**OS（H）实验1-1**

**树莓派启动过程和原理调研报告**

**一. 树莓派的启动流程**

对于传统计算机，一个专门用于存储引导加载操作系统的BIOS将用来加载操作系统。而树莓派省略了这一步——它将引导文件直接放在了SD卡中。需要注明的是，树莓派有多种启动方式，第一种默认的是SD卡启动，还有USB或GPIO启动，但其原理都应是一样的，因此本人只叙述实验中用到的SD卡启动。

树莓派中央的BROADCOM2837B0芯片里有一个只读存储器ROM，当为树莓派供电时，它将首先加载这个ROM中的程序——出场时已经写死不可更改，使其在它硬链接的一个RISC上执行代码——这称为启动的**第一阶段**。ROM中的代码会挂载SD卡中FAT32文件系统，使其可以进行第二阶段的启动。在这期间，树莓派的CPU和RAM并没有启动。

**第二阶段**，树莓派读取SD卡中名为bootcode.bin的文件，并用它将bootloader.bin加载到GPU的L2 Cache里——这将启动RAM，并可以在SD卡中检索GPU固件start.elf，然后运行该固件，启动GPU——ARM CPU依旧没有启动，它仍处于复位状态。接下来，将由GPU中的一个特殊系统VideoCore IV完成启动过程。

进入**第三阶段**，GPU这时接管任务，读取SD卡中名为config.txt的文件——它的用途类似于我们上文谈到的BIOS，里面存储着启动时的各项参数，诸如ARM CPU主频、GPU主频等。加载完config.txt后，start.elf仍旧在工作——它寻找cmdline.txt（如果存在），一个包含内核运行参数的文件；最后，start.elf加载kernel.img，将RAM移交给CPU，结束CPU的复位状态，开始正式运行操作系统。

**二. 树莓派启动与传统PC启动的区别——以x86启动为例**

上文已经提到，BIOS是PC机中主管CPU、GPU、RAM等硬件初始化的，而树莓派所不同于此，而与嵌入式系统有所相似的是，它将这些任务都交给了Bootloader程序。 而一般地，x86还会有一个CMOS——记录各项硬件参数的一个嵌在主板上的存储器。

x86的BIOS是第一个启动的程序，它检查计算机硬件是否完备以能够基本运行（这称为硬件自检），然后寻找能够启动的硬盘，并在这个硬盘里读取它的MBR——启动管理程序，这是在硬盘上安装操作系统时操作系统设置的。接下来，MBR加载操作系统的核心文件，核心文件将接管接下来的工作，读取并启动操作系统。

一般地，PC启动时使用的是RAM和CPU——GPU更多的是一个图形加速硬件。而在树莓派中，GPU掌管了操作系统启动的绝大多数任务，而只在最后启动CPU。

从一，二两个部分的解释中，可以基本看全二者启动的不同之处了。

**三. 树莓派启动所使用到的文件系统**

第一部分提到的文件系统是**FAT32**——内含从启动的第二阶段到第三阶段中用到的全部文件。

接下来，由于我们启动的是Linux内核，所以内核会加载SD卡中的第二个分区——**ext4**文件系统，启动init完成系统初始化。（当然了，如果你树莓派里装的是Windows 10，那么这个ext4估计要改成NTFS了）。

**参考文献：**

**[1] config.txt**

[**https://www.raspberrypi.org/documentation/configuration/config-txt/README.md**](https://www.raspberrypi.org/documentation/configuration/config-txt/README.md)

**[2] HOW-THE-RASPBERRY-PI-BOOTS-UP**

[**https://thekandyancode.wordpress.com/2013/09/21/how-the-raspberry-pi-boots-up/**](https://thekandyancode.wordpress.com/2013/09/21/how-the-raspberry-pi-boots-up/)

**[3] x86: The Boot Process**

**https://docs.oracle.com/cd/E19683-01/817-6958/hbsysboot-30637/index.html**