

**汇编语言与逆向技术课程实验报告**

**实验四：ARM平台HelloWorld**

****

学 院 网络空间安全学院

专 业 信息安全

学 号 2211044

姓 名 陆皓喆

班 级 信息安全

**一、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”。

**二、实验内容**

以下步骤以在华为鲲鹏云服务器上执行为例。

**1.创建 hello 目录**

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

mkdir hello

cd hello

**2.创建示例程序代码hello.s**

执行以下命令，创建示例程序源码 hello.s。

vim hello.s

代码内容如下：

.text

.global \_start

\_start:

mov x0,#0

ldr x1,=msg

mov x2,len

mov x8,64

svc #0

mov x0,123

mov x8,93

svc #0

.data

msg:

.ascii "Hello World!\n"

len=.-msg

**3.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。 通过上述代码运行，可以看出，编写的 hello-world 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。

as hello.s -o hello.o

ld hello.o -o hello

./hello

**三、代码解析**

**.text;标识后面有指令代码**

**.global \_start;** **声明了一个名称为“\_start”的全局符号**

**\_start:;** **此处是“\_start”标签，为程序的入口点**

**mov x0,#0;** **把立即数0存到寄存器x0里**

**ldr x1,=msg;** **将字符串msg（"Hello World!\n"）的首地址储存到寄存器x1里**

**mov x2,len;** **将字符串msg（"Hello World!\n"）的长度储存到寄存器x2里**

**mov x8,64;** **将立即数64储存到寄存器x8里**

**svc #0;** **这是一个特殊操作码，终端程序的执行并转移控制权到操作系统。在这里是调用操作系统的系统调用打印字符串"Hello World!\n"**

**mov x0,123;** **将立即数123存储到寄存器x0里**

**mov x8,93;** **将立即数93存储到寄存器x8里**

**svc #0;** **这里是再次调用操作系统的系统调用，用于退出程序**

**.data ;定义数据段**

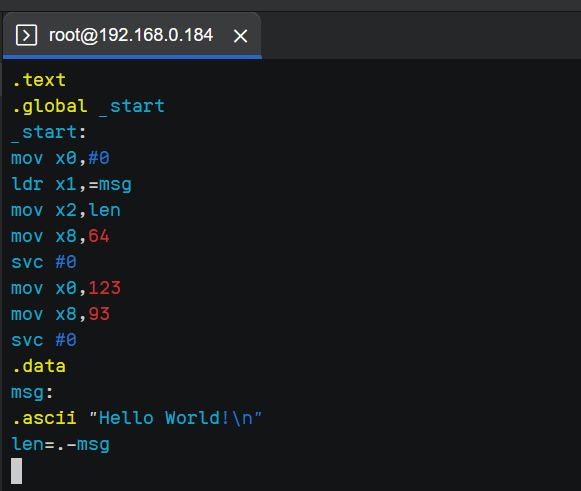
**msg: ;** **定义一个名为“msg”的字符串**

**.ascii "Hello World!\n";定义字符串常量"Hello World!\n"**

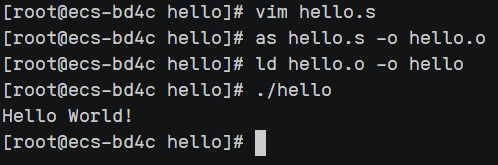
**len=.-msg;** **“.-”符号用于计算当前位置和msg标签之间的字节数，即字符串msg的长度**

**四、运行截图**

1、“hello.s”的编写



2、运行结果截图



**五、思考题**

**同样的代码能否在x86平台运行，为什么？**

**答：不能在x86平台上运行。因为代码中使用了国产的鲲鹏处理器的特有的一些命令与寄存器，比如说svc、ldr等等。在x86平台中没有这些特殊的命令与寄存器，所以不能够运行。**