**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)**

76066001\_2018\_jac@ubuntu:~/76066001-lab/gxemul$ readelf -h 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: 37164 (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

76066001\_2018\_jac@ubuntu:~/76066001-lab/readelf$ readelf -h 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

这因为这个工具(readelf)和objdump命令提供的功能类似，但是它显示的信息更为具体，并且它不依赖BFD库(BFD库是一个GNU项目，它的目标就是希望通过一种统一的接口来处理不同的目标文件），如果使用我们readelf程序只能打印出来他的section header的信息，而且我们程序是简单的程序。但是readelf工具是系统提供的，所以它可以解析我们的内核文件（可执行的文件）。