《openEuler内核编程》

课程讲稿

第一章 第1讲

操作系统简史

软件所制

第一章 第1讲 操作系统简史

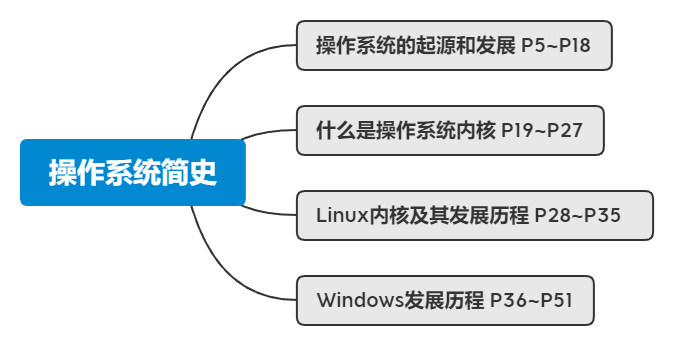
**学时：**2学时

**教学目的：** 了解自计算机发明以来操作系统的简要发展历史。了解什么是操作系统内核，对Linux的发展及现状有基本的认识。最后通过学生熟悉的Windows操作系统来讲解一个操作系统的进化过程。

**课程时间线：**



**知识框图：**

****

**课外参考读物：**

1. <https://zhuanlan.zhihu.com/p/19796979>
2. https://www.oschina.net/news/111410/linux-market-growth
3. https://lkml.org/lkml/2020/5/17/314
4. https://www.oschina.net/news/100042/linux-september-2018-stats

**PPT讲稿：**



这一章一共有5讲内容，今天的内容是第一讲，比较轻松，和大家聊聊操作系统的历史，然后着重讲一下Linux的历史，因为我们的课程是基于Linux系统教学的。



这一节也是分两部分，前面一部分先和大家介绍一下操作系统的历史，以大家最熟悉的Windows为例，说一下现代操作系统的发展。后面会讲一下Linux的内核，感受一下Linux的魅力。以及Linux的发展历史。

1. （这里还是以ENIAC作为起始，因为讲的是操作系统的起源而不是计算机的起源。）

世界上第一台计算机于1946年的情人节，也就是2月14日，诞生于美国的宾夕法尼亚大学，这个是由莫齐来和他的学生艾克特设计的，这个计算机非常大，重30吨，占地面积140平方米。这个电脑的名字叫做ENIAC（Electronic Numerical Integrator and Calculator），英文的意思是电子数值积分和计算机。这个东西当时就一个功能，计算导弹的弹道，确保打出去的炮弹能打中目标。所以这个东西可以说是我们现在一切计算机的祖宗了。其实很多领域都是这样，最早都是在军工领域，然后慢慢的渗透到民用。像互联网，最早就是阿波罗登月时期的一个内部局域网，后来才发展为全世界的互联网。像ENIAC这一类的早期电脑，全部都是手动操作的，各种人工插线，后来进步了一些，开始用了打孔纸带。

这个时期的计算机还根本没有操作系统的概念，自动化的程度也不够，由于体积大比较笨重，应用的范围也比较窄。具体操作就是人拿着纸带放到输入机，得到程序和数据，再输入到计算机，拿到结果后放到输出机，输出结果。这中间的步骤全部都是人工操作，可以看成是人工操作系统。为了让大家更容易理解，我举一个例子，假设有一个医生，他能看病，然后患者来了，需要先挂号，再看病，看完了医生给处方，给完了处方，病人再拿着处方去买药。这里医生就是计算机，患者就是计算机的用户。

