



北京航空航天大学
BEIHANG UNIVERSITY

操作系统 Lab1 - 1 课上测试



- **考试时间 14:00 ~ 16:00**
- 每次课上测试题目分为**基础测试**和**附加测试(选做)**两部分
- 每道题单独评分，每题满分均为100分
- 各个Lab的得分计算公式是： $\text{Lab基础分值} * (\text{课下成绩} * 0.6 + \text{课上exam成绩} * 0.4) / 100$
- 通过附加测试将会给予额外加分 (**附加题成绩 ≥ 60**)



Lab1课上基础题



Step1:创建lab1-1-exam分支

- `cd ~/学号/`
- `git checkout lab1`
- `git add .`
- `git commit -m "xxxxx"` (填写修改或可提示自己的信息)
- `git checkout -b lab1-1-exam` (务必注意有参数-b)



Step2:完成lab1基础题代码编写



Step3:提交更改

- `cd ~/学号/`
- `git add .`
- `git commit -m "xxxxx"` (填写修改或可提示自己的信息)
- `git push origin lab1-1-exam:lab1-1-exam`



Step4:提交结果

```
remote: [ Compile success! readelf found. ]
remote: [ PASSED:2 ]
remote: [ TOTAL:2 ]
remote: [ You have passed readelf testcase 1/2 ]
remote: [ PASSED:2 ]
remote: [ TOTAL:2 ]
remote: [ You have passed readelf testcase 2/2 ]
remote: [ You got 100 (of 100) this time. Wed Mar 16 21:52:09 CST 2022 ]
remote:
```

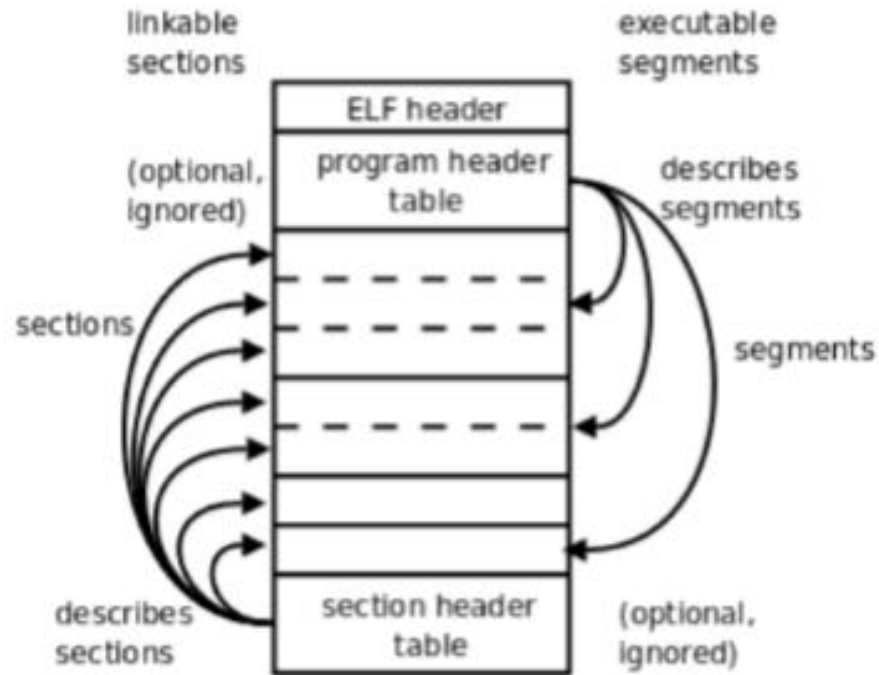
正确完成实验要求后提交代码，可以看到如上图所示评测结果，得到100分。



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。





Lab1-1-exam测试题目

题目内容：修改readelf.c中解析ELF文件的代码，readelf函数需要输出ELF文件的Section Header中序号为2和3的表项（第一个表项序号为0）中所记载的相应偏移量sh_offset和加载起始地址sh_addr信息（此数据成员指明本段内容应当加载到的虚地址）。

格式要求：对每个Section Header，输出格式为：“**Read : %d:0x%x,0x%x\n**”，三个标识符分别代表序号、偏移量sh_offset和加载起始地址sh_addr。

ps:相应信息可查看ELF手册根据括号内解释进行确认。

ppss: 本次exam和Extra测试使用的ELF文件均为小端格式



Lab1-1课上附加题



Step5:创建附加题分支（选做）

- `git checkout lab1` (回到lab1分支下)
- `git add .`
- `git commit -m "xxxxx"` (填写修改或可提示自己的信息)
- `git checkout -b lab1-1-Extra` (务必注意有参数-b)



