

## 一、RISC-V 工具链

下载并安装 UC Berkeley 提供的 toolchain, Github: <https://github.com/riscv/riscv-tools>

注意: 按照网站指示下载, 并进行交叉编译。如果下载过慢, 也可以网盘下载:  
<https://pan.baidu.com/s/1eR80gMY>

网盘下载后, 执行下述命令。

```
$ tar xzvf freedom.tar.gz
```

```
$ cd freedom
```

```
$ git submodule update --init --recursive
```

由于安装过程需要安装 sbt, 如网速过慢, 解决方法可参考: <https://cnrv.io/>。

交叉编译时间比较长, 请耐心等待。本 lab 需要使用的是 RISC-V 的 64 位工具链 riscv64-unknown-elf-gcc。

## 二、RISC-V 指令级模拟器

完成 RISC-V rv64I (64 位基本集) 部分 ISA 的模拟, 需完成:

1. 从网盘目录 lab2.1 下载作业: <https://pan.baidu.com/s/1o8wXJHg> 密码: sann
2. 需要实现的 RISC-V rv64I 的部分指令, 可参考其中的 RISCv-simple-greencard.pdf。如需要查阅其他指令编码, 可参考 riscv-ISA-list.pdf。
3. 利用 RISC-V 工具链 riscv64-unknown-elf-readelf 完成用户程序的分析, 并能够自动加载和结束用户程序的运行。初始化时, 需要对 SP 和 PC 设置初始值, 并将指令段和数据段加载到内存。为简化起见, memory map 可自行设计, 并将各段数据加载到对应地址。
4. 只需要模拟用户程序(即 main 函数), 对于可执行文件中各种初始化和库代码无需模拟。
5. 提供指令模拟器模板, 可根据模板修改, 也可自行编写代码。

## 三、检查要求

1. 使用 RISC-V 提供的工具链编译 lab1.1 中所写的程序 (数目不限), 能够完成 elf 格式可执行文件的自动分析, 并将所需代码和数据自动加载到内存。

注意: 编译 C 语言代码时, 编译参数可参考:

riscv64-unknown-elf-gcc -Wa,-march=rv64i -o xxx xxx.c, 以保证生成的用户程序代码中尽量使用 rv64I 中的指令。可以根据需要改变支持的指令集。

2. 正确执行代码并结束, 可查看寄存器结果和内存结果。
3. 支持单步模式, 并支持单步寄存器和内存查看。
4. 如果提供图形化界面, 可加分。
5. 如果实现完整的 ISA, 或实现其他扩展集, 可加分。
6. 如果使用助教编写的测试程序并执行正确, 可加分。