1947年，美国贝尔实验室发明了晶体管，直接导致了50年代中期的计算机更新换代，也就是第二代计算机。第一台晶体管计算机是1954年在贝尔实验室诞生的，一共用到了800个晶体管。这个计算机的体积已经明显减小了。u当病人在挂号或者在取药的时候，医生就只能在那傻坐着干等，造成了巨大的浪费。早期计算机运行速度慢的时候还好，当速度越来越快的时候，挂号取药的这点时间可能比看病的时间都长了。所以后来出现了批处理系统。就是在计算的输入输出两侧分别增加了磁带机。把成批的待计算任务输入到磁带机，然后计算机读取磁带，输出结果，再由磁带机输出到输出机，完成批量结果的输出。就是医院升级了，一次可以挂10个病人的号，然后在医生门口排队看病，10个人全看完以后，大家再一起排队取药。这样医生的工作效率就提高了。这个就是批处理系统。但是这样还有一点点问题，就是这10个人排队挂号、排队取药还是需要时间的，这段时间医生还是只能等待，没办法继续看病。

后来又发展出了脱机批处理系统，就是新增一个卫星机，和计算机保持脱离状态，卫星机的任务就是把任务、结果和磁带进行互转，主机只负责磁带的读取和输出，这个就是有点并行的意思了。就是说医院又升级了，专门雇了一个人负责组织排队挂号，排队取药。这个人的任务就是负责把一批一批的病人送到医生门口，然后把一批一批的病人从门口接走。这样的话计算机就不在需要因为外围的一些因素而停下来了。大家要知道早期的计算机非常昂贵，计算时间非常宝贵，能省一点是一点。

这时候看起来没什么问题，医生的工作时间都占满了。但有的时候医生没问题，患者可能有问题，比如一个患者刚进来看了一半的病，突然要上厕所，出门了。这时候医生只能是等着，其他别的患者也只能等着。这不是笑话么？

我们换一个角度来看任务的执行过程，假设内存里面有两个程序，A和B。正常按照顺序运行没什么问题。但是其中A程序在运行的过程中，被挂起，比如等待用户输入或者读取数据等其他原因。挂起的这段时间CPU是没事做的，只能等挂起结束才会继续运行A程序。A运行完了再运行B程序。这种叫单道程序系统。所以为了改善这个问题后来设计了新的系统，叫做多道程序系统，A程序挂起的时候，CPU别闲着，先去运行B，A可以运行了再去运行A，这样操作可以让CPU满负荷运载。这种情况在计算机系统里是经常出现的，比如等待从磁盘读取数据，等待从键盘获取输入等等。多道程序系统可以保证计算机只要有活干，那么就一定在干活。

然后到这里感觉设计的已经非常好了，可以让医生24个小时不休息的干活，看起来简直完美。但日子好了人们的需求也开始多了。这时候就有来看病的患者说了，每次来看病都得排队，能不能不排队啊？

于是计算机科学家就发明了分时系统。所谓分时系统，就是将多个程序同时装入内存，达到内存复用。同时时间上也可以复用，一个程序运行一会，挂起，运行别的程序，如此反复。就像一个医生说，来，你们十个人，都进来。于是一个人看病1分钟，看没看完不要紧，必须换下一个人。当把每个人单次看病的时间逐渐细分，比如0.1秒，这样在每一个病人看来，都是医生在给自己实时的看病，这样大家都很高兴。

但就是这样，还是有人不满意。为什么呢？因为有一次医院来了一个急诊病人，必须要抢救。可是医生这里前面还排队了100个患者等着看病呢。等轮到急诊患者的时候，人家已经去世了。所以系统缺少对这种紧急任务的处理。

后来为了应对这种情况，发明了实时系统。当一个紧急任务来了必须优先执行，立即抢占CPU，其他任务挂起，运行完成后再回到之前的任务。这类系统用于飞机飞行，导弹控制，对于实时性要求高的场景。这种系统对于紧急事件会优先响应。

早期的操作系统非常多样化，每个硬件厂商都有自己的系统，甚至每出一个新产品搭配的系统都是不一样的。一直到二十世纪六十年代，IBM开发了System/360系列机器，这些机器性能不一，但有着统一的操作系统：OS/360。这是操作系统发展的一个里程碑，IBM用行动告诉人们，计算机和它的系统是可以拆分开的，这个东西现在我们司空见惯，在当时却是一个非常重要的发明。

