操作系统的启动

18340225 钟婕

一、 基础知识：

1、 引导系统：

装入内核以启动计算机的过程

2、 引导程序（引导装载程序）Boot：

2.1 定义：

绝大多数计算机系统都有一小块代码，即为引导程序或者称之为引导装载程序。这段程序代码能够**定位**内核，将它**装入**内存，开始执行。有些计算机系统，例如 PC 机，采用两步完成：一个简单的引导程序从**磁盘**上调入一个较复杂的引导程序，而后后者再装入内核。

2.2 作用：

引导程序可以完成一系列任务。通常，一个任务是要通过运行诊断程序来确定机器的状态。如果诊断通过，程序可按启动步骤继续进行，那么此时，系统的所有部分都可以被初始化，包括从 CPU 寄存器到设备控制器，以及内存的内容。

基于上述引导与准备过程，操作系统得以启动。

除此之外，由于操作系统的适用性不同，对于一些小型操作系统例如在手机、游戏控制台的操作系统保存在 ROM 中；而对于一些大型操作系统例如 Windows、MAC OS、Unix 等，引导程序保存在固件中，而操作系统保存在磁盘中。

为了运行计算机系统，必须初始化 CPU 且在固件系统中启动执行引导程序。如果操作系统本身也在固件系统中，那么引导程序可以直接启动操作系统，否则，引导程序必须逐步地从固件或者磁盘中装载更聪明地程序，直到操作系统本身被装入内存并且执行。

3、 BIOS：

基本输入输出系统。其实，它是一组固化到**计算机**内**主板**上一个 **ROM 芯片**上的**程序**，它保存着计算机最重要的基本输入输出的程序、开机后自检程序和系统自启动程序，它可从 CMOS 中读写系统设置的具体信息。其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。

二、 操作系统的启动过程：

宏观过程：

1、BIOS 程序首先将存储设备的引导记录（Boot Record）载入内存，并执行引导记录中的引导程序（Boot）；

2、引导程序会将存储设备中的操作系统内核载入内存，并进入内核的入口点开始执行

3、后操作系统内核完成系统的初始化，并允许用户与操作系统进行交互

细致分析：

1、 BIOS 程序执行：

CPU 加电后初始化：

把 CPU 所有寄存器的值设为默认值，除了将 CS 寄存器的值改为 0xFFFF，将其他寄存器的值都置为 0。根据 CS 和 IP 寄存器的值就可以找到指令的物理地址 **0xFFFF:0x0000**，也就是 0xFFFF0。在 CPU 初始化之后，CPU 就可以找到初始时指令的物理地址，并且从这个位置开始执行，此时这里存放了一条 JMP 指令，指令跳转到 BIOS 的真正启动代码处。

BIOS 加电后自检：

跳转到 BIOS 的真正启动代码处之后，BIOS 首先会执行 POST（即加电后自检）。加电自检主要是检测系统中一些关键设备是否存在和能否正常工作。如果检测到出现问题，那么主板会发出蜂鸣，启动中止。如果没有问题，屏幕就会显示出 CPU、内存、硬盘等信息。

BIOS 复制后继续执行：

BIOS 程序在执行完成必要的开机自检和 CPU 初始化后，会将自己复制到从 **0xA0000** 开始的物理内存中并继续执行。

BIOS 加载引导扇区：

BIOS 开始搜寻可引导的存储设备(即根据用户指定的引导顺序从软盘、硬盘或是可移动设备)。如果找到，则将该存储设备中的引导扇区读入物理内存 **0x7C00** 处，并且跳转到 物理内存 0x7C00 处继续执行。

2、 Boot 程序执行：

简而言之,其实Boot程序执行的主要目的是为了加载loader程序。

Boot 程序利用了 BIOS 提供的中断服务程序，读取软盘文件系统中的根目录，同时在根目录中搜寻 **loader.bin** 文件，如果找到，则将整个 loader.bin 文件加载到从地址 0x1000 开始的物理内存中，并且程序跳转到 0x1000 地址处开始执行。

