* 1. 操作系统做了什么

大家好，今天我给大家带来的是操作系统原理课的第一讲 操作系统概述，在这一讲里头，我们将学习以下这些内容 首先我们通过一个程序的执行过程 来看一下，操作系统做了什么，然后我们介绍操作系统的定义和它的作用 操作系统是一个软件，那么相对于其它的软件 它有哪些特征呢？有哪些特点呢 另外我们介绍一些典型的操作系统架构 通过介绍Windows、 Unix和Linux操作系统的架构 让我们有一个感性的认识，看一看操作系统都有哪些功能 它们之间有哪些关联，最后我们介绍一下操作系统的分类 首先我们先看一下 操作系统做了什么，这是一个 简单的C语言程序helloworld 我们想通过这个程序的执行过程，来看一下操作系统 对这个程序的执行过程有哪些支持，我们先来看一下程序 这个程序里头主要功能是调用了一个puts函数 它的作用是在标准的输出设备上，也就是我们说的显示器上 显示字符串"hello world" 好下面我们来看一下这个程序的执行过程 首先我们要启动程序执行 用户要告诉操作系统，执行helloworld程序 大家可以想一想，有哪些方式 可以告诉操作系统要执行某个程序 比如说，我们可以在命令行键入相应的命令 我们也可以通过鼠标双击helloworld程序的图标 通常告诉操作系统执行一个程序的方法 不止一个，操作系统接到了用户的请求之后 就会根据用户提供的文件名 到磁盘上找到这个程序的相关信息 找到信息之后，会去检查这个程序 是不是一个可执行文件，因为有的时候我们 的一个文件它不是可执行文件，那么操作系统就不能执行这个文件 就会去报错，那么如果是一个可执行文件，操作系统才能正确地执行 在检查完类型之后，操作系统会根据程序首部信息 来确定代码和数据在这个可执行文件当中的位置 并计算出它相应的磁盘块地址 关于可执行文件，有哪些 常见的格式呢，大家可以去查一下相关的资料 那么我们知道Windows操作系统下，可执行文件的格式呢是PE格式 在Linux操作系统下，可执行文件的格式呢是ELF格式 ELF，大家可以去自行了解一下这方面的知识 为了执行这个helloworld程序 操作系统首先要创建一个新的进程 并将helloworld程序的可执行文件格式 映射到该进程结构 表示由该进程来执行这个helloworld程序 做完了这件事情，操作系统就把控制权 交给了调度程序，我们假设调度程序 正好选中了helloworld程序，那么 由操作系统为这个helloworld程序设置CPU 上下文环境，并跳到程序的开始之处 准备执行这条这个程序，那么下一个指令周期就是执行helloworld程序了 helloworld程序的 执行的时候，当执行第一条指令的时候 会发生什么事情呢？实际上就是发生了 缺页异常，那么为什么呢？ 因为我们知道，程序在执行的时候，先要把程序的代码和数据读入内存 CPU才能去执行，但是helloworld程序的代码和数据还没有读入内存呢 所以这个时候呢，硬件机制就会 捕获出缺页异常，并且 把控制权交给操作系统 操作系统管理了计算机系统中的 内存，有很多的内存，比如说，我们说 有页式存储管理方案的话，那么内存就有很多很多的物理页面 操作系统的内存管理模块，就会分配一页空闲的物理内存 并且根据前面计算出的磁盘块地址 把helloworld程序的代码读入内存 从磁盘读入内存，然后 继续执行helloworld程序，有的时候程序很大 一页内存还不够，因此在这个执行过程中会多次地产生缺页异常 然后去到磁盘读入程序 到内存，这个过程会有多次 下面helloworld程序执行puts函数 那么puts函数的作用，是在标准的输出设备上 显示字符串，那么这个标准输出设备，通常就是我们的显示器或者说屏幕 那么puts函数呢，它实际上呢是一个系统调用 是由操作系统来完成这个功能 所以控制权又转回到操作系统 操作系统找到要将字符串送到哪一个显示设备 那么通常这个设备呢是由一个进程控制的，所以操作系统 把要写的字符串实际上是送给了这个进程 那么控制设备的进程，会告诉设备的窗口系统，它要显示字符串 窗口系统确定这是一个合法的操作，然后将字符串转换为像素 把像素写入设备的存储映像区 以下就是视频硬件的工作 视频硬件将像素，转换成显示器可以接收的一组控制信号或数据信号 那么显示器再去解释这个信号，激发液晶屏 其结果，OK，我们在 屏幕上就看到了字符串"hello world" 那么以上就是一个，非常简单的程序的一个执行过程 从启动到执行，到结束 从这个过程当中，我们得到什么结论呢？ 我们看到，在CPU上执行一个用户程序 这个用户程序会不时地去请求操作系统的服务 因此在CPU上，时而运行的是用户程序 时而呢，是操作系统程序在运行，我们来看一下 helloworld程序，在执行的时候呢，需要操作系统来创建进程 因此呢转到操作系统创建进程，进程创建完了以后呢，那么从操作系统返回 那么接着执行这个helloworld程序 那么出现了缺页异常以后呢，控制权又转回了操作系统 操作系统呢分配物理内存，给这个用户程序 然后呢又转向用户程序接着执行 然后helloworld程序又要去执行puts函数，又转向了操作系统 那么这是从用户程序，在执行 过程中，不断请求操作系统服务这个角度来看 程序执行的过程，如果我们从另外一个角度来看用户程序的执行呢？ 实际上呢，是操作系统 在执行过程中呢，如果选中了一个程序，那么 就去通过调度选中这个程序，去执行这个程序 程序执行过程中呢，会不断地去陷入操作系统 由操作系统完成一些服务，然后再通过调度再选中程序，接着执行 实际上是这样一个过程，那么 从上面的介绍过程，我们看到在一个程序的 执行过程中，操作系统负责了这个程序的启动过程 负责了这个程序执行的过程，同时在执行的 过程中，不断地去为用户程序的执行 提供各种各样的支持

1.2 操作系统的定义和作用

下面我们来介绍一下操作系统的定义及作用 那么操作系统是什么？怎样理解它的作用？ 操作系统是什么呢？ 操作系统是计算机系统中的一个系统软件 这个系统软件呢是一些程序模块儿的结合 这些程序模块相互协作、 相互配合来完成以下这些功能 首先，它们能以尽量 有效、 合理的方式组织和管理计算机的 各种软硬件资源。 为了 支持程序的执行，操作系统 要合理地组织计算机的工作流程，控制程序的执行 并在程序执行的过程中 向程序提供各种各样的服务功能。 第三 操作系统必须采用各种各样的方式使得用户能够非常灵活、 方便地使用计算机 采用各种各样的技术使整个的计算机系统能够高效率地运行 这就是操作系统的一个定义 那么操作系统完成各种各样的功能 我们解读一下儿在这个定义当中有些关键的 关键的词汇 第一个词是有效 那么操作系统有管理各种各样的资源 那么这些资源像 CPU，那我们总是希望 CPU 不空闲，能够充分利用 CPU，有很多很多的程序在 CPU 上跑 那么还有内存、 磁盘，我们希望这些内存、 磁盘呢 应该能够充分利用这些空间，能够得到很好的管理 比如说各种各样的设备，我们希望这些设备能够充分地忙碌起来 总而言之，我们希望通过 操作系统能够把各种各样的资源的利用率提高 以提高整个的操作系统，整个的计算机系统的效率 第二个关键词是合理 因为要管理资源，所以呢各种各样的策略 就会有各种各样的策略。 