顺着IBM的思路走，在1965年的时候，AT&T贝尔实验室要建立一套多用户、多层次、多任务的操作系统。想整合当时所有的先进技术，实现远程拨号登录主机，用户可以在终端对计算机进行使用。但是在当时那个年代，目标过于激进，进度缓慢，最后还是被砍掉了。其实历史总是不断地在重演，产品要适应时代，反过来时代也能成就产品。远一点的比如平板电脑，微软早在上个世纪90年代就发明了，那时候比尔盖茨还是一个年轻小伙。但是由于那个年代计算机都没有普及，平板就显得过于超前。近一点的谷歌的Google glass，在2012年发布，当时大家以为世界要变了，但是受限于功耗、续航、价格等一些问题的限制，谷歌眼镜仅仅流行于一些极客中，没有普及到广大的群众，最后谷歌也把这个项目停掉了。这些产品都是因为太超前，而没有取得成功。

虽然贝尔实验室放弃了这个项目，但项目组其中一个人，名叫Ken Thompson，因为自己工作需要，想写一个小的作业系统，那么开发这个系统花了多长时间呢？答案是一个月。这个系统叫什么名字呢？答案是Unix。当他完成Unix的时候，他自己绝对想不到，这个系统会成为计算机世界的基石，这一年他26岁。由于这个贡献，后来他获得了图灵奖。又过了四年，1973年，他又发明了C语言，并把Unix进行了重写，并在1974年正式向外界披露了这个操作系统。顺便说一下，他还参与了Go语言的开发。

时间走到了1979年，Unix已经到了第七版，虽然系统是Ken Thompson开发的，但是版权还是归AT&T所有，出于商业考虑，他们竟然决定收回Unix的版权。这里学校受到了很大影响。这就好比微软和苹果打架，双方打得热火朝天，各有损伤，但是其他相关的小公司却经不起市场的波动，全都倒闭了。那么学校没有了教学用的操作系统，拿不到源代码，肯定是不行。

这时候有一位教授叫谭邦宁，决定自己写一个和Unix很像的操作系统，但又不能有版权问题。于是他闭着眼睛，也就是一眼Unix代码都不看，自己写了一个操作系统，专门用于教学的。这个系统叫Minix，就是Mini Unix的意思。由于并不是为了广大用户日常使用，所以系统的功能并不算强大，都是点到为止，教学够用就行了。不要小看这个几十年前的Minix操作系统，现在很多国内外的高校都还在用这个系统作为操作系统课程的示例。

那Minix在操作系统的历史长河中只能算是一个小插曲，那为什么我要单独拿出来讲一下呢？为的是引出我们这门课程的主角，Linux操作系统。Linux的诞生和Minix有着直接关系。



Linux是芬兰人Linus（李纳斯）发明的。他的外公是一位统计学教授，家里有一台电脑，李纳斯受外公的影响很小就已经会编程了，对计算机也非常的感兴趣。后来他在上大学的时候曾经花了一个暑假的时间读了一本书，名叫《操作系统：设计与执行》，这本书对他产生了很重要的影响，书籍的作者就是刚才提到的谭邦宁教授。再后来李纳斯购买了自己的第一台电脑，同时安装了Minix操作系统。但Minix还是有很多的问题，最大的问题就是没有终端。就像我们现在远程登录服务器需要一个终端程序一样，现在看起来稀松平常的东西，在当时可没有。于是李纳斯实在受不了，决定自己写一个。牛人和普通人就是不一样，普通人遇到些困难可能会绕着走，甚至往回走，牛人遇到困难直接干上去。要知道写一个终端需要从硬件层面开始设计，对个人的编程功力和耐心是极大的考验。耗时两个月后，李纳斯完成了。完成了以后李纳斯没有就这样停下来，而是继续添加磁盘驱动、文件系统等功能，一个操作系统的雏形已经出现了。

最终在1991年9月17日李纳斯发布了Linux第一版操作系统，一共10239行代码，现在看起来对一个操作系统来说太少了，但这可是李纳斯一个人的工作量。从1991年的1月购买自己第一台电脑，还是分期付款的，到发布Linux，李纳斯只用了不到10个月的时间，这个时候他只有21岁。大家现在的年龄和李纳斯相仿，如果给你10个月的时间，你能做出多大的成就呢？



