1. Linux的老大哥Unix的历史

早期的计算机体格非常庞大而且架构使用困难，输入设备只有卡片阅读机（在卡片上以打洞的方式输入程序相关的信息，然后在插入卡片阅读机来将信息读入主机运算）。这种方式如果程序有一点写错了就得重新打卡，加上主机少，用户多光是等待时间耗去很多时间。

之后，计算机硬件与操作系统有了改良，引入了键盘进行信息的输入，不过主机少，多人使用的时候还是要等待。好在1960年代初麻省理工学院发展了“兼容分时系统（Compatible Time-Sharing System, CTSS）”，它可以让大型主机通过提供多个终端机（terminal）以联机进入主机的方式，来利用主机的资源进行运算工作。但是当时先进的主机大概也只能提供30不到的终端机。

为了加强大型主机的功能，以让主机资源可以提供更多使用者来利用，在1965年前后，由贝尔实验室（Bell）、麻省理工学院（MIT）及奇异公司（GE）共同发起Multics的计划。不过在1969年前后，计划进度落后，资金也短缺，在认为Multics计划不能成功之后，贝尔实验室退出了计划的研究工作。而其中有一位参与Multics计划的人员Thompson因为自己需要开发一个小的操作系统以提供自己需求。所以，他以汇编语言写出了一组核心程序，包括一些核心工具程序以及一个小文件系统。这个核心程序将Multics庞大复杂的系统简化了不少，实验室里朋友都戏称它为Unics。这个系统就是Unix的原型。这个文件系统重要概念就是所有的程序或系统装置都是文件。

因为Unics系统非常好用在贝尔实验室广为流传，而且经过了数次改版。但是Unics是汇编语言编写的，移植性不好，每次安装到不同机器需要重新编写汇编语言，很不方便。后来，Dennis Ritchie将B语言重新改写成C言语，Thompson与Ritchie合作将Unics使用C语言重新改写，最后发行出Unix的正式版本。

因为Unix的高度可移植性与强大性能，深受很多商业公司（AT&T、IBM、HP等公司）的喜爱，纷纷在自家主机上搭配了自己的Unix操作系统。由于当时生产计算机的硬件公司没有统一标准，因此只有配合自家计算机硬件开发合适的Unix系统。另外，没有厂商针对个人计算机设计Unix系统，所以在早期没有支持个人计算机的Unix操作系统。早先的Unix系统只能与服务器或者大型工作站划等号！

到了1979年，AT&T推出的System V第七版Unix，这一版本最重要特色就是可以支持x86架构的个人计算机系统，也就是System V可以在个人计算机上安装运行了。由此AT&T在商业上考虑下想将Unix的版权收回去。所以，AT&T在1979年发行的第七版Unix中，特别提到了“不可对学生提供原始码”的严格限制。于此同时在业界造成了紧张气氛，引爆了很多商业纠纷。

这一版权严重影响了学习教Unix核心原始码相关学问的教授教学。其中有一位叫Andrew Tanenbaum（谭宁邦）教授，为了课程能够继续，谭宁邦教授开始动手写Minix核心程序，并且强调Minix能与Unix兼容，在1986年终于完成，次年出版Minix相关书籍。但是Minix不是完全免费的需要购买磁盘/磁带，当然Minix原始码也会附带在磁盘上，学者可以学习Minix的核心程序。这个特色对Linux的起始阶段有着很大关系。

1. GNU计划发展史

在1984年，Richard Mathew Stallman（史托曼）发起了GNU计划，这个计划目的是：建立一个自由、开发的Unix操作系统（Free Unix）。但是建立一个操作系统谈何容易，而且当时的GNU是仅有自己一个人单打独斗，实在太麻烦了。因为操作系统太过复杂，史托曼开始着手在Unix上面运行的小程序。起初史托曼参考Unix上面现有的软件，依据这些软件作用开发出功能相同的软件，开发期间不看其他软件原始码，避免吃官司。因为史托曼开发的软件GNU软件都是免费且功能和专利软件差不多，于是更多的人转向使用GNU软件，逐渐GNU打开了知名度。在1985年，史托曼为了避免GNU所开发的自由软件被其他人所利用而成为专利软件，所以他和律师草拟了通用公共许可证（General Public License, GPL）。

