

实验二：移植U-boot和挂载NFS文件系统

实验目的

1

掌握U-boot的配置、编译和运行

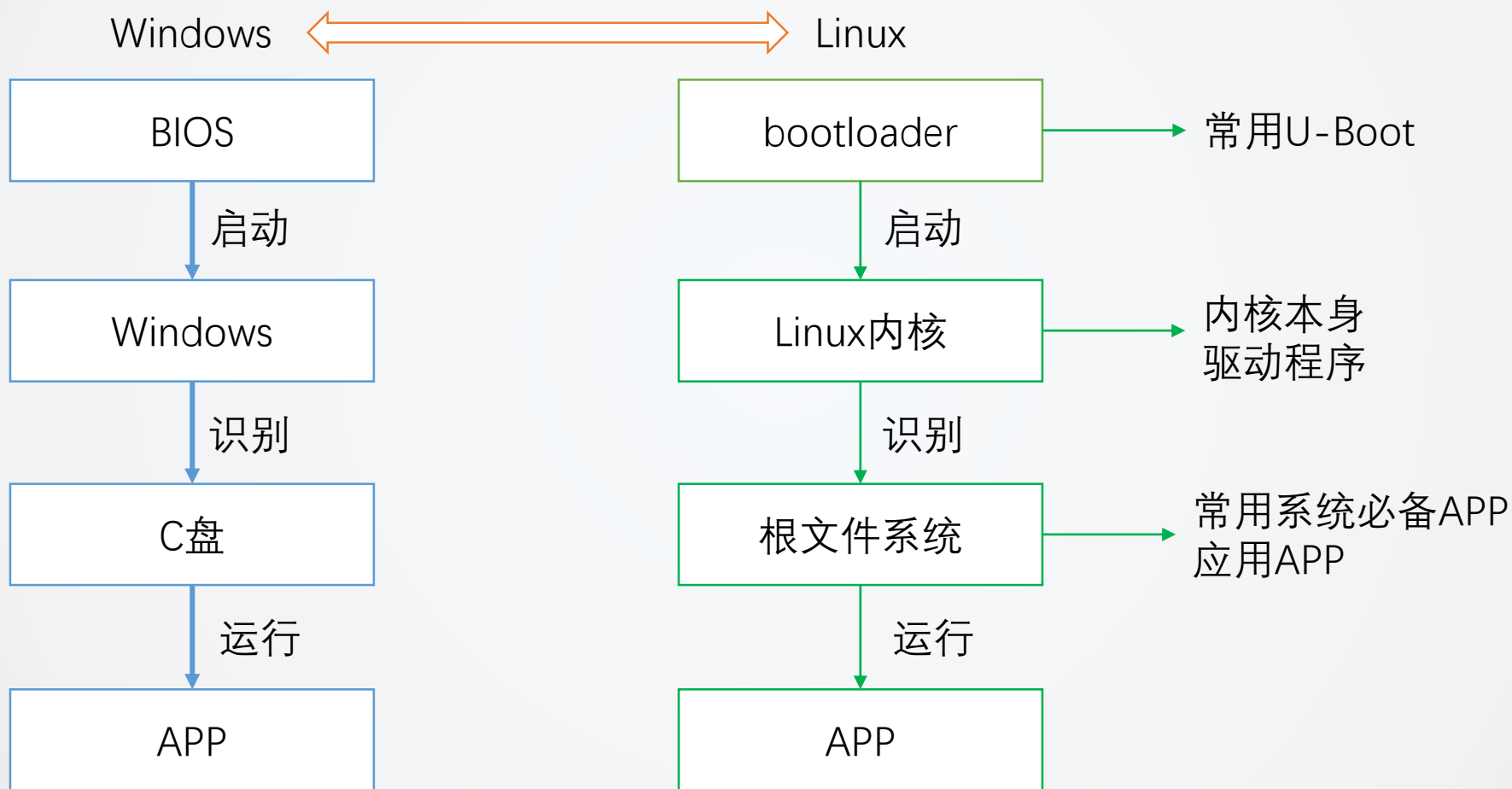
2

掌握NFS（网络文件系统）的配置和挂载

1

嵌入式Linux启动流程

