COEN 175

Week 1 - Phase 1

TAs

Stephen Tambussi

Email: stambussi@scu.edu

Office Hours: Tuesday 1:00 - 2:00PM / Wednesday 4:00 - 5:00PM (Heafey Atrium)

Jordan Randleman

Email: jrandleman@scu.edu

Office Hours: Tuesday 9:10-10:10AM / Thursday 9:10-10:10AM (Heafey Atrium)

Introduction

- Building a C compiler in 6 phases!
  - Jordan: Monday 2:15-5pm, Tuesday 5:15-8pm
  - Stephen: Monday 5:15-8pm, Tuesday 2:15-5pm
- **There are labs on MLK day (January 16th), same times. You can attend either section.**
- Submissions
  - Tar file uploaded to camino
  - Typically due on Sundays at 11:59PM (-1pt for every minute late!)
- Advice
  - Keep up with lecture material
    - ***Or you will fail miserably and have to spend another $20K for an additional quarter at SCU***
  - Read the entire assignment (PDF & slides!) very carefully (most questions can be answered there)
  - Start early
    - ***A 2 week lab does not mean screw off for a week and a half then struggle at the last minute***
  - Write your own test cases
    - Dr. Atkinson’s provided ones do ***NOT*** cover all possibilities – you must fully exercise your own code!
- Must run and compile on linux servers
  - Make sure that you use the ORIGINAL makefile when doing so! – all of your own changes must be reverted prior submission!

How our Compiler Works

- Read in from standard input
- Write to standard output

- Running
  - ./scc < *example.c* > output.out OR ./scc use cltr+d for eof
- Testing
  - diff output.out *example.out*
    - Program is correct if there is no output

Recommended Directory Structure

```
├── phase1
    ├── lexer.l
    ├── Makefile
├── examples
    ├── example1.c
    ├── example1.out
    ├── example2.c
    └── example2.out
├── CHECKSUB.sh
```

Submitting

1. Create tar file (phase1.tar) containing your source code
2. Run CHECKSUB to make sure tar file works
   a. ./CHECKSUB.sh phase1.tar examples.tar
      i. If you get a permission error at this point run: chmod 770 CHECKSUB.sh
3. Submit tar file to camino

- Run CHECKSUB before you leave lab today to make sure that you do not have issues running it later
- If CHECKSUB does not compile your submission, you will receive a 0.

Phase 1 - Lexical Analysis

- Write a lexical analyzer using flex
  - The Makefile will convert your “.l” file into a “.cpp” file for you, with the C++ code that flex created from your “.l” regex/C++ code pair instructions
- Print out all lexical constructs (tokens) recognized from standard in
- All whitespace, comments, illegal characters to be ignored
  - All rules on Assignment document
  - You will only be given lexically correct Simple C code in your tests for this phase

- Example
  - Standard in: **123**
  - Standard out: **integer 123**
- Due Sunday January 15th, 11:59PM

Hints

Comments

- Don’t bother trying to write a regular expression for a comment
- Write a function to scan a comment by hand
- Use yyinput() to read in a character

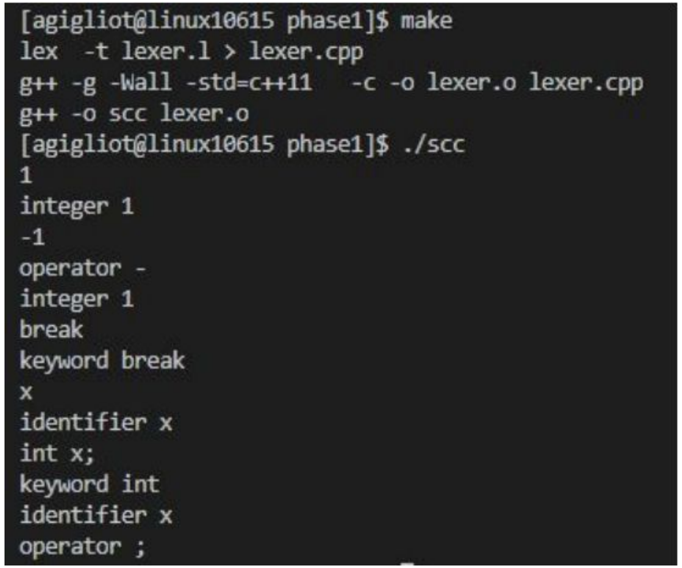
Strings

- Need to escape quote to properly match them
  - \” to match quotes
- Check lecture slides for more explanation
  - ***This will be a recurring theme throughout all of this quarter!***