那么这些管理资源的这种策略是不是公平，是不是合理呢？ 那么如果不公平、 不合理，则会产生什么样的问题呢？ 我们呢也要解决这样的问题。 第三个关键词是方便使用 我们要和操作系统打交道 通常希望能够非常容易地使用 操作系统提供给我们的各种接口。 两个典型的接口是 一个是用户界面，也就是通过命令行，用户和操作系统打交道 还有一个呢是编程接口，也就是程序员 通过开发软件，在软件中与操作系统打交道 那么这两种接口都希望能够容易使用、 容易学习 操作系统的作用是什么？ 我们总结出典型的操作系统的三个作用 第一个作用，操作系统是资源的管理者 操作系统是资源的管理者是从什么角度来看呢？ 是从自底向上，也就是从操作系统内部的角度来看 操作系统是资源的管理者。 哪些是操作系统该管的资源呢？ 这都是我们熟悉的硬件资源和软件资源 CPU, 内存 各种各样的设备，包括我们说的 I/O 设备，打印机啊 扫描仪啊，磁盘、 时钟 还有像网卡等等。 那么还有 很多软件资源，比如说磁盘上有很多很多的文件 有很多很多的管理信息也是操作系统所管理的 所以操作系统是资源的管理者。 从这样一个角度出发 我们来看一下怎么去管理各种各样的资源 操作系统为了管理资源，首先 需要跟踪记录每一种资源的使用状况 那么这就是通过数据结构 一个软件有数据结构、 算法 这些成分，所以数据结构是用于记录资源的使用情况 那么 一个数据结构都记录哪些信息呢？我们可以记录哪些资源是空闲的 哪些资源已经分配出去了，分配给谁使用了 并且允许使用多长时间等等这些信息 有了数据结构之后 我们还需要算法，也就是说，操作系统要确定资源分配策略 操作系统要决定把资源分给哪一个进程用 给它用多久，给它用 用多少，这都是所谓资源分配策略 那么资源分配策略通常呢有几个典型的 一个是静态分配策略。 所谓静态分配策略就是说一个程序在执行前 就把它所需要的各种资源拿到手 那么这个程序在执行过程中就不再需要新的资源了 就可以一直执行完成。 当然这里头有一个问题 这个资源在执行过程中有的时候不需要用资源的时候呢，这个资源就被 浪费了。 因为它拿着资源，别的进程不能用 因此，静态分配策略会导致整个资源的浪费 因此，很多的操作系统，或者说绝大多数操作系统都采用的是一个动态分配策略 那么动态分配策略就是说在一个 进程执行的过程中，什么时候需要资源 什么时候提出申请。 那么操作系统就会根据目前系统中资源的 使用状况，也就是刚才所说的数据结构 来决定这个资源是否分配给它，是否分配给它 那么这样的好处就是，资源的利用率得到了充分的提高 充分的提高。 我们举一个典型的例子 比如说我们是一个页式的存储管理系统 那么内存中有，我们把内存，物理内存划分出很多的物理页面 那我们可以用一个非常简单的数据结构位示图，或者位图 bitmap 来记录这个物理内存的使用情况 一位对应了物理内存的一页 如果这一页是空闲的，那么我们这里头就记为 0 如果这一位是分出去了，那我们就记为 1 那么我们的分配算法就实际上就是在这个数据结构上来执行 那么下面就要去实施资源的分配和回收 那么分配的过程实际上就是 按照策略，按照刚才制定的这种资源分配策略选中 要给分配资源的这个进程，然后去决定分配给它多少 然后去修改相应的数据结构。 比如说刚才，那么如果我们把一页 内存分配给了一个进程，那么这个位示图就会把这个原来的 0 改成 1 而回收，当一个进程使用完资源之后，那么就要回收，系统要把它回收 那么回收的时候呢，也是要修改相应的数据结构，比如说就把原来的 1 换成 0 这就是实施资源分配的和回收的过程 除此之外，操作系统 还要采用各种各样的技术来提高资源的利用率 当有多个进程都要使用资源的时候呢，还要保护资源 那么一个进程使用资源的过程不被另外一个进程所干扰 如果在一个系统中有多个进程 都对资源有申请，那么产生了冲突的情况下，需要操作系统来协调 这就是操作系统怎样管理一个资源的一个一般性的过程 那么我们以后所有的资源其实都符合 这样的一般性的过程，当然有一些特殊的问题需要解决 从资源管理的角度 那么我们来看，操作系统呢总结出操作系统的五大基本功能 进程/线程管理，有的时候也称为 CPU 管理，因为它们进程和线程是在 CPU 上执行 这里包括了进程线程的状态设计、 控制操作 同步互斥、 通信还有调度 存储管理主要是做 存储空间的分配和回收 完成地址的转换，还有存储保护、 内存扩充 文件管理，那么当然是 主要包括了文件目录管理，文件的各种各样的操作 磁盘空间的管理，还有文件的存取控制管理 设备管理呢，主要在设备的驱动程序 设备的分配、 回收过程 还有为了提高设备的各种各样的资源利用率呢采用的各种技术，比如说缓冲技术等等 那么一个操作系统在使用过程中，必然提供 给用户非常方便使用的接口，因此呢，作为操作系统呢我们要讨论 接口的，典型的接口，系统命令，还有编程接口 操作系统的第二个作用呢 是向用户提供各种服务。 那么这些服务呢是方便用户使用的 那么我们可以说 操作系统是各种系统服务的一个提供者 那么这个系统服务呢，如果操作系统 提供功能更强大，更容易使用的这样的一系列的命令或系统调用 我们就认为这个操作系统提供了更好的服务。 那有哪些典型的服务呢？ 我们来看一下，典型的像进程的创建呀 执行啊，文件和目录的各种操作呀 各种 I/O 设备的使用啊，还有提供各种各样的统计信息 这里简单地举了几个例子，大家有那么一个感性认识 操作系统的第三个作用呢是对机器硬件的一个扩展 具有一个扩展能力。 我们来看一个 示意。 那么我们在这里头我们知道操作系统 实际上是硬件基础上的第一层软件 那么操作系统完成各种各样的功能 在操作系统之上是应用程序在执行 操作系统为应用程序的执行提供了，像我们说的 各种各样的接口，用户的接口和编程的接口 那么这些就组成了一个虚拟机器界面。 操作系统之下呢 是硬件，也就是操作系统屏蔽了一些硬件的细节 用户程序不需要直接和硬件打交道 通常它都是通过操作系统来控制这些硬件的 执行，那么这是我们说操作系统是硬件基础上的第一层软件 那么为什么要把，用操作系统把硬件屏蔽呢？ 那我们来看一下与硬件相关的一些工作 是非常复杂和繁琐的。 我们有两个操作 我们现在介绍两个操作，那么想问一下，这两个操作当中的哪一个操作 你觉得简单，你觉得很方便 第一个操作呢是说 从一个文件当中读一个数据块 第二个操作呢是说你要移动磁头，要等待这个磁头放下 那么我们对程序员来讲，第一个操作 显然要比第二个操作更简单，更方便使用 那我们来看看如果你要考虑第二个操作，你要实现第二个操作 那你要考虑哪些问题，是不是很复杂，是不是很琐碎？ 我们举一个非常 简单的例子，软盘 I/O 操作 虽然软盘呢我们现在都很难见到了 但是因为软盘和硬盘很相似，但是它又非常简单，所以我们就举 它作为例子。 