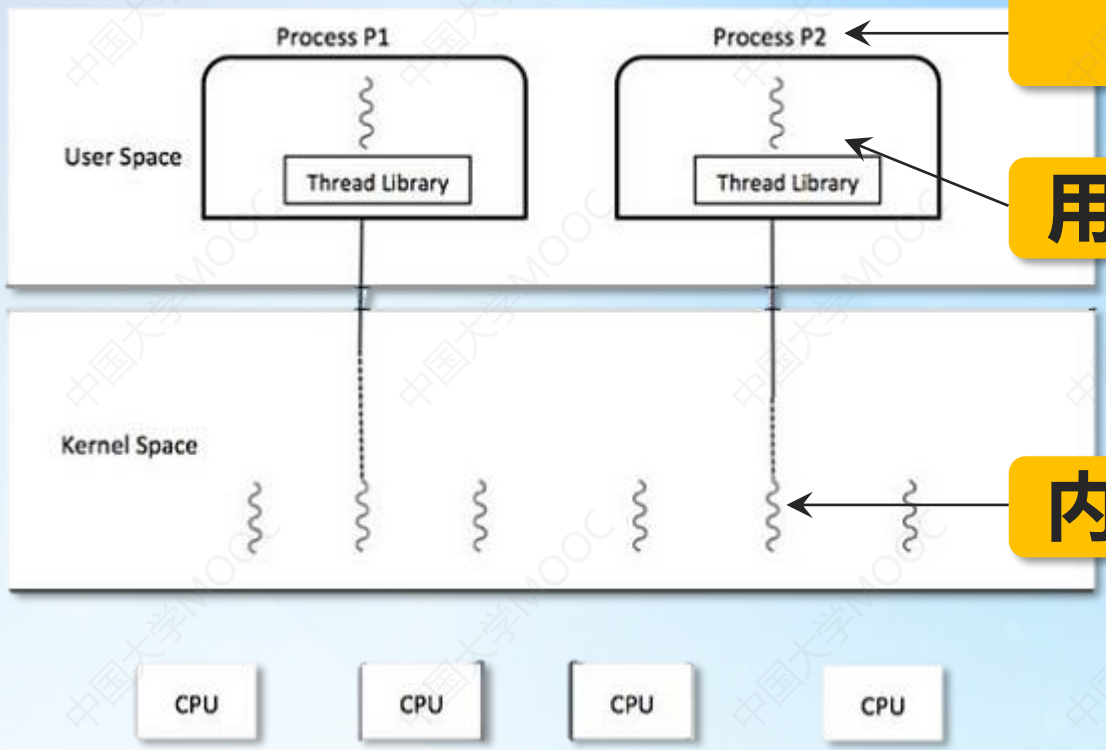
一对一的关系将一个用户级线程映射到一个内核级线程。

线程模型--多对多关系



多对多关系线程模型是任意数量N的用户线程到相等或者小于N的内核线程的多路复用。

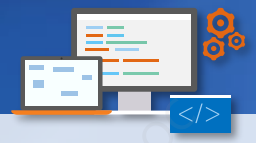
Linux中进程和线程标识符



进程pid

用户级线程utid

内核级线程ktid



Linux线程编程举例

```c
#include <string.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <sys/syscall.h>

#define gettid() syscall(__NR_gettid)

pthread_t ntid;

void *printids(void *s)
{
    pid_t      pid;      /* 进程号       */
    pid_t      ktid;     /* 内核级线程号 */
    pthread_t  utid;     /* 用户级线程号 */

    pid  = getpid();     /* 获取当前进程号   */
    ktid = gettid();     /* 获取内核级线程号 */
    utid = pthread_self();  /* 获取用户级线程号 */

    printf("%s pid %u  ktid %u  utid %u  (0x%x)\n", s, (unsigned int)pid,
           (unsigned int)ktid, (unsigned int)utid, (unsigned int)utid);

    pause();
}
```

Linux线程编程举例

```
26  int main(void)    // 主线程
27  {
28      /* 主线程main()调用pthread_create创建一个子线程 */
29      pthread_create(&ntid, NULL, printids, " new thread:");
30      /* 主线程main()调用printids()打印自己的线程号和所属的进程号 */
31      printids("main thread:");
32      sleep(1);
33
34      return 0;
35  }
```