我们这一门课是基于openEuler操作系统来讲的。openEuler是华为在2019年12月31日正式开源的操作系统。目前版本基于Linux 4.19内核。这个操作系统主要用在服务器端，但是经过裁剪或定制用在嵌入式设备也是可以的。我们讲操作系统，其实最重要的还是讲内核。那么什么是操作系统内核？它和操作系统有是什么关系呢？这些内容我们之后会逐步的探讨。

我们日常最常用的，接触最多的还是Windows系统。我从Windows1.0开始，一直到它最新的版本，给大家简单介绍一下，让大家有一个感性的认识。



微软的Windows操作系统第一个版本1.0于1985年11月发布，这个是微软第一个含有GUI界面的操作系统。这个版本的操作系统基于DOS。窗口之间还不支持重叠。这个窗口虽然简陋，但是已经有了滚动条的概念，这个概念一直延伸到现在。



1987年11月，微软发布了Windows2.0版本，新版本的Windows不同窗口可以互相遮挡了。



Windows3.0是微软非常重要的一个版本，发布于1990年5月，从这个版本开始，Windows被大众熟知。这一版本，加入了著名的纸牌游戏。



两年后的1992年3月，微软发布了Windows3.1版本，相对上一个版本，只有一些特性上的更新。这也是基于dos的最后一版Windows操作系统。因为这个时候微软在憋一个大招。



这个大招就是Windows NT 3.1。一个全新NT内核的操作系统，采用了全新的技术。这个操作系统发布于1993年7月。这和我们现在用的Windows系统形貌上已经差不多了。甚至这里的一些早期图标，在现在的系统中还能找到。



两年后微软又发了一个大招，Windows 95。发布于1995年8月。微软引入了窗口底部的任务栏，经典的开始菜单也有了。文件浏览窗口也出现了。这个设计，一直延续到今天。当年Windows95发售的时候，人都是排着大长队去抢购。



三年之后，1998年，微软又发布了一个重磅的版本，Windows98和95采用相同的内核。但是在视觉显示上面进行了大幅的优化



2000年，微软发布了Windows2000。增加了大量新特性，包括支持热插拔。



同年，微软还发布了Windows me。稳定性，资源占用都不好。是十分失败的一款产品，以至于很多人都不知道。



仅过了一年，微软就王者归来，发布了Windows XP，寿命最长的一款操作系统。一直到2014年微软在停止对其技术支持。但是遇到一些威胁性较高的漏洞，还是会打上补丁。目前仍然有人在用xp，尤其是一些较大的机构。



在XP收获了巨大成功以后，微软直到2007年才发布下一款操作系统，Windows Vista。当时这个系统的华丽界面拥有这毛玻璃效果，和xp相比就是质的飞跃。刚出来的时候有种第一次看见苹果手机的感觉。但是华丽的代价就是系统资源占用高。当时的主流机器根本跑步起来，卡的要死。为了体验这个操作系统，必须要花费大量金钱去升级硬件。大家罕见的为了操作系统去升级硬件。但是操作系统的兼容性还是不太好，主流的软件还是运行在xp系统上的，放到Vista上面经常跑不起来。总体来看，算不上一款成功的操作系统。



痛定思痛，两年后2009年微软发布了Windows 7。这款操作系统我相信大家应该都用过。视觉上保留了Vista的风格，但更加清爽。软件兼容性也好了很多。最重要的是，对于硬件的要求终于没有那么高了。于是这款系统一下子就普及开了。直到十几年后的今天，还是有人在用Windows 7。



微软有一个特点，每做出一款优秀的操作系统，下一款一定不怎样。2012年，Windows 8诞生。这款操作系统完全违反了用户的直觉。最经典的开始菜单竟然没有了，变成了一个Metro界面。同时微软做平板电脑的心不死，这款系统还支持触屏操作，所以各方面也做了相应的优化。甚至专门推出了平板电脑Surface。视觉化效果也都是扁平化的。但是消费者不买账啊，那段时间很多辅助软件都有一个功能，就是恢复经典的开始菜单。以至于在win8的升级版，8.1中，官方不得已恢复了开始菜单。



