

# 数字逻辑与处理器基础作业——汇编程序设计

2023 年 4 月 3 日

## 一、作业概述

本次作业要求同学们在 MARS 模拟器上,将指定的 C/C++代码手动编译成 MIPS 汇编指令,然后汇编,运行,调试,完成能够解决给定问题的汇编程序设计。本次作业的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。

本次作业分为两个部分:基础练习和实战应用。基础练习部分中,会对 MIPS 常见汇编操作进行练习,学习 MIPS 汇编程序编写的基本方式。实战应用部分中,会根据给出 3 个不同算法对应的 C 语言代码,编写算法功能完整的汇编程序。

## 二、提交要求

本次作业的提交要求如下:

作业在 2023 年 4 月 30 日 23:59 前用一个压缩包提交到网络学堂,压缩包名称:“学号\_姓名.xxx”。“xxx”为压缩包格式,推荐用 zip 格式,其他常见压缩格式如 rar 和 7z 等也可以。压缩包解压后需要包含:

一个“实验报告.pdf”文件,内容要求言简意赅,按照作业内容分为两个部分:基础练习部分简述程序测试功能即可,实战应用部分只需要介绍寄存器和变量之间的分配关系、过程调用时进行的操作和最终程序输出即可;

一个“exp1”文件夹，其中包含：一个“exp1\_1.asm”汇编代码文件，一个“exp1\_2.asm”汇编代码文件，一个“exp1\_3.asm”汇编代码文件和一个“exp1\_4.asm”汇编代码文件，总计 4 个文件；

一个“exp2”文件夹，其中包含：一个“sssp\_bellman.asm”汇编代码文件，一个“sssp\_dijkstra.asm”汇编代码文件和一个“sssp\_dfs.asm”汇编代码文件，总计 3 个文件。

上述所有文件名（含扩展名）中出现的字母**均要求小写**。

注：本次作业中的部分成绩会由评测脚本直接给出。**如果因为提交格式错误导致评测脚本无法正常找到代码进行评测，相应部分的分数会被扣除。**

### 三、注意事项

1. 本次实验中所有的 MIPS 代码均提供了相应的 C/C++代码（在 exp1 和 exp2 文件夹中），需要和所提供的代码对应，**不可另外使用其他 C/C++代码**。
2. 本次作业须独立完成，**严禁任何形式的抄袭**（形式包括但不限于复制同届或往届学生的代码、复制网络开源代码、复制由 ChatGPT 生成的代码等）。抄袭行为一经发现，本次作业记 0 分，并按照学校相关规章制度严肃处理。

### 四、负责助教联系方式

如果对本次作业有任何疑问或建议，可以联系本次作业负责助教：

陈佳煜：jiayu-ch19@mails.tsinghua.edu.cn

杜禧瑞：dxr22@mails.tsinghua.edu.cn

## 第一部分 基础练习

### 练习 1-1：系统调用（exp1\_1.asm）

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.cpp` 代码内容主要包括：

- 申请一个 8 byte 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 + 10`。
- 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。

提示：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 (exp1\_2.asm)

用 MIPS 语言实现 exp1\_2.cpp 中的功能并提交汇编代码, 尽量在代码中添加注释。exp1\_2.cpp 代码内容主要包括:

- 将输入值  $i$  取相反数, 将输入值  $j$  取绝对值。(  $j$  为非 0 整数)
- 从变量  $i$  开始, 循环  $j$  轮, 每轮  $i = i+1$

输出格式要求: 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。

## 练习 1-3: 数组, 指针 (exp1\_3.asm)

用 MIPS 汇编指令实现 exp1\_3.cpp 的功能并提交汇编代码, 尽量在代码中添加注释。exp1\_3.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  $a$  中的元素逐个加 1, 并且逆序存储在  $a$  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 (exp1\_4.asm)

本节提供了计算斐波那契数列的 C++ 语言代码 exp1\_4.cpp, 请翻译成 MIPS 代码。

## 第二部分 实战应用

### 问题定义

在加权图上，将路径上的边权之和定义为该路径的长度。单源最短路径（Single-Source Shortest Path, SSSP）问题是计算从一个给定的源点出发到其他顶点的最短路径的问题。本次作业中的程序具体需要解决的问题如下。