不论什么软件都必须编译成二进制文件后才能够执行，史托曼就想如果能写出一个不错的编译程序，那不是大家都需要的软件吗？因为当时C语言编译软件版本太多而且都是专利软件。因此他便开始写起了C语言的编译软件，这个软件就是现在都非常有名的GNU C Compiler（gcc）。开始GCC编写的并不顺利，为此他将原先写过的Emacs编辑器写成可以在Unix上个面跑的软件，并公布原始码。Emacs是一种程序编辑器，用户在编写程序过程中进行程序语法检验，此功能深受程序设计师喜欢，当时史托曼凭借Emacs以磁带出售赚了一些钱，后而开始全力撰写其他软件，并成立自由软件基金会（FSF，Free Software Foundation）。在1990年GCC终于完成了，同时他还写了C函数式库（GNU C library）和用来操作操作系统的基本接口软件BASH shell。

1. Linux的发展历史

一说到Linux操作系统大家都知道创始人Linus Torvalds（托瓦兹）。托瓦兹的外祖父是赫尔辛基大学的统计学家，他的外祖父为了让他学点东西，所以从小就将托瓦兹带到身边管理一些微计算机。在这个时期托瓦兹学习了硬件架构和汇编编程语言。1988年托瓦兹顺利进入赫尔辛基大型就读计算机科学系。在就学期间他接触到Unix操作系统，因为但是赫尔辛基大型只有一部最新的Unix系统仅供16个终端机。光是等待Unix使用时间就耗去很长时间，无法满足托瓦兹的需求。于是他就想自己搞一部Unix来玩，但当时Unix不允许想学生提供原始码，要自己写核心程序谈何容易！不过托瓦兹知道了一个类似Unix的操作系统且兼容Unix上的工具软件，而且还能在Intel 386机器上运行。就是前面提到的谭宁邦教授撰写的Minix系统。托瓦兹购买了最新的Intel 386的个人计算机后，立刻安装了Minix操作系统并学习了Minix核心原始码，了解核心程序的设计理念。

虽然Minix操作系统非常好用，但是谭宁邦教授不愿意对其功能加强和维护。导致使用Minix操作系统的工程师对功能方面是欲求不满，于是托瓦兹就想为什么不自己写一个操作系统。他开始编写起了核心程序，好在这是已经有了GNU提供bash工作环境软件和Gcc编译程序等自由软件，让托瓦兹顺利编写了核心程序。他参考Minix的设计理念和书上的程序代码，然后使用GNU自由软件将核心程序代码与386电脑结合在一起，最终写出了他想要的核心程序，并真的能够在386电脑上跑起来还可以读取Minix的文件系统。这个事情令他非常高兴，他也希望这个程序能够获大奖一些修改建议，于是他便把核心程序放在网络上提供大家下载，同时在BBS上面贴了一则消息：他说他完成了一个小小的操作系统，这个核心程序可以在386机器上运行。这则新闻引起了很多人注意，他们也去托瓦兹提供的网站上下载了这个核心程序并安装。因为托瓦兹放置核心程序那个FTP网站目录为Linux，为此大家都称这个核心程序为Linux了。

同时，托瓦兹为了让自己的Linux能够兼容于Unix系统，在Unix系统上能够运行的软件在Linux上也希望能可以运行。于是，他开始参考标准的POSIX规范，开始修改Linux核心程序。因为POSIX标准是针对Unix与一些软件运行时候的标准规范，只要依据这些标准规范来设计的核心程序与软件，理论上就可以搭配在一起执行了。Unix遵循了POSIX标准规范，Linux也遵循了POSIX标准规范。因此，Unix上运行的软件在Linux上也可以运行了，这一步导致Linux的使用率大大增加。

因为Linux放在网络上供大家下载使用，有更多的人帮忙编写相关的驱动程序和软件，对于Linux核心所欠缺的项目，他总是“先求有且能跑，再求进一步改良”的心态，让Linux使用者和志工得到相当大的鼓励，因此Linux获得巨大进步，发布的版本也是非常快，快到一个星期不到就推出新的版本了。另外为因应这种随时都有程序代码加入状况，于是Linux便逐渐发展成具有模块的功能。即某些功能独立于核心外，在需要的时候才加载到核心中，大大增加了Linux核心的可维护能力。越来越多的人加入到Linux大军之中，逐渐发展成以个虚拟团队，团队里的人们都是来自世界各地彼此都没有见过面。在1994年发布了Linux的核心正式版本version 1.0。这一版本还加入了X Window System的支持。1996年发布了2.0版本、2011年发布3.0版本、到了2015年有了4.0版本，Linux发展非常迅速，如今各大中小互联网企业基本都设计Linux操作系统。此外，托瓦兹指明了企鹅作为Linux的吉祥物。