第三个周，已经看了不少关于编译的书及论文，主要内容是虚拟机实现与代码生成部分，难度比较大，看很多遍才能勉强理解。特别是代码生成部分，各种语句的该如何生成代码，为什么要这样设计，都值得好好地学习和理解。

为了能够全面掌握从源代码到汇编代码的翻译流程，以及汇编代码的执行过程，想要实现一个虚拟机，用于解释生成的中间代码。但是虚拟机的指令该如何选择，指令该如何执行还没有太多的概念。因此还需要看更多的虚拟机实现的内容才能一窥究竟。

1.工作内容

1.看编译器相关的书籍，了解中间代码生成的算法。

2.学习符号表的相关内容，理解符号表在编译器中的地位。

3.了解语法制导的编译技术，以及单趟编译的工作流程。

4.了解了虚拟机的相关内容。

2.完成情况

1.学习了几种语句的代码生成，如if-else语句，while语句，逻辑语句等。

2.知道了符号表在编译器中的重要地位，但是该何时创建符号表项，以及其应该拥有哪些属性，暂时没有太多头绪。这需要在实现阶段根据词法分析、语法分析过程中需要的信息进行确定。

3.了解了单趟编译的内容，将其与多趟编译流程进行对比，发现它更加高效，但是其将多个过程压缩在了一起，因此实现的难度也比多趟编译要大。

4.

3.存在的问题

1.随着对编译器了解的深入，发现其涉及的内容十分复杂，而且前后关联特别大，例如符号表的实现就穿插在了词法分析和语法分析中，而且这两个步骤存在很多的细节，很容易就被绕晕了。

2.将各种语句翻译成中间代码的算法，看似不是特别难，但是因为实际编程中通常是各种复杂的语句互相嵌套，因此如何保证生成的代码互相不干扰，还没有理清头绪。