■ 嵌入式Linux系统，就相当于一套完整的PC软件系统



1 嵌入式Linux启动流程

■ 嵌入式Bootloader

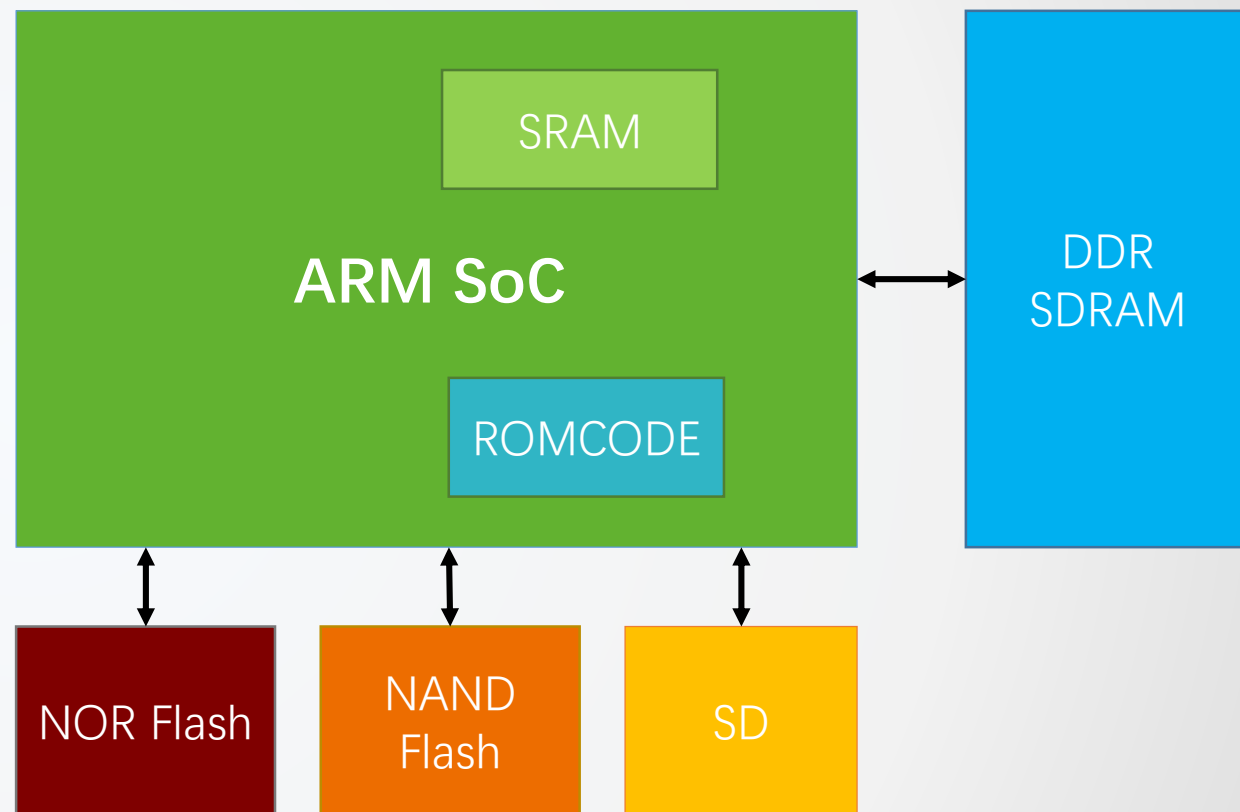
- 检测硬件是否正常
- 加载操作系统镜像到RAM
