操作系统 作业1

- 王华强
- 2016K8009929035

作业指导

> 请在一个posix兼容的环境（unix，linux，windows cmd、mac等）编译执行附件小程序，并试着分析每个变量所属的段（section），可以用objdump 等进行验证。

> 作业参考附件，请给出详细分析，包括分析内容和objdump输出结果，不能只列出结果！

实验背景: ELF文件

ref: https://www.cnblogs.com/lxq20135309/p/5551658.html

实验环境

Windows subsystem of Linux. LTS 18.04.

gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)

结果

> 问题: C语言中指针会被初始化在何种位置?

myname(指针)

已初始化全局变量, 在.data中.

"Bao Yungang"

不可变更的字符串, 在.rodata中.

gdata

未初始化全局变量, 在.bss中.

bdata[16]

已初始化全局变量, 在.data中.

ldata

ldata是在运行时分配的数组, 所用的空间在栈中. 数组指针的位置也在栈中.

ddata

ddata是在运行时创建的指针, 保存在栈中. 指针指向的空间由`malloc()`分配, 分配在堆中.

分析验证

先直接使用`objdump`查看符号表.

使用查看符号表/静态符号表命令

```
objdump -t
objdump -T
```

截取相关部分如下:

```
0000000000201060 g     O .bss   0000000000000080              gdata
0000000000201020 g     O .data  0000000000000008              myname
0000000000201010 g     O .data  0000000000000010              bdata
```

ref: https://www.jianshu.com/p/863b279c941e

之后使用gcc逐步进行编译, 查看编译后的汇编代码. 参见实验课件1.

编译结果分析如下:

汇编器结果分析

```
gcc -S -o addr_space.s addr_space.c
```

"Bao Yungang":

```
        .section        .rodata
.LC0:
        .string "Bao Yungang"
```

"gdata: %llX\nbdata.....":

```
.section        .rodata
        .align 8
.LC1:
        .string "gdata: %llX\nbdata:%llX\nldata:%llx\nddata:%llx\n"
```

bdata, gdata:

```
        .section        .data.rel.local,"aw",@progbits
        .align 8
        .type   myname, @object
        .size   myname, 8
myname:
        .quad   .LC0
        .comm   gdata,128,32
        .globl  bdata
  ......
```