F.I.R.E Final Report

Graham Patterson (gpp2109)
Frank Spano (fas2154)
Ayer Chan (oc2237)
Christopher Thomas (cpt2132)
Jason Konikow (jk4057)

The limits of my language are the limits of my world. – Ludwig Wittgenstein, Tractatus Logico-Philosophicus

Table of contents

1. Introduction

- *Include your language white paper.*

F.I.R.E (File Input Reinterpretation Engine) is a programming language inspired by AWK, Bash & C. It seeks to provide some of the robust pattern matching and script-manipulation functionality of these languages with a more palatable syntax.

1.1 Motivation

AWK & Bash remain essential weapons in the arsenal of any programmer seeking to manipulate text files in a Unix-like environment. They are robust and integrate

well with other Unix tools, but they are also artifacts of a time before the total dominance of C in the landscape of programming language design. As a result, both languages feature syntax that is alien to generations of programmers reared on C and C-like languages like Java, C++, Python and et al. Bash, for all its utility, has gained a reputation for fiddly and non-intuitive syntax rules (ever leave an extraneous space in a `makefile` or `.bashrc`? You know what I'm talking about).

Since C is so well regarded and so well-known, we have made it the inspiration for much of our syntax. If you code in C or a language inspired by C, much of FIRE will feel as comfortable as a cardigan.

1.2 Target Audience

The kind of programmer that will find FIRE useful spends many hours in the terminal. He or she or they will likely have to manipulate dozens if not hundreds of text files. They may be a systems administrator, sifting through system logs, or a scientist working through huge data sets. F.I.R.E empowers this target user with tools to expedite tasks like pattern matching (via our robust regex library, built into the language), the opening and creation of files, and more.

2. Tutorial

- *A short explanation telling a novice how to use your language.*

This tutorial assumes a certain degree of familiarity with programming languages, compilation and the command-line. We recommend brushing up on basic git commands like `git pull` and with docker as well. Use git clone to download a local copy of our repository, named `plt-f18`:

`git clone https://github.com/pattersongp/plt-f18.git`

2.1 Environment Setup

It is highly recommended that you use the Docker container utilized by our development team when compiling FIRE programs. Based on a Docker container built by Programming Languages & Translators TA John Hui, the Dockerfile for the container lies in the root directory of our repository, with corresponding directions provided by the `Readme.md` file. Those directions are reiterated here:

2.1.1 Docker Configuration

For help with installing Docker on your machine, please consult [Docker's documentation](#)

1. Ensure that you are in the root of the repository: `cd plt-f18`
2. Build a docker image via the following command: `docker build -t fire .` Note that a `.` is required at the end of the command.
3. To execute the container, use the following command: `docker run --rm -it -v \`pwd\`:/home/fire -w=/home/fire fire`

Please note that these commands may vary slightly depending on the shell you are using. Also note that in the third command, a bash variable `pwd` is employed – this prints your local directory. It can be substituted with the absolute or relative path of the root directory of the repository.

2.1.2 Dependencies

If you elect not to utilize a Docker container, please ensure you have the following packages installed on your machine:

```
ocaml
menhir
llvm-6.0
llvm-6.0-dev
m4
git
aspcud
ca-certificates
python2.7
pkg-config
cmake
opam
```

Utilize an appropriate package manager for your machine to install the above, e.g. `homebrew` on a mac or `apt-get` on a Debian-based Linux distribution.

2.2 Building the compiler

For all subsequent sections of our tutorial, if you are intending to use our docker container, ensure you are inside the container by running the 3rd command

listed in Section 2.1.1 — otherwise you will run into compilation error messages constantly.

Navigate to the source folder of the repository, via `cd src`. Then run `make` to build the compiler of our language, `fire.native`. Note that `make` runs an automated test suite, compiling valid and invalid FIRE programs and outputting whether or not they have succeeded or failed. This is a highly useful tool for anyone working on extending or testing the language, but if your goal is only to write FIRE programs, you can safely terminate the automated test suite with `CMD + D`.

