

数字逻辑与处理器作业

——汇编程序设计

2022 年 4 月 11 日

一、作业内容

本次作业要求同学们在 MARS 模拟器上，将指定的 C++ 代码翻译成 MIPS 汇编指令，然后编译，运行，调试，并通过测试。目的在于：理解汇编语言如何完成高级语言描述的算法，了解 MIPS 处理器的硬件结构如何实现指令的需求，同时学会如何编写调试汇编程序。

1. 基础练习

练习 1-1：系统调用

练习使用 MARS 模拟器中的系统调用 `syscall`，使用 `syscall` 可以完成包括文件读写，命令行读写（标准输入输出），申请内存等辅助功能。系统调用基本的使用方法是

- 向 `$a*` 寄存中写入需要的参数（如果有）
- 向 `$v0` 寄存器中写入需要调用的 `syscall` 的编号
- 使用“`syscall`”指令进行调用
- 从 `$v0` 中读取调用的返回值（如果有）

更多具体的使用方法可以参照 MARS 模拟器的 Help 中的相关内容。

`exp1_1_sys_call.cpp` 代码内容主要包括：

- 申请一个 8byte 整数的内存空间。
- 从“a.in”读取两个整数。
- 向“a.out”写入这两个整数。

- 从键盘输入一个整数 i 。
- $i = i + 1$ 。
- 向屏幕打印这个整数。

输入输出文件格式： 输入文件名为“a.in”，输出文件名为“a.out”，输入输出文件均使用二进制格式。文件中提供了“a.in”作为测试样例，包含 1 和 10 两个整数。对于打开文件的 mips 指令，**只读对应 $flag = 0$ ，只写对应 $flag = 1$ ，可读可写对应 $flag = 2$**

练习 1-2：循环，分支

用 MIPS 语言实现 exp1_2_loop.cpp 中的功能并提交汇编代码，尽量在代码中添加注释。exp1_2_loop.cpp 代码内容主要包括：

- 将输入值取绝对值，存在变量 i, j 中
- 从变量 i 开始，循环 j 轮，每轮 $i = i + 1$

练习 1-3：数组，指针

用 MIPS 汇编指令实现 exp1_3_array.cpp 的功能并提交汇编代码，尽量在代码中添加注释。exp1_3_array.cpp 代码内容主要包括：

- 输入数组 a 的长度 n
- 任意输入 n 个整数
- 将数组 a 逆序，并且仍然存储在 a 中
- 打印数组 a 的值

提示： MIPS 中系统调用 9 和 C 语言中的 new 作用类似。使用该指令开辟 n 个整数的空间，传入参数为 $n * 4$ ，返回值为空间首地址

练习 1-4：函数调用

本节提供了计算汉诺塔的 C 语言代码，请补充完成 MIPS 代码，逐步完成函数调用的编译，使得其可以完成计算 $Hanoi(n)$ 的任务。（在实验报告中完成即可，不需要提交相应汇编程序）

int Hanoi(int n)	
{	Hanoi : #将参数放入\$a0
if (n == 1)	addi \$t0 \$0, 1
	bne \$t0 \$a0 Next
{	addi \$v0 \$0 1 # 返回1
return 1;	jr \$ra
}	Next:
else	(保护现场)
{	addi \$s1 \$0 1
return 2 * Hanoi(n - 1) + 1;	(调用Hanoi(n-1), 返回值存在\$v0)
}	add \$s1 \$v0 \$s1
	add \$s1 \$v0 \$s1
}	add \$v0 \$s1 \$0
	(恢复现场)
	jr \$ra

2. 作业要求

作业用一个压缩包提交，压缩包名称：“学号_姓名.7z”。推荐用 7z 格式，其他常见压缩格式也可以。

压缩包打开后需要包含：

一个“实验报告.pdf”文件，

一个“exp_1_1.asm”，一个“exp_1_2.asm”，一个“exp_1_3.asm”

注意所有的 MIPS 代码需要和 C 语言代码对应，不可使用其他 C 程序。实验 2 要求的输出格式：最终程序运行结束时结果储存在寄存器 \$v0 中。没有按要求输出将会酌情扣分。

二、附录

C++代码

exp1_1_sys_call.cpp

```
1. #include "stdio.h"
2. void main()
3. {
4.     FILE * infile ,*outfile;
5.     int i,max_num=0,id;
6.     int* buffer;
7.     buffer = new int[2];
8.     infile = fopen("a.in","rb");
9.     fread(buffer, 4, 2, infile);
10.    fclose(infile);
11.    outfile = fopen("a.out","wb");
12.    fwrite(buffer, 4, 2, outfile);
13.    fclose(outfile);
14.    scanf("%d",&i);
15.    i = i + 1
16.    printf("%d",i);
17. }
```

exp1_2_loop.cpp

```
1. #include "stdio.h"
2. void main()
3. {
4.     int i,j,temp;
5.     scanf("%d",&i);
6.     scanf("%d",&j);
7.     if (i<0){i=-i;}
8.     if (j<0){j=-j;}
9.     for(temp=0;temp<=j;++temp)
10. {
11. i += 1;
12. }
13. printf("%d",i);
14. }
```

exp1_3_array.cpp

```
1. #include "stdio.h"
2. void main()
3. {
```

```
4.  int *a, n, i, t;
5.  scanf("%d",&n);
6.  a = new int [n];
7.  for(i=0;i<n;i++)
8.  {
9.      scanf("%d",&a[i]);
10. }
11. for(i=0;i<n/2;i++){
12.     t = a[i];
13.     a[i] = a[n-i-1];
14.     a[n-i-1] = t;
15. }
16. for(i=0;i<n;i++) printf("%d ",a[i]);
17. }
```