- 设置不同的启动方式

■ 常见的启动方式

- NOR/NAND flash启动
- 从SD卡启动
- 从网络加载Linux启动

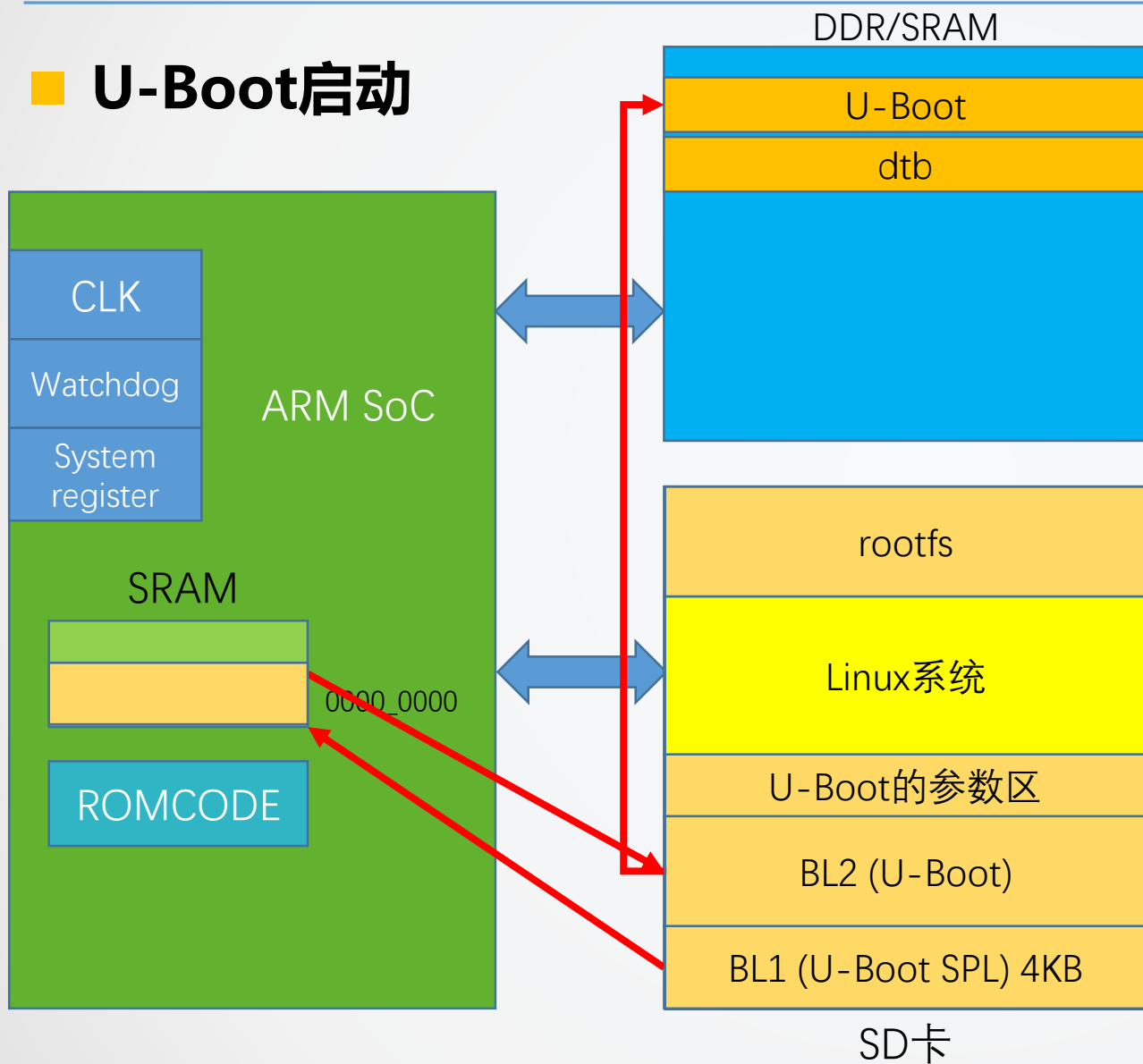
请思考

U-boot是系统上电后运行的第一行代码?