那么硬盘比它要复杂得多，所以我们更多的问题需要去解决 有这么一个非常简单的软盘控制器 那么这个软盘控制器呢可以执行 16 条命令 这 16 条命令呢实际上的 长度是不一样的，是从 1 到 9 个字节的，长度不等 而这些命令包括了什么呢？我们简单来看一下，比如说读写数据 然后移动磁头臂，格式化磁道 还有些初始化呀，检测软盘的状态呀，复位呀等等等等 那么这些呢都是非常细节的一些命令，我们很难记住 那么这些命令呢要把它写入到 设备的控制寄存器里头去来控制这个设备的执行 我们以这个 read 命令为例，read 命令呢 需要 13 个参数，那我们还要记住 13 个参数 每个参数呢，像 read 命令的 13 个参数呢包括了你要读取的磁盘地址 磁盘的扇区数，磁道的扇区数，还有很多跟 物理介质相关的一些物理特性，软盘的一些物理特性，你都要了解 因此对于程序员来讲就很不容易来知道每一个细节 另外呢，这个软盘 工作的是不是正常，是不是完成了你的这种控制要求？ 因此这个软盘的这个控制芯片呢还会返回给 23 个状态或者出错信息，这些信息呢 又记录在了 7 个字节的地方，7 个字节的设备 状态寄存器里头。 那么还要了解这 23 个 不同的状态，什么样的情况，发生了什么事情 除此之外，还有一个很特殊的 就是程序员还要保持步进电机的一个开关状态 那么我们知道软盘不能总是在 启动的状态，那么时间久了，软盘就会损坏 因此呢，如果长时间不用，它就会停止 但是一旦要用，读数据了，要写数据了，要启动这个软盘 让它旋转到一定的转速那才能去读 因此呢，程序员要注意这样一些步进开关的这种 步进电机一个开关的一些状态，还要考虑这些细节 从以上介绍其实我们已经看出来了，说如果让我们来 编程的时候，考虑这么琐碎、 复杂的事情，那么程序员呢会非常非常头疼 所以我们说与硬件相关的这些工作特别地琐碎 特别地繁杂，而且不同的硬件呢这些还不一样。 如果我们换了一个 驱动器的，驱动器，那么这个命令就可能是另外的样子，每条命令就不一样，参数也是不一样的 因此，操作系统就做了这样一个事情 操作系统就是在硬件之上 在应用程序与硬件之间 建立了这么一个等价的一个扩展机器，虚机器 虚拟机，那么通过对硬件的抽象 那么提高了整个系统的可移植性 那么对于程序员来讲呢，他去对底层硬件编程 更麻烦，而提供了这样一个操作系统这一层之后，那么 程序员比对这个编程就会比底层硬件的这种更容易 那么我们举一个例子来 描述一下这个图所表达的含义 我们通常我们买，可以买一套房子 那么这套房子呢我们是一个毛坯房，毛坯房，那么你去看房子的时候一看这个墙 非常地难看，凸凹不平，都是那个砖嘛，凸凹不平 那怎么办呢？那这时候呢我们就需要装修 那么装修当然就是说先用腻子把墙抹平，然后再刷上 几遍漆，这样的话你就可以住进非常温馨的家里去了 但是这个装修一般来讲，不可能是说每家每户自己来装修 通常在我们这儿，都是有个装修队，有一个装修公司来帮助你来装修 那么这就是说装修公司起的作用就像操作系统起的作用一样 那么它把这些比较琐碎、 复杂的一些硬件屏蔽在它之下 然后向用户提供了一个非常良好的，非常优雅的、 清晰的 一致的这样一个接口，那么供用户去使用 那么我们说操作系统是对硬件机器的一个扩展 为用户程序提供了一个虚拟机器 好，那么我们最后再 考虑这个操作系统的三个作用，再强调一下 第一个作用，就是有效地管理计算机系统的各种资源 第二个作用呢，就是通过接口，命令接口 编程接口，为用户提供各种各样的服务功能 第三个作用呢，就是屏蔽硬件细节，为用户提供一个 扩展的虚拟机器

1.3操作系统的主要特征

下面我们介绍操作系统的特征。 操作系统 是一个软件，那么它与其它软件相比较，它具有哪些特点呢？ 总结一下，操作系统的特征呢 有四个主要的特征。 第一个是并发 第二个是共享，第三个是虚拟，第四个是随机 操作系统的第一个特征是并发 所谓并发 指的是操作系统必须具有这样的能力 能够处理多个同时性的活动 操作系统能够处理多个同时性的活动 但是当你处理多个同时性的活动的时候呢 就会带来很多的问题，也就是给操作系统的设计呢带来一定的复杂性 比如说，我有多个活动，这些活动呢是 一会儿操作系统管理这个活动，一会儿操作系统支持另外一个活动，所以活动和活动之间呢- 要有切换 切换的时候呢需要操作系统做很多的事情 另外，如果有很多的活动，比如说，很多的 程序都在内存，那就需要对内存进行保护，使得这些活动之间呢互相不干扰 通常我们在解决一个问题的时候呢，我们需要 多个活动相互配合来解决这个问题。 因此 这些活动之间就具有一个依赖的关系，那么这些相互依赖的活动之间，那我们需要 解决它的同步等问题。 因此，由于并发会给 操作系统的管理带来很多的问题，那么需要操作系统解决这些问题 这周呢，我们来再看一下，如果计算机系统中 同时存在多个程序在执行，那么它究竟是什么样的一个场景呢？ 在单 CPU 的情况下，我们说只一个 CPU 而言 那么在这个 CPU 上呢，所谓同时执行多个程序呢是一种宏观上的 观察，在宏观上，这些程序好像都在执行着，都在运行着 但是在微观上呢，由于只有一个 CPU，所以任何时刻 只有一个程序真正在 CPU 上执行，那我们可以说 这个，这些程序是在 CPU 上是轮流执行的 当然这个轮流的过程也需要操作系统提供相应的支持 并发是操作系统的一个最典型的特征 那么但是还有一个与并发非常相似的一个概念叫并行 这个我们也经常说并行，那这里头我们稍微区分一下 我们所说的并行，多指的是不同的程序 同时在多个硬件部件上执行 也就是它会有硬件的多个部件的支持。 我们举一个 日常生活中的例子。 如果是跑马拉松，大家都一块儿在跑 都在往前跑，那么实际上呢，这是一个并发的场景 大家都在并发地往前跑。 但是如果是 110 米栏 那么每个运动员都有自己的一个跑道 那么他只能在这个跑道上去跑，那么这个呢就是并行 所以我们通过这样一个小例子呢，大家对并发和并行这个概念呢有一个 区分，不要混淆这两个概念。 当然平常我们在说的时候呢，有的时候 并发和并行呢有的时候不太区分，这也是可以的，只要大家能够理解 意思就可以。 操作系统的第二个特征是共享 所谓共享是由于操作系统是管理资源的 那么这些资源呢数量又是有限的 用户也要使用资源，操作系统为用户提供服务的时候呢也用到这些资源 因此我们说，所谓共享就指的是操作系统与 多个用户的程序共同使用计算机系统中的资源 那么，对资源的使用有不同层面层次的要求 对操作系统而言，要求操作系统对资源管理要进行合理的 分配和使用，那么使得这些资源能够充分利用 在一段时间内，交替地被多个进程使用 那么如果我们对 资源的使用再进一步提出不同层次的要求呢，我们来看一下 有些资源呢它是互斥使用，我们称之为互斥共享 比如说像打印机，打印机给一个进程用就不能 同时再给另外一个进程用。 当然，一个进程用完了可以分给另外一个进程用 所以这种资源就叫互斥共享 而另一种层次上，那么就是同时共享。 