继续痛定思痛，微软在2015年发布了最新的操作系统Windows 10。各方面大幅更新，比如第一次引入了智能助手。同时Windows 不再像之前的系统一样每隔几年出一个新版。而是有新的系统功能，直接以补丁的形式更新。目前Windows 10 和刚发布时候的Windows 10 已经有了很大差别。截止到2020年的3月底，Windows10的活跃设备已经达到了10亿台。



这里是Windows这些年的一个发展历程时间表。操作系统那么多，为什么拿Windows来举例，因为Windows是目前使用人数最多，应用范围最广的操作系统，有良好的群众基础。这里面列的系统，大家或多或少都接触过或者使用过。了解历史才能创造未来。



下面说一下操作系统的内核



学习操作系统，首先要搞明白的第一件事就是什么是操作系统内核？它和操作系统有什么关系？我们假设新配置了一台电脑，电脑由最基本的硬件组成，这里列了CPU、内存和硬盘。实际上要更多。配置好电脑后我们开机肯定是不能用的，必须要有操作系统。硬件只是一个骨架，操作系统才是计算机的灵魂。于是我们安装了一个操作系统，接着我们借助操作系统，安装了各种软件，一台电脑就算配置好了。我们的用户软件和操作系统打交道，像申请内存，对按键做出反映，都需要操作系统提供接口。这个就属于是软件和系统之间的通信。同样的，操作系统收到了应用软件的请求后，需要往更底层去进行交互，比如从硬盘的某一块读取数据，开辟一段新内存空间等等。这里就属于是操作系统和底层硬件的通信了。这个时候，其实操作系统对于我们普通用户，或者说普通软件的开发者来说，依旧像是一个黑盒子，通过操作系统提供的api，可以完成很多事情。我们这一门课，就是要把这个黑盒子给拆开，看看里面到底有什么奥秘。这里简单的先拆一下，总体上由三部分组成，一部分是内核，直接和计算机硬件打交道，在硬件之上，有对外的接口，这是第二部分。这些接口等于说利用了内核的功能，进行了更加抽象的封装，提供给外部。同时利用内核的功能，还有系统软件，比如一些系统库，编译器，可视化系统等等。系统软件一般都是系统自带的。显而易见的，操作系统里最重要的就是内核部分。我们这节课就是要深入的研究一下它。



下面我用类比的方式来看看内核像什么。作为操作系统的核心，内核就像政府一样，会自发的做做一些事情，做一些有用的事情，维护着系统内的环境，比如时不时的清理一下垃圾。内核也像交通警察，各类程序就像是从各个方向驶来的车辆，内核要保证所有车辆安全驶过，井井有条，对于不那么听话的程序做好协调工作，要是真出了车祸也能尽快的恢复现场。同时内核也像基建，在操作系统内建好一条条的高速公路，让信息的传输畅通无阻，让每一个应用都可以快速的调用所需的资源。



具体来说内核又像是什么呢？首先内核也是软件，源码经由编译，得到二进制程序，然后执行指定的命令。内核也像一个驱动程序，必须要理解硬件，懂得对每一种不同硬件的操作方法。从用户的角度来看，内核也像一个服务器，用户向内核发起请求，得到回复。另一方面内核有更高的特权，可以直接访问用户空间，反过来则不可以。

内核大体上可以分类三种，单体内核、微内核和混合内核。单体内核自己承担了操作系统全部的功能，像是一个多面手，所有的活都自己干。微内核则是只负责系统底层最核心的一些功能，其他的功能都独立，每个部分负责自己的事。混合内核介于两者之间。