1 嵌入式Linux启动流程

■ U-Boot启动



(1) ROM: 初始化存储器接口、建立存储映射、根据CPU外部管脚/eFuse值来判断系统的启动方式。

(2) SD卡前4KB的代码复制到芯片内部的SRAM中执行，映射SRAM到零地址，然后在这4kB代码中进行各种初始化、代码复制、重定位等工作。

(3) 加载U-Boot

1 嵌入式Linux启动流程

■ **U-Boot参数:** bootm 0x60003000 - 0x60800000

➤ Linux内核编译

```
ulmage LOADADDR=0x60003000
```

```
...
```

```
LD vmlinux ①
```

```
SORTTAB vmlinux
```

```
SYSMAP System.map
```

```
OBJCOPY arch/arm/boot/Image ②
```

```
Kernel: arch/arm/boot/Image is ready
```

```
GZIP arch/arm/boot/compressed/piggy_data ③
```

```
AS arch/arm/boot/compressed/piggy.o ④
```

```
LD arch/arm/boot/compressed/vmlinux ⑤
```

```
OBJCOPY arch/arm/boot/zImage ⑥
```

```
Kernel: arch/arm/boot/zImage is ready
```

```
UIMAGE arch/arm/boot/ulmage ⑦
```

```
Image Name: Linux-5.15.30
```

```
Created: Wed May 11 21:05:01 2022
```

```
Image Type: ARM Linux Kernel Image  
(uncompressed)
```

```
Data Size: 5008888 Bytes = 4891.49 KiB = 4.78 MiB
```

```
Load Address: 60003000
```

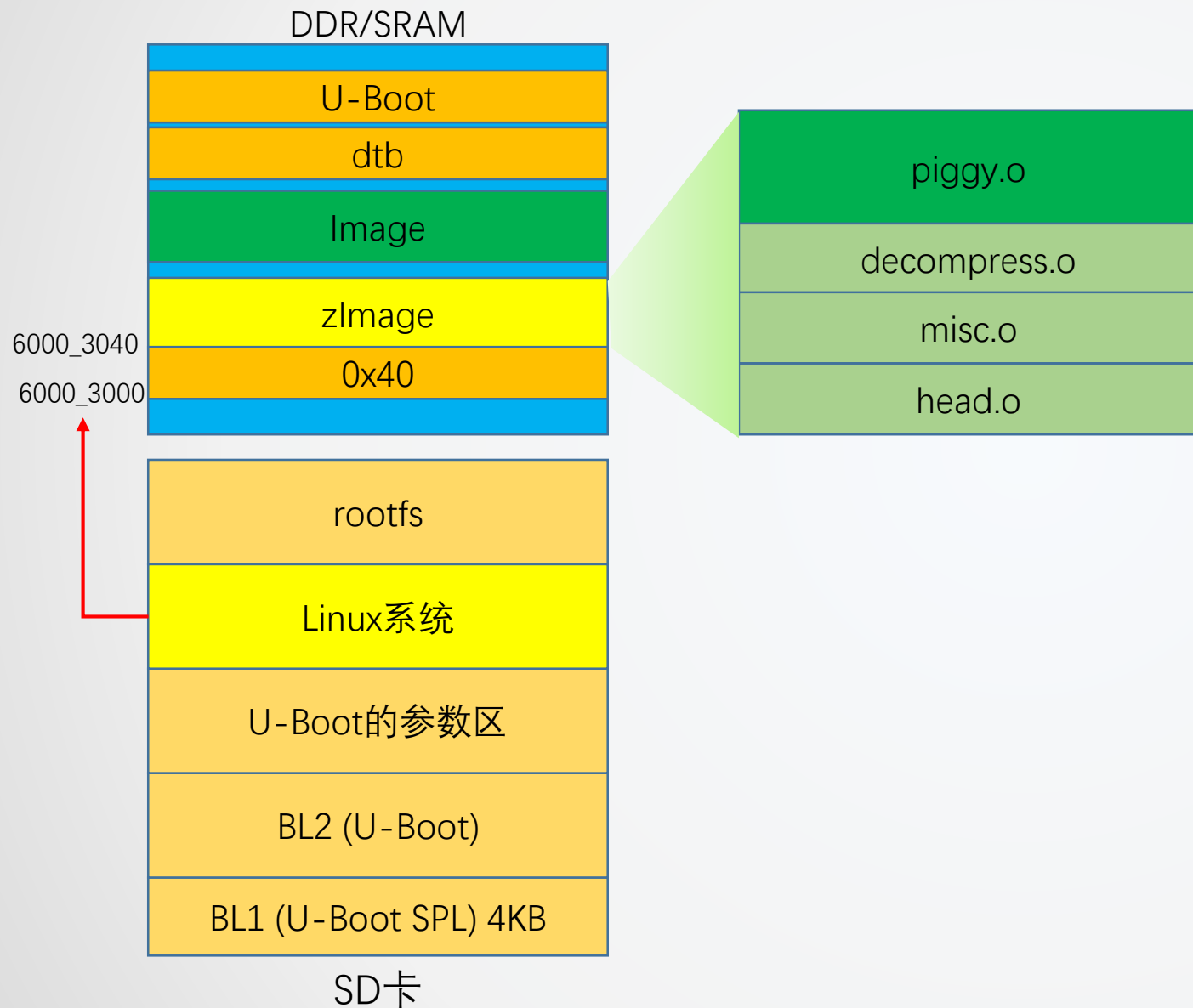
```
Entry Point: 60003000
```

```
Kernel: arch/arm/boot/ulmage is ready
```

```
root@cs-virtual-machine:/home/cs/linux-5.15.30#
```



