浙江大学

本科实验报告

课程名称：嵌入式系统

姓　　名：胡亮泽

学　　院：计算机科学与技术学院

系：计算机科学与技术系

专　　业：计算机科学与技术

学　　号：3120102116

指导教师：王总辉

2015 年　4　月　15　日

浙江大学实验报告

课程名称：嵌入式系统 实验类型：综合

实验项目名称：嵌入式软件编程技术

学生姓名：胡亮泽 专业：计算机科学与技术 学号：3120102116

同组学生姓名：王谦 指导老师：王总辉

实验地点：曹西-501 实验日期：2015 年 4 月 15 日

一、 实验目的和要求

1. 编写和交叉编译简单 C 语言程序、汇编程序、混合程序和内联式汇编
2. 用三种方法编写并验证 SUM(M:N)程序

二、 实验内容和原理

用 3 种方法实现 SUM(M:N)：

1. 同组同学学号的后两位，数字小的为 M，数字大的为 N
2. 输出：屏幕上打印 SUM(M:N) = XXXX
3. 用 C 语言实现 SUM(M:N)
4. 用汇编语言实现 SUM(M:N)
5. 用混合语言实现 SUM（M:N）
6. 混合语言编译时，要求采用 Makefile

三、 实验结果(源代码)

首先使用 C 语言编程：

```
#include <stdio.h>


int main(){

    printf("16xx\n");

    return 0;

}
```

直接调用 printf 函数即可完成输出功能。

接下来使用汇编语言进行编程：

```
.data
    id:
    .string "16xx\n"
    length = . -id
.text
.global _start

_start:

#movl length, %eax
#sub  $1,      %eax
#movl %eax,       length


movl $length, %edx
movl $id,     %ecx

movl $4,      %eax
movl $1,      %ebx

int $0x80

movl $0,      %ebx
movl $1,      %eax


int $0x80
```

直接定义一个字符串 id, 输入学号信息。

定义一个 length 变量，其值为字符串的长度。

最后调用 0x80 号中断，传入参数进行输出。其中 write 功能的功能号为 4，我们将 4 传入 eax 寄存器，stdout 文件类型为 1，所以将 1 传入 ebx 寄存器，edx 代表字符串缓冲区的大小，传入 length，ecx 代表缓冲区位置，传入 id 字符串。

成功输出学号信息后，将 1 传入 eax 寄存器中，1 为退出功能号，再次调用 0x80 号中断退出程序。

最后是混合编译：

我们先用 C 语言编写主要的 main 函数

```
#include <stdio.h>

extern void sum(char * a, char * b);

int main(){

    char *a = "16";
    char *b = "xx\n";


    sum(a,b);
    return 0;

}
```

在 C 语言程序中，我们用 extern 定义了一个外部函数 sum，这个函数就是一会儿我们要用汇编语言实现的函数。在这里，我们定义两个参数为 char *指针，用以传入字符串地址。

在 main 函数中，我们给两个字符串指针赋值，分别是组员的学号，再作为参数传入函数 sum 函数后，打印出学号并返回。

接下来是使用 AT&T 汇编语言编写的 sum 函数：

```
.data

    length1:
        .int 2
    length2:
        .int 3

.bss
    ID1:
        .space 4,0
    ID2:
        .space 4,0

.text
.global sum

sum:

    movl %esp,    %ebp

    movl 4(%ebp) ,    %eax

    movl %eax, ID1


    movl 8(%ebp), %eax
    movl %eax,    ID2


    movl length1, %edx
    movl ID1,     %ecx
```

```
    movl $4,       %eax
    movl $1,       %ebx

    int $0x80


    movl length2,     %edx
    movl ID2,      %ecx
    movl $4,       %eax
    movl $1,       %ebx

    int $0x80

    movl $1,   %eax
    movl $0,   %ebx

    int $0x80
```

在程序的一开始，我们定义了两个变量 length1 和 length2，其值分别为 2,3,代表两个学号字符串的长度，第二个字符串因为有一个换行符所以长度为 3：

```
    length1:
        .int 2
    length2:
        .int 3
```

接下来，我们在.bss 段定义了两个变量 ID1 和 ID2，用以存储作为参数传入的学号字符串

```
.bss
    ID1:
        .space 4,0
    ID2:
        .space 4,0
```

用.space 预留 4 个字节的空间，因为实验平台上字符串指针的大小为 4 个字节。

接下来，通过堆栈读取两个参数

```
    movl %esp,     %ebp

    movl 4(%ebp) ,    %eax

    movl %eax, ID1
```

由于压栈时会压入 IP 寄存器和 SP 寄存器的内容，所以读取第一个参数时需要将初始地址加上 4。接下来先将内容存入 eax 寄存器中，再存入 ID1 变量中。ID2 变量通过同样的方式赋值，但是需要将初始堆栈地址加上 4.

随后，我们通过调用 0x80 号中断显示字符串，并再次调用中断退出程序。

四、 讨论与心得

本次实验的收获主要有以下几点：

1. 学会如何使用 AT&T 汇编语言，了解该语言和 Intel x86 汇编的区别

2. 熟悉混合编译的方法，包括函数的调用以及使用堆栈进行参数的传递方法。
3. 更加熟练的使用开发板