3、 loader 程序执行：

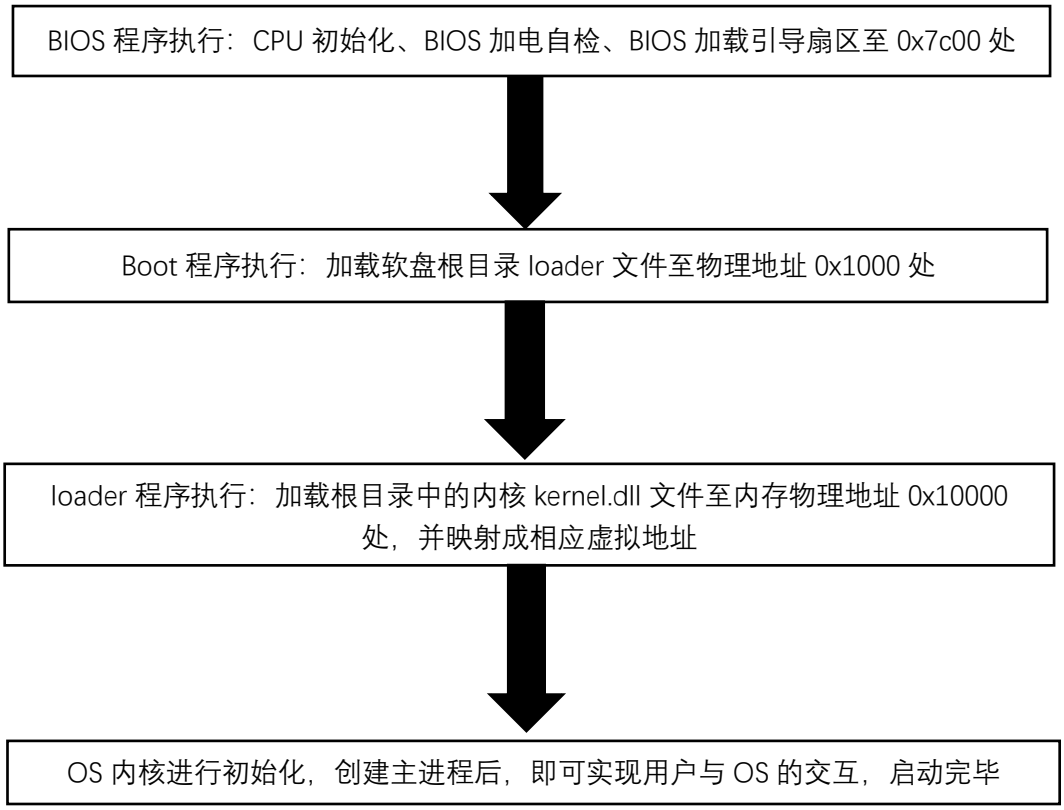
类似于 Boot 程序，loader 程序的执行目的也是为了将其他程序加载到内存中，loader 程序是为了将操作系统内核加载到内存中。除此之外，Loader 程序还负责检测内存大小，为操作系统内核提供保护模式执行环境等工作。

loader 程序在软盘的根目录中搜寻 kernel.dll 文件，找到后将整个文件加载到地址为 **0x10000** 的物理内存中，然后根据分页机制，将该物理内存地址映射成虚拟地址 **0x80000000**，完成后，程序跳转到 kernel 程序的入口地址处执行。

4、 **操作系统内核初始化程序：**

内核初始化过程主要是初始化处理器和中断、各个管理模块、最后创建主进程。接着启动控制台程序，这样就可以实现用户与操作系统的交互，进行应用程序的执行了。

总结上述的操作系统启动的总体过程：



三、常见操作系统启动

Windows 的启动过程包括以下几个阶段：

启动自检阶段：

1、这个阶段主要是读取 BIOS ，然后内存，CPU，硬盘，键盘等设备进行自检。这个阶段在屏幕上显示就是自检的那些打印信息。

屏幕显示：自检的打印信息

初始化启动阶段：

2、这个阶段根据 BIOS 指定的启动顺序，找到可以启动的优先启动设备，比如本地磁盘，CD Driver ， USB 设备等等，然后准备从这些设备启动系统。

屏幕显示：黑屏

Boot 加载阶段：

3、这个阶段首先从启动分区(比如 C 盘) 加载 Ntldr ，然后 Ntldr 做如下设置：

4、内置内存模式，如果是 x86 的处理器，并且操作系统是 32 位，则设置为 32-bit flat memory mode，如果是 64 位操作系统 + 64 位处理器，则设置为 64 位内存模式。

5、启动文件系统

6、读取 boot.ini 文件

屏幕显示：黑屏，如果按 F8 或者多系统时会显示启动选项菜单。

检测和配置硬件阶段：

7、这个阶段检查和配置一些硬件设备，它们分别是：

—系统固件，比如时间和日期

—总线和适配器

—显示适配器

—键盘

—通讯端口

—磁盘

—软盘

—输入设备（如鼠标）

—并口

—在 ISA 总线上运行的设备

屏幕显示：黑屏

内核加载阶段：

8、在内核加载阶段，Ntldr 将首先加载 Windows 内核 Ntoskrnl.exe 和 硬件抽象层 (HAL). HAL 有点类似于嵌入式操作系统下的 BSP（Borad support

package)，这个抽象层对硬件底层的特性进行隔离，对操作系统提供统一的调用接口，操作系统移植到不同硬件时只要改变相应的 HAL 就可以，其它的内核组件不需要修改，这个是操作系统通常的设计模式。

