苏州大学实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机学院 | | 年级专业 | | 计科 | | 姓名 |  | 学号 |  |
| 课程名称 | | 汇编语言程序设计 | | | | | | | 成绩 |  |
| 指导教师 | | 胡沁涵 | | 同组实验者 | | 无 | | 实验日期 | 2025.6.18 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验四 鲲鹏处理器汇编 |

1. 实验目的

1、了解GNU ARM汇编代码环境的搭建、配置和编译运行；

2、了解华为鲲鹏云服务器环境中进行汇编语言的编写、编译和运行；

3、了解ARM平台上实现C语言代码中内嵌汇编代码的方法。

1. 实验内容

1、实验准备

（1）登录华为云，参考相关PPT，购买合适的**鲲鹏云服务器**；

（2）使用SSH工具以root用户登录鲲鹏云服务器，使用Linux命令，创建一个用户账号，采用stu+学号作为用户名，录入stu2027405000；切换到该账号后，执行ls /home命令，运行结果请截图；

2、验证型实验

（1）实现ARM平台精简指令集（RSIC）编写的hello-world程序的编译和运行。

代码如下（hello.s）：

|  |
| --- |
| .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 |

利用汇编器完成编译和运行，相关Linux命令如下：

|  |
| --- |
| as hello.s –o hello.o  ld hello.o –o hello  ./hello |

（2）上述代码存在某些错误，请完成调试，并解释原因。

（3）过C语言代码内嵌汇编代码，将一个整数类型值，以字节为单位从小尾端转到大尾端或者相反的功能。例如小尾端时32bit整数值用16进制表示为0x12345678，将其以字节为单位转换为大尾端存储后，该值为0x78563412。

代码如下（globalBuiltin.c）：

|  |
| --- |
| /\* globalBuiltin.c\*/  #include <stdio.h>  int main()  {  int val=0x12345678;  \_\_asm\_\_ \_\_volatile\_\_(  "mov x3,%1\n"  "mov w3,w3, ror #8\n"  "bic w3,w3, #0x00ff00ff\n"  "mov x4,%1\n"  "mov w4,w4, ror #24\n"  "bic w4,w4, #0xff00ff00\n"  "add w3,w4,w3\n"  "mov %0,x3\n"  :"=r"(val)  :"0"(val)  :"w3","w4","cc"  );  printf("out is %x \n",val);  return 0;  } |

利用GCC编译器完成编译和运行，相关Linux命令如下：

|  |
| --- |
| 步骤 1 预处理  **gcc -E globalBuiltin.c -o globalBuiltin.i**  步骤 2 编译  **gcc -S globalBuiltin.i -o globalBuiltin.s**  步骤 3 汇编  **gcc -c globalBuiltin.s -o globalBuiltin.o**  步骤 4 生成可执行文件  **gcc globalBuiltin.o -o globalBuiltin**  步骤 5 运行可执行文件  **./globalBuiltin** |

3、应用型实验

（1）通过C语言代码内嵌汇编代码，要求输入一个正整数n，计算1~n之和，最后输出结果。

c代码如下（sum.c）：

|  |
| --- |
| #include <stdio.h>  int main()  {  int n=0;  int sum=0;  scanf("%d",&n);  printf("n=%d\n",n);  //下面是嵌入汇编部分  \_\_asm\_\_ \_\_volatile\_\_(    :"=r"(sum)  :"r"(n)  :"x3","x4","cc"  );  printf("sum is %d \n",sum);  return 0;  } |