比如说，我们有一段代码 我们有磁盘上的文件，那么两个或两个以上的进程 想共享一段代码，这是可以的 或者说两个或两个以上进程都打开同一个，打开不同的文件 处理不同的文件，那么这些不同的文件都放在磁盘上，那么这个磁盘呢是共享的 所以我们大致要了解一下，对于资源的这种共享呢，分成了互斥共享 和同时共享这两个方面。 共享就会给操作系统又带来管理上的一个复杂性 因为操作系统要使得这种资源的分配达到一个最优化，这个是比较难的 同时呢，操作系统还要考虑如何来保护资源 使得不同的进程之间不会互相干扰 操作系统的第三个特征呢是虚拟 这个是非常重要的一个特征 所谓虚拟呢，往往指的是 一个物理的实体，比如物理的，比如 CPU 或者是内存 把它映射为或者转换为若干个对应的逻辑实体 所以通常我们会说，操作系统 把时间分成了若干片段，分时了 或者我们把空间分成很多区域，分空间了 通过这样一些手法，那么使得一个物理的实体 给它转换成或者对应成多个逻辑的实体 通过这样一个手段，那么实际上呢是为了提高 计算机系统的资源的利用率，所以这是我们的目的。 通过虚拟 提高的是资源的利用率。 这样我们再举几个典型的例子，在操作系统当中如何来体现虚拟性 CPU，假定只有一个物理 CPU，一个单 CPU 那么，在单 CPU 上又跑了很多的进程，又运行了很多进程 实际上每个进程拿到的是什么呢？每个进程实际是在虚拟 CPU 上在跑，我们可以这么去理解，虚拟 CPU。 当然啦，这是宏观上 那么微观上呢，每次只有一个进程在一个 CPU 上去跑 那么，存储器。 存储器呢我们是这样来 解决这个问题的。 操作系统给每个进程分配了一个地址空间 这地址空间呢是一个虚拟地址空间 是每个进程一个，各个进程之间是相互独立的这么空间 这个空间里放了这个进程的代码啊、 数据啊、 堆啊、 栈啊等等的这些信息 那么所以每个进程都有自己的地址空间，那这个是逻辑的。 那么在物理上，那么它会 共享，就共用一块物理内存。 另外呢，我们有显示器显示设备，在显示设备上我们可以开多个窗口 或者是虚拟终端，那么这样的话呢，一个显示设备其实可以显示不同在不同的 这个窗口或者虚拟终端显示不同的内容，那么这个呢也是一个虚拟性虚拟化的一个结果 所以操作系统的这个虚拟特性是我们比较，应该讲都比较了解熟悉的 操作系统的最后一个特征呢是随机性 所谓随机就是说，在操作系统运行过程中，在用户程序运行过程中都会发生各种各样的事件 操作系统必须随时对这些事件进行相应的处理 所以我们说，操作系统必须随时对 以不可预测的次序发生的事件进行响应并处理 由于有随机性，其实我们会看到这样一些 结果。 那么，进程的运行速度 是不可预知的。 因为我们有好多进程在运行，我们不知道什么时候是这个 进程运行，它运行多长时间，所有的一切都在操作系统的掌控之下。 因此 多个进程在并发执行，这种并发执行，我们所看到就是 "走走停停"，一会儿上 CPU 执行，一会儿呢暂时，一会儿又上 CPU 执行 所以用户也无法预知每个进程的运行推，呃，运行的推进的这个快和慢，这是无法预知的 其次，那么，如果有随机性 就会带来另外一个问题，就是，在操作系统运行过程中如果出现了错误 很难重现，也就是，不能重现或者很难重现 系统在某个时刻的一个状态，因为这种状态的组合太多，所以很难够 重现，那么这样的话就给操作系统的调试带来了很大的困难，带来了很大的困难 所以呢，这是操作系统的最后一个特征，就是随机性

1.4典型的操作系统的架构

下面我们介绍典型操作系统的构架， 那么我们主要举典型的像 windows UNIX, Linux 这些典型的操作系统的一个架构我们来介绍， 主要是通过这个介绍希望大家对一个操作系统 有哪些典型的功能呢有一个感性认识。 我们首先介绍 windows 架构， 这是一张 windows 的这个体系结构的图， 那么这张图当中呢，我们首先要明确 这有一条比较深的实线， 那么在这个线之上，我们称之为用户态 之下我们称之为内核态，那当然了，这个 整个 windows 架构呢，既有一部分在用户态执行的 内容，也有很多，就是更多的功能是集中在了内核态。 在内核态里有哪些模块呢，或者有哪些功能呢？ 首先我们看一下，有一个硬件抽象层 那么这个硬件抽象层呢实际上是和具体的 计算机打交道的一些屏蔽了这样一些细节在这一层里头， 然后是内核层，内核模块， 内核模块呢主要是处理一些不同的 体系结构之间的一些，相关的一些重要的功能，比如说 线程的管理，线程的基本的一些管理，线程的调度。 还有一些中断的处理，还有一些陷入或者叫 异常的处理，那这些还包括一些同路都在 Kernel 这里头去做 然后我们可以看到这是设备驱动程序层， 那么设备驱动程序这个模块呢其实是完成一些和设备打交道的，比如说像 硬件的设备驱动程序，各种各样的硬件设备驱动程序，还有呢像文件 系统驱动程序还有一些过滤器驱动程序等等。 这些，这个模块呢是 图形与窗口系统，那这也是 windows 非常特殊的地方， 中间呢一大块我们叫做执行体， 执行体呢首先提供了各种各样的功能，像我们看到有进程线程管理， 有虚存的管理器，还有安全监测器，还有 I/O 管理器 还有一些对象管理器，等等等等。 除此之外它还提供了一组 函数，这组函数呢实际上是供用户态程序去调用执行的， 那么这组函数呢是通过了 NTDLL.DLL 这个，这一层 提供给用户程序使用，那么在 用户态我们可以看到有一些系统支持进程， 还有一些服务进程，还有我们大部分的一些应用程序。 有一个特殊的叫做子系统，环境子系统环境子系统中比较典型的就是 win32 的 环境子系统，那么 win32 的环境子系统呢，实际上是把一些 内核的一些系统调用，一些函数经过 了封装，封装成一些特定的形式，提供给用户去使用，编程的时候使用 那么这张图呢可能比较复杂，我们把这张图 简化一下，简化成 这样一个就把几块给它明晰出来，那么我们还看到用户态 内核态，用户态呢有这么几，用户态呢有系统进程，服务进程，用户进程和环境子系统。 那么这上面有一层就是动态链接库，那么实际上是提供了各种各样的函数给 上面执行，在内核态呢，有几大块 硬件抽象层，内核，设备驱动程序，执行体，还有图形窗口 模块，那么这里头和硬件 相关的模块，一个是硬件抽象层， 一个是内核，不同的体系结构呢是在内核中区别的， 那么相同体系结构的不同的计算机呢是在 HAL 层区别的。 那么有了这两层就基本上 那是把所有的硬件的相关处理都封装在这两层里头。 也就是其他的像设备驱动程序，执行体 图形窗口系统这些模块，就不和硬件打交道了。 那么执行体对外提供了各种各样的 函数支持，那么通过内核态可调用接口和系统服务的分发器 上面和动态链接库对象相接，然后为对象提供各种各样的服务。 这就是简化以后的一个 体系结构，那么我们再简化一下， 那个还复杂，再抽象出来，那么就抽象成这样一个 非常简单的图。 底层是硬件，在硬件之上是操作系统的扩展。 但是呢，操作系统要为用户程序提供各种服务嘛，所以有一专门有一层，我们叫做系统功能- 调用层。 那么这个系统功能调用层之上是应用程序，所以我们也可以把它再抽象成这样的一个图。 