2.3 Compiling a FIRE Program

Consider this very basic FIRE program, `hello.fire`:

```
func int main = () => {

    str s = "Hello World";
    sprint(s);

}
```

As you may have gathered, this program outputs "Hello World" to `stdout`. To compile it, ensure you are in the `/src` directory of your copy of the FIRE repository, and run:

`./fire.native hello.fire`

(Please ensure you have hello.fire located in your source directory, or that you reflect its current path accurately when supplying it as an argument).

The compiler will now generate an executable, `hello.flames`, as well as several intermediate files used in compilation, usually with the extension `.ll` and or `.s`. These artifact files can be deleted without repercussion as they are only useful in the process of building an executable.

As the above example illustrates, compiling a file is much the same as it is with other languages — supply a source file to the compiler as an argument, and the compiler will generate an executable binary. Running the above will output `Hello World!`, as expected.

2.4 Clean Up

If you would like to clean up binaries or recompile the FIRE compiler, run
`make clean` while inside the `src` folder. Make clean not only removes the
existing binary for the FIRE compiler, it removes all the artifacts of
compilation (like `.ll` files) and will clean up artifacts generated the the test
suite as well.

3. Language Reference Manual

- *Include your language reference manual.*

4. Project Plan

- *Identify process used for planning, specification, development and testing*
- *Include a one-page programming style guide used by the team*
- *Show your project timeline*
- *Identify roles and responsibilities of each team member*
- *Describe the software development environment used (tools and languages)*
  Include your project log

4.1 Our Process

4.1.1 Weekly Meetings

Our team conferred in twice-weekly sprints throughout the semester, with one
meeting on Tuesday afternoons and the other on Thursday afternoons. The Tuesday
meetings took place between 2 and 3 pm, and were often our weekly opportunity to
sort out logistics, assign tasks for the week, and engage in high-level planning
about the direction our language would take during development. Our Tuesday
meetings were followed by a half-hour to hour meeting with our TA and mentor John
Hui. The timing of our Tuesday meetings proved to be hugely helpful, as we were
able to formulate questions and concerns that we could almost immediately field
to John for input.

Our Thursday meetings were comprised of long coding sprints, from 4 to 6 pm,
though in practice we often began work earlier than the specified time. Though we
frequently did work outside of the sprints, sometimes remotely and separated from

one another, these Thursday sprints gave us a chance to confer and collaborate in person, keeping us on the same page.

Numerous times in the semester we broke the cadence of twice weekly meetings to meet every day of the week, most notably around the development of the `Semant.ml`.

4.1.2 Team Communication

Our team utilize a slack channel to manage communications between our five-member group. This proved indispensable, primarily because Slack was a familiar technology for almost all of us, and had useful features like support for Code Blocks and a mobile app. Helpfully, all of our members were responsive to messages and replied promptly, which enabled us to quickly pivot or assign tasks to one another. Also of note is the fact that our TA John Hui joined our Slack, which enabled us to quickly ping him with questions (and enable him to respond, almost as quickly).
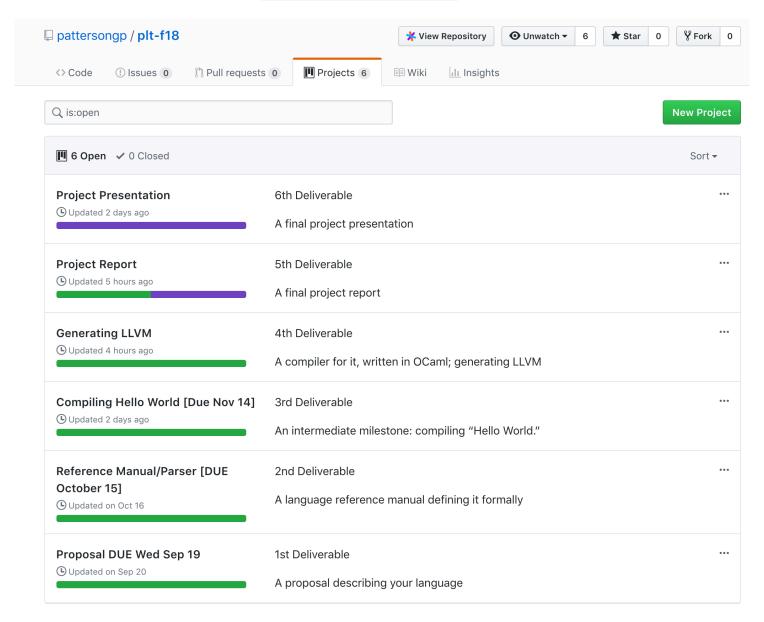
4.1.3 Git

