OPERATING SYSTEMS

LAB 1 DESIGN REVIEW

张远航 甘睿彤

Where to place the bootblock

- “在MIPS 架构下，启动程序被放在了启动设备（U 盘，硬盘等）的第一个扇区，系统启动时该扇区被loadboot 自动加载在内存地址0xa0800000 处”

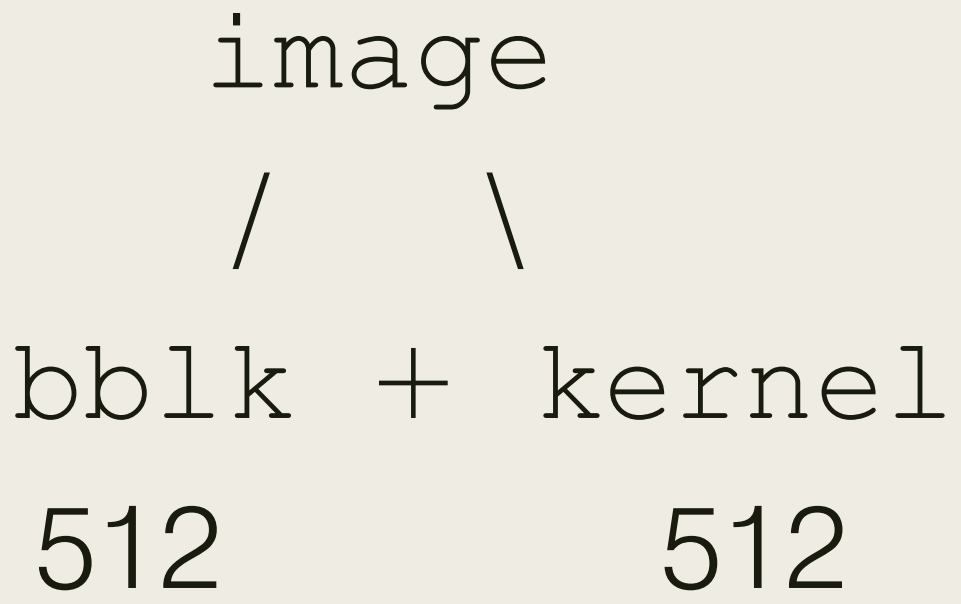
How to move kernel from disk to memory & Where to place the kernel in the memory

- 怎样从磁盘读内核?
  - PMON中的读盘函数
  - 三个参数
    - 读取字节数 512
    - SD卡内部偏移量 512 (bblk—kernel)
- 将内核放在内存什么位置?
    - 读取目的地址（内存中） 0xa0800200
      - 原因：bootblock占了512(200H)字节

Where the kernel entry point is

- “将内核读取在内存的0xa0800200 处，而kernel.c中main 函数的偏移地址为0x6c, 那么在bootblock.s 读完内核之后就需要跳转到0xa080026c 处执行”

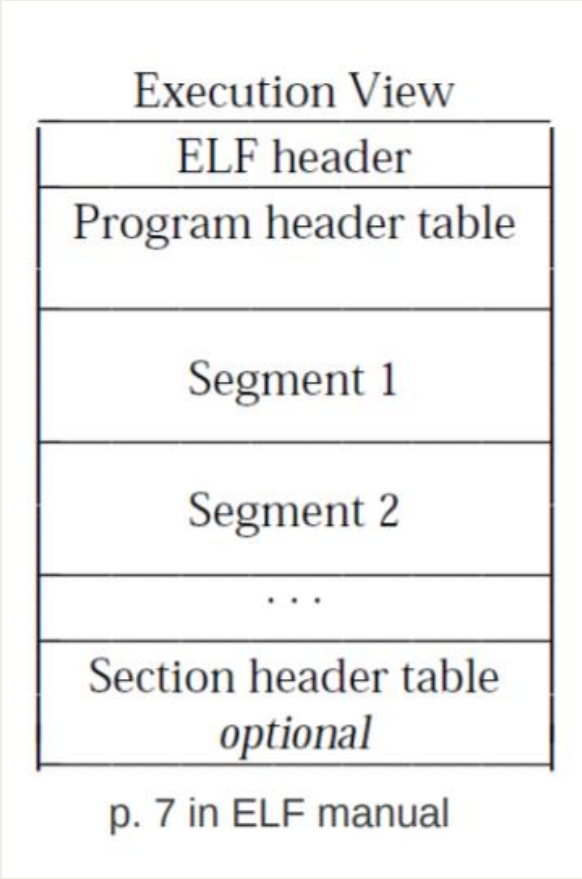
可执行文件bootblock和kernel在内核镜像的什么位置存放?



可执行文件(ELF)的特点？怎样读取可执行文件？

- read in ELF header
- read each program header
  - *Executable segments:*
    `p_type == PT_LOAD`

`PT_LOAD`	The array element specifies a loadable segment, described by `p_filesz` and `p_memsz`. The bytes from the file are mapped to the beginning of the memory segment. If the segment's memory size (`p_memsz`) is larger than the file size (`p_filesz`), the "extra" bytes are defined to hold the value 0 and to follow the segment's initialized area. The file size may not be larger than the memory size. Loadable segment entries in the program header table appear in ascending order, sorted on the `p_vaddr` member.



p. 7 in ELF manual

How to create `image`

- 利用ELF头信息找到首个代码段
  - *assuming at least one program i.e.* `e_phnum > 0`
  - *1st program occurs at (Program Header Start Offset + Num headers * Program Header Size):* `e_phoff + e_phentsize`
- 这一代码段应当放在image的什么位置?
  - *Sec 2.1, "**Base Address**" section of ELF documentation*
  - *"To compute the base address, one determines the memory address associated with the lowest* `p_vaddr` *value for a* `PT_LOAD` *segment. One then obtains the base address by truncating the memory address to the nearest multiple of the maximum page size."*

How to create `image` (cont'd)

- 需要做padding
  - *force padding by writing* `p_memsz > p_filesz`
  - *at the end, pad to nearest sector*