（2）将实验二中的凯撒密码一题的程序，使用鲲鹏处理器格式汇编指令进行迁移。

加密对照表：

明文字母表：abcdefghijklmnopqrstuvwxyz

密文字母表：DEFGHIJKLMNOPQRSTUVWXYZABC

加密用例：

明文字符串：Hello World XyZ123

密文字符串：HHOOR WRUOG XBZ123

1. 实验步骤和结果

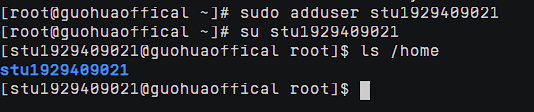


图 1实验准备

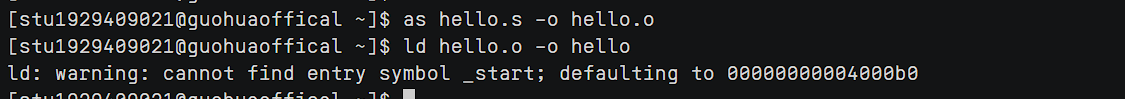


图 2验证型实验错误信息

错误原因：

1. 汇编代码缺少 \_start 标签，是\_start 而不是 start
2. ldr x2, =len 而不是 mov x2, len
3. write 在 ARM64 是 64（不是 x86 的 1）。
4. exit 在 ARM64 是 93（不是 x86 的 60）。