单内核像一把超级瑞士军刀，什么事情都能做，什么事情干的都好，哪里需要我，我就到哪里去。这样的设计优点是所有组件都在同一地址空间，性能非常好。但也有缺点，就是结构非常复杂，对维护、设计者的要求很高，容错性差。经典的DOS系统，早期的Windows 95、98系统。还有Linux都是单体内核。



微内核的设计思路和单体内核相反。对于不同的任务模块进行了解耦，操作系统由很多内核组成，每个内核负责一个功能。就像一个超级全的工具箱，里面每个工具都有自己的用途，需要干什么活拿什么工具，就算一个工具坏掉了也丝毫不影响其他工具。微内核最大的优点就是内核之间不再相互交错，每个内核有自己的内存空间，就算一个内核出了错，也不会对其他内核产生影响。常见的操作系统里，微内核的系统不多，大家比较熟悉的可能就是华为的鸿蒙OS以及谷歌的Fuchsia。

1. 混合内核介于两者之间我们用的新版Windows系统，以及新版的macOS都属于混合内核。混合内核吸纳融合了宏内核与微内核的优势。但是在后期技术人员维护的时候，就比较困难，因为它介于微内核和宏内核之间，对一些数据的传递以及权限的差异，很容易混淆。

openEuler是基于Linux开发的，所以属于是单体内核操作系统。前面说过，单体内核大而全，所有东西都在一起，维护和扩展都比较麻烦。Linux为了规避这个问题，采用了内核模块的设计。虽然是单内核，但是对于不同的任务还是分成了不同的模块来处理，模块与模块之间独立，支持动态加载，规避单内核的劣势。Linux内核包含五大子模块分别是：内存管理、进程管理、进程间通信、虚拟文件系统、网络接口。有了这五大模块，操作系统的根基就有了。就像一棵大树一样，只有树根够粗壮，那么上面的树干才能稳固，才能开花结果。



总结一下Linux内核的设计理念。第一是要简化，对于相同的指令集要复用，避免不必要的资源开销，包括总线协议、外设接口的实现都要简化。第二是标准化，统一接口标准，支持异构硬件和不同应用，只有标准化才能提高效率，减少无谓的冗余设计。第三是专业化，要体现出计算机科学的专业性，比如内存管理算法、调度算法等等，用高效的手段去解决实际问题。

1. 略。

下面来看一下Linux经过了多年的发展，都发生了哪些变化。Linux是芬兰人Linus（后称李纳斯）在自己21岁的时候发明的。那个时候操作系统还是Unix的天下。李纳斯将自己的第一台电脑安装了Minix操作系统，这个操作系统是mini Unix的缩写，但这个操作系统还是有很多的问题，最大的问题就是没有终端。就像我们现在远程登录服务器需要一个终端程序一样，现在看起来稀松平常的东西，在当时可没有。于是李纳斯实在受不了，决定自己写一个。牛人和普通人就是不一样，普通人遇到些困难可能会绕着走，甚至往回走，牛人遇到困难直接干上去。要知道写一个终端需要从硬件层面开始设计，对个人的编程功力和耐心是极大的考验。耗时两个月后，李纳斯完成了。完成了以后李纳斯没有就这样停下来，而是继续添加磁盘驱动、文件系统等功能，一个操作系统的雏形已经出现了。最终在1991年9月17日李纳斯发布了Linux第一版操作系统，一共10239行代码，现在看起来对一个操作系统来说太少了，但这可是李纳斯一个人的工作量。从1991年的1月购买自己第一台电脑，还是分期付款的，到发布Linux，李纳斯只用了不到10个月的时间，这个时候他只有21岁。大家现在的年龄和李纳斯相仿，如果给你10个月的时间，你能做出多大的成就呢？



到今天Linux已经快30岁了。围绕Linux系统已经有了无数的操作系统发行版，在服务器端也占有着极大的市场率。移动端我们常用的安卓操作系统也是基于Linux的。目前（截至2020年5月19日）最新版的Linux 已经到了5.6.13了。当前Linux的代码规模已经超过了2500万行，估值超70亿美元。