1 嵌入式Linux启动流程



(4) bootm解析ulImage的0x40(64)字节的数据头，解析出指定的加载地址。

(5) 运行解压缩代码head.o\misc.o\decompress.o，解压出真正的内核镜像Image

(6) 加载Image，运行Linux内核真正的代码

1 嵌入式Linux启动流程

```
## Booting kernel from Legacy Image at 60003000 ...
Image Name:   Linux-5.15.30
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    5008928 Bytes = 4.8 MiB
Load Address: 60003000
Entry Point:  60003000
Verifying Checksum ... OK
## Flattened Device Tree blob at 60800000
Booting using the fdt blob at 0x60800000
Loading Kernel Image
Loading Device Tree to 7fe6e000, end 7fe74700 ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 5.15.30 (root@cs-virtual-machine) (arm-linux-gnueabi-gcc (Ubuntu 9.4.0-lubuntu1~20.04.1) 9.4.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #5 SMP Tue May 10 21:53:37 CST 2022
CPU: ARMv7 Processor [410fc090] revision 0 (ARMv7), cr=10c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
OF: fdt: Machine model: V2P-CA9
OF: fdt: Ignoring memory block 0x80000000 - 0x80000004
Memory policy: Data cache writeback
Reserved memory: created DMA memory pool at 0x4c000000, size 8 MiB
OF: reserved mem: initialized node vram@4c000000, compatible id shared-dma-pool
cma: Reserved 16 MiB at 0x7ec00000
Zone ranges:
  Normal   [mem 0x0000000060000000-0x000000007fffffff]
Movable zone start for each node
Early memory node ranges
  node    0: [mem 0x0000000060000000-0x000000007fffffff]
Initmem setup node 0 [mem 0x0000000060000000-0x000000007fffffff]
CPU: All CPU(s) started in SVC mode.
percpu: Embedded 15 pages/cpu s30156 r8192 d23092 u61440
Built 1 zonelists, mobility grouping on. Total pages: 130048
Kernel command line: root=/dev/nfs rw nfsroot=192.168.232.140:/home/rootfs,v3 init=/linuxrc console=ttyAMA0 ip=192.168.232.100
printk: log_buf_len individual max cpu contribution: 4096 bytes
printk: log_buf_len total cpu_extra contributions: 12288 bytes
printk: log_buf_len min size: 16384 bytes
printk: log_buf_len: 32768 bytes
printk: early log buf free: 14828(90%)
```

内核引导阶段stage1:

判断内核是否支持该架构、判断内核是否支持该板型、建立一级页表、禁止ICache、Dcache、使能MMU、复制数据段、清BSS、保持CPU ID到processor_id变量、调用start_kernel

内核启动阶段stage2:

获取Uboot传递给内核的信息、设置体系结构相关的环境、初始化内存结构、开启MMU、建立页表、初始化各种硬件（GPIO、串口、I2C、网口、USB），加载各个硬件的驱动……