下面我们看一下 UNIX 架构， UNIX 操作系统呢 它呢，来看看先从，自顶向下来先看一个非常粗的，那我们先看一下 这个层次架构当中呢，里头是硬件 在硬件之上呢我们叫内核，叫内核，就是操作系统内核。 然后内核之再外一圈呢，实际上就是系统调用接口，系统调用接口。 在系统调用接口之上呢，在外圈呢就是 UNIX 各种各样的命令和这些库。 我们这个比较熟悉啦，那我们来看看内核有什么？ 内核呢 这些功能我们刚才在介绍 windows 的时候呢，其实也碰到 这些功能，只是它们的关联，它们的这个关联是不太一样的。 我们来看一下， 首先我们看一下同样有用户态，有内核态，在 内核态里有哪些功能呢？ 我们有一个进程控制的子系统，这里头包括了调度 也包括了进程间的通信，还有些存储管理，内存管理。 然后我们有文件系统，文件系统， 我们有各种各样的设备驱动程序，那么在 UNIX里的设备驱动呢分成了块设备的设备驱动和 字符设备的设备驱动。 同样 我们在UNIX里头也有硬件的控制层，硬件的控制层， 这里头跟硬件打交道的比较多。 在接近在这个 内核的最上层，事实上我们又看到了我们所熟悉的系统的 调用接口，系统调用接口，而系统调用接口呢 我们是提供给用户程序用的，所以用户程序呢 可以和直接和系统调用接口打交道，也可以通过一些 库，一些函数库来和这个 系统调用接口打交道，这就是 UNIX 操作系统的一个架构， 现在我们来再看一下 LINUX 的架构， 但这是一个只把 LINUX 内核组件展示出来了， 同样我们看到了内核态和用户态， 那么LINUX里头有哪些功能呢，内核里头有哪些功能呢？ 我们看到了进程和调度，进程，调度。 有虚存的管理，有物理内存的管理， 我们这还看到了文件系统， 当然啦设备也有管理，那么也有 各种各样的设备驱动程序，字符设备驱动程序，块设备驱动程序，网络的设备驱动程序。 通常我们操作系统支持网络，所以专门有一个网络的 模块，按照网络体系结构编写的这个模块，支持网络功能。 那么还有两个比较 重要的模块或者功能呢，一个呢是陷入和 这个异常的一个处理，一个呢是中断处理，这些呢都是和硬件 相关的一些从操作系统角度上，对硬件的一些发生的一些事件的一个处理 的功能，当然我们也有一个接口 系统调用接口，所以在这里头，也是整个内核的最上面 我们有一个系统调用接口和前面介绍的两个操作系统都非常的相似， 那最后我们再简单的看一下安卓的架构， 那么安卓操作系统我们给出的就是一个整体架构 那么这个整体架构当中呢那么分了四层。 这四层呢最底层呢是 LINUX 内核，当然这是我们这门课里头其实关注的应该是内核的内容。 那么在 LINUX 内核里头包括了一些进程管理 内存管理啊，一些安全啊，还有一些网络啊驱动，在 内核之上是一些系统库和安卓运行时系统， 在这之上呢是我们的应用程序框架，那么开发人员可以 使用这种内核提供的这种，各种各样的 API 来建立这个框架。 最后再上面呢是安卓一些的应用程序， 这是安卓的一个操作系统的一个整体架构。 通过我们上述的介绍 那么希望大家能思考一下，从 WINDOWS、 LINUX、 UNIX 的这种系统架构当中，我们得到了哪些结论，我们看到了哪些功能？ 我们看到这些功能是怎样关联的，那么从安卓的架构中，我们又得到了什么样的结论呢？ 希望大家能够思考一下。

1.5 操作系统的分类1

下面我们介绍一下操作系统的分类 啊，各种各样的分类都是存在的，那么我们呢主要是介绍传统的分类 那么我们要简单提一下，我们教材里提出的这个 Tenenbaum 的这种啊分类 操作系统是不断在发展的 那么它的发展是由不同的因素来驱动的 这些因素呢我们归纳总结出：首先是硬件技术的 发展，硬件发展了对操作系统就有新的要求 还有就是应用需求的发展 然后是软件新技术的出现 所以我们说操作系统的发展是随着计算机硬件技术 应用需求以及软件新技术的出现而发展的 为什么要不断的进展？那么因为 操作系统要能够充分的利用硬件，能够为用户提供 更好的服务，所以操作系统必须以这个为目标来不断的发展 从过去到现在 操作系统实际上是总在不断的往前发展，那么 早期很早的时候是大型计算机，那么有大型计算机操作系统 后来呢，又有了个人计算机啊，个人计算机呢又从简单到复杂，所以又有 不同的操作系统出现。 当有了网络的 供支持之后，那么我们的操作系统就要支持网络的功能 后来移动计算出现了，那么操作系统要 变得能够支持各种各样的移动 设备上的各种就管理啊，各种管理功能 云计算和我们现在的 泛在计算或者是说物联网时代，一直到今天 非常火的机器人，那么都需要操作系统 进行相应的改改变，然后去支持这些 不同的环境，不同的环境。 好！那么我们 看一下操作系统的这些分类，我们主要是谈的是 传统操作系统分类。 那这传统操作系统分类呢主要是要 通过这个时间啊，年代啊，年代，我们来看一下 那么这些分类当中呢，其实希望大家掌握就是说每一 种操作系统或者每一类操作系统，它的主要的特征或它追求的 目标是什么样的？它的工作方式又是什么样子？我们要抓住这 两点。 一个是它的追求的目标，每一种操作系统追求的目标不太一样 有一种它的，而且是每个操作系统的工作方式也略有不同 我们从这两个角度来抓住它的特质，抓住它的最特别的地方 我们先来看批处理操作系统 这是一个最早出现的操作系统。 那么批处理操作系统我们现在可能 很难见到啊，并不是不存在而是我们平常日常生活我们用起来很难用到它 啊，很难用到它。 在一些大型的商务网站或者是一些大型公司里头它还是用到 批处理操作系统的。 那我们首先呢来通过我的介绍啊 大家也充分的发挥一些想象力，来看一下一个批处理的操作系统是怎么工作的？ 啊，它的工作方式是这样的，那么首先呢 用户要将自己的作业交给一个专门的 人员来处理，那么这个人呢我们称之为系统操作员，所以用户将作业交给系统操作员 系统操作员将许多用户的作业 收集起来，组成一批作业然后把它输入到计算机系统中 那么这样的话在计算机系统中就会形成一个作业流 这个作业流可以自动转接，是一个连续的自动转接的作业流 这个时候呢，再启动操作系统 让操作系统自动的、 依次的来执行每个作业 那么作业的执行结果，统一由操作员 提交给用户，那么这是一个批处理操作系统的工作方式 这里头呢我们可以看到，用户对自己的这个作业是不能够干预的 那么批处理操作系统它的存在呢，主要的一个目标是什么？主要追求的目标是什么？ 第一个是提高资源的利用率，第二个是能够增加 作业处理的吞吐量，就单位时间里处理越多的作业越好 那我们来看看批处理操作系统的一些 简单的细节啊。 那么首先什么是作业？ 啊在批处理操作系统中的一个作业呢包括了几部分 包括了用户的代码程序，这个程序需要处理的数据 以及这个作业怎么执行的一个 规格说明书我们就把它称之为作业说明书 使用一种特定的叫作业控制语言编写的 它和我们现在的脚本很相似啊很相似 那么它描述的是这个作业，怎么样去处理的一个步骤？ 