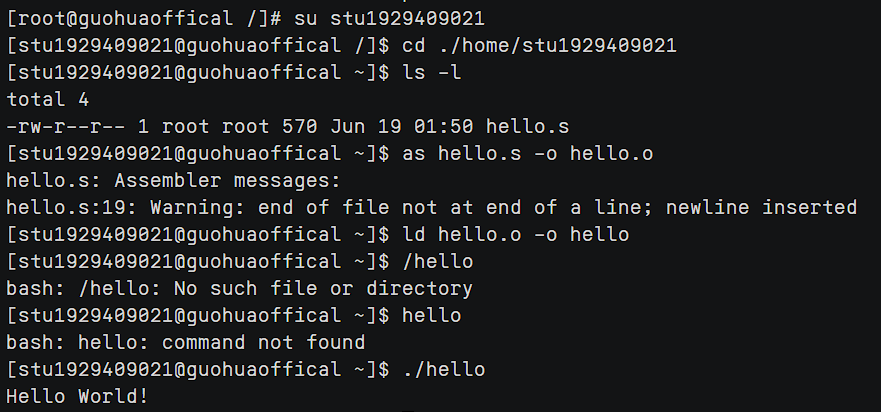


图 3验证型实验输出信息

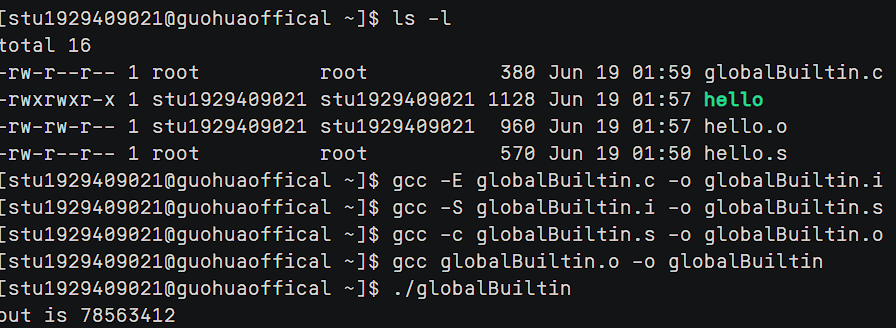


图 4验证型实验(globalBuiltin.c)输出信息

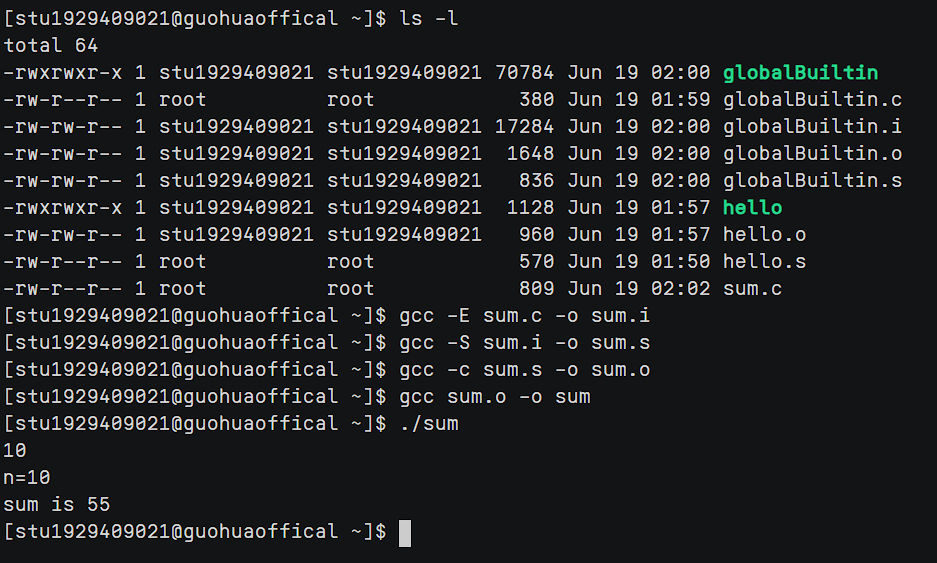


图 5应用型实验(sum.c)输出信息

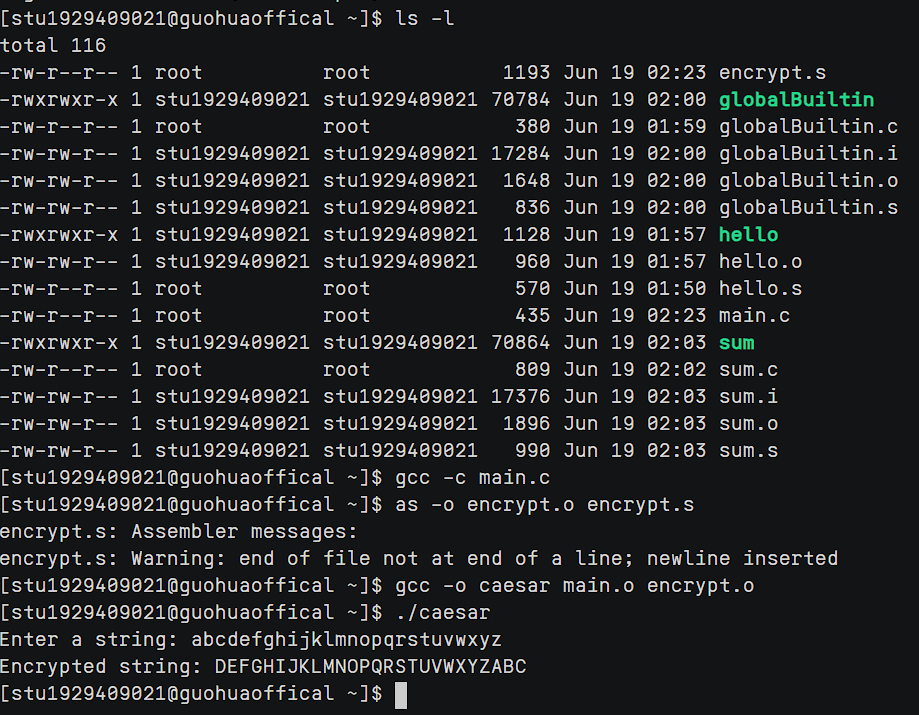


图 6凯撒密码程序输出信息

1. 实验总结

通过本次实验，我掌握了GNU ARM汇编环境的搭建与配置，熟悉了交叉编译工具链的使用，能够编写、编译和运行ARM汇编程序。在华为鲲鹏云服务器上，我实践了汇编语言的开发流程，包括编写代码、使用`as`和`gcc`工具链编译链接，并验证了程序在ARM架构下的正确性。此外，我深入理解了C语言内嵌ARM汇编的方法，通过`\_\_asm\_\_`关键字实现混合编程，优化关键代码性能。实验过程中，我体会到汇编语言对硬件的直接控制能力，以及内嵌汇编在提升C程序效率中的作用，为后续底层开发奠定了基础。