

ERISC 程序可视化——大作业报告

一、 小组成员

🚩 软件 93，刘怡豪，2018010560

🚩 软件 93，周航，2018010224

🚩 软件 92，翁兆天，2018010288

二、 程序设计思路

本项目采用面向对象程序设计思路，将程序拆解为不同对象的组合：

1. Computer（计算机）类，完成 ERISC 指令的运行、流程控制与位图文件的输出；
2. Register（寄存器）类，完成 ERISC 指令集中与寄存器有关的指令；
3. Memory（内存）类，完成 ERISC 指令集中与内存有关的指令；
4. Stack（栈）类，完成 ERISC 指令集中于栈有关的指令；
5. Input（输入）类，完成输入文件的解析与 ERISC 指令行的获取。

在程序运行时，首先将输入文件 (*.risc) 送到 Input 类中解析，针对每一行都生成一条相应的 ERISC 指令，并将其放入向量存储；然后 Computer 类不断读入 Input 类中存储的 ERISC 指令并运行，换句话说，即根据不同的指令调用 Register 类、Memory 类或 Stack 类相应的成员函数，并控制 Input 类行序号的跳转，如此之后，便同时完成了指令运行与流程控制。

得益于类良好的封装特性，本项目模块之间弱耦合，在总体程序框架设计结束之后，便可以容易地进行分工合作。同时本项目使用诸如重载运算符、异常传播等技巧，大大降低了代码的冗余度，结合详细的注释，使可读性极大提高。本项目另一方面尤其注重输入的鲁棒性与提示信息的完备性，以此给予用户更友好的体验。特别地，本项目对跨平台做了兼容处理，使该程序可以既可以运行在 Windows 上也可以运行在 Linux 上。

三、 基本功能完成情况

本项目完成了所有的基本功能，包括：

1. 内存访问指令（Memory 类）：load, store
2. 栈操作指令（Stack 类）：push, pop
3. 寄存器计算指令（Register 类）：mov, add, sub, mul, div, rem, and, or
4. 控制指令：行标识、无条件跳转(jal)、条件跳转(beq, bne, blt, bge)、函数调用(cal)、

函数返回 (ret)

5. 指令操作可视化: .bmp 文件输出 (draw)、.txt 文件输出 (end)
6. 必做任务: task1.risc, task2.risc, task3.risc, task4.risc

四、 扩展功能说明

本程序扩展了以下功能, 包括:

1. 寄存器计算指令 (Register 类): xor (异或)、not (位反)
2. 输入文件 (.risc) 鲁棒性:

只要输入文件符合语法, 本程序可以处理任意数量多余的空格。

3. 输入文件 (.risc) 注释:

本程序支持输入文件以双斜杠“//”单行注释与井号“#”单行注释。

4. 输入文件 (.risc) 错误位置追踪:

得益于异常向上传播的机制, 本程序能够追踪输入文件中哪一行出现了哪一类型的错误, 并及时输出显示。

5. 输出文件覆盖机制:

输出文件位于“output”文件夹中, 每次程序运行会自动删除之前旧的输出文件。

6. 程序运行时间计时:

本程序能够计量指令运行的时间, 并输出显示。

7. 跨平台性:

本项目既可以在 Windows 上编译运行也可以在 Linux 上编译运行, 其中在 Windows 上定义宏 WINDOWS, 在 Linux 上定义宏 LINUX。同时本项目使用更通用的字符集 UTF-8, 保证各平台都可以正常显示中文字符。

五、 编译环境与时间效率

本项目使用 VS2019 进行开发, 最终以 Release、x64 模式输出可执行程序。经过计时统计, 本程序每秒大约能执行 ERISC 指令 4,000,000 万条, 而完成任务 4 (筛法寻找素数) 大约需要时间 4s, 因此满足时间复杂度 $O(n\log(\log n))$ 。另外本程序输出文本文件大约需要 1s 的时间 (以上数据在不同硬件、软件平台上可能不同, 因此仅供参考)。

同时, 本项目也可以使用 g++ 进行编译, 编译命令为 (Windows 平台):

```
g++ -std=c++11 *.cpp -o ERISC -O3 -D WINDOWS
```

或 (Linux 平台):

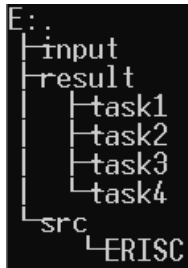
```
g++ -std=c++11 *.cpp -o ERISC -O3 -D LINUX
```

若中文出现乱码，则说明显示的字符集编码不是 UTF-8，可以用以下编译命令解决：

```
g++ -std=c++11 *.cpp -o ERISC -O3 -D WINDOWS -fexec-charset=GB2312
```

六、 程序输入与输出

提交文件夹结构如下图所示：



在提交文件夹根目录下有名为 `ERISC.exe` 的可执行文件，可以直接使用，双击打开之后，输入“`input/task*.risc`”，即可获得输出结果。输出结果保存在“`output`”文件夹下，并且新的输出结果会覆盖旧的输出结果。

七、 小组分工情况

👤 刘怡豪：程序框架设计、指令集实现

👤 周航：位图文件输出

👤 翁兆天：ERISC 文件输入