那么批处理的话体现在批，啊所以成批，所谓成批就是 通常这一批作业是由若干作业组成的，用户 提交完作业之后，只能等待处理结果不能干预自己作业的执行 那么当操作系统对批作业进行处理的时候呢 实际上是对这一批作业的每一个作业进行相同的一个处理过程 都是从磁带读入用户作业和 编译链接程序，然后呢去编译链接 用户作业生成可执行程序，然后再去启动执行这个程序 并且执行然后输出结果。 所以 所有的作业都是按每个作业都是按这样一个过程来完成的 这是一个典型的啊，典型的一个一个作业 的一个这个这个结构，当然这一个作业结构是用卡片的 啊，早期现在我们见不到了，早期呢我们的输入是用卡片输入的 叫卡片机，卡片读取机。 那么每个程序啊，代码呀 数据啊，都都穿在这个卡片上，就是在卡片上输入 啊，那么这个卡片呢就包括了你的一些基本描述 和你的这个 Fortran 的编译、 程序还有一些 装入，运行的这些命令还有相应的一些数据，这是一个作业的一个典型的结构 那么当把这一批作业 送到计算机上的时候呢，这里就说需要通过输入输出设备来完成 那么输入输出设备有个特点，特别慢所以我们说是慢速的输入输出 如果我们说一台功能强大的计算机 来去处理这些慢速的输入输出，那么就会造成 在输入输出的时候 CPU 实际上是空闲的，因为这时候呢是 磁盘那其他的部件在工作， CPU 是没事可干的，因此 当批处理系统中出现了这样一个问题的时候呢 那么我们就开始来寻找一个好的坚决方案 早期的一个解决方案是什么呢？就是说增加计算机 不是一台计算机，增加其他计算机，这些计算机呢我们称之为卫星机，啊卫星机 那么卫星机的作用呢，它就是完成 对用户的输入、 输出这样一些工作 在完成这些工作之后呢，把这个输入的内容或者输出的内容 都暂时存放在磁带啊或者磁盘上 那这就是卫星机，所以卫星机也是计算机它的功能稍微弱 功能较弱一点要少一点，那么它的工作主要是为了完成输入输出 好！这是一张早期的批处理的一个 示意，那我们就看到了 1401 是比较便宜 的功能比较简单的计算机，那么就是我们刚才说的卫星机 那么它们呢就是完成输入作业 输出结果的作用，所以也就是说这台 1401 负责的是卡片机 那这台 1401 就负责的是打印机啊就分别负责不同的输出设备。 那么这个 7094 是功能非常强大的一台计算机，那么这台计算机呢是要充分利用它要 处理作业的。 因此我们来看到这个系统操作员会把在 1401 上输入的这一批作业，放在磁带上然后把磁带送给这个 7094 然后呢这个 7094 上操作系统在跑，把作业依次去执行 结果呢也放在磁盘上磁带上，然后再由操作员去把它送到 啊应用一个卫星机来做最后的输出工作，所以这是早期的批处理操作系统的一个示意 那么批处理操作系统呢我们看到经常会说单道批处理和多道批处理，这两有什么区别呢？ 所谓单道批处理、 多道批处理是指的在 7094 这台计算机上一次处理 多少个作业，如果一次处理一个作业，这个作业处理完了 再去选择第二个作业进行处理，就叫单道批处理 那我们也可以说，一次选多个作业 然后呢，选，比如说选上 3、 4 个， 4、 5 个然后去处理，那么这叫多道批处理系统 那我们当然更，更多的是谈的是多道批处理系统。 因为这是支持多道程序设计的 那么在实现批处理系统的时候呢，通常采用的一种技术叫做 spooling 技术 spooling 技术是 1961 年英国 曼彻斯特大学在一台 Atalas 计算机上啊做的实验 那么，spooling 这个词呢，来源于这样一个一个描述，这个描述呢是 Simultaneous Peripheral Operation On-Line 那么它的翻译过来呢，叫做同时的外部设备联机操作 那么，为什么要用这种技术呢？它和我们刚才那张啊 描述的图最大的区别在于，它是在一台机，计算机上 完成输入、 处理和输出 在一台计算机上，那么怎么完成呢？是用磁盘做缓冲 将输入、 计算、 输出分别组织成独立的任务流 在一台计算机上执行，使得输入、 输出和计算真正并行 我们刚才在，回想一下，刚才是用卫星机完成输入 和输出，那就是两个不同的计算机，然后用主机完成计算 那么，有了这个技术这是软件技术，就会把所有的工作在一台计算机上执行 所以叫联机操作。 那么 和刚才的脱机相比，相比较而言，所以这种 spooling 技术 有的时候我们又称之为：假脱机。 也就是说，它不是真的脱机，它是假脱机 好，这是 spooling 技术。 那么 spooling 技术或者 spooling 系统的工作原理是什么？那么，它把 一一个完整的过程，对一个作业处理的完整过程，分成了几，几个阶段 第一是用户作业加载到 系统当中，就是输入。 那么放到哪里呢？放到了 在磁盘上的一块空间，叫输入井。 所以我们说第一个 一项一个任务呢，是把用户作业送到磁盘上的输入井 那么，磁盘输入井会有很多的作业。 因此，第二项任务呢就是 按照某种调度策略选择几个作业搭配得当，啊，长的短的 这个不同的啊性质作业，然后把它调入内存去执行 那么，第三项任务呢，就是在控制这个程，作业的执行，那么这个作业 的执行呢，把它送到什么呀？送到输出井啊，输出井呢是磁盘上的另一块区域 那么专门去存放每个作业的输出结果 最后，那么需要一项任务，专门 负责把输出井上的数据啊结果 送到打印机上去。 那么这样有这么 4 个 4 项任务，这 4 项任务 在同一台计算机上呢，并发的去执行，啊并发的去执行 那么它的好处是说，当一个作业在主机上执行的时候，它要真正输入的时候 不需要到卡片机上去读入数据，因为所有的输入数据已经送到了输入井 所以，它就只要从输入井上去读数据就可以了。 它输出的时候呢，也不需要把结果直接送到打印机 那么，它只要把结果送到输出井就可以了，那么这样的话呢就提高了 整个计算机系统的一个效率，啊就是资源利用率 那么，我们再看 一下，引入了这样一个技术之后，一个多道批处理系统 在一台计算机上呢，变成什么样了呢？那我们看，上面还是我们传统的脱机，对吧？脱机 那么，我们就变成了在一台计算机上的一个，整个的处理 过程。 而这些处理过程呢是由输入进程 作业调度程序、 作业控制进程和输出 进程相互配合共同完成的，在一台计算机上 所以，输入进程从输入设备上把各种各样的 作业，读入到输入到磁盘上，输入井 作业调度进程呢，去从输入井选一些作业啊，搭配得当的作业，把它 送到内存，然后作业控制进程呢，来控制作业的执行过程 中间呢，因为不同的作业在内存，还需要进一步的去调度，谁上 CPU 谁暂时不上 CPU ，当然它的输出结果是送到 输出井。 如果它还想在运行过程中想输入，就从输入井去读数据 就可以了。 最后，输出进程就把结果送到了输出设备 因此呢，上面就是脱机，下面就是 spooling 技术，就是假脱机 那么，我们为什么讲这个技术？ 那这个技术因为很古老了，那么现在我们还用吗？ 实际上呢，在我们现代的计算机系统中 我们还在用 spooling 技术，因为，在我们做打印的这种服务的时候 我们通常的是采用的是 spooling 技术来实现这个打印的 那么我们来介绍一下，啊，比如说我们有一个 Word 文件 我们有 100 页，那这时候我们就把它啊，这个按了这个打印的这个按钮 那么这时候我们会看到什么？