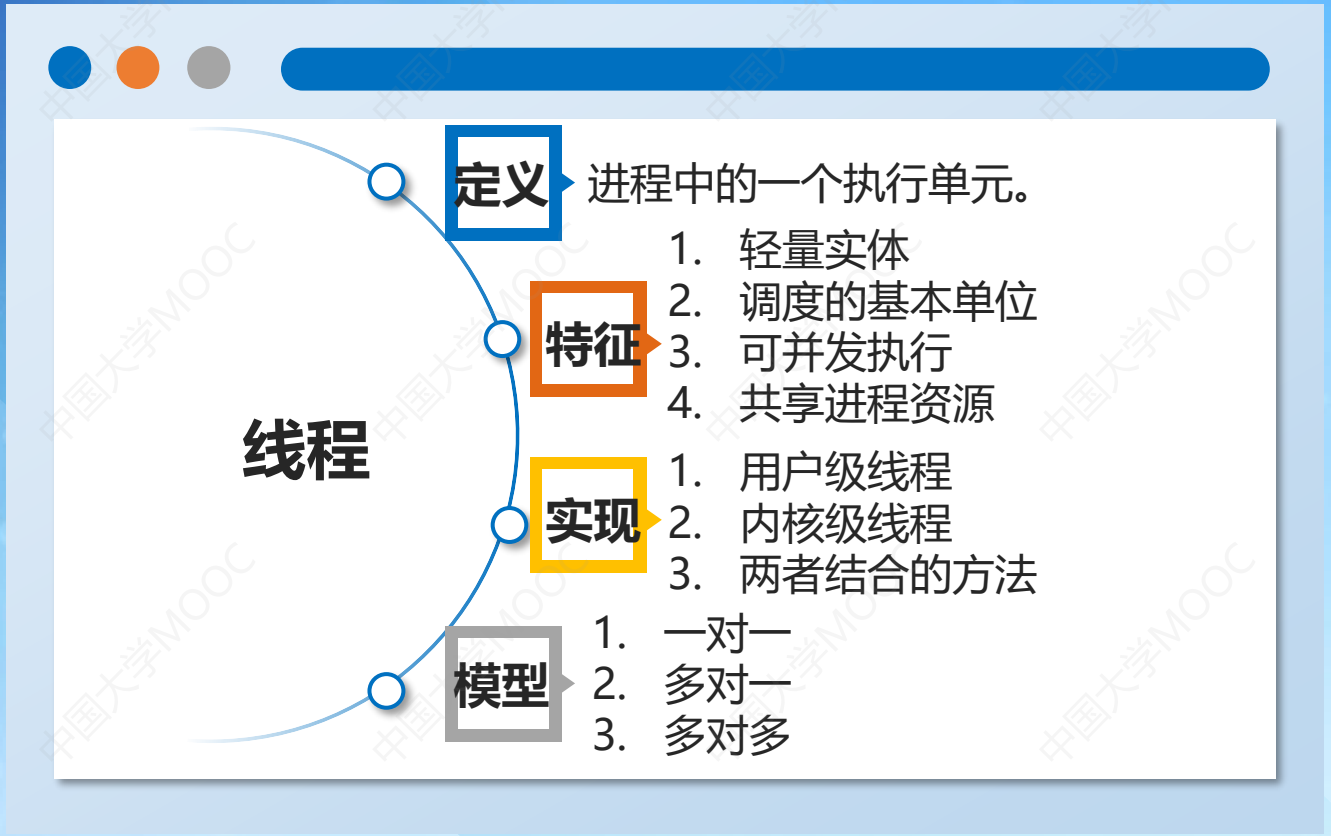
编译命令

```
gcc  -lpthread
thread_ex.c  -o
thread_ex
```

运行结果

```
[hds@localhost OS]$ ./thread_id
 new thread:  pid 5871   ktid 5872   utid 770463488  (0x2dec5700)
main thread:  pid 5871   ktid 5871   utid 778790720  (0x2e6b6740)
```

小结


线程
定义
进程中的一个执行单元。
特征
1. 轻量实体
2. 调度的基本单位
3. 可并发执行
4. 共享进程资源
实现
1. 用户级线程
2. 内核级线程
3. 两者结合的方法
模型
1. 一对一
2. 多对一
3. 多对多