

编译实验 - 实验感想

22371027-汤睿璟

回顾一学期的编译实验，可谓是充满遗憾。最开始想着要生成 MIPS 并且把自己能够完成的优化都尽可能地给做了，但是不论是时间还是从对编译器的设计理解，都不允许我生成 MIPS。如果还有将来的话，我一定会做的更好。

0.1 生成 MIPS 心得

在生成 MIPS 的时候，或许是理论学习不扎实，或许是别的原因，我在编译器的设计上与目标需求越走越偏。等到意识到不得不重构的时候，正是数据库大作业和编译双重打击的时候。最终，我完成了图着色寄存器分配，基本上都快完成 MIPS 的生成了，这时才发现一直没有考虑函数的形参，也发现了图着色算法也一直没有考虑函数形参，回想起来后悔不已，要是之前考虑上了，或许 MIPS 就已经生成出来了。最终放弃了 MIPS。

0.2 关于代码优化的感悟

代码优化是一件很有意思的事情。理论就摆在哪儿，实现起来差异却很大。这一点一直让我感到很困惑。拿 mem2reg 来说，看起来耗不起眼的优化，在实现起来的时候却十分复杂，计算支配集和支配边界的算法也是在近几十年才提出的。可能还有相当多的优化算法在教材里面根本就没有出现过，但是却在工业界广泛使用。我想这一点或许也和实际比较接近：之后学到的很多算法或许都会投入到实际使用中，而远远不想之前想象的什么图算法、排序算法这一类这么理论，简洁。实际上拿到的数据格式可能是一种十分复杂的数据格式，而需要进行相当多的加工才能将算法融入到实际应用当中。

我在暑假参加龙芯杯的时候，就发现龙芯教育推出了 32 位龙架构编译器编译出来的代码我就发现，编译出来的代码我似乎看不懂。现在回想起来，恐怕是做了循环展开，并且展开地相当大。在使用 Clion 写编译器的时候，Clion 的 clang-tidy 对代码的静态解析也相当强，数据流解析做得相当全面。而且不仅如此，在实现如此复杂的代码静态分析之后还能保持相当高的代码分析速度，这现在看来也是难以想象的事情。或许是我把编译优化想象得太简单了吧，但是编译能够做到的事情细想起来真的非常强大，而且背后蕴含的理论似乎并不简单。

就算不说工业界现在采用的编译优化，哪怕是课本上的编译优化，也有相当多的一部分没有涉及；即使是涉及到了也难以想象具体的实现方式。尽管最终没能生成 MIPS，但是在和同学进行的优化编译优化的讨论中也能学到不少内容。或者说

是算法吧。之后如果真的和编译器打交道的话，或许还不一定是和 MIPS 这一类运行在比较老的机器上的机器语言打交道，可能还会有更多的事情发生，比如和 GPU 或协处理器有关的机器语言，或者是和

0.3 关于课程设计的一些思考

课程设计总体来说已经很完美了。只是有一些小地方或许可以做的更好？比如，符号表的输出顺序和作用域的起始位置顺序相同这一点，个人感觉比较不符合“栈式符号表”的意思。尽管我们做的符号表不一定是栈式的，但是作用域构成的结构到底还是呈栈式的，应当在以结束顺序作为输出顺序或许会较为合适？

0.4 写在最后

编译写得确实比较烂，不过成绩烂归成绩烂，学到东西了就好。这学期逼着自己用 C++ 和 CMake 硬着头皮把编译器写下去了，尽管很不符合 Modern C++ 的设计理念，不过到底还是更加熟悉 C++ 了一点。从这个角度来说倒也是足够了。