



计算机科学
与技术系

实验一 词法分析和语法分析

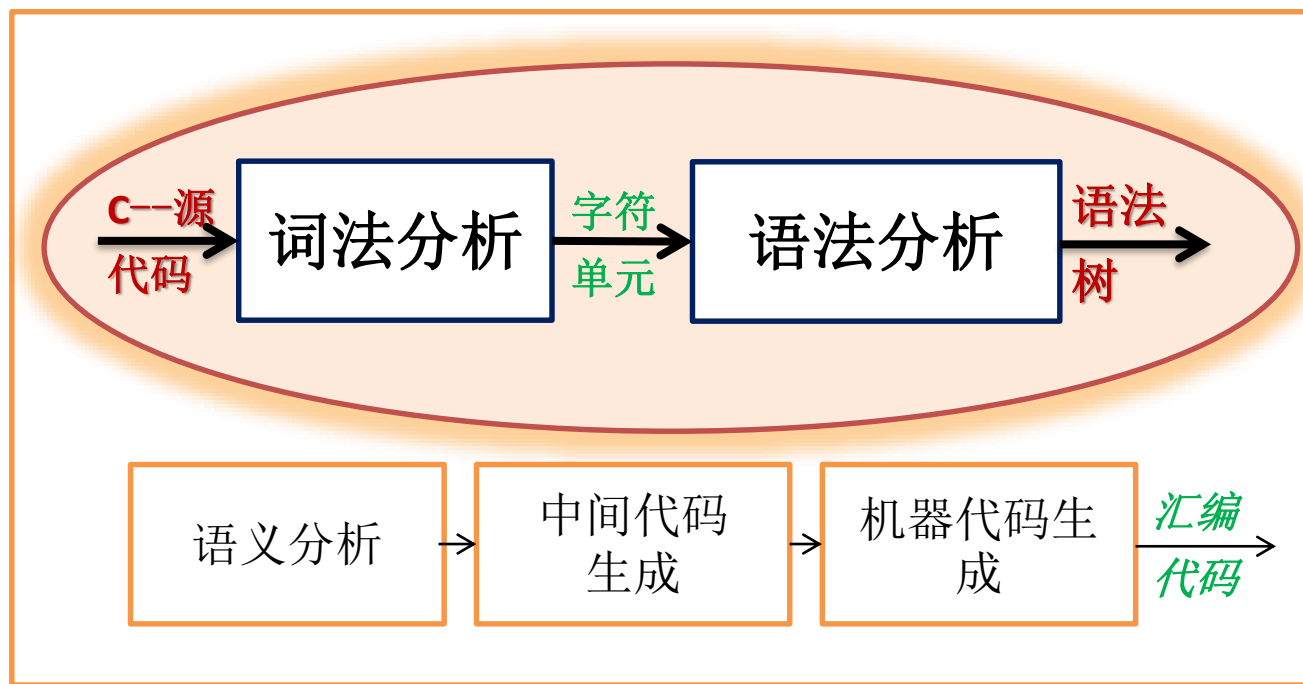
老师：戴新宇

助教：宁天昊，邱丰羽

{ningth, qiufy}@nlp.nju.edu.cn

概要

- 实验任务
 - 必做
 - 选做
- 实验提交
 - 提交内容
 - 提交方式
- 实验检查
 - 运行环境
 - 代码查重



实验任务

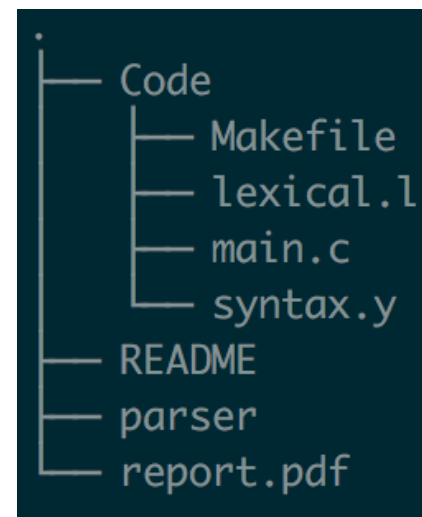
- 准备工作
 - 熟悉C--语言的文法（Appendix_A_实验语法.pdf）
- 词法分析
 - **必做**：能够查出C--源代码中的词法错误
 - **选做**：识别八进制（012 ✓，081 ✗）、十六进制整数（0x89/0xA b/0XaB ✓，0xG5 ✗），识别指数形式的浮点数（1E-2/01.2E+34/ 43.e-3/.5e02 ✓，9.8E7.6 ✗）
- 语法分析
 - **必做**：能够查出C--源代码中的语法错误
 - **选做**：正确处理两种风格的注释
- 没有词法和语法错误的情况，则打印语法树

