Unix、Windows、Linux

典型操作系统中的IPC机制

下面呢，我们介绍一下典型操作系统当中的一个 IPC 机制

进程同步/通信实例

UNIX

管道、消息队列、共享内存、信号量、信号、套接字

Linux

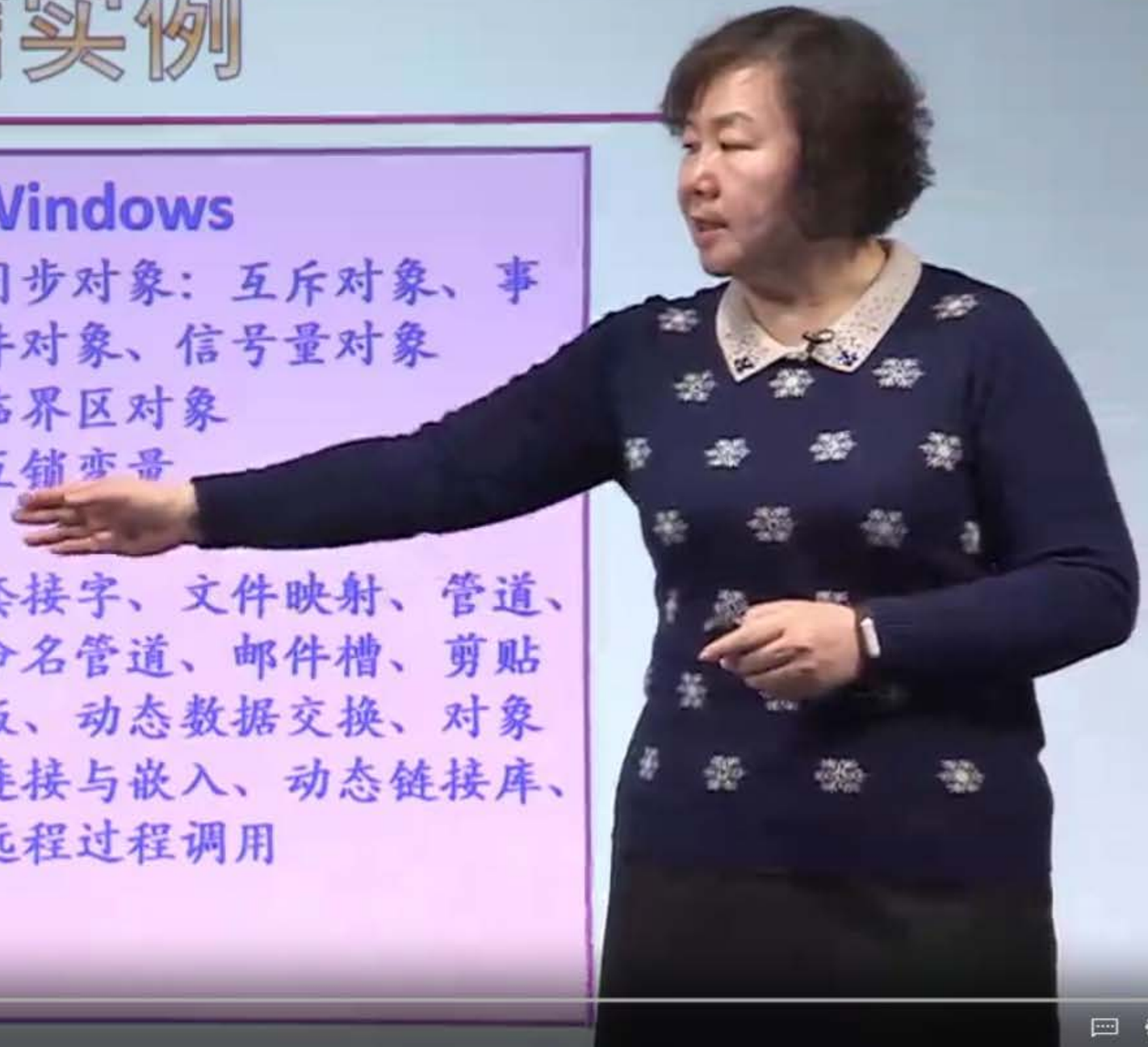
管道、消息队列、共享内存、信号量、信号、套接字

内核同步机制：原子操作、自旋锁、读写锁、信号量、屏障、BKL

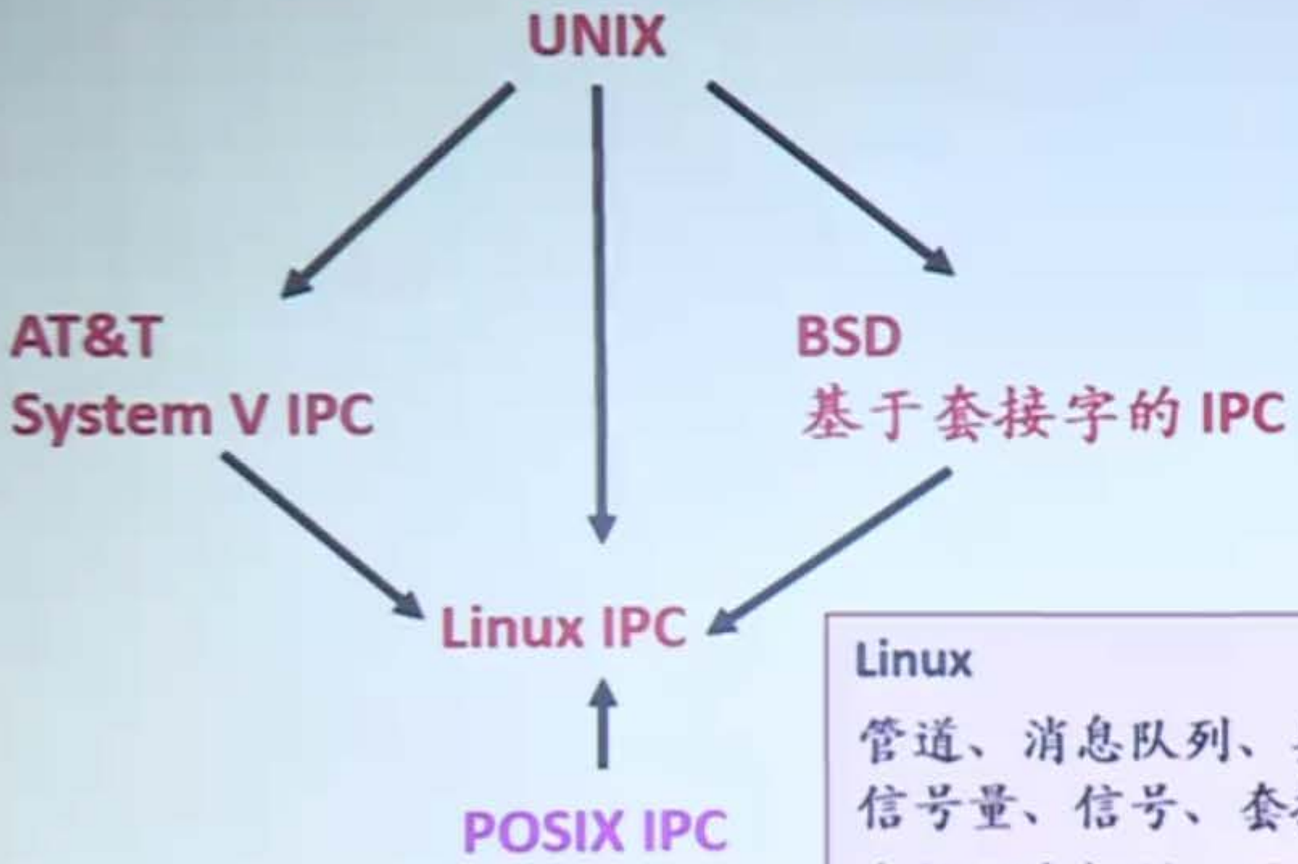
Windows

同步对象：互斥对象、事件对象、信号量对象
临界区对象
互锁变量

套接字、文件映射、管道、命名管道、邮件槽、剪贴板、动态数据交换、对象连接与嵌入、动态链接库、远程过程调用



LINUX的进程通信机制



Linux

管道、消息队列、共享内存、信号量、信号、套接字

内核同步机制：原子操作、自旋锁、读写锁、信号量、屏障……

LINUX内核同步机制

- 原子操作
- 自旋锁
- 读写自旋锁
- 信号量
- 读写信号量
- 互斥体
- 完成变量
- 顺序锁
- 屏障
- …

那我们这里头我们就介绍其中两种，一种呢是叫原子操作

原子操作(1)