Step6:完成lab1-1附加题代码编写（选做）



Step7:提交更改（选做）

- `cd ~/学号/`
- `git add .`
- `git commit -m "xxxxx"` (填写修改或可提示自己的信息)
- `git push origin lab1-1-Extra:lab1-1-Extra`



Step8:提交结果 (选做)

```
remote:
remote: [ readelf.c found ]
remote: rm -rf *.o
remote: rm readelf
remote: rm: cannot remove 'readelf': No such file or directory
remote: Makefile:15: recipe for target 'clean' failed
remote: make: *** [clean] Error 1
remote: gcc -I./ -c main.c
remote: gcc -I./ -c readelf.c
remote: gcc main.o readelf.o -o readelf
remote: [ Compile success! readelf found. ]
remote: [ PASSED:1 ]
remote: [ TOTAL:1 ]
remote: [ You have passed Conflict and Overlay testElf testcase 1/3 ]
remote: [ PASSED:1 ]
remote: [ TOTAL:1 ]
remote: [ You have passed Conflict and Overlay testElf testcase 2/3 ]
remote: [ PASSED:2 ]
remote: [ TOTAL:2 ]
remote: [ You have passed Conflict and Overlay testElf testcase 3/3 ]
remote: [ You got 100 (of 100) this time. Thu Mar 17 10:57:35 CST 2022 ]
remote:
```

正确完成实验要求后提交代码，可以看到如上图所示评测结果，得到100分。
lab1-1-Extra测试内容和lab1-1-exam并无逻辑依赖，不必基于lab1-1-exam完成

此时lab1-1-Extra测试通过，**可以获得lab1-1课上测试的额外加分。**



题目背景

一个可执行文件或共享目标文件的程序头表(program header table)是一个数组，数组中的每一个元素称为“程序头(program header)”，每一个程序头描述了一个“段(segment)”或者一块用于准备执行程序的信息。一个目标文件中的“段

(segment)”包含一个或者多个“节(section)”。程序头只对可执行文件或共享目标文件有意义，对于其它类型的目标文件，该信息可以忽略。在目标文件的文件头(elf header)中，e_phentsize 和 e_phnum 成员指定了程序头的大小。

在实际加载过程中，我们在考虑程序头表中定义的程序加载目标地址和偏移量之外，还需要考虑欲加载位置的页面是否已经存在，如果存在，是否与其中已加载内容存在冲突（即存在两段欲加载地址区间存在重叠，可见下页示例图），在我们的MOS系统中，页面的大小是4KB。



测试说明

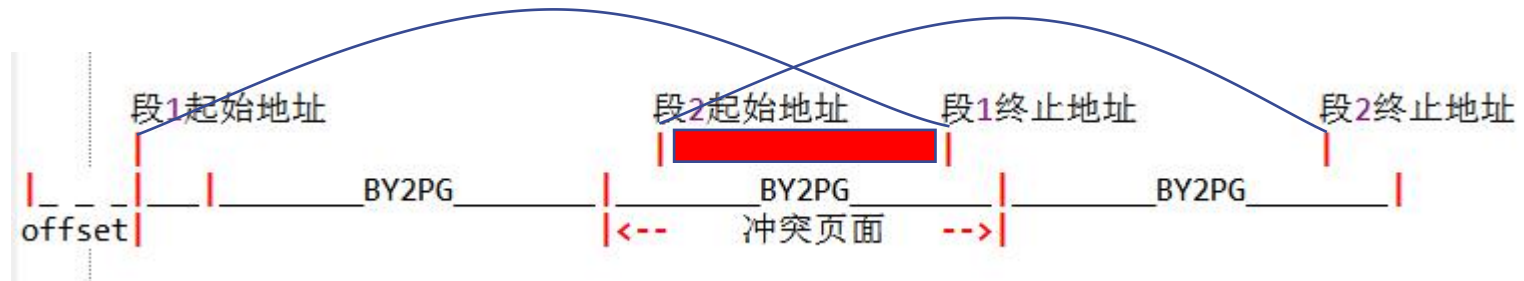


此处即为共用页面的起始地址，左闭右开，该地址一定页对齐

题目背景



示例1：共用页面（BY2PG代表页面大小）



示例2：冲突页面（BY2PG代表页面大小）



题目背景

对于共用页面情况给出定义，记两个段目标加载位置为 l_1, r_1 和 l_2, r_2 ，若 $\text{ROUNDDOWN}(r_1, \text{BY2PG}) = \text{ROUNDDOWN}(l_2, \text{BY2PG})$ ， ROUNDDOWN 为向下对齐，则称这两段存在共用页面