实验提交方式

- 提交平台: <http://cslabcms.nju.edu.cn>
 - 使用个人账号登录, 独立完成
 - 各位选课同学列表已经导入, 请完善个人信息
 - 上述过程中的任何问题请联系助教或老师
 - 无特殊情况不接受其他提交方式
- 提交第三周的项目作业
 - Project1 词法分析和语法分析
 - 所有内容打包并压缩, 命名为学号+"_"+lab1.zip/rar/tar.gz, 如141220001_lab1.zip
- 截止日期: **4月9日, 23:59:59**, 请尽量不要在此之前的几分钟提交, 网络有风险

实验提交内容

- 压缩包内容至少包含
 - 源代码（*.l, *.y, 其他.h, .c文件）
 - 尽量遵循我们发布的代码规范
 - 可执行文件（命名为parser）
 - 可以提供一個Makefile
 - 实验报告
 - 最好是pdf，可选docx，暂不支持doc
 - 报告内容包括个人信息、完成的功能点，实现方法，用到的数据结构表示，编译运行方法，实验总结等
 - 不要贴大段代码，篇幅不超过4页
 - README可选



Makefile示例

GNU make手册：<http://www.gnu.org/software/make/manual/make.html>

***** 遇到不明白的地方请google以及阅读手册 *****

编译器设定和编译选项

CC = gcc

FLEX = flex

BISON = bison

CFLAGS = -std=c99

编译目标：src目录下的所有.c文件

CFILES = \$(shell find ./ -name "*.c")

OBJS = \$(CFILES:.c=.o)

LFILE = \$(shell find ./ -name "*.l")

YFILE = \$(shell find ./ -name "*.y")

LFC = \$(shell find ./ -name "*.l" | sed s/[^/]*\\.l/lex.yy.c/)

YFC = \$(shell find ./ -name "*.y" | sed s/[^/]*\\.y/syntax.tab.c/)

LFO = \$(LFC:.c=.o)

YFO = \$(YFC:.c=.o)

Makefile示例（续）

```
parser: syntax $(filter-out $(LFO),$(OBJS))
        $(CC) -o parser $(filter-out $(LFO),$(OBJS)) -lfl -ly

syntax: lexical syntax-c
        $(CC) -c $(YFC) -o $(YFO)

lexical: $(LFILE)
        $(FLEX) -o $(LFC) $(LFILE)

syntax-c: $(YFILE)
        $(BISON) -o $(YFC) -d -v $(YFILE)

-include $(patsubst %.o, %.d, $(OBJS))

# 定义的一些伪目标
.PHONY: clean test

test:
        ./parser test.cmm

clean:
        rm -f parser lex.yy.c syntax.tab.c syntax.tab.h syntax.output
        rm -f $(OBJS) $(OBJS:.o=.d)
        rm -f $(LFC) $(YFC) $(YFC:.c=.h)
        rm -f *~
```

实验检查

- 检查环境
 - Ubuntu 16.04
 - GCC 5.3.x
 - GNU Flex 2.6.0
 - GNU Bison 3.0.4
- 事实上，如果不使用较冷门或非正式、稳定版本，基本不会有兼容的问题，你甚至可以在 macOS 上完成

实验评分

- 你可以先试试实验手册上的测试样例，我们会使用其他类似的测试样例进行检查并评分
- **必做部分（80%）**
 - 词法分析（40%）
 - 语法分析（40%）
- **实验报告及代码风格、实现方式等（20%）**
- **选做部分（bonus 15%）**
 - 识别八进制、十六进制整数（5%）
 - 识别指数形式浮点数（5%）
 - 识别两种注释风格（5%）

代码查重

- 参考网上的任何代码请注明出处！
- 注意区别参考与抄袭，任何形式的代码抄袭都是不允许的！抄袭者与被抄袭者本次实验都记为0分！



- 祝你好运！