

思路

学号：19373440 姓名：王雨飞

开始前的分析

- 1.看起来比 lab3 简单一点(在后来看了**实验指导**之后，应该比 lab3 简单不少,可惜我发现晚了)
- 2.准备部分重构 lab4 以前的代码，主要是判断 token 类型这块，以前都是用一个 char[] 储存 token，这个 char[] 也必须要包含所有的信息(例如 token 的内容，类型等)，这就导致这个 char[] 内容太多了，写起来十分费劲，而且需要分割字符串。所以打算以后使用一个结构体来储存 token，也方便以后的修改。
- 3.打算尝试继续用连续的数字来作为虚拟寄存器的值，虽然感觉实现起来非常麻烦。

写代码

重构

- 1.重构 token 的储存方式，是一个比较轻松的活，但是代码量已经很大了，还是花了些功夫。

使用数字寄存器

- 1.如果继续使用数字寄存器，那么一个很大的问题就是：跳转命令的 label 是未知的。
- 2.所以我的想法是：将条件变量相关(包括相关的所有代码块)，先设立一个队列，把他们存进去，当然此时 label 后面的寄存器是未知的，所以将这些未知的寄存器标记出来。然后先读入输入，等到条件变量相关的代码结束之前(此时即将回到 main 函数)，我们知道了每一个条件 Block 的寄存器的号，所以将队列中的内容一并输出，把未知的寄存器写入已知的 Block 的寄存器的号码，就可以达成目的。
- 3.想法自然是美好的，但是在这里我并没有这么实现，原因有二：其一是最近的学业压力比较大，这其实写起来较为复杂；其二是我不知道有 and 和 or 的指令，这是因为我没有好好看 lab7 的实验指导，我根据样例来判断输出，所以在此时的我的眼中，这个 lab4 是十分复杂的，有很多跳转，会很晕。

使用字符串寄存器

- 1.因为不知道 Block 对应的寄存器号码，而 br 指令是先输出的，所以干脆在 br 指令这里设置好要跳转的 Block 的名字。因为以数字作为寄存器是需要保证连续，而我们不能推断出 Block 的寄存器数字，所以使用字符串命名来解决。
- 2.我**依然没有仔细阅读实验指导**，但还是基本实现了样例 3 的样例输出，我确信我应该差

不多能过掉所有样例，但果然还是会些问题(尤其当我自己创建了几个新的测试文件，发现问题不算少)。

发现了实验指导中的 and 和 or 指令

- 1.想给看到这一步之前的自己来一刀。有了这个指令，大幅度地降低了逻辑上的难度。
- 2.有了前面的铺垫，比较有效率地拿下了 lab4 的样例，并且提交。

Debug

- 1.lab3 的函数调用写的有点问题，lab4 才发现。
- 2.要把所有的寄存器都换成字符串寄存器，所以在之前的数字顺序寄存器那里，每一个前面都加了个 x。
- 3.icmp 的指令写错了，后来通过问别人解决了。
- 4.栈 pop 有些问题，本来该写 if 的，结果写成了 while。

感想

目前做的 lab 来看，lab4 难度应该仅次于 lab3，不过在知道 and 和 or 指令后就轻松了不少。

我感觉我的信息搜集能力并不太好，icmp 指令写错了，and 和 or 指令也不知道，就稀里糊涂地去写了，以后应该更注重搜集资料、并且更仔细地阅读实验文档。

好消息是 lab5 这块，理论课居然讲到了。希望之后的 lab 可以轻松些，不然软工大作业就完蛋了。