Linux从发布之初就是开源软件，遵循GPL协议。这个协议的意思就是如果你用了我的开源代码写了东西，你也得继续开源。GPL加持下的Linux就像瘟疫一样，迅速蔓延，保证了Linux的活力，直到今天，为Linux贡献过代码的程序员已经超过2万人。李纳斯本人仍然在参与Linux的维护。Linux的代码管理基于Git，顺便提一下，Git也是李纳斯发明的。后来Linux逐渐发展起来后有代码管理的需求，李纳斯选择了BitKeeper软件，这软件开始是免费的，后来开始收费。这回李纳斯又不爽了，于是自己开发了Git，现在风靡全球。现在你再去看BitKeeper的官网，会发现BitKeeper已经变成了一个开源软件，可以免费下载和使用，官方的讨论组也是异常的冷清，实在是让人感慨。不知道李纳斯下一次出手又会把谁打死呢？



Linux从发布之初到现在已经经过了近三十年，版本也是几经更迭。这张时间线可以一目了然的看出变化。Linux的版本编号规律主要分为三个阶段。第一个阶段是1.0之前，从0.01开始。按现在的说法是属于创业阶段，没什么规矩，差不多就行。第二个阶段是1.0到2.6之间，也就是从94年开始，Linux的版本号开始变得正规化了。版本号由3位组成，第一位表示大幅转变的内核，第二位表示内核产生了重大的修改，第三位表示轻微修改的内核。所以从94年到2011年我们看到后期都是2.x的版本，一直到2.6。按照命名规则，其实从2.0以来内核就没有过根本性的更改，所以第一位的意义也不太大了。从3.0开始就是第三个阶段的编号规律。第一位是主版本号，第二位是内核版本，第三位是安全补丁。目前最新版是5.6。我们的课程是基于Linux 4.19版本的，大家注意一下。



Linux作为一款开源操作系统，以为我们可以根据自己的需要进行修改和定制。比如我想做一个小型化的嵌入式物联网设备，不需要可视化的模块，就可以删掉，内核的安全性要加强，我也可以自己去更改原始代码，甚至新增一个模块也是可以的。基于这一点优势，市场上出现了无穷无尽的定制和衍生操作系统。我举一个最简单的例子。现在你打开阿里云，搜索云服务器，可以选择服务器操作系统。可选择的一共有10个，除了一个Windows Server，剩下的全都是Linux的不同发行版。另外国外非常强大的云计算服务商aws提供的操作系统里除了Windows Server也全部都是基于Linux，这足以说明Linux的强大和应用的广泛性。在服务器市场已经不是占据半壁江山了，而是处于统治地位。



在其他领域Linux的身影也无处不在，目前使用最广泛的安卓操作系统，就是基于Linux内核修改，进行二次开发得来的。事实上很多大公司，比如谷歌、英特尔、红帽都是Linux内核的主要代码贡献者。近年来物联网发展迅速，每个设备硬件都不尽相同，使用环境差别也非常大，所以对操作系统的定制化要求非常高。这个时候Linux可定制、高度模块化的优点又显现出来。所以可以看到，目前市面上几乎所有的物联网设备、智能设备、消费级的嵌入式设备都是基于Linux的。我想李纳斯当年做梦也没有想到，自己业余时间随手开发的操作系统在三十年后变的无处不在，并持续的发光发热吧。



除了定制化和模块化外，Linux对于不同指令集的支持也是非常广泛的。如果你拿到一份Linux的源码，打开根目录下的arch文件下，就能看到当前的系统都支持哪些指令集了。我这里的例子是4.19版本的Linux源码。可以看到目前支持24中不同的指令集，包括最常用的x86，arm。以及开源指令集riscv等等。基于这个特点我们也可以推断出来，Linux上的应用程序移植起来也是非常方便的。

到这里我们对Linux的历史以及发展都有了一个初步的了解。但还是比较浮于表面，下面我们要揭开Linux内核的面纱，去深入的学习和剖析内核。

最后放一张Linux内核开发组人员的合照，中间躺着的那位就是李纳斯。希望有一天能看到在座的各位和李纳斯的合影。

1. 略