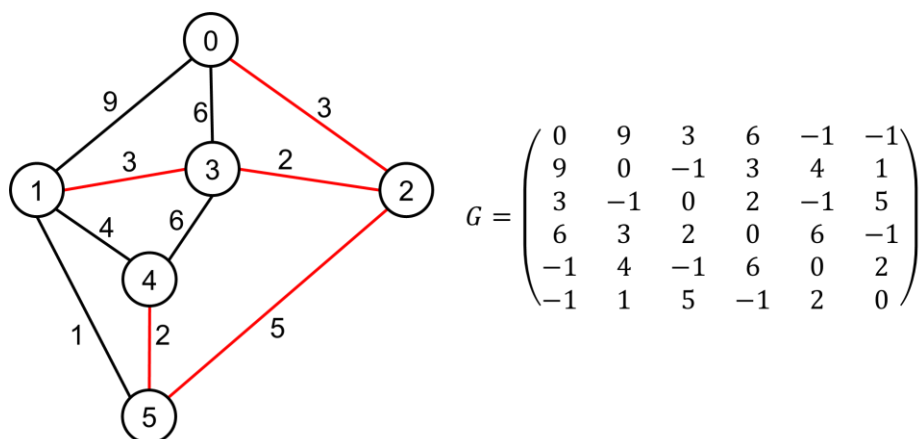
给定一个共 $n$ 个顶点（ $3 \leq n \leq 32$ ）的无向图的邻接矩阵 $G = \{g_{ij}\}$ ,  $i, j = 0, 1, \dots, n-1$ ，其中 $g_{ij}$ 满足如下关系：

$$g_{ij} = g_{ji} = \begin{cases} 0 & \text{如果 } i = j \\ w_{ij} > 0 & \text{如果 } i \neq j \text{ 且 顶点 } i \text{ 和 顶点 } j \text{ 之间有一条权重为 } w_{ij} \text{ 的边} \\ -1 & \text{如果 } i \neq j \text{ 且 顶点 } i \text{ 和 顶点 } j \text{ 之间没有边相连} \end{cases}$$

计算从源点（顶点 0）到其他所有顶点的最短路径长度。测试输入数据保证所有边权为正整数，且任意无环路径长度不会超过 4 byte 有符号整数的上限（即 0x7FFFFFFF）。

### 问题示例

如下图所示，给定一个共 6 个顶点的无向图的邻接矩阵 $G$ ，计算源点（顶点 0）到其他所有顶点的最短路径长度。



此无向图中的最短路径树用红色标出，具体为：

0 到 1 的最短路径  $0 \rightarrow 2 \rightarrow 3 \rightarrow 1$ （长度  $3 + 2 + 3 = 8$ ）；0 到 2 的最短路径  $0 \rightarrow 2$ （长度 3）；0 到 3 的最短路径  $0 \rightarrow 2 \rightarrow 3$ （长度  $3 + 2 = 5$ ）；0 到 4 的最短路径  $0 \rightarrow 2 \rightarrow 5 \rightarrow 4$ （长度  $3 + 5 + 2 = 10$ ）；0 到 5 的最短路径  $0 \rightarrow 2 \rightarrow 5$ （长度  $3 + 5 = 8$ ）。

因此，对于此示例，程序的输出为：8 3 5 10 8（输出格式详见“输入/输出格式”部分）。

## 输入/输出格式

输入文件为小端的二进制文件，其中最开始的 4 byte 为整型变量  $n$ ，之后的 4096 byte 为一个大小为 1024 的整型数组 `graph`，用以表示  $32 \times 32$  的邻接矩阵：其中前 128 byte 为邻接矩阵第一行，之后 128 byte 为邻接矩阵第二行，依此类推。在这个  $32 \times 32$  的“邻接矩阵”中，只有前  $n$  行中的每行前  $n$  个数是有效的。具体输入格式可参考提供的测试文件 `test.dat`，其中包含“问题示例”部分所给出的例子对应的数据。

输出直接打印到屏幕，共 1 行，包含  $n-1$  个由空格隔开的整数，表示顶点 0 到其他所有顶点的最短路径长度（按照邻接矩阵中顶点的顺序依次输出；如果相应顶点不与顶点 0 连通，则到该顶点的最短路径长度输出 -1）。**请勿在输出末尾额外输出空行。**

## 实验内容

本部分作业提供了解决单源最短路径问题的三种不同算法的 C 语言代码：Bellman-Ford 算法（`sssp_bellman.c`）、Dijkstra 算法（`sssp_dijkstra.c`）和 DFS 算法（`sssp_dfs.c`），相应代码位于 `exp2/c/` 路径下。

同时提供了三种算法的相应主函数的汇编语言实现，需要各位同学补全三个汇编代码文件中相应的算法函数部分。提供的汇编文件名与 C 文件名对应，补全后无需更改文件名。相应代码位于 `exp2/asm/` 路径下。

补全的代码部分中，需要有必要的注释来对代码进行解释说明，以便于助教评分。

本部分作业对相应算法原理的掌握不做要求，相应算法原理可以参考《数据与算法》等课程的相应课件。