



一、 目标

本次作业的目标是为 Seal 编写一个目标代码生成器，并结合前三个项目，实现一个完整的 Seal 在 x86-64 下的编译器，实现对 Seal 语言文件编译到汇编文件的目标。本次作业是本学期最困难也最具挑战性的一次作业，在对于结果的检查中，我们不限产生汇编代码内容格式，只要求能够对产生的汇编文件汇编之后，产生的二进制结果能够按照编程的目标执行即可。

二、 说明

代码生成器需要使用课程作业二的语法分析和课程作业三的语义分析。代码生成器目标是生成 x86-64 汇编代码(在代码生成过程中无需进行错误恢复——所有的错误程序将被编译器的前段所检测)。

三、 文件说明

- cgen.cc

这个文件将包含代码生成器中所有的代码。代码生成器的入口是 `Program::cgen(ostream&)` 方法，其在 AST 根部被调用。除了常见的常量外，我们还提供了用于发出 x86-64 指令的函数、用于编码字符串、整数、浮点数和布尔值的框架等，你可以根据需要使用他们。

- cgen.h

这是代码生成器的头文件，这里面可以定义各种需要的辅助的类或者函数。

- emit.h

这个文件包含各种用于发出 x86-64 指令的代码生成宏。你可以修改这个文件。

- seal-decl.h、seal-stmt.h、seal-expr.h

该文件中包含 AST 节点的类声明。你可以将数据成员或方法声明添加到本文件中，其中方法的具体定义需添加到 cgen.cc 中。

- cgen_supp.cc

本文件包含代码生成器的通用支持代码。你可以按照需求进行添加，但不要改变已经存在的代码。

四、 设计

代码生成器有许多实现方式，一种可供参考的方式是参考 C 文件编译产生的汇编文件。鉴于 Seal 语法和 C 的相似性，你可以尝试对一个 C 文件利用 `gcc -S <文件> -fno-asynchronous-unwind-tables -o <输出文件>`，编译出汇编代码文件作参考。

在设计代码生成器的过程中，可能会遇到的问题和挑战有

- ①你的代码生成器必须能对 Seal 进行编译工作
- ②你应该对 Seal 程序运行时的语义有一个清晰的认识。
- ③你应该基本理解 x86-64 指令集
- ④你应该决定代码中产生的哪些不变量应该被观察（比如保存哪些寄存器，哪些寄

存器被覆盖、变量存储地址等等)

对于变量的存储,有一些建设性的建议。对于临时变量的值,可以存储在栈上,并且通过一个 map,对每一个临时变量与其地址相对应。

对于寄存器,设想可以利用一个类来实现对空闲寄存器的管理。

其他的包括如何传参、如何返回函数值等,建议采用类似 C 语言标准的方式,即 Int、Bool 和 String 参数将顺次放在%rdi,%rsi,%rdx,%rcx 中,返回值放在%rax 中,而 Float 类型的参数将顺次放在%xmm0,%xmm1,%xmm2 中。推荐阅读《深入理解计算机系统》第 2、3 章,来增进对问题的理解。

另外,因为 Seal 和 C 的语法相近,也推荐利用 gcc 编译器,对一些 C 程序进行编译为汇编文件(指令为 gcc -S <源文件> -o <目标文件>),来参考 gcc 的思想。

此外,printf 这个内置函数的相关部分已经给出,请根据注释按照你的想法把它缺失的部分补全。

五、 测试和 debugging

因为我们开发的环境是 x86-64 环境,所以一个最好的办法就是对产生的汇编代码以 gcc <代码文件> -o main 来进行编译,然后用 ./main 来执行。为了方便测试,在一开始我们就约定 printf 作为一个不可重定义的函数。具体的我们可以利用 printf 函数对某些运算结果进行输出,由于 Seal 中的 Int 类型是 64 位的,故 printf 函数的占位符将会是 %lld。

另外,我们给了 5 个样例文件(Helloworld、斐波那契数列、辗转相除法求最大公约数、求正切值、求群元素的指数),利用 judge.sh 脚本可以实现快速测试。最终在评分时,将继续使用这五个算法,但是其中的数据会改变。

我们给出了一个“标准编译器”的样本,但是汇编代码结果仅供参考,因为在寄存器分配、变量存储等各个方面,每个同学的策略可能都不一样,所以此次作业,我们仅对最终执行结果做评价。

六、 如何评分

我们将利用代码生成器对一些 Seal 文件(如五中所述)进行编译,产生汇编文件,然后利用 gcc 汇编为二进制文件执行,通过观察运行结果来评分。

七、 关于 printf

这个函数我们强制要求实现如下(代码部分为 void Call_class::code(ostream &s)的实现),要求在调用前,将 printf 所有参数中,Float 格式的数目赋给%eax(利用 emit_irmovl),然后将%rsp 减 8,而后再调用 printf。例如 printf("a=%f",1.1),目标代码的结果会有

```
subq $8,%rsp
movl $1,%eax
call printf
的语句段。
```

八、 文件提交要求

要求将原作业目录下的所有文件,放置在一个名为<学号>的目录下,并且将整个目录打包为<学号>_<姓名>.tar 格式。

九、 提交截止时间

请同学们在 2021.1.10 晚上 23:59 之前,将结果提交到 ftp(将会查看文件创建时间, 2021.1.10 晚上 23:59 之后的均视为迟交, 结果按 0 分处理)。