

Advanced Programming languages

Programming Project 3

Project #3 - A Parser Generator Project.

yacc/lex Compiler Front End.

You will rewrite the grammar from your compiler into a left-recursive format typical of LR parsers. Implement a parser for the LR version of the grammar using yacc and lex. I will expect you to have four separate files for your parser: a .l file from lex and a .y file from yacc, a header file with necessary constants for token types, and a `makefile` for the project.

In class we will discuss the use of lexical analyzer/parser generators and you will be provided with examples of simple programs which you will extend.

Input and Output

There will be no user interaction with the program. The program will be invoked 8 times from a shell script as demonstrated in class (see below in deliverables). Input will be via command line parameter of the file to be parsed, specified in the script file. The final legal program must demonstrate proper identifier processing, nested expression processing and control constructs nested inside others. Output will be to the console reporting either a successful parse or a syntax error message.

Other Policies

All projects will be individual.

Deliverables:

You will submit all your work in a .zip file to the DropBox for the course on elearning.uwf.edu. Place *all your files* in a single directory named

```
<firstInitial><lastname>-p<projNumber>.
```

For instance, my submission would be `jcoffey-p3.zip` that folder into a .zip file and upload it to the appropriate dropbox.

So, submit a single zip file containing a single folder, with the following items in the folder:

- An updated version of the grammar of your language.
- Your lex and yacc programs, the header file, and the makefile
- A User's manual regarding exactly what this submission does and how to run it (and any problems or deficiencies the submission might have)
- Four examples each of both legal and illegal programs of increasing complexity. The programs should be of increasing complexity, with the last demonstrating all features of your language.
- A script that runs all legal and illegal programs as a batch, citing success or flagging the error.