- 不可分割，在执行完之前不会被其他任务或事件中断
- 常用于实现资源的引用计数
- atomic_t

```
typedef struct { volatile int counter; } atomic_t;
```

原子操作API包括：

```
atomic_read(atomic_t * v);
atomic_set(atomic_t * v, int i);
void atomic_add(int i, atomic_t *v);
int atomic_sub_and_test(int i, atomic_t *v);
void atomic_inc(atomic_t *v);
int atomic_add_return(int i, atomic_t *v);
```

原子操作(2)

- 例子

```
atomic_t   v = atomic_init(0);

atomic_set(&v, 4);
atomic_add(2, &v);
atomic_inc(&v);
printk("%d\n", atomic_read(&v);
```

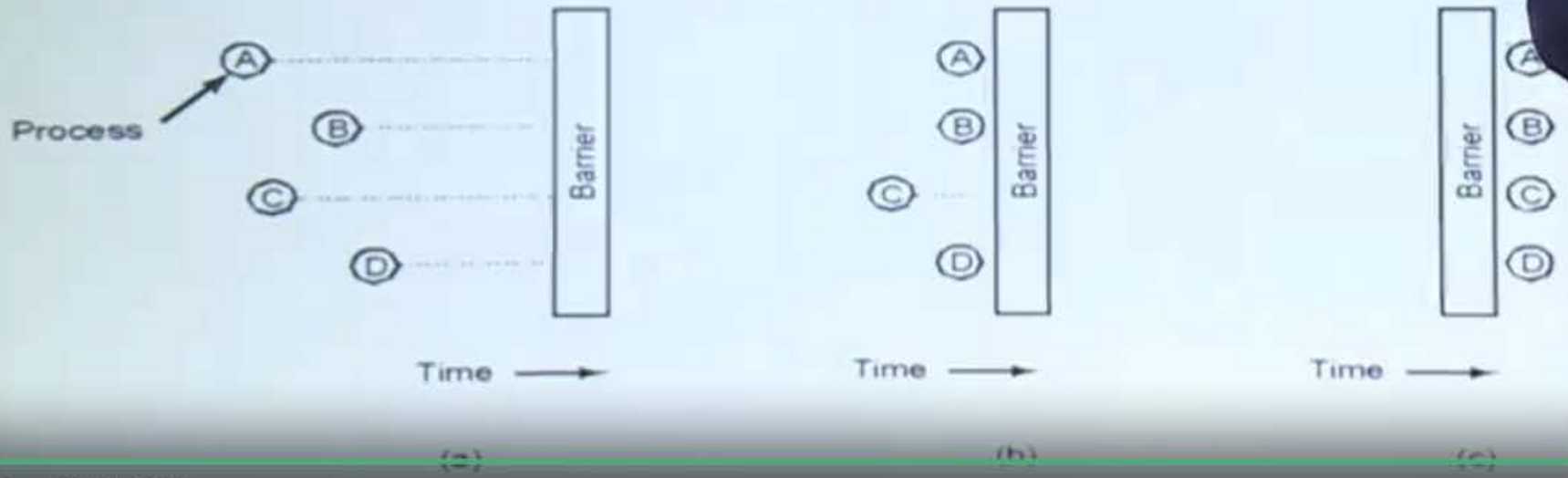
```
int atomic_dec_and_test(atomic_t  *v)
```

屏障(BARRIER)