我们会看到，一般情况下啊，是在屏幕的右下角 我们会看到一个显示说，打印几页、 打印几页、 打印几页嘛 那实际上，打印机很慢，它不会马上把这个打印结果啊，这个输出来 那么，当底下这个显示结束之后，我们就可以关闭这个 Word 文件了，就可以干别的事情去了 当然，如果这个还没有完成，我们去关闭这个文件，就会提示你说，这个打印还没有完成 但是，如果它结束了，可是打印机上确实没有出东西 那么，什么原因呢？就是说，它的打印的完成是说把这个打印的 内容，送到了一个队列里头去，我们把它称之为打印请求队列 就是说，把这个，其实也就是在磁盘上的一个目录啊，就送到这个队列 里头去，当这个队列里有打，有要打印的文件的时候呢 那么，这个时候操作系统会提供一个打印进程，专门来管理这个打印队列 当这个队列不空的时候，有打印的这个文件的时候，这个进程就被启动啊，被启动 去分别一个一个的，把文件送到打印机上 当然，如果没有打印的请求，这个队列是空的 那么，这个打印进程呢就可以去睡眠，就可以去等待啊，去等待 所以，现代操作系统中，这个打印过程通常也采用的是什么呀？ spooling 技术。 所以，我们为什么介绍它呢？也是因为说它还继续发挥作用

1.6 操作系统的分类2

下面我们介绍分时操作系统 虽然说我们现在基本上看不到分时操作系统了 但是呢它是最早的一个交互式的系统，和我们现在的使用计算机的方式非常相像 那么分时操作系统它的工作方式是这样的，得有一台主机 然后这台主机连了很多很多的终端。 所谓终端呢 实际上只有键盘啊，显示器啊，其实它没有任何的 计算能力，没有 CPU，也没有什么啊？存储，也没有存储 所有在这上这个敲入的命令都直接送到这个主机上 当用户呢就敲入命令呢，然后呢等待回答 等待回答之后呢，根据回答的结果呢再输入命令，实际上是这样一种工作方式 那这里头呢就是说一台主机同时要为多个终端去服务 那么就必须采用一定的技术。 一个非常典型的 就是时间片，啊，时间片，因为要轮流为每个终端用户服务的话，那么 操作系统会将 CPU 的时间划分为若干个片段，我们通常叫时间片 这个词大家都很能理解。 那么操作系统呢 就是以时间片为单位，轮流为每个终端用户去服务，而且每次 给你一个时间片，那么时间片到了，那么这个 终端，就不为这个终端服务，就为另外一个终端用户服务了，是这样一个过程 那么这个做法为什么能够非常好地 工作，让每个用户都非常高兴地在计算机前去使用计算机呢？ 实际上它是利用了一个人的错觉，因为每个人在 做操作的时候会比较慢，计算机的处理能力很强，那么 它一两秒钟处理的能力，就处理完的事情，你呢可能敲半天，敲半天。 所以在这种情况下 利用了这样一个人的错觉，使得用户呢 感觉到说每个，这个计算机为我自己服务，没有去为其他用户服务 就是这样一个错觉。 那么这个呢实际上也就是说把一台 计算机变成了多个虚拟的计算机，就是说 每个用户在一个虚拟机上去运行，也可以这么去理解 好，那么在分时系统，和我们现在是一样的，就是说，现在是交互式系统 交互式系统追求的主要目标是什么呢？就是相应时间，就是敲一个命令 发出一条命令，然后它得到回答，这个时间就 是一个响应时间，当然我们希望，每个用户都希望，响应时间是越短越好 越短越好。 那么这样的话呢，操作系统呢才能够为很多的 用户去服务，让用户感觉到操作系统为自己服务一样 那那个时候呢计算机很贵，所以呢不可能说一个 我有一台计算机这就是批处理系统，那台计算机呢是分时系统，通常呢那个时候的 操作系统都是叫通用操作系统。 也就是把分时和批处理的处理结合起来 向用户提供各种服务。 那么基本 的原则是这样的，分时优先，批处理在后，为什么呢？因为 用户是和计算机打交道的，它有一个心理的一个这个 感觉，如果计算机没有响应他，时间很久不响应他，他会 着急，会烦，所以呢我们希望说分时在先 然后呢有一些对时间要求不强的一些作业，可以呢 作为批处理作业处理，可以放在后台 所以我们可以说分时在优先，那么就是前台 前台呢放了很多，就处理各种需要频繁交互的作业 而后台呢实际上是放一些这个 时间要求不高的作业，当然这个我们白天的时候 大家都上班了，那么这个时候分时模式起作用 但是如果中午大家都午休了，吃饭了，或者晚上大家都下班了，那么这个时候就可以 使得大批的这个批处理作业去执行 那这样让计算机能够时时刻刻保持着运转 这就是所谓通用操作系统，分时和批处理的一个结合 另一种计算机 的操作系统的类型呢就是实时操作系统 所谓实时操作系统呢它有对时间的这种这个，更有更严重的要求 我们来看。 所谓实时指的就是计算机能够及时响应外部事件的请求 在规定的严格的时间范围内，完成对事件的处理，一定要求把处理完成。 那么完成处理实际上是控制各种各样的实施设备 和实施任务协调一致地去工作 我们典型的像汽车，汽车 汽车电子，那么这里头应该是个实时系统 为什么呢？当你踩刹车的时候，当你要做各种各样的操作的时候必须 很快去完成这项工作，否则就会出现其它的问题 那么实时操作系统呢当然也分成不同类型 第一类呢我们叫做实施过程控制 那么包括了工业控制、 航空、 军事控制等等等等 第二类呢叫实时通信系统，或者实时信息处理系统，像我们 说的一些电讯，自动交换机，银行，银行系统 飞机订票，股市行情的这些都属于实时的信息处理系统，那么当然它也是要求有实时性的 那么实时操作系统在设计的时候它所追求的目标是什么 呢？主要是，第一，对于外部请求在严格的时间范围内要有响应 要有处理。 第二个呢是高可靠性，因为 一个实时系统往往是运行在一个环境里头，那么这个环境 要求这个系统能够对环境要有适应性 所以要有高可靠，比如说像汽车电子 汽车，那么你要如果不可靠，那你就会带来其他问题，对吧？ 那我们在这里头再强调一下 实时系统当中的这个最关键因素和参数呢就是时间 所谓时间呢，那么我们知道，比如说举个例子，就是汽车的 装配线，这是讲一个工业过程控制这样一个系统。 那么汽车装配线上呢比如说我们要 让一个汽车去完成一个汽车的组装工作，组装工作 那么如果没有某一个环节没有在严格的实时时间范围内处理，那就会造成什么？比如说 焊接机器人，如果焊，焊接，没焊完，这个时间就过去了 那么这种就带来了一个隐患，将来汽车在行驶过程中就可能出问题呀 所以呢这个时间是非常关键的，但是这个时间呢 其实又分成了对时间要求的一个硬性的和一些 软性的要求。 那么所谓硬性的要求呢，就是我们通常称之为硬实时系统 所谓硬实时系统呢，就是说某个动作绝对 必须在规定的时间这个范围内，或者某个时刻完成 像刚才我们说的这种汽车装配线上，安装轮胎 也好，去焊接什么也好，那么这些东西都要有严格的这个时刻 或者时间范围来决定。 