本文档主要是用于自己学习LLVM及其他编译优化的东西，便于设计IR及后端流程

* **LLVM后端流程和几种后端表示**

后端流水线：

先是IR转每个BB为selectionDAG，

随后，根据这个DAG做指令选择以及合法化（两者应该放在一起看，大概就是得到了目标平台的对应代码DAG）

在这一步之后，就是instruction schedule，大概就是根据DAG得到一个序列化的指令流。这一步其实可以简单的根据DAG的拓扑排序就好了。

随后则是寄存器分配，寄存器分配本身有些在instruction selection那边就已经做了的，比如有些目标平台的指令会要求某些寄存器，这些寄存器在执行这些指令之前是不能存放有用的东西的。

另外在这里需要做一些比如copy这样的指令的合并。做这个算法之前还要什么活跃变量分析巴拉巴拉。

如果溢出了，那么需要弄到栈上面去。

然后目标指令都有了，寄存器也都有了，基本上这个东西也就可以生成汇编指令了。

（有一说一，光是这些实现一波就已经很不容易了）

其他：

调用约定（我不太清楚这个，但是既然LLVM把调用约定也专门作为一个模块，那么不能写死在代码里面。）

* **指令选择：**

听说指令选择是占用后端流水线最多时间的部分了，大约有宏替换，树covering，以及DAG covering等等。（我看看文献研究一下）

* **寄存器分配**