

lab1-extra内容介绍

评测截止日期：2020/3/22-23:00

lab1-extra内容介绍

完成与提交实验的方式

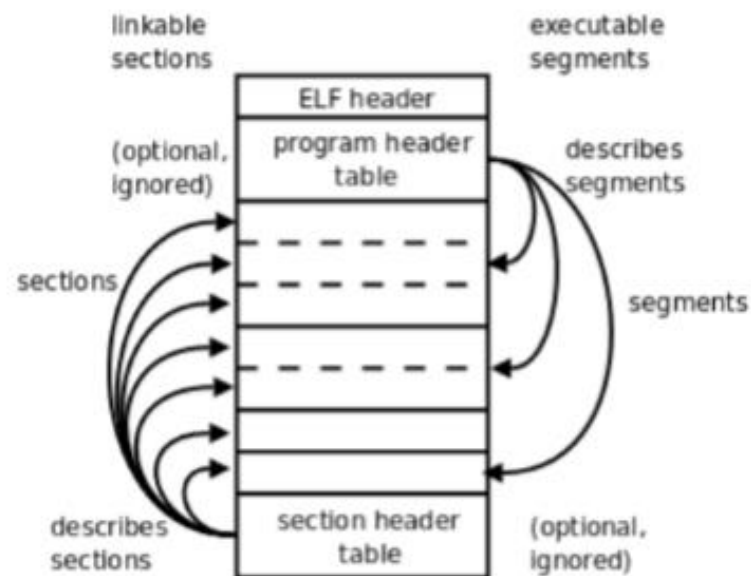
1. 使用git branch检查并确保自己在lab1分支下
2. 将本地改动提交，以防切换分支造成混乱（即add与commit操作）
3. git checkout -b lab1-extra #新建并切换到名称为lab1-extra的分支，注意大小写
- 4. 在lab1-extra分支下完成实验内容**
5. git add --all #接下来这三步用来提交实验代码
6. git commit -m "xxx"
7. git push origin lab1-extra:lab1-extra

lab1-extra内容介绍

题目背景

ELF文件从整体来说包含5个部分：

1. ELF Header，包括程序的基本信息。
2. Program Header Table，主要包含程序中各个Segment的信息。
3. Section Header Table，主要包含各个Section的信息。
4. Segments，就是各个Segment。
5. Sections，就是各个Section。



lab1-extra内容介绍

题目背景

ELF 标识，即 ELF 文件头结构最开始的 **16 个字节**，作为一个数组 `e_ident[EI_NIDENT]`。它的各个索引位置的字节数据有固定的含义，如右图所示。

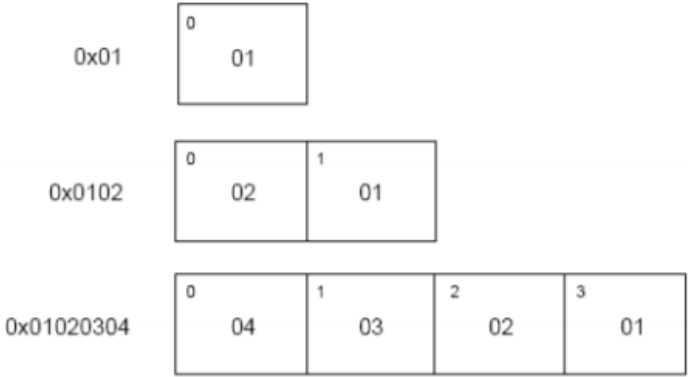
其中 `e_ident[EI_DATA]` 部分，指明了目标文件中的数据编码格式，0 对应非法编码，1 对应小尾端编码，2 对应大尾端编码。

名字	值	用途
EI_MAG0	0	文件标志
EI_MAG1	1	文件标志
EI_MAG2	2	文件标志
EI_MAG3	3	文件标志
EI_CLASS	4	文件类别
EI_DATA	5	编码格式
EI_VERSION	6	文件版本
EI_PAD	7	补齐字节开始位置
EI_NIDENT	16	<code>e_ident[]</code> 数组的大小