但是呢还有一类对时间要求很 很快，但是呢又没那么严格，相对而言，那么我们叫做软实时系统 软实时系统呢实际上就是说，有一些实现，最终实现可能 偶尔违反，在某些点上违反了这个最终实现，没有在规定的时间内去做到 但是呢这是偶然的行为，还是可以接受的 我们举一个例子就是，就是媒体播放 就视频播放啊，MP3 这种播放，音频播放 那么播放一个，一段这个这个电影 那么这个电影的播放的过程中如果偶尔有一帧，两帧没有 达到实时的这种这个，这个处理，那么可能会看电影的时候可能有点卡 但是呢也就是偶尔，那么你会对这个电影的整体的理解是 没有问题，不受影响的，所以呢我们也接受这种情况 这就是所谓硬实时系统和软实时系统 那么这些系统有了以后，后来慢慢慢慢到了上个世纪 80 年代就出现了个人计算机 那么个人计算机也需要操作系统，而个人操作系统大家知道那时候呢 是计算机在某一时刻只为一个用户服务，所以相对比较简单 所以在个人计算机操作系统刚刚出来的时候，那么实际上对 界面友好、 使用方便 还有上面提供很丰富的应用软件是有 要求的。 那么对于操作系统内部的功能其实没有太多的要求 慢慢慢慢个人计算机系统又往前发展了，从简单到复杂 当所有的计算机能联网的时候 那么这个时候主要出现了一个叫做网络操作系统， 所谓网络操作系统就是说，在计算机联网的之上 在网络当中有各个计算机，对吧？有各个计算机，每个计算机都有操作系统，这些操作系统- 必须支持 按网络体系结构协议标准开发的这种网络模块，网络模块 而网络模块呢主要完成的是网络的管理 然后计算机之间，在网络当中的计算机之间的通信，安全， 包括资源共享和 各种各样的提供各种各样的网络应用 啊网络应用。 那这时候有了网络之后一些新的要求。 所以网络操作系统的追求目标是：第一，在网络上各个计算机之间能够相互通信。 第二，能够大家共享这个网络上的各种资源。 再之后，实际上呢就 对操作系统的分类呢，我们就介绍一种分布式操作系统啊。 当然分布式操作系统主体呢，它建立在分布式系统之上的，是为分布式系统开发的。 那么所谓分布式系统呢，可能是以计算机网络为基础，就是把计算机联网； 也可能是多处理器，现在的多 CPU 的这种计算机系统，这也是分布式系统的一个典型的。 所以基本的特征就是说处理能力，就是 CPU 的这种计算能力是分布在不同的计算机上的。 而分布式操作系统，那么它就在这种分布式系统之上 开发的一个操作系统，而这个操作系统的特点是什么呢？它是一个统一的操作系统。 而允许啊，就是通过操作系统的这个运行， 允许若干台计算机上的这个，这个完成同一个任务。 那么这些，若干台计算机是相互协作相互配合来共同完成一项任务。 那么操作系统把各种各样的任务分布在，在系统当中的任何一台计算机上都是可以执行的。 这种分配可以自动去实现，然后包括任务的分配，包括 这个调度，包括负载的这个均衡等等，都是操作系统 要解决的问题。 有了分布式系统以后 那么处理能力就增强了，速度也快了，可靠性增强了， 同时呢对用户来讲就具有透明性，因为用户不知道自己的任务在哪一台计算机上执行。 但是由于有这么多的这个好处了， 那么对操作系统的这个设计就带来了一个更大的一个复杂性。 另外一种操作系统， 我们称之为嵌入式操作系统，因为嵌入式操作系统是为嵌入式系统开发的。 所谓嵌入式系统的话呢，其实呢嵌入式系统是在各种各样的设备、 装置当中完成了某些 特定功能的硬件系统，啊软硬件系统啊软硬件系统。 比如说我们说汽车，汽车里头有一个电子嵌入式系统。 手机、 电视机，类似于像这个什么 MP3 播放器 ， 像冰箱啊，彩电微波炉其实都有了。 那么 我们所说的嵌入式系统呈现在我们面前的呢，其实不是计算机啊不是计算机， 它是一个大设备当中，或者大的装置当中的一个部分啊一个部分。 而这些嵌入式系统通常工作在一个反应式 或对处理时间有严格要求的环境中，就是像汽车。 好，那么在为嵌入式系统开发的 嵌入式操作系统呢，那么它的主要特点就是说对整个 嵌入式系统以及它所操作的控制的，各种各样的这种 部件装置来，这些都是它的资源，来统一地协调， 统一地调度，统一地去控制，那么这就是嵌入式操作系统要做的事情。 好，这就是操作系统的一个 某一种传统啊，传统意义上操作系统的分类。 那在我们的教材里头呢，其实还给了另外一种分类，但是我们称之为 TANENBAUM 的分类。 那么这个分类呢，也非常有意思，大家可以看一下：大型机、 服务器、 多处理机、 个人计算机、 掌上计算机、 嵌入式、 传感器节点、 实时操作系统、 智能卡操作系统。 实际上好像感觉就是从大到小这样一个分类， 因为分类是什么，什么角度都是可以的，那么我们现在呢 这个分类呢我们就不详细介绍了。 大家呢因为书上都写了，所以大家呢去读一下教材啊。 了解一下每一种啊分类啊，每一种操作系统它都在支持哪些应用程序的执行。 这里呢我们简单的再介绍一下智能卡操作系统。 因为这个也比较好玩，对吧？那么智能卡呢实际上 是一个卡片，卡片上有一个芯片啊有一个 CPU 芯片。 那么智能卡因为它这个太小了，所以呢它的这个 运行过程中的能耗要有严格的要求，不能太耗电啊不能太耗电。 而呢 其另外呢，它的这个空间呢也很小，也不能有很多的存放的，存储空间，所以 耗电和空间都是有严格的限制的。 那么因此 在智能卡上通常我们实现的这个应用呢，一般都是单项功能。 比如说电子支付啊等等等等。 在这个卡上呢其实应该有一个专用的小操作系统啊 小操作系统，我们来看一个小例子啊，比如说有一些智能卡，它是面向 Java 的， 因此呢在这个智能卡的这个 ROM 上呢，我们会有一个非常小的一个 Java 虚拟机解释器。 当 Java 程序被下载到这个卡里的时候， 就由这个 Java 虚拟机解释器呢来解释执行。 有一些卡上呢如果允许它多，同时处理多个 Java 程序，那么这就 是多道程序设计，也就是我们要支持多个程序之间的一些切换啊一些问，一些处理。 那么这个时候多个 程序执行还需要一些调度知识， 因此当两个或多个程序同时运行的时候呢， 对于这样一个智能卡上的资源管理，还有这些程序之间的保护 就都成为了一个突出的问题，那么这些问题呢都是需要操作系统 来处理解决的。 到这儿呢我们就把第一讲操作系统概述介绍完了。 那么我们呢大家提一些要求，在学完 这一讲之后呢，希望大家能够对本讲的重点有所了解。 那么首先呢我们希望大家掌握操作系统的概念， 理解操作系统它的不同的作用。 然后去理解操作系统的主要特征。 另外我们希望大家能够掌握一项重要的操作系统技术，就是 SPOOLing 技术。 另外希望大家能够了解操作系统的架构， 操作系统的分类。 大家在课下呢还要重点阅读教材的相关内容， 我们这里给出了，比如说第 1 章希望大家去读1.1、 1.2、 1.4。 那么第 10 章也有一些相关内容是 10.2.5 ，这是跟 Linux 相关的一些内容。 还有第 11 章跟 Windows 相关的内容 11.3.1。 重点呢大家要掌握我们提出的这几个概念。 一个是操作系统的定义，操作系统的三个作用， 操作系统的四个特征：并发、 共享、 虚拟、 随机。 我们还要掌握 SPOOLing 技术的工作原理 以及它的一些用途，还有呢我们 要大致了解一下 Windows、 Linux、 UNIX 的架构，了解在这些操作系统当中有哪些主要的功能。 好，今天我们的课就讲到这儿，谢谢大家。