

编译器安装说明

1、安装程序运行环境

本编译器的运行环境如下：

```
gcc 9.4.0
g++ 9.4.0
make 4.2.1
flex 2.6.4 bison 3.5.1
clang 10.0 llvm 10.0
```

安装方法一

通过提供的Dockfile文件安装Docker容器，配置Docker容器环境。

下面仅提供在VSCode的Docker使用方法。

- 在VS Code中安装Docker和Remote - Containers扩展
- 将Dockerfile放在准备用来存放代码的文件夹中
- 点击左下角蓝色或者绿色的箭头部分。
- 在VS Code顶端中选取Remote - Containers : Open Folder in Containers选项，选取在容器中打开的本地文件夹，然后选取From 'Dockerfile'选项。
- 等待建立镜像和容器，建立完毕后该文件夹即在容器中打开。成功标志是左下角双箭头处显示"DevContainer:Existing Dockerfile"。

安装方法二

按照如下语句逐条执行

更新apt-get安装系统管理指令

```
apt-get update
apt-get install sudo -y
```

更新apt

```
sudo apt update -y
```

安装wget

```
sudo apt install wget -y
```

安装 gcc g++ make

```
sudo apt install build-essential -y
```

安装flex, bison

```
sudo apt install flex bison -y
```

安装 clang 和 llvm

```
# ubuntu 20.04 默认版本 10.0
RUN sudo apt install clang -y
RUN sudo apt install llvm -y
```

安装locate

```
RUN sudo apt-get install mlocate -y
RUN sudo updatedb
```

2、编译源码

记录所有生成的.o文件

```
OBJS = yacc.o \
       lex.o \
       ParseT2AST.o \
       ASTnodes.o \
       TypeSystem.o \
       CodeGen.o \
       symbolTable.o \
       semanticAnalyse.o \
       ObjGen.o \
       main.o \
```

确定编译参数

```
LLVMCONFIG = llvm-config-10
CPPFLAGS = `$(LLVMCONFIG) --cppflags` -std=c++14
LIBS = `$(LLVMCONFIG) --libs`
```

对yacc和lex的源文件进行编译，生成yacc.cpp和lex.cpp

```
yacc.cpp: yacc.y
        bison -d -o $@ $<

yacc.hpp: yacc.cpp

lex.cpp: lex.l yacc.hpp
        flex -o $@ $<
```

检查现有文件是否有更新

```
ParseT2AST.cpp: ParseT2AST.h

ASTnodes.cpp: ASTnodes.h

TypeSystem.cpp: TypeSystem.h

CodeGen.cpp: CodeGen.h

symbolTable.cpp: symbolTable.h
```

```
semanticAnalyse.cpp: semanticAnalyse.h CodeGen.h  
  
ObjGen.cpp: ObjGen.h  
  
main.cpp: ObjGen.h semanticAnalyse.h ParseT2AST.h
```

对每个.cpp生成.o文件

```
%.o: %.cpp  
    g++ -c $(CPPFLAGS) -o $@ $<
```

生成可执行文件 compiler

```
compiler: $(OBJS)  
    g++ $(CPPFLAGS) -o $@ $(OBJS) $(LIBS)
```

以上内容都封装在Makefile文件之中，在编译时仅需要输入

```
make
```

即运行makefile编译源码，生成可执行的编译器compiler