汇编语言与逆向技术实验报告

Lab8- ARM Hello World

学号：2112514　姓名：辛浩然　专业：信息安全、法学

一、 实验步骤

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。

```
as hello.s -o hello.o
ld hello.o -o hello
./hello
```

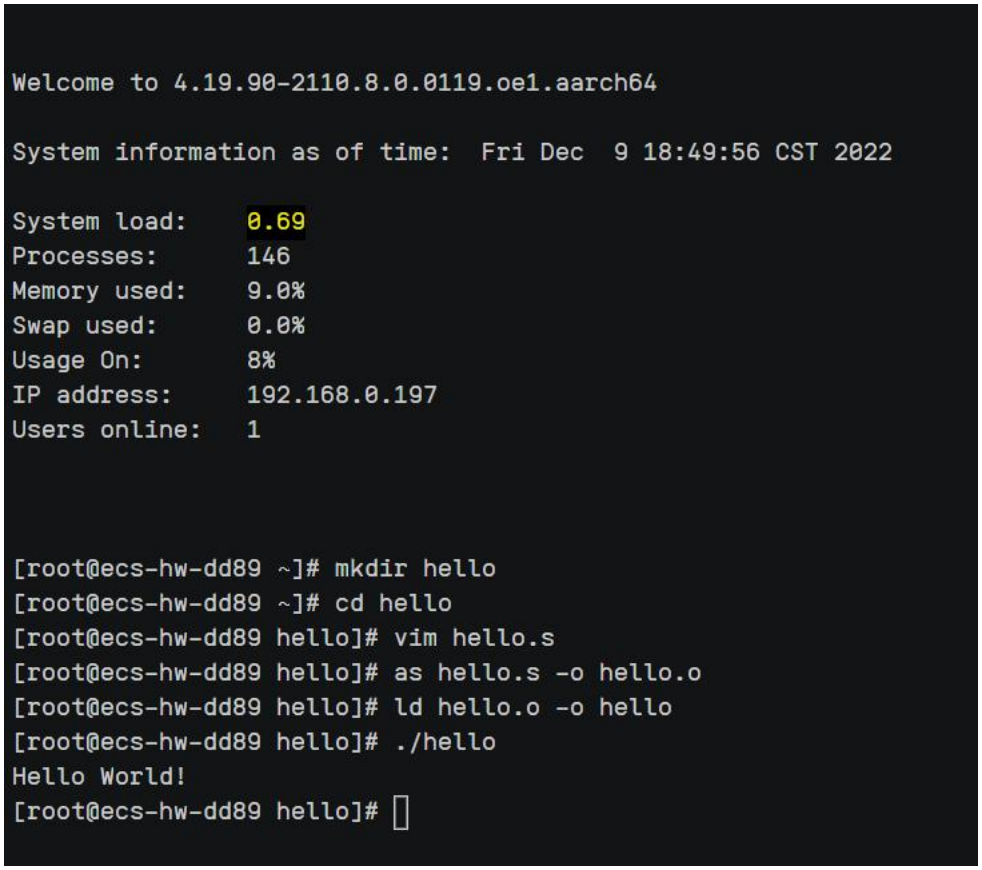
二、 汇编语句解析

```
.text  @ 代码段
.global _start
@ globl 用来定义一个全局的符号,global _start 让_start 符号成为可见的标识符,这样链接器就知道跳转到程序中的什么地方并开始执行程序
_start:
@ _start 是一个函数的起始地址,也是编译、链接后程序的起始地址
mov x0,#0   @ 函数第一个参数，文件描述符 fd 为 0
ldr x1,=msg   @ 字符串地址存入寄存器 x1
mov x2,len    @ 字符串长度存入寄存器 x2
@ write 函数参数为：int fd, const void *buf, size_t count; x0,x1,x2 分别为这三个参数
mov x8,64
svc #0
@ 使用软中断指令 svc 来进行系统调用,调用 write 函数,将字符串在终端输出

mov x0,123   @ exit 函数参数 status 的值，存入寄存器 x0
mov x8,93
svc #0
@ 使用软中断指令 svc 来进行系统调用,调用 exit 函数,结束进程

.data  @ 数据段
msg:
.ascii "Hello World!\n"  @ 定义一个字符串
@ data 段有一个标号 msg，代表字符串"Hello,world!\n"的首地址
len=.-msg
@ len 的值由当前地址减去符号 msg 所代表的地址得到，即字符串的长度
```

三、 实验截图


Welcome to 4.19.90-2110.8.0.0119.oe1.aarch64
System information as of time: Fri Dec 9 18:49:56 CST 2022
System load: 0.69
Processes: 146
Memory used: 9.0%
Swap used: 0.0%
Usage On: 8%
IP address: 192.168.0.197
Users online: 1
[root@ecs-hw-dd89 ~]# mkdir hello
[root@ecs-hw-dd89 ~]# cd hello
[root@ecs-hw-dd89 hello]# vim hello.s
[root@ecs-hw-dd89 hello]# as hello.s -o hello.o
[root@ecs-hw-dd89 hello]# ld hello.o -o hello
[root@ecs-hw-dd89 hello]# ./hello
Hello World!
[root@ecs-hw-dd89 hello]#