lab1-extra内容介绍

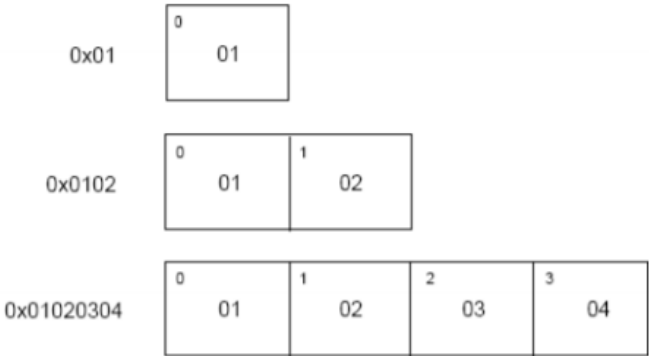
题目背景

小尾端编码：在一个多字节的数据中，“权值”比较小的低位数在低字节。



小尾端编码数据存储示意图（左侧为数据，右侧为数据在内存中分布）

大尾端编码：在一个多字节的数据中，“权值”比较大的高位数在低字节。



大尾端编码数据存储示意图（左侧为数据，右侧为数据在内存中分布）

lab1-extra内容介绍

lab1-extra测试题目

题目内容：修改 `readelf.c` 中的代码，使得我们的`readelf`函数可以根据ELF Header中`e_ident[EI_DATA]`记录的信息对elf文件的编码格式进行判断，**既可以正确解析小尾端编码的elf文件，也可以正确解析大尾端编码的elf文件**。需要输出 elf 文件的所有 Program Header的序号(从0开始)及其所记录的相应的Segment的 `filesize`和`memsize`信息。

格式要求：对每个Program Header，输出格式为：`"%d:0x%x,0x%x\n"`，三个标识符分别代表**序号**、**filesize**和**memsize**。

lab1-extra内容介绍

正确结果样例

当同学们正确完成了对大小尾端elf文件的解析后，使用编译得到的readelf可执行程序，应该即可以正确解析testELF（小尾端elf文件），也可以正确解析的vmlinux（大尾端elf文件）

```
jovyan@b5d98ddbe03c:~/117373439-lab/readelf$ ./readelf testELF
0:0x120,0x120
1:0x13,0x13
2:0xa44,0xa44
3:0x11c,0x124
4:0xc8,0xc8
5:0x44,0x44
6:0x44,0x44
7:0x0,0x0
8:0xec,0xec
jovyan@b5d98ddbe03c:~/117373439-lab/readelf$ ./readelf ../gxemul/vmlinux
0:0x18,0x18
1:0x8b90,0x8b90
```

注意，图中结果只是示例，可能与你的输出结果不同，请使用readelf工具比较判断自己是否输出正确。(readelf -l <elf-file>)

lab1-extra内容介绍

实验提示

- 同学们可以参考《see mips run linux》第10、11章内容完成实验。
- 有关ELF文件的更多知识，可以参考《ELF手册》了解。

lab1-extra内容介绍

完成与提交实验的方式

1. 使用git branch检查并确保自己在lab1分支下
2. 将本地改动提交，以防切换分支造成混乱（即add与commit操作）
3. git checkout -b lab1-extra #新建并切换到名称为lab1-extra的分支，注意大小写
4. **在lab1-extra分支下完成实验内容**
5. git add --all #接下来这三步用来提交实验代码
6. git commit -m "xxx"
7. git push origin lab1-extra:lab1-extra

lab1-extra内容介绍

评测结果说明

正确完成实验要求后提交代码，可以看到如右图所示评测结果，得到100分。

其中，testcase 1/6~3/6测试对小尾端elf文件的解析。testcase 4/6 ~6/6测试对大尾端elf文件的解析。

评分细则：readelf.c编译成功，可得10分。

小尾端elf文件解析成功可得60分。

大尾端elf文件解析成功可得30分。

```
[ Compile success! readelf found. ]
[ PASSED:25 ]
[ TOTAL:25 ]
[ You have passed readelf testElf testcase 1/6 ]
[ PASSED:24 ]
[ TOTAL:24 ]
[ You have passed readelf testElf testcase 2/6 ]
[ PASSED:32 ]
[ TOTAL:32 ]
[ You have passed readelf testElf testcase 3/6 ]
[ PASSED:9 ]
[ TOTAL:9 ]
[ You have passed readelf vmlinux testcase 4/6 ]
[ PASSED:7 ]
[ TOTAL:7 ]
[ You have passed readelf vmlinux testcase 5/6 ]
[ PASSED:16 ]
[ TOTAL:16 ]
[ You have passed readelf vmlinux testcase 6/6 ]
[ You got 100 (of 100) this time. Tue Mar 3 13:39:02 CST 2020 ]
```

至此，lab1-extra实验任务全部说明完毕
祝实验顺利

评测截止日期：2020/3/22-23:00

注意：经课程组讨论，lab1-extra通过**并不一定保证**能获得lab1课上测试加分
等疫情结束返校后还可能再进行一次lab1课上测试，请以后续通知为准