挂载根文件系统，启动init进程

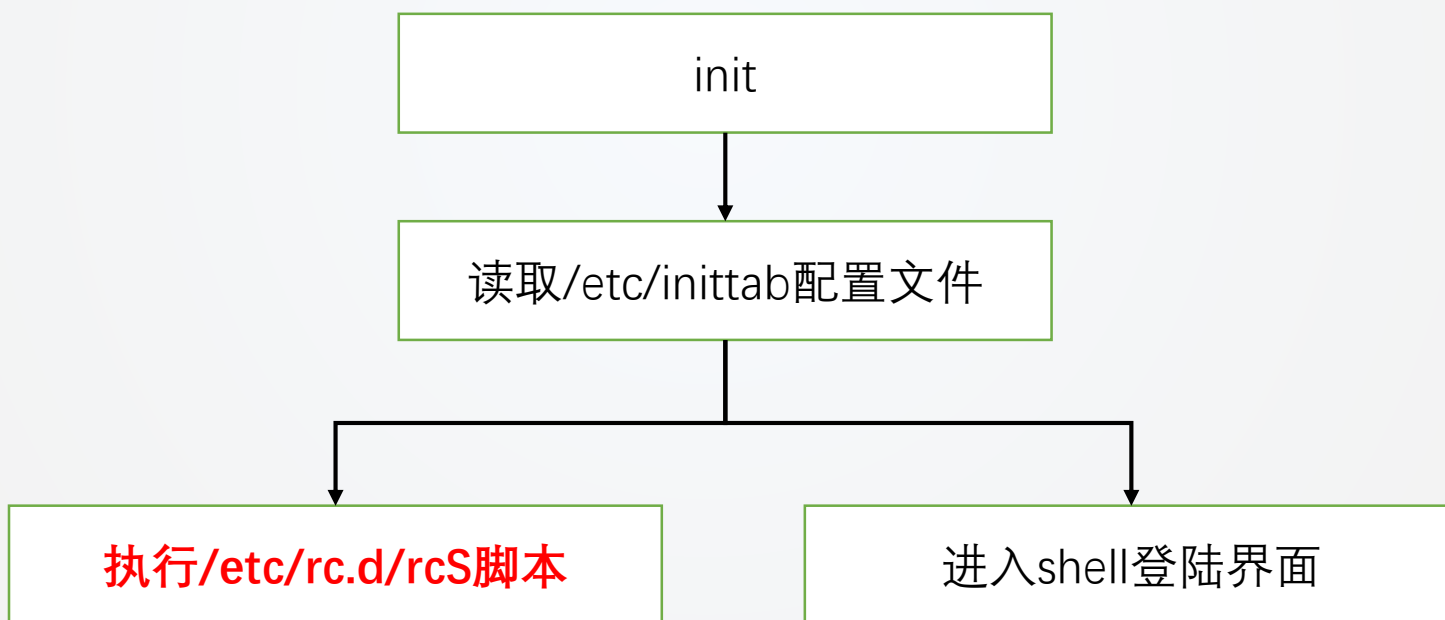
1

嵌入式Linux启动流程

■ 挂载根文件系统目的：

- 安装适当的内核模块，以便驱动某些硬件设备或启用某些功能
- 启动存储于文件系统的init服务，以便让**init服务**接手后续的启动工作。

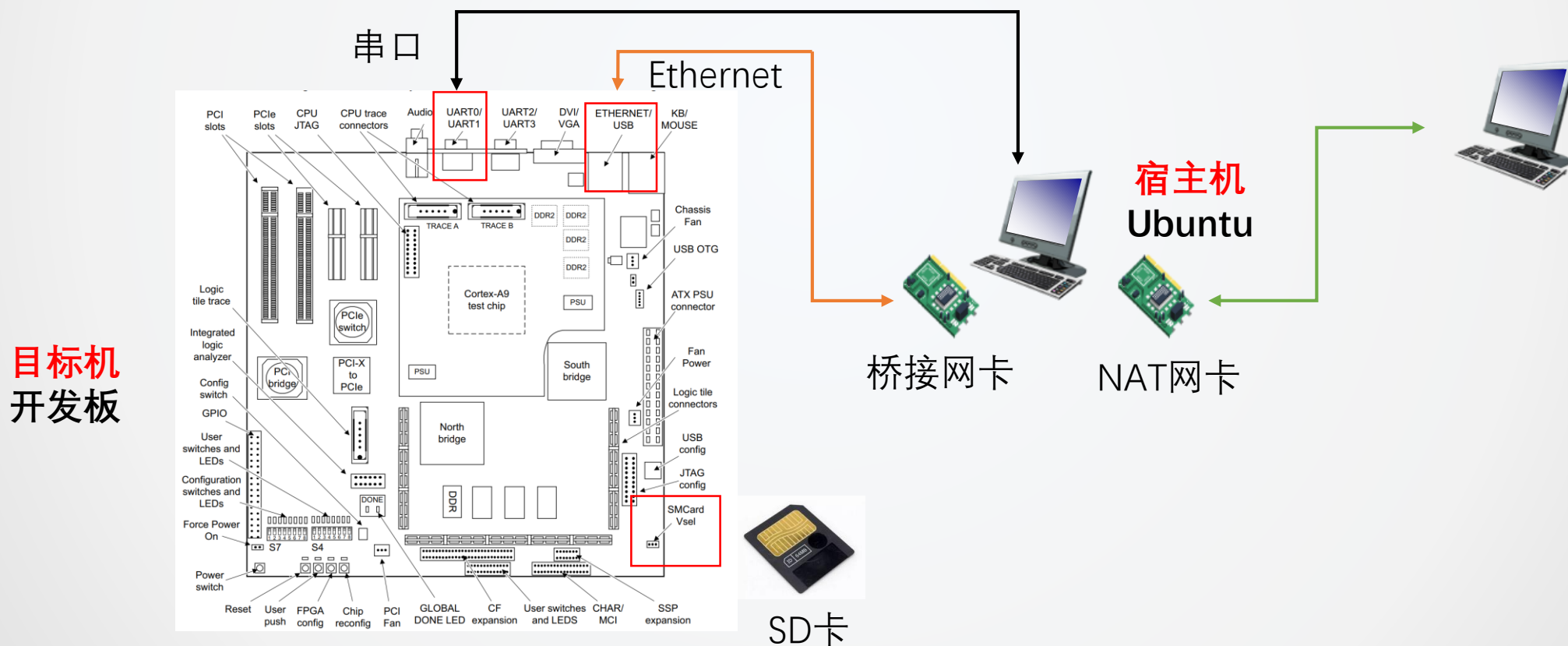
■ init服务初始化过程



2 NFS文件系统

■ NFS: Network File System网络文件系统

- 镜像格式的文件系统必须先烧录到硬盘后才能挂载，适用于出货产品。
- NFS采用C/S架构，客户端通过网络去挂载服务器端上的文件系统，适用于调试阶段。



实验步骤

- 编译运行u-boot
- 设置QEMU网络功能
- 使用u-boot自动化加载Linux内核和SD卡根文件系统
- 开启NFS服务、修改u-boot、Linux支持NFS、挂载NFS根文件系统

实验要求及分数说明

■ 实验任务：

1. 完成U-Boot的编译和运行，并能引导Linux和挂载SD卡根文件系统。
2. 修改U-Boot代码，使得在U-Boot启动阶段打印自己的学号和姓名。
3. 完善根文件系统，使得在进入shell之前打印欢迎信息、你的学号和姓名。

■ 实验分数：

- 回答实验报告中的问题：20%
- 完成任务1：40%
- 完成任务2：20%
- 完成任务3：20%

作业提交

■ **提交内容：**实验报告（有模板）

■ **截止时间：**

实验课后一周内提交至HITsz Grader 作业提交平台，具体截止日期参考平台发布。

- 登录网址：： <http://grader.tery.top:8000/#/login>
- 推荐浏览器： Chrome
- 初始用户名、密码均为学号，登录后请修改

注意

上传后可自行下载以确认是否正确提交

**同学们
请开始实验吧！**