



《操作系统实验》 实验报告

(实验一)

学院名称：数据科学与计算机学院

专业（班级）：17 软件工程 2 班

学生姓名：张伟焜

学号：17343155

时间：2019 年 3 月 9 日

实验一：AT&T汇编实验

一. 实验目的

1. 熟悉Linux下使用AT&T汇编的编译链接命令
2. 初步掌握使用AT&T汇编程序设计技术

二. 实验要求

用AT&T汇编语言编写一程序求一组数(12,4,6,7,80,34,54,46,3,23)最小值和均值，并将在屏幕上输出。

三. 实验方案

安装VMware虚拟机，装入Linux系统。在Linux环境下进行AT&T汇编。将问题转化成C++问题，写出代码如下：

```
#include<iostream>

using namespace std;

int main() {
    int data[] = { 12, 4, 6, 7, 80, 34, 54, 46, 3, 23 };
    int sum=0;
    int min=data[0];
    int mean;
    for (int i = 0; i < 10; i++) {
        sum += data[i];
        if (data[i] < min)
            min = data[i];
    }
    mean = sum / 10;
    cout << "The minimun value is " << min << endl;
    cout << "The mean value is " << mean<< endl;
    return 0;
}
```

根据C++代码思路进行AT&T代码的编写。

AT&T 编译

```
as --32 -o hw1.o hw2.s
```

AT&T 链接

```
ld -dynamic-linker /lib/ld-linux.so.2 -lc -m elf_i386 -o hw1 hw1.o
```

四. 实验过程

1. 安装虚拟机，安装 Linux 系统。

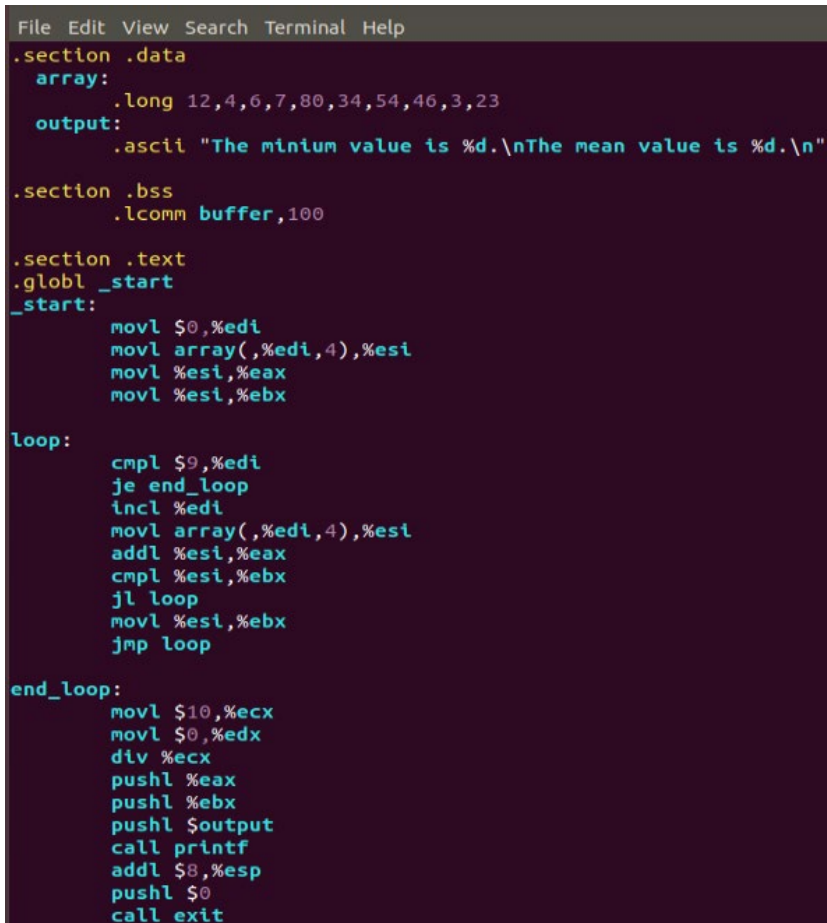
2. 配置环境。目的是在 AT&T 汇编中调用 C 函数库。

为了 64 位 GCC 能编译 32 位程序，需安装相应的适配库：

gcc-multilib, g++-multilib, module-assistant

```
sudo apt-get install gcc-multilib
```

