

**汇编语言与逆向技术课程实验报告**

**实验四：ARM汇编输出helloworld**

****

学 院 网络空间安全学院

专 业 信息安全

学 号 2212046

姓 名 王昱

班 级 信息安全班

1. **实验目的**
   1. 理解GNU ARM 汇编代码运行环境的搭建、配置及编译运行，掌握在华为鲲鹏云服务器上进行环境配置
   2. 命令行输出“HelloWorld”
2. **实验原理**

华为鲲鹏云主机、openEuler20.03 操作系统；

1. **汇编语句解析**

**.text //**定义一个代码段，处理器开始执行代码的地方，代表后面是代码****

**.global \_start**

**\_start: //程序的入口点**

**mov x0, #0 //将寄存器x0的值设置为0，准备作为参数传递给系统调用**

**ldr x1, =msg //将msg的地址加载到寄存器x1，该符号指向存储在.data段中的字符串"Hello World!\n"**

**mov x2, len //将字符串的长度加载到寄存器x2**

**mov x8, 64 // 将系统调用号加载到寄存器x8，表示要执行的操作是写入标准输出**

**svc #0 // 将参数传递给操作系统执行，这里表示中断号**

**mov x0, 123 // 设置寄存器x0的值为123**

**mov x8, 93 // 设置系统调用号为93，表示要执行的操作是退出程序**

**svc #0**

**.data //表明接下来是数据段**

**msg:**

**.ascii "Hello World!\n" // 这是一个标签，它表示一个字符串的地址。.ascii 是一个伪指令，它表示将字符串的 ASCII 码存储在内存中储存“helloworld”字符串**

**len = . - msg // 计算字符串的长度，“.-”表示当前指令的地址减去msg的地址，即字符串的长度。**

**因此，这段代码的作用是先将"Hello World!\n"字符串写入标准输出，然后以123作为参数退出程序。**

1. **实验过程和运行截图**
2. **创建 hello1 目录**

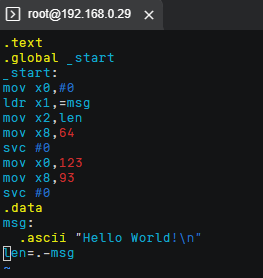
执行以下命令，创建 hello1 目录，存放该程序的所有文件, 并进入 hello 目录。

1700044371974

1. **创建示例程序代码hello1.s**

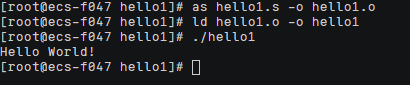
**7c76630c40322c2a7dda335ab97f89e**

执行以下命令，创建示例程序源码 hello1.s：



1. **进行编译运行**

保存示例源码文件，然后退出 vim 编辑器。在当前目录中依次执行以下命令，进行代码编译运行。（ESC进入命令行后输入:wq）



通过上述代码运行，可以看出，编写的 hello-wolrd 示例程序已经在华为鲲鹏云服务器上通过编译和运行，并成功输出结果。

1. **思考题**

**问：同样的代码能否在x86平台上运行，为什么？**

**不一定。同样的代码能否在x86平台上运行，取决于代码的语言、编译器和操作系统的兼容性。x86和ARM是两种不同的处理器架构，它们有不同的指令集、寄存器和寻址模式。如果代码是用一种高级语言编写的，比如C或Java，那么只要有相应的编译器或解释器，就可以将代码转换为适合不同平台的机器码。但是如果代码是用一种低级语言编写的，比如汇编语言，那么就需要根据不同的架构进行修改，才能在x86平台上运行。**

1. **心得与体会**

**在本次实验中学习了ARM汇编语言基本用法，了解了不同的处理器架构和汇编语言的特点和差异，比如x86和ARM 的指令集、寄存器和寻址模式，感受到了ARM汇编语言的魅力。**