矩阵运算

- 一种同步机制(又称栅栏、关卡)
- 用于对一组线程进行协调
- 应用场景

一组线程协同完成一项任务，需要所有线程都到达一个汇合点后再一起向前推进



本讲重点

- 管程
  - 如何保证互斥
  - 如何保证同步——条件变量及wait/signal
  - Hoare管程
  - MESA管程
- 进程间通信
  - 消息传递、共享内存、管道
- Pthread中的同步机制
- Linux的IPC机制

本周要求

- 重点阅读教材

  第2章相关内容：2.3.6~2.3.9

- 重点概念

管程　Hoare管程　Mesa管程
条件变量　wait/signal
Pthread中的互斥锁与条件变量
共享内存　消息传递　管道
Linux：原子操作　屏障　读写锁

0:00 下面呢，我们介绍一下典型操作系统当中的一个 IPC 机制 这里头呢，我们列举出来了 Unix、 Linux 还有 Windows 里头，它们都有哪些 同步机制，有哪些通信机制？ 那当然了，我们这里头重点呢，简单的啊挑这个 Linux 做一些介绍 Linux 的进程兼通信机制呢，它是 继承了 Unix 的相关的内容，就是 Unix 的 IPC 它继承了 同时呢，Linux 呢它还继承了 系统 5 的 IPC 机制，继承了 BSD 的基于套接字的的 IPC 机制，所以 Linux 呢，它继承了 Unix 系统 5 还有伯克利版本的 各种操作系统 IPC 机制的。 同时呢它也是基于 POSIX 的标准的 IPC 机制，那么设计了 Linux 的 IPC 机制 在此之后呢，它又增加了它自己的一些进程兼通信机制 就得到了现在这样一个结果 那我们来看一下说管道啊，消息队列呀，共享 内存啊，信号量，信号啊，还有套接字啊，那这些呢实际上是 可以用户程序可以使用的，也就换句话说用户程序 可以通过接口来完成这样 一些同步，互斥或者是通信等功能 那么在内核里头，操作系统内核里头有自己需要的同步控制机制 那么这里头包括了原子操作啊，自旋锁 读写锁，还有信号量，屏障等等。 那这个我们前面介绍过自旋锁的概念 那么介绍读者写者问题的时候呢，我们也介绍了读者写，读写锁啊读写锁 好，那么我们来看一下 在 Linux 的内核里头的同步机制，这里头有原子操作 自旋锁，有的时候呢还可以叫读写自旋锁 有信号量，有读写信号量，还有互斥体啊用的不同 然后呢有完成变量，有顺序锁，有屏障等等，这里还没有列全 那么之所以列这些是想说，第一，这些机制都是内核里头自己都要用到的机制 当然有些内容它会开放接口给用户程序用 那么它呢，大部分情况下自己去用这些机制 那么为什么这么多种呢？这就如同我们有一个工具箱 工具箱里头有各种各样的这种钳子，各种型号的 各种各样的这个十字改锥，那么螺丝刀，那么螺丝刀里头我们知道有 一字头儿的，十字头花子头儿的，那么还有大大小小不同的这个尺寸的 其实在 Linux 的内核里头也是一样，我们可以用一些比较 精准的这样的一些同步机制来完成 比较重要的工作，那么可能性能效率都会好 所以提供了很多种，根据需要挑选合适的一种 那我们这里头我们就介绍其中两种，一种呢是叫原子操作 原子操作呢，它的主要体现在这个操作呢是不可分割的 在这个操作没有完成之前不会被 其它的事件，任务被打断 那么这种原子操作通常会去用于对资源的引用技术，因为我们知道 有一个变量 I ，把这个变量加 1 ，比如说 I++ 其实这个 I++ 呢当然是一条语句，但是到了汇编这一层就是变成多条指令了 因此任何一个指令执行完了都会被中断 因此即便是 I++ 这么一个很简单的一条语句 也会被啊事件打断。 当然我们有了原子操作 那么对于这种引用技术就可以适用于这些场景