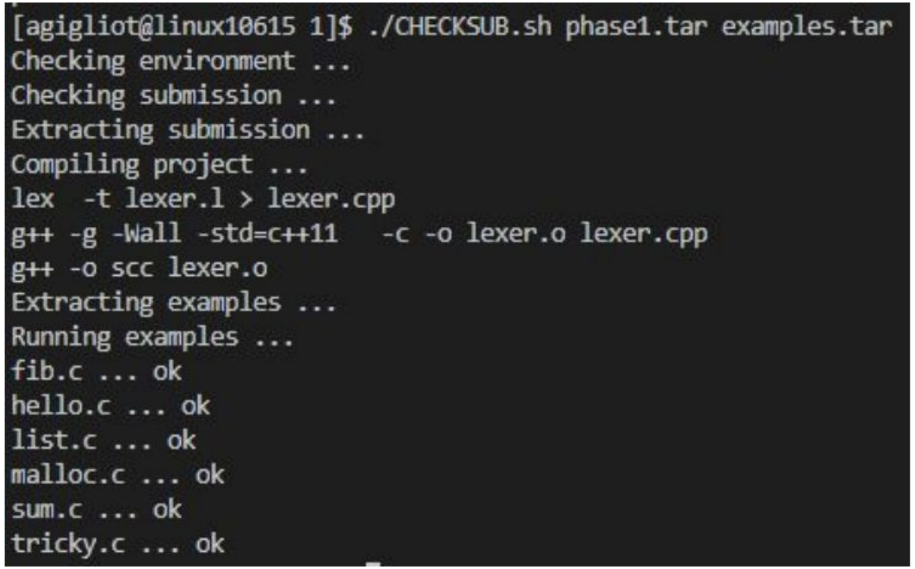
General

- Incrementally develop your solution: write one rule at a time and then test

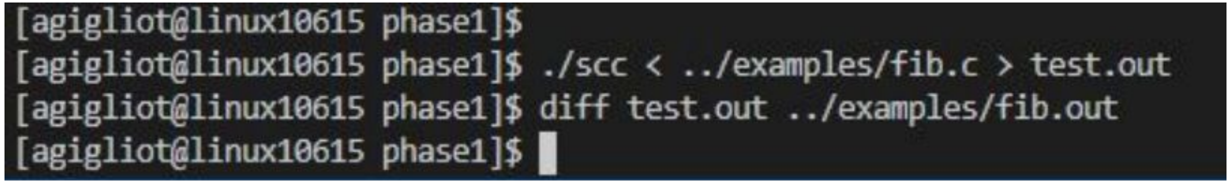
Examples



```
[agigliot@linux10615 phase1]$ make
lex  -t lexer.l > lexer.cpp
g++ -g -Wall -std=c++11   -c -o lexer.o lexer.cpp
g++ -o scc lexer.o
[agigliot@linux10615 phase1]$ ./scc
1
integer 1
-1
operator -
integer 1
break
keyword break
x
identifier x
int x;
keyword int
identifier x
operator ;
```



```
[agigliot@linux10615 1]$ ./CHECKSUB.sh phase1.tar examples.tar
Checking environment ...
Checking submission ...
Extracting submission ...
Compiling project ...
lex  -t lexer.l > lexer.cpp
g++ -g -Wall -std=c++11   -c -o lexer.o lexer.cpp
g++ -o scc lexer.o
Extracting examples ...
Running examples ...
fib.c ... ok
hello.c ... ok
list.c ... ok
malloc.c ... ok
sum.c ... ok
tricky.c ... ok
```



```
[agigliot@linux10615 phase1]$
[agigliot@linux10615 phase1]$ ./scc < ../examples/fib.c > test.out
[agigliot@linux10615 phase1]$ diff test.out ../examples/fib.out
[agigliot@linux10615 phase1]$ 
```