9、接下来 Ntldr 从 HKEY_LOCAL_ MACHINE\SYSTEM\CurrentControlSet 下读取这台机器安装的驱动程序，然后依次加载驱动程序。

驱动程序加载完成后，Windows 做如下设置：

10、 创建系统环境变量

11、 启动 win32.sys ，这个是 Windows 子系统的内核模式部分。

12、 启动 csrss.exe，这个是 Windows 子系统的用户模式部分。

13、 启动 winlogon.exe

14、 创建虚拟内存页面文件

15、 对一些必要的文件进行改名，（主要是驱动文件，如果更新后，需要在下次重启前改名）

屏幕显示：显示 Windows logo 界面和进度条

登录阶段：

16、 这个阶段会做如下几件事：

17、 启动机器上安装的所有需要自动启动的 Windows 服务

18、 启动本地安全认证 Lsass.exe

19、 显示登录界面

屏幕显示：显示登录界面

四、疑难知识点：

1、 BIOS 中主要存放的程序有哪些？

BIOS 中主要存放的程序包括：**自诊断程序**（通过读取 CMOS RAM 中的内容识别硬件配置，并对其进行自检和初始化）、**CMOS 设置程序**（引导过程中，通过特殊热键启动，进行设置后，存入 CMOS RAM 中）、**系统自动装载**

程序（在系统自检成功后，将磁盘相对 0 道 0 扇区上的引导程序装入内存使其运行）和**主要 I/O 驱动程序**和**中断服务**（BIOS 和硬件直接打交道，需要加载 I/O 驱动程序）。

2、 BIOS 自检结束后的启动顺序是怎样的？

BIOS 需要有一个外部存储设备的排列顺序，排在前面的设备就是优先启动、引导的设备。这种排序叫做“启动顺序”（Boot Sequence）。可以在 BIOS 的操作界面中设置启动顺序。

BIOS 按照“启动顺序”，把程序控制权转交给排在第一位的存储设备。根据用户指定的引导顺序从软盘、硬盘或是可移动设备中读取启动设备的 MBR，并放入指定的位置（0x7c000）物理内存中。

3、 主引导记录：

计算机读取“启动顺序”中排在第一的设备的第一个扇区，也就是读取最前面的 512 个字节。如果这 512 个字节的最后两个字节是 0x55 和 0xAA，表明这个设备可以用于启动；如果不是，表明设备不能用于启动，程序控制权于是被转交给“启动顺序”中的下一个设备。这最前面的 512 个字节，就叫做”主引导记录”（Master boot record，缩写为 MBR）

主引导记录只有 512 个字节，放不了太多东西。它的主要作用是，告诉计算机到硬盘的哪一个位置去找操作系统。

主引导记录由三个部分组成：

（1） 第 1-446 字节：调用操作系统的机器码。
（2） 第 447-510 字节：分区表（Partition table）。
（3） 第 511-512 字节：主引导记录签名（0x55 和 0xAA）。

4、 分区表：

分区表的长度只有 64 个字节，里面又分成**四项**，每项 16 个字节。所以，一个硬盘最多只能分**四个一级分区**，又叫做“主分区”。每个主分区的 16 个字节，由 6 个部分组成：

（1） 第 1 个字节：如果为 0x80，就表示该主分区是激活分区，控制权要转交给这个分区。四个主分区里面**只能有一个是激活的**。

（2） 第 2-4 个字节：主分区**第一个扇区**的物理位置（柱面、磁头、扇区号等等）。

（3） 第 5 个字节：主分区类型。

（4） 第 6-8 个字节：主分区**最后一个扇区**的物理位置。

（5） 第 9-12 字节：该主分区第一个扇区的逻辑地址。

（6） 第 13-16 字节：**主分区的扇区总数**。

5、 补充细节：

在阅读资料过程，本人对 BIOS 的执行过程不太理解，因此找到以下资料，详细介绍了 BIOS 的执行过程。

BIOS 启动细节：

a)按下电源开关，电源就开始向主板和其它设备供电；当芯片组检测到电源已经开始稳定供电了，它便撤去 RESET 信号；CPU 马上就从地址 FFFF:0000H 处开始执行指令，放在这里的只是一条跳转指令，跳到系统 BIOS 中真正的启动代码处。