那这里头我们列举出来了各种各样的原子操作，比如说给一个变量加 加 1 啊，给一个变量加一个值啊等等等等，都可以去用它 下面我们来看一下原子操作的这个例子啊，我们声明了一个 原子啊，类型的这么一个变量 V，它的初值是 0 那么这里头呢，我们可以把这个 V 加 2 ，可以加 2 也可以直接啊加 1 啊，加 1 有这样一些操作啊，这样一些操作 那么下面呢我们再介绍 Linux 另外 一种机制，同步机制，叫做屏障啊，屏障 有的时候也叫栅栏，或者叫关卡。 那么这种机制呢主要是用于 对一组线程进行协调的。 那么什么情况下用这种机制呢？比如说有这么一个场景 有一组线程它们共同完成一项任务 那么要求所有的线程都到达了某一个汇合点之后 让它们再一同向前进。 可能有的线程先执行到了这个点，那它就得等着 然后等着其它的所有的线程 都到达了这个汇合点然后再从这个点，再往前出发 经常我们比如说，在矩阵运算当中，比如说我们要做 一个非常数量非常大的，规模非常大的一个矩阵运算 那么这个矩阵运算呢，我们可能要做很多次的迭代 那么在每一次迭代的过程中，由于一个矩阵很大 我们又把这个矩阵呢，划分成了很多很多的啊小矩阵 然后呢，让每一个线程去完成一个小矩阵的计算 那么每个，每一次迭代的话，每个小矩阵都 计算完了，那么就才能进入下一次迭代。 假定是这么一个场景 那么有的线程运行得快一点，有的线程运行得慢一点，但是 当这一次迭代，每个线程都完成了它们这个 小矩阵的计算，那么这一次迭代才能算全部完成 这种情况下呢，我们就可以用这种屏障的机制，来 把这些线程让它们达到这个一个汇合点之后再往前走，所以这个 第，某一次迭代实际上就是这个汇合点 那么这一次的迭代完成了，那么这个到了汇合点才能继续进行下一次迭代 那我们教材上有这么一张图，这张图非常形象啊，大家可以看一下 在某一个时刻，现在啊有这么 4 个进程 这 4 个进程呢我们可以看到 A、 B、 C、 D 有的跑得快一点，有的跑得慢一点 到了下一个时刻，我们可以看到 B 和 D 啊，都，还有 A 都已经到达了汇合点了，但是 C 还没有完成，所以呢 A、 B、 D 它们都在那等待，等待 C 到达汇合点 那么当 C 到达汇合点之后 它们 4 个进程再通过这个屏障再重新地继续执行啊，这就是屏障的作用 我们这一讲呢 主要包括的内容呢是管程。 大家要了解 如何来保证互斥，管程如何来解决同步问题 那么解决同步问题主要是条件变量，以及在条件变量上的 wait / signal，当然还可以有 modify 和 broadcast 操作 那我们的管程呢，主要介绍了两种管程 一种是 Hoare 管程，一种是 MESA 管程 这两种管程的语义是不一样的，那么希望大家能够了解

它们的语义 同时呢，还希望大家能够对 Hoare 管程和 MESA 管程怎么去应用有了解 怎么样用管程的方法来解决我们所碰到的一些问题 比如说生产者消费者问题，读者写者问题，还有其它的一些同步互斥问题 我们还介绍了进程间通信，那么进程间通信的话呢主要是 介绍了消息传递，共享内存和管道几种通信方式 那么对于这个线程库 Pthread 它的同步机制，我们也进行了介绍 这里头呢，在实现的时候，在用这个 Pthread 来实现一些进程之间同步互斥问题的时候 那么我们要注意它和用 Hoare 管程或者 MESA 管程所不同的地方 最后呢，我们简单地介绍了两种 Linux 的 IPC 机制 而 Linux 的还有很多的这种同步机制我们没有介绍 那么介绍了其中的，了解了啊，一个是自旋锁 读写锁还有这个信号量，还有我们刚才介绍的原子操作 还有这个我们说的屏障啊，这些呢都是大家可能会用到的 比较熟悉会用到的。本周的教材阅读呢就是这些啊 就这么几个内容，虽然非常少也就是没有几页，可能有 7、8 页的，但是内容非常多啊非常多 重点的概念呢，这里头列出来了，希望大家对这里头的重点概念一定要把它把握住 好，本讲的内容呢就到这里啊，谢谢大家！