## 测试环境

Linux version 4.4.0-18362-Microsoft (Microsoft@Microsoft.com) (gcc version 5.4.0 (GCC) )

bison (GNU Bison) 3.0.4

flex 2.6.4

## 程序清单说明

.

├── analysisError.c

├── analysisError.h

├── codeanalysis.y //yacc语法分析、语义分析、代码生成

├── lex.l //lex词法分析

├── Makefile

├── sematic.txt

├── symbolTable.c //符号表管理

├── symbolTable.h

├── transform.out //可执行程序

├── util.c //一些辅助函数定义

├── util.h

├── 单元测试

│   ├── a.out

│   ├── symbolTable.c //符号表

│   ├── symbolTable.h

│   ├── test

│   ├── test\_lex.l

│   ├── test\_parser.y

│   └── test\_sym.c

├── 张帅帅组课程设计报告.docx

├── 测试用例

│   ├── array\_index.pas //数组越界

│   ├── base.out

│   ├── base.pas //较复杂的基准测试，覆盖大部分测试要点，无错误

│   ├── base.pas.c

│   ├── gcd.pas //gcd测试

│   ├── gcd.pas.c

│   ├── lexerror.pas //词法错误

│   ├── quick\_sort.pas //快排

│   ├── quick\_sort.pas.c

│   ├── redeclare\_undeclare.pas //未定义和重定义

│   ├── sematic.pas //语法错误

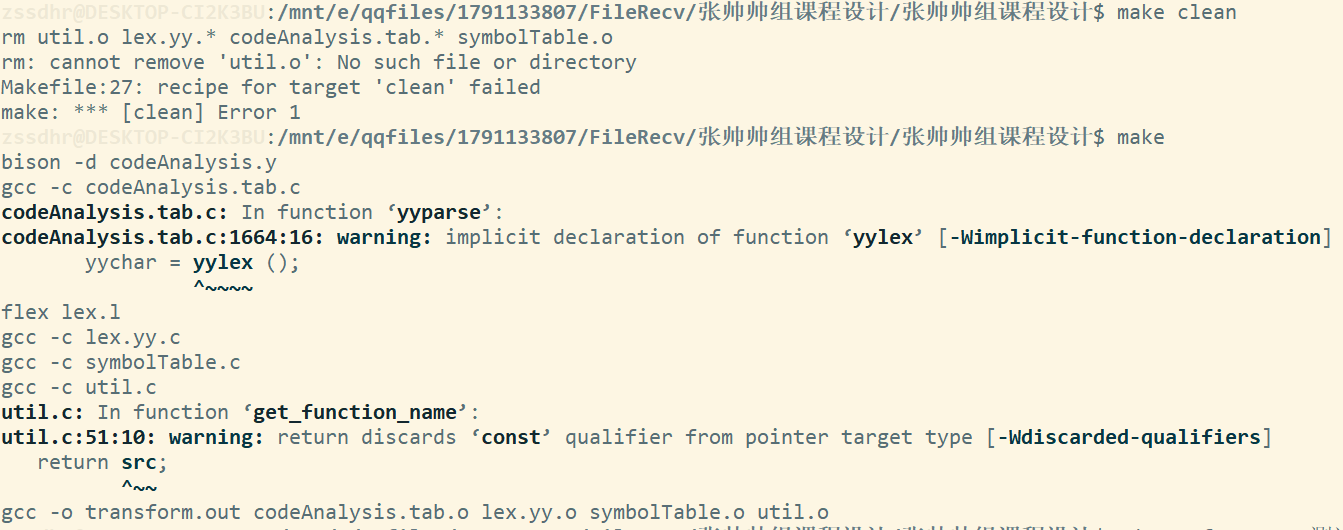
│   └── type\_check.pas //各种类型错误

├── ~$程序使用说明.docx

└── 程序使用说明.docx

## 程序使用

1. 你可以跳过这一步，直接进入第2步执行可执行程序。也可以手动编译：进入文件所在文件夹，先输入make clean命令进行清理，然后和输入make命令，进行编译。

