在最开始，我想先向编译之路上给我提供了很大帮助的刘传助教和徐睿远助教提出感谢。

或许是因为喜欢写代码，对于编译实验，我是比较喜欢的。从最开始的词法分析、到语法分析、再到代码生成，我都是在往前赶工，超前完成作业，把时间留到后面的优化。

编写编译器对我加深对编译技术的理解起到了决定性作用。或许在词法分析还不能体现，但在写递归下降语法分析器时，一开始我是完全无从下手，但经历这么一个过程，现在回想，便感觉是无比简单的事情。

因为对于我来说主要过程还是在于优化，接下来我围绕优化过程展开。

首先是中间代码上，我一开始并不知道LLVM有多好使，而是自己写中间代码。我重构过一次中间代码，在代码生成作业时，我中间代码的设计基本是按当前需求进行设计，种类繁多而且很难统一，生成代码时或许不用想太多问题，但在优化过程中它很难操控的弊端便显现了。从“生成代码”到“优化代码”的需求改变让我不得不重构中间代码，在之后的一版中间代码中，我基本是按照四元式构建的，当时还没有看到郭佬的文档，但看到之后发现与郭佬很多都不谋而合。

构建中间代码过程中，是很难超前想到怎么构建会更加利于之后的优化，只能是先尝试，然后遇到细节，发现或许那样更好，再更改，再尝试，再更改。我目前对最后这版中间代码大体满意，但对于数组存取上还有改进空间。我会用[] z x y表示z=x[y]，[]= z x y表示z[x]=y，但我现在想改成\*z x与&z x，即直接用x变量存取z地址元素，这样好处有三：一是对地址的加减可以进行表达式优化了；二是地址操作的中间代码更加简洁，便于处理；三是在函数内联时，也更好直接赋值地址。

优化我写了SSA和GVN，写它们的一大动机是我看到我的中间代码上，有很多明显冗余的赋值指令，比如如果我进行一次d=a+b+c; 我会先把a+b结果计算出来后，再赋值给一个t1，再与c相加。当时想写这个于是第一次去了编译实验答疑，感谢刘传学长给我梳理了一遍SSA插phi和填phi等一些原理和细节实现，让我没费多大劲就把它写成了。

但可能现在想想当时没费多大劲还是时间比较充裕，过程中其实还是遇到了一些BUG要细细de.比如phi函数确实是容易出错的地方，当时我曾对着几个程序把流图画在纸上一个个变量进行推导。

后来时间突然变得紧张，想赶紧把寄存器分配写掉。由于计组发生了一些小概率事件，让我本来就紧张的时间变得更加紧张。这两天把一些ddl赶完，想赶紧开始写。我写的是比较简单的图着色，并不画出图而是进行完活跃变量分析后直接在支配树上进行分配，即郭佬在文档中写的方法。前天写了一天，昨天上午又写了半天，写好之后，De了一些小bug后，通过了一半的测试点，通过的测试点有极其明显的提升，大概有70%左右，我预期的也是这样。但还是有BUG，比如有一个莫名奇妙的BUG，加入一段不可能进入的语句使得结果居然不同。我又一次画出流图推导，推导了很长时间最后发现原来是phi函数寄存器上有冲突，有一个move $9 $10,move $10 $9，应该同时执行但我这样会丢失一个的信息。这让我想起刘传学长之前跟我提到过这个问题，我实现过程中注意了这个细节，对变量进行了拓扑排序。但现在又分配了寄存器，仅仅在变量上拓扑排序是不行的，而应该是在寄存器级别上的进行拓扑排序，想了一下，可能需要重构一些相关代码。我先取巧采用随机分配寄存器的方式，在概率上避免了大部分这种情况。

但除了这个BUG，还有多少BUG是我无法估计的。De了昨天一下午加晚上，还是有BUG。我知道我用的时间可能已经超出预算，但是还是不甘。确实发现mips架构寄存器很多所以不太需要特别纠结细节，所以简单的图着色同样有明显的效果，比起SSA其实是性价比更高的。我已经写了SSA，起到了大概30%左右的优化，但不写更容易且优化效果更好的寄存器分配还是有些可惜。大概估计了一下，从目前通过的测试点与之前的比对来看，如果写成，绝对可以排到很靠前的排名。这也同样符合我的预期，简单的图着色+SSA应该就可以超过大部分人。今天早晨起来还是不甘，又DE了一会，但确实还是不行。

我知道没有时间再放到这上面了，便以写这个总结的方式进行割舍吧。我感觉再给我一些时间，也就两三天，把寄存器踏踏实实写完，会让我更加满意。如果我大三选择这门课程肯定会有更多时间可以写我想写的更多优化，但既然是这样选择就要有所割舍。感觉编译优化很大一部分就是如何最好地调配寄存器，但我却连最简单的还没有时间实现了，算是个遗憾吧。

但同样，我不能忽视过程上带给我的成长，没有直接利用LLVM而是自己设计中间代码，以及写SSA，DEBUG时在纸上画出流图一步步推导，都让我对整个过程有了更深入的理解，以后再做类似工作便会很轻松。以及，更普遍地，让我对编译需要进行的工作以及工作状态有了一个了解。

对于课程，我真心认为编译的课程设计在所有课程里算非常合理的一个。我喜欢这门课程，我认为它给了学生自由的选择空间，可以自由发展。同时评测除了有时候有些慢，以及如果写出死循环的BUG交上去要等它触发TLE为止等好久以外，不设计什么罚时或等待机制，以及还设置辅助测试返回错误信息，感觉都比一些其它课程的设置更加大气。

如果可以，我可能会在假期参加编译比赛弥补没写完想写的优化的遗憾。如果做的比较好，我也愿意申请当下一年的助教。但现在，只能暂时地先和编译说再见了，虽然还会要准备实验考试，但终究不能花很多时间在上面了。