在此基础上，给出冲突页面的定义，若 $l_2 < r_1$ ，且题目数据保证 l_1 远小于 r_1 ， r_2 远大于 r_1 ，即对于冲突页面的情况，只需要考虑形如前页示意图的情况。



Lab1-1-Extra测试题目

题目内容：修改readelf.c中的代码，使得我们的readelf函数可以检查ELF文件中各个程序段是否存在共用页面、加载区间冲突的情况。如果不存在上述情况，则需要输出 ELF 文件的所有segment header的**序号（第一个表项序号为0）**及其相应的**filesize**和**memsize**信息。如果存在上述情况，则不输出相应信息，改为输出出现上述情况的第一个页面起始地址（**虚拟地址**）。

格式要求：

正常输出：对每个segment 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。

共用页面：输出格式为：**"Overlay at page va : 0x%x\n"**，标识符代表存在共用现象的第一个页面的起始地址。

冲突页面：输出格式为：**"Conflict at page va : 0x%x\n"**，标识符代表存在冲突现象的第一个页面的起始地址。

ps：本次exam和Extra测试使用的ELF文件均为小端格式



Lab1-1-Extra评测用例限制

对于不存在上述情况的数据，其segment总数 < 5 。

对于存在冲突或共用的数据，有且仅有一处共用或冲突，segment总数 ≤ 3 。

冲突样例仅存在形如示意图所示型冲突，不存在诸如一个段欲加载区间完全包含在另一个段欲加载区间中的情况，不存在冲突起始位置与冲突终止位置不在一页上。

请同学们自行考虑如何构造含有共用和冲突页的ELF文件，以便进行本地测试。



Lab1-1本地测试教学

请同学们充分进行本地测试后，再提交评测！
以下内容介绍如何对lab1-1开展本地测试。



Lab1-1本地测试教学

1、随意编写一个语法上正确的C程序，此处不妨命名为a.c

```
git@19373153:~/git/19373153/readelf$ vim a.c
```

```
int fib(int n)
{
    if (n == 0 || n == 1) {
        return 1;
    }
    return fib(n-1) + fib(n-2);
}

int main()
{
    int i;
    int sum = 0;
    for (i = 0; i < 10; ++i) {
        sum += fib(i);
    }

    return 0;
}
```



Lab1-1本地测试教学

2、使用gcc命令将我们编写的a.c编译为a.o

```
git@19373153:~/git/19373153/readelf$ gcc -o a.o a.c
```

```
git@19373153:~/git/19373153/readelf$ ls
Makefile  a.c  a.o  kerelf.h  main.c  main.o  readelf  readelf.c  readelf.o  testELF  types.h
```



Lab1-1本地测试教学

3、使用**系统自带的**readelf检查我们生成的a.o是否为小端文件，之后便可以使用a.o进行类似课下对testELF进行的测试了

```
git@19373153:~/git/19373153/readelf$ ./readelf testELF
Read : 2:0x0,0x0
Read : 3:0x154,0x8048154
```

```
git@19373153:~/git/19373153/readelf$ readelf -h a.o
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                                ELF64
  Data:                                      2's complement, little endian
  Version:                               1 (current)
  OS/ABI:                                UNIX - System V
  ABI Version:                           0
  Type:                                  DYN (Shared object file)
  Machine:                               Advanced Micro Devices X86-64
  Version:                               0x1
  Entry point address:                   0x1040
  Start of program headers:              64 (bytes into file)
  Start of section headers:              14632 (bytes into file)
  Flags:                                  0x0
  Size of this header:                   64 (bytes)
  Size of program headers:               56 (bytes)
  Number of program headers:              13
  Size of section headers:               64 (bytes)
  Number of section headers:              29
  Section header string table index:      28
```




Lab1-1本地测试教学

4、运行在readelf文件夹下运行make指令，之后调用**我们自己写的**readelf进行测试即可，如果一切正常，对于lab1-1-exam，可以看到类似右面的输出：

说明本地测试通过，你的readelf程序基本工作正常。此时可以通过git push提交评测了。

如果评测服务器返回错误，请仔细阅读输出信息，修改相应的readelf.c代码，然后再次进行本地测试后再提交评测。

```
git@19373153:~/git/19373153/readelf$ ./readelf testELF
Read : 2:0x168,0x8048168
Read : 3:0x188,0x8048188
```




北京航空航天大学
BEIHANG UNIVERSITY

下面请同学们开始做题
有问题可以随时提问

祝实验顺利！