

OS lab1 上机实验报告

一、思考题

Thinking 1.1

也许你会发现我们的 `readelf` 程序是不能解析之前生成的内核文件(内核文件是可执行文件)的, 而我们之后将要介绍的工具 `readelf` 则可以解析, 这是为什么呢? (提示: 尝试使用 `readelf -h`, 观察不同)

二者数据的存储大小端不一致, 内核文件是大端存储, 而 `testelf` 文件是小端存储。

```
17373126_2019_jac@stu-113:~/17373126-lab$ readelf -h ./gxemul/vmlinux
```

ELF Header:

```
  Magic:   7f 45 4c 46 01 02 01 00 00 00 00 00 00 00 00 00
  Class:                               ELF32
  Data:                               2's complement, big endian
  Version:                           1 (current)
  OS/ABI:                             UNIX - System V
  ABI Version:                        0
  Type:                               EXEC (Executable file)
  Machine:                            MIPS R3000
  Version:                            0x1
  Entry point address:                0x80010000
  Start of program headers:          52 (bytes into file)
  Start of section headers:         36716 (bytes into file)
  Flags:                              0x50001001, noreorder, o32, mips32
  Size of this header:                52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:          2
  Size of section headers:           40 (bytes)
  Number of section headers:          14
  Section header string table index: 11
```

```
17373126_2019_jac@stu-113:~/17373126-lab$ readelf -h ./readelf/testELF
```

ELF Header:

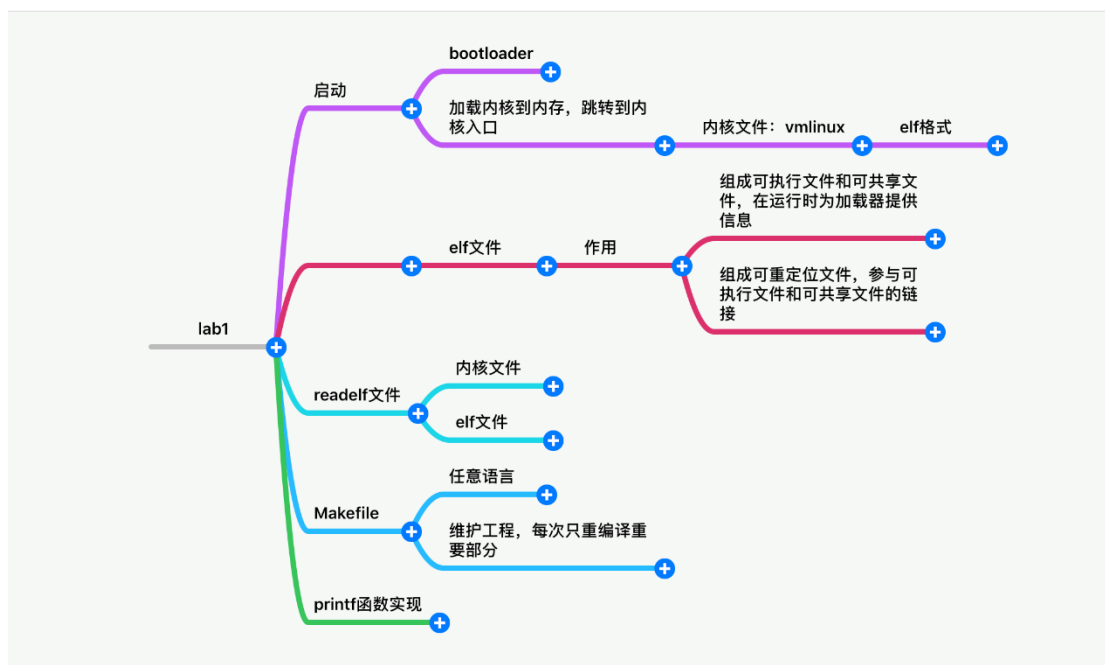
```
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                               ELF32
  Data:                               2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                             UNIX - System V
  ABI Version:                        0
  Type:                               EXEC (Executable file)
  Machine:                            Intel 80386
  Version:                            0x1
  Entry point address:                0x8048490
```

Start of program headers:	52 (bytes into file)
Start of section headers:	4440 (bytes into file)
Flags:	0x0
Size of this header:	52 (bytes)
Size of program headers:	32 (bytes)
Number of program headers:	9
Size of section headers:	40 (bytes)
Number of section headers:	30
Section header string table index:	27

Thinking 1.2 main 函数在什么地方？我们又是怎么跨文件调用函数的呢？

main 函数在 init/main.c 这个源文件中，通过编译生成.o 文件，然后通过 linker 对所有目标文件进行连接，链接后填补连接前单易目标文件调用函数语句的地址空缺

二、实验难点图示



三、体会与感想

感觉每一步都很迷茫，跟着实验指导书虽然能做出来，但是不知道为什么这样做就是对的，网上的资料也很少，只能一种一种方法去试验。

四、残留难点

基本全部吧，没有多少是完全懂的