Our Project utilized git & Github for version control and project management. We made liberal use of branches, but mandated that pull requests into master required approval from at least one other team member. That team member was responsible for a) vetting the changes made and for b) providing a rationale for their approval of the pull request. Branches were made not only for major features like the semant or the test-suite, but also for LRM and final project, and also for subsets of a feature divided among multiple teammates. This project proved to be a great excuse to develop a familiarity with Git and its unbelievably robust feature set.

4.1.4 Github Projects

Beyond availing ourselves of Github's free hosting, our project also leveraged several of Github's project management tools. Github Projects is a kanban board built into every Github repository. It allowed for our team Manager, Graham Patterson, to break down our project into tasks and subtasks, with major milestones separated into individual projects and subtasks into 'notes' that existed within those projects. These notes would progress from a section titled `To do` to one titled `In Progress` as we began implementation. As we finalized the task, it would be promoted to the `Done` section of the Project.

When all of the notes reached this final section, the project would be deemed complete and work would begin on the next Project.

Below are screenshots of our Github Projects page and some notes from an individual project, titled `Compiling Hello World`:

📖 pattersongp / **plt-f18**

[✳ View Repository] [👁 Unwatch ▾ 6] [★ Star 0] [🍴 Fork 0]

<> Code   ⓘ Issues 0   ⑂ Pull requests 0   ▥ Projects 6   📖 Wiki   📊 Insights

🔍 is:open   [New Project]

▥ 6 Open   ✓ 0 Closed   Sort ▾

Project Presentation   6th Deliverable   ⋯
🕐 Updated 2 days ago
▬▬▬▬▬▬▬▬   A final project presentation

Project Report   5th Deliverable   ⋯
🕐 Updated 5 hours ago
▬▬▬▬▬▬▬▬   A final project report

Generating LLVM   4th Deliverable   ⋯
🕐 Updated 4 hours ago
▬▬▬▬▬▬▬▬   A compiler for it, written in OCaml; generating LLVM

Compiling Hello World [Due Nov 14]   3rd Deliverable   ⋯
🕐 Updated 2 days ago
▬▬▬▬▬▬▬▬   An intermediate milestone: compiling "Hello World."

**Reference Manual/Parser [DUE   2nd Deliverable   ⋯
October 15]**
🕐 Updated on Oct 16   A language reference manual defining it formally
▬▬▬▬▬▬▬▬

Proposal DUE Wed Sep 19   1st Deliverable   ⋯
🕐 Updated on Sep 20
▬▬▬▬▬▬▬▬   A proposal describing your language

Github Projects is surprisingly robust. It allowed us to link notes to individual git commits, combining task management with git's version history tools to allow us to more granularly and effectively track progress. It was an essential part of our workflow.

4.2 Style Guide

We followed the below style guide when writing code in FIRE. Our goal was primarily readability, though some of our test suite programs broke the these rules to properly vet features.

- Individual lines of code cannot exceed 80 characters.
- Employ `camelCase` when naming variables and functions.
- Blocks of non-trivial code should have terse but useful comments.

- Git commit messages should be substantive; e.g. "Fixed map bug in semant.ml" instead of "fixed bug".

Many of these style principles come from our experience coding in other languages, and proved to be useful in the context of coding in FIRE as well.

