

工作报告

张其良¹

中国科学技术大学 计算机科学与技术学院

January 16, 2014

¹929434288@qq.com

目录

1	获得位置的方法	1
1.1	AST 层次上	1
1.2	LLVM IR 层上	2

1 获得位置的方法

1.1 AST 层次上

```
1 std::pair<std::pair, std::pair> Pair;
2 ForStmt *FS = *sFS;
3 //第一种。
4 //SourceManager SM = Context.getSourceManager().
5 SourceRange SR = FS->getSourceRange();
6 SourceLocation SLB = SR.getBegin();
7 SourceLocation SLE = SR.getEnd();
8 PresumedLoc PLoc1 = SM.getPresumedLoc(SLB);
9 PresumedLoc PLoc2 = SM.getPresumedLoc(SLE);
10 Pair.first.first = PLoc1.getLine();
11 Pair.first.second = PLoc1.getColumn();
12 Pair.second.first = PLoc2.getLine();
13 Pair.second.second = PLoc2.getColumn();
14 //llvm::outs() << PLoc1.getLine() << " " << PLoc2.
    getLine() << "\n";
15 //第二种。
16 FullSourceLoc FullLocation1 = Context.getFullLoc(FS->
    getLocStart());
17 FullSourceLoc FullLocation2 = Context.getFullLoc(FS->
    getLocEnd());
18 Pair.first.first = FullLocation1.getSpellingLineNumber()
    ;
19 Pair.first.second = FullLocation1.
    getSpellingColumnNumber();
20 Pair.second.first = FullLocation2.getSpellingLineNumber
    ();
21 Pair.second.second = FullLocation2.
    getSpellingColumnNumber();
22 LocRange[FS] = Pair;
23 llvm::outs() << LocRange[FS].first.first << "□□" <<
    LocRange[FS].first.second << "\n";
24 llvm::outs() << LocRange[FS].second.first << "□□" <<
    LocRange[FS].second.second << "\n";
```

获取指定 SourceLocation 对应的标识符的开始位置和结束位置,

```
1 //获得语言的类型。
2 const LangOptions &Lang = context.getLangOpts()
3 SourceManager &SM = context.getSourceManager();
4 SourceLocation End = SR.getEnd(); //得到一个语句最后一个标识符
5 //如果是宏获得宏开始和结束的位置,
6 const std::pair<SourceLocation, SourceLocation> ExpansionLoc(SM
```

```

    .getExpansionRange(End));
7   const SourceLocation Ebegin = ExpansionLoc.first;
8   //宏结束的位置宏中最后一个标识符位置,
9   const SourceLocation EEnd = ExpansionLoc.second;
10  //得到最后一个标识符的长度
11  SourceLocation EndLoc(clang::Lexer::getLocForEndOfToken(EEnd,
    1, SM, Lang));
12
13  bool Invalid = false;
14  //得到指向给定位置的指针
15  const char *EEndptr = SM.getCharacterData(EndLoc,&Invalid);

```

1.2 LLVM IR 层上

clang -g -emit-llvm -S hello.c -o a.ll, 加上参数产生.ll 文件时, 保留了对应的行信息, 若获取的话可以从 DebugInfo 入手.