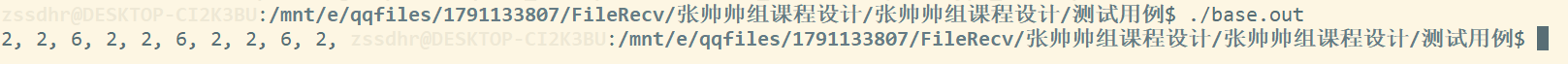


1. 在编译成功后，生成可执行文件transform.out，运行方式为./transform.out 源程序文件名。



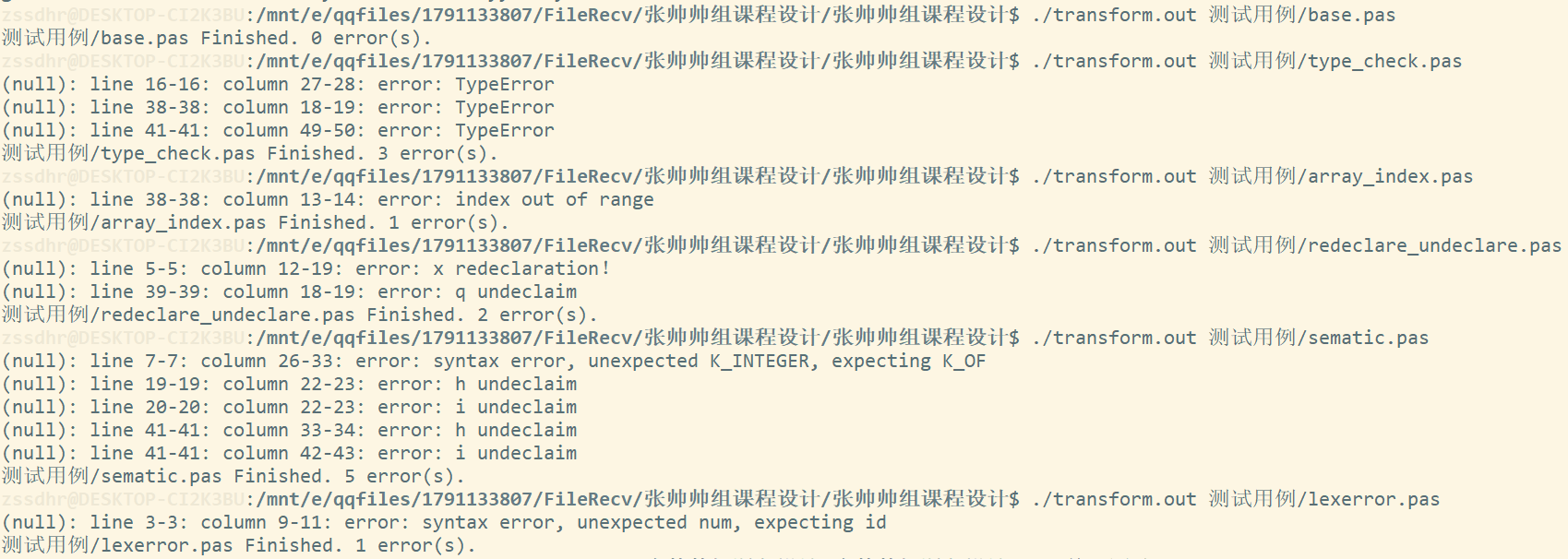
3.编译成功的话会提示编译完成，没有出现错误，并输出到相应文件夹。将.c文件中的printf和scanf手动改正（%s, %d格式）后可用gcc编译运行。





4.如果源Pascal代码中有错误会出现报错，并输出报错信息以及具体出现错误的地方，方便用户进行代码修改。

5.我们在测试时，进行了数组越界测试、未定义、重复定义测试、语法错误测试等，程序都会进行相应的报错。



6.单元测试，进入单元测试文件夹，执行gcc symbolTable.c test\_sym.c -o test编译，然后执行./test可对符号表进行单元测试。