b) 系统 BIOS 的启动代码首先进行 POST(Power－On Self Test，加电后自检)。由于 POST 是最早进行的检测过程，此时显卡还没有初始化，如果系统 BIOS 在进行 POST 的过程中发现了一些致命错误，例如没有找到内存或者内存有问题，那么系统 BIOS 就会直接控制喇叭发声来报告错误，声音的长短和次数代表了错误的类型；在正常情况下，POST 过程进行得非常快，几乎无法感觉到它的存在。POST 结束之后就会调用其它代码来进行更完整的硬件检测。

c) 接下来系统 BIOS 将查找显卡的 BIOS。前面说过，存放显卡 BIOS 的 ROM 芯片的起始地址通常设在 C0000H 处，系统 BIOS 在这个地方找到显卡 BIOS 之后就调用它的初始化代码，由显卡 BIOS 来初始化显卡。此时多数显卡都会在屏幕上显示出一些初始化信息，介绍生产厂商、图形芯片类型等内容，不过这个画面几乎是一闪而过。系统 BIOS 接着会查找其它设备的 BIOS 程序，找到之后同样要调用这些 BIOS 内部的初始化代码来初始化相关的设备。

d) 查找完所有其它设备的 BIOS 之后，系统 BIOS 将显示出它自己的启动画面，其中包括有系统 BIOS 的类型、序列号和版本号等内容。

e) 接着系统 BIOS 将检测和显示 CPU 的类型和工作频率，测试所有的 RAM，并同时在屏幕上显示内存测试的进度。可以在 CMOS 设置中自行决定使用简单耗时少或者详细耗时多的测试方式。

f) 内存测试通过之后，系统 BIOS 将开始检测系统中安装的一些标准硬件设备，包括硬盘、CD-ROM、串口、并口和软驱等设备，另外绝大多数较新版本的系统 BIOS 在这一过程中还要自动检测和设置内存的定时参数、硬盘参数和访问模式等。

g) 标准设备检测完毕后，系统 BIOS 内部支持即插即用的代码将开始检测和配置系统中安装的即插即用设备。每找到一个设备之后，系统 BIOS 都会在屏幕上显示出设备的名称和型号等信息，同时为该设备分配中断、DMA 通道和 I/O 端口等资源。

h) 到这一步为止，所有硬件都已经检测配置完毕了，多数系统 BIOS 会重新清屏并在屏幕上方显示出一个表格，其中概略地列出了系统中安装的各种标准硬件设备，以及它们使用的资源和一些相关工作参数。

i) 接下来系统 BIOS 将更新 ESCD(Extended System Configuration Data，扩展系统配置数据)。ESCD 是系统 BIOS 用来与操作系统交换硬件配置信息的一种手段，这些数据被存放在 CMOS(一小块特殊的 RAM，由主板上的电池来供电)之中。通常 ESCD 数据只在系统硬件配置发生改变后才会更新，所以不是每次启动机器时都能够看到“Update ESCD… Success”这样的信息。不过，某些主板的系统 BIOS 在保存 ESCD 数据时使用了与 Windows 9x 不相同的数据格式，于是 Windows 9x 在它自己的启动过程中会把 ESCD 数据修改成自己的格式。但在下一次启动机器时，即使硬件配置没有发生改变，系统 BIOS 也会把 ESCD 的数据格式改回来。如此循环，将会导致在每次启动机器时，系统 BIOS 都要更新一遍 ESCD，这就是为什么有些机器在每次启动时都会显示出相关信息的原因。

j) ESCD 更新完毕后，系统 BIOS 的启动代码将进行它的最后一项工作：即根据用户指定的启动顺序从软盘、硬盘或光驱启动 MBR。在这个过程中会按照启动顺序顺序比较其放置 MBR 的位置的结尾两位是否为 0xAA55，通过这种方式判断从哪个引导设备进行引导。在确定之后，将该引导设备的 MBR 内容读入到 0x7C00[1]的位置，并再次判断其最后两位，当检测正确之后，进行阶段 1 的引导。

五、总结：

通过这次的热身学习，对于操作系统有了初步的了解，在网上查询资料、阅读文献的过程遇到一些困难，例如对于一些缩写的含义、一些专有名词的理解不清楚等，在自学过程中，不断查询这些含义，虽然还不能算理解，但最起码了解。虽然只是一个热身项目，但是已经点燃了自己的学习激情，不会的东西有很多，需要再努力好好研究学习，不能荒废了。本次自学过程中，查看了一些文献的同时观看了一些相关的视频讲解，这大大有助于我的理解。在之后的学习过程可以继续尝试这种方式！

六、参考文献：

- https://blog.csdn.net/wchstrife/article/details/78879554
- 《operating system concepts》（seventh edition）by Abraham Silberschatz