report

问题1

请参照 Environment 的LLVM 配置或者其他资源在你的机器上下载LLVM 11.0的源码，编译它；编译成功后，请将包含有LLVM工具可执行文件的路径，如llvm-install/bin加入到环境变量PATH中。 请记录下：

1）你的机器配置(cat /proc/cpuinfo)、操作系统版本号(uname -a)、你下载的LLVM的版本号(llvm-config --version)、你编译采用的配置模式Release/Debug、编译耗费的时间、编译时占用的内存、编译后占的硬盘资源大小等

机器配置：8个

- processor : 0 (0-7)
- vendor_id : GenuineIntel
- cpu family : 6
- model : 158
- model name : Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
- stepping : 9
- microcode : 0xffffffff
- cpu MHz : 2801.000
- cache size : 256 KB
- physical id : 0
- siblings : 8
- core id : 0
- cpu cores : 4
- apicid : 0
- initial apicid : 0
- fpu : yes
- fpu_exception : yes
- cpuid level : 6
- wp : yes
- flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm pni pclmulqdq dtes64 est tm2 ssse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 movbe popcnt aes xsave osxsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt ibrs ibpb stibp ssbd
- bogomips : 5602.00
- clflush size : 64
- cache_alignment : 64
- address sizes : 36 bits physical, 48 bits virtual
- power management:

操作系统版本号：

Linux DESKTOP-LP1A2G2 4.4.0-19041-Microsoft #488-Microsoft Mon Sep 01 13:43:00 PST 2020 x86_64 x86_64 x86_64 GNU/Linux

llvm版本号：

6.0.1

编译采用的配置模式：

Release

编译耗费的时间：

42min

编译时占用的内存：

在资源管理器中内存(总8GB)使用情况维持在60%左右，实际使用应为100MB-几GB数量级

编译后占用的硬盘资源大小：

384M

2）在编译中遇到的问题、截屏，以及你的解决对策。希望大家在平台的讨论板块进行提问与交流。

问题：在编译过程中报了几个warning，忘记截图了。

问题2

请参考 LLVM Command Guide，使用LLVM的命令来编译 至少3个特征不同的 C程序，输出对应的 LLVM IR文件（.ll），请分析总结C源程序的语法结构与

LLVM IR之间的对应关系。（请将编写的C程序以及输出的LLVM IR文件作为附件上传。）

程序1 hello.c

简单的helloworld程序，hello.ll中显示了.ll文件的基本结构框架。

程序2 type.c

包含了大量类型转换，type.ll中显示了LLVM对类型的控制，同时可以看到，printf中待输出的字符串都以全局变量的形式保存。

程序3 BiTree.c

其中包含了结构体，函数调用，控制语句，指针等，.ll文件中显示了将代码分成一个个基本块来实现，以及对结构体等的实现。

问题3

请给出以下 C 语言代码到 LLVM IR 的人工翻译，将人工翻译的结果保存到fib.ll, 并使用lli命令测试其结果。请记录在你人工翻译中非一次性成功的代码片段，总结其对应的正确翻译规则。

在翻译过程中遇到的最主要问题是变量的使用，前面数次使用lli命令测试时报错，都是变量序号，类型使用不当的报错，在逐一修改过后，正确执行。使用echo $?，输出正确结果33。

问题4

请结合第2和3题涉及的IR特征，根据 LLVM IR及其构建 提供的线索，熟悉 LLVM IR的相关表示以及 IRBuilder 的工作逻辑，并总结你的理解。为日后开展实验做好准备。

在第2题输出LLVM IR文件时，尝试使用了-O3选项，发现生成的.ll文件中代码量，明显减少，但同时可读性也有所降低。LLVM IR表示，在不使用优化选项时，与源代码的结构基本一致，在变量声明等管理上更为合理规整。

在代码结构上，通过基本块及其前驱，便能读懂程序的思路，可读性明显优于汇编代码。

在第3题翻译过程中，使用变量的过程中，需要时刻把握每一个变量的类型和含义，应先转换为三地址代码，再进行翻译，能提高翻译效率。