3. 根据 C++ 代码写出 AT&T 汇编代码



```
File Edit View Search Terminal Help
.section .data
array:
    .long 12,4,6,7,80,34,54,46,3,23
output:
    .ascii "The minium value is %d.\nThe mean value is %d.\n"

.section .bss
    .lcomm buffer,100

.section .text
.globl _start
_start:
    movl $0,%edi
    movl array(,%edi,4),%esi
    movl %esi,%eax
    movl %esi,%ebx

loop:
    cmpl $9,%edi
    je end_loop
    incl %edi
    movl array(,%edi,4),%esi
    addl %esi,%eax
    cmpl %esi,%ebx
    jl loop
    movl %esi,%ebx
    jmp loop

end_loop:
    movl $10,%ecx
    movl $0,%edx
    div %ecx
    pushl %eax
    pushl %ebx
    pushl $output
    call printf
    addl $8,%esp
    pushl $0
    call exit
```

4. 编译、链接、运行及结果

```
willken@ubuntu:~/os$ vim hw1.s
willken@ubuntu:~/os$ as --32 -o hw1.o hw1.s
willken@ubuntu:~/os$ ld -dynamic-linker /lib/ld-linux.so.2 -lc -m elf_i386 -o hw1 hw1.o
willken@ubuntu:~/os$ ./hw1
The minium value is 3.
The mean value is 26.
willken@ubuntu:~/os$
```

五. 实验总结

虽然有上学期学习MIPS汇编的经验，但真正开始学习AT&T汇编时还是遇到了一些问题。我认为，两个汇编语言最大的一个区别就是源寄存器和目标寄存器左右位置的区别。此外，在配置环境时，由于没有安装相应的库，导致C函数库不能正常调用，也浪费了一些时间，这启示我在之后的实验中要注意环境的配置。

在将C++代码转化为汇编语言时也遇到了一些问题。在求解最小值时，先假设第一个数最小，然后遍历后边的九个数，若比min小，则将其作为最小值记录下来。此时的遍历次数为除第一个之外的就个数。此时，再要求和是就会少加一个数，所以，应先有sum=array[0]的操作。综上，有：

```
movl array(%edi,4), %esi
movl %esi,%eax          #sum=array[0]
movl %esi,%ebx          #min=array[0]
```

这样，找出最小的数和求和都可以统一通过后9个数的遍历完成。

总体来说，先写出C++代码，再“翻译”成汇编语言还是节约了不少时间，大大提高了实验效率。

【参考文献】课件《00-1AT&T 汇编语言初步》

【源代码】

```
.section .data
array:
    .long 12, 4, 6, 7, 80, 34, 54, 46, 3, 23
output:
    .ascii "The minium value is %d.\nThe mean value is %d.\n"

.section .bss
    .lcomm buffer, 100

.section .text
.globl _start
```

```
_start:
    movl $0,%edi
    movl array(,%edi,4),%esi
    movl %esi,%eax
    movl %esi,%ebx

loop:
    cmpl $9,%edi
    je end_loop
    incl %edi
    movl array(,%edi,4),%esi
    addl %esi,%eax
    cmpl %esi,%ebx
    jl loop
    movl %esi,%ebx
    jmp loop

end_loop:
    movl $10,%ecx
    movl $0,%edx
    div %ecx
    pushl %eax
    pushl %ebx
    pushl $output
    call printf
    addl $8,%esp
    pushl $0
    call exit
```