4.3 Project Timeline

Milestone	Date
Proposal	Sep 19th
Reference Manual	Oct 15th
Parser	
Semant	Nov 15th
Hello World	Nov 15th
Generate LLVM	
Project Report	Dec 14th
Project Presentation	Dec 19th

4.4 Team Roles

The below are the roles assigned to us at the inception of the project. As is natural with a project of this complexity, and with a team of this size, people came to take on other roles beyond their prescribed ambit as need required:

Role	Team Member	UNI
Manager	Graham Patterson	gpp2109
System Architect	Jason Konikow	jk4057
System Architect	Frank Spano	fas2154
Tester	Ayer Chan	oc2237
Language Guru	Christopher Thomas	cpt2132

4.5 Tooling

Languages: OCaml, C, Bash Version Control: Git Repository Management: Github Testing: Bash, FIRE Editors: Vim, VS Code, Sublime Text, Xcode Platforms: MacOS 10.14 Mohave, Ubuntu (via Docker, VirtualBox & Parallels Desktop) Documentation: Markdown, Macdown Editor Communication: Slack

4.6 Project Log

GITHUB COMMITS HERE — REPLACE DAY OF DEMO. Current Commits are from 12/14, pulled from master.

commit 6410255c4bdc50c0e249ecac7301a1f8b65cc3bc Merge: 411cf3e 52b8098 Author: Lord-Left sethosayher@gmail.com Date: Fri Dec 14 23:49:23 2018 -0500

```
Merge branch 'master' of https://github.com/pattersongp/plt-f18
```

commit 52b8098f4a1a081a57777d017ab7c81a6c4cc739 Merge: f6b1069 4c39595 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Fri Dec 14 11:57:58 2018 -0500

```
Merge pull request #84 from pattersongp/file-delim

Adds atoi
```

commit f6b10699f17e8564f3fcc457b90a41825b9a526c Author: Graham Patterson pattersongp.usmc@gmail.com Date: Fri Dec 14 11:22:36 2018 -0500

```
Deletes for loop tests (#85)
```

commit 4c39595d39dd603515a0d23fd354ab656d34db75 Author: Graham Patterson gpp2109@columbia.edu Date: Fri Dec 14 10:38:21 2018 -0500

```
Adds atoi
```

commit f97851a0a12fcdcfcabe156ca539d489d87e5a57 Merge: d13ec4e d6244fa Author: fspano118 fas2154@columbia.edu Date: Thu Dec 13 16:29:20 2018 -0500

```
Merge pull request #83 from pattersongp/strcmp

Strcmp
```

commit d6244fa23553b7b1284a696bd6957bdca60c5981 Merge: c121e53 d13ec4e Author:
Graham Patterson pattersongp.usmc@gmail.com Date: Thu Dec 13 16:28:41 2018 −0500

> Merge branch 'master' into strcmp

commit c121e53c19365c839c227a0233dc0426aecf015b Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Thu Dec 13 16:27:23 2018 −0500

> Adds strcmp

commit d13ec4e649e72ecca3901a8b4a90349c90598ab0 Author: Jason Konikow
jkon1513@gmail.com Date: Wed Dec 12 18:14:36 2018 −0500

> fixed unop tests (#82)

commit b9c5f66429903c2d594bf54ab396addf6b0d9449 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Wed Dec 12 17:49:38 2018 −0500

> Adds support for keys() functionality (#81)

commit 8fb94dbea2bb5ff0939818ac9121e8989f20e769 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Wed Dec 12 17:07:25 2018 −0500

> Adds support for keys() functionality

commit f173c17009379b79c246471bf62b5aa13b428981 Merge: 59a06db f404f9e Author:
fspano118 fas2154@columbia.edu Date: Wed Dec 12 14:23:24 2018 −0500

> Merge pull request #80 from pattersongp/2darr
>
> Adds support for multi-dimensional arrays and `split` functionality

commit f404f9ee28a7d8cea1d0f76cad03b2a1bd02e38b Merge: d722d50 59a06db Author:
Graham Patterson pattersongp.usmc@gmail.com Date: Wed Dec 12 14:18:59 2018 −0500

> Merge branch 'master' into 2darr

commit d722d504f3b156660f5e2094b4719e7635a25534 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Wed Dec 12 14:16:32 2018 −0500

Adds support for multi dimension arrays

commit 74969330f010abf346502ace6c09479cfd04339c Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Dec 12 13:56:51 2018 −0500

Adds split() function to fire interface

commit 6e7c6b034085617a71fdebb8ad8283962aa7b8d4 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Dec 12 13:34:39 2018 −0500

Adds C lib functionality for split() function

commit 3fbd6ec44e171051335707e46eed69a949798b1f Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Dec 12 11:08:48 2018 −0500

Adds Add/Get for IntArray() functions

commit 59a06db13326332cf77ddb840f57207bcff41e35 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Dec 11 09:28:31 2018 −0500

Adds len() function for array length (#79)

commit e63622877a5715cbe134d47a28f0e4807940d255 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Dec 11 08:45:44 2018 −0500

Fixes bug for issue 72 (#78)

commit e77fb5a8944fa9162638e389894eb6bd1c88fc99 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Dec 11 08:07:48 2018 −0500

```
Testfixes (#77)

* Fixes fail-assign2 testcase

* Fixes fail-binop5

* Fixes fail-formals3

* Fixes fail-identifier3 test

* Fixes fail-if{1,4,5}

* Fixes fail-func3

* Fixes fail-if6
```

commit bccbfdd2ace92e8234a57c2538eda15461d025d4 Merge: fbf58ff 1197ed2 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Dec 11 06:54:37 2018 −0500

```
Merge pull request #75 from pattersongp/filter

Filter implemented in arrlib.c and codegen.ml
```

commit fbf58ff94420dd8639a83af56c64f3bb74296f83 Author: Christopher Thomas sethosayher@gmail.com Date: Tue Dec 11 06:53:16 2018 −0500

```
Valid Test Cases added to Test Suite (#76)

* started testall.sh, removed outdated array tests

* Starts refactoring test suite

* partially modified testall.sh for FIRE, need to add lib checks

* created checklist for test modifications

* wrote 3 new assignment tests and ref output files

* update testall.sh to check for required libs

* testall.sh succesfully runs tests

* added assign and binop fail cases to tests
```

* add fail cases for relational binops in tests

* added clean function with -c option to testall

* corrected typo in expected error mssgs

* add fail cases tests for identifiers

* add fail case tests for formals into test suite

* added fail case tests for unop into test suite

* added fail test cases for if stmt, for stmt, and while stmt. created test.md to track unchecked errors

* added fail cases for func, no main, print, and sprint into test suite

* corrected typos, completed fail cases

* fixed typo

* removed old test files

* Made slight change to testall.sh to pipe contents of test script provided as argument into the fire.native executable. This seems to fix an earlier issue where the fire.native executable would hang when the testall.sh script would be run with a fire test file as an argument.

* Editing testall to reflect FIRE compilation process.

* Working fix for valid test cases for testall.sh script. More test cases with outputs needed.

* Fixed typos caused by merge conflict.

* Added valid tests for addition.

* Added valid tests for various arithmetic operations.

* Added Graham's recursive fib program to test suite; fixed base case bug.

* Added first batch of valid programs that utilize functions. Checks for scoping issues, etc.

```
* Added next batch of valid programs testing functions. Tested nested functions.

* Removed debug statements from testall.sh.

* Added final batch of simple function tests.

* Wrote tests using recursive gcd algorithim. Based off tests by Graham Patterson.

* Added first batch of array tests.

* Wrote some additional tests for arrays that retrieve values from one type of array
from another type of array; e.g. from str-int to int-str.

* Wrote a test that tests nested retrieval of values.

* Added output file for Graham's general array test (test-arr-all.fire)
```

commit 411cf3e1283ef65d4eecef6d8035aa17c16c48c3 Author: Lord-Left
sethosayher@gmail.com Date: Mon Dec 10 21:59:15 2018 -0500

```
Wrote Scaffolding of Final Report.
```

commit 1197ed2eec7557c39e98957178e39f0b974c5fa4 Author: frank spano
fspano@tripcents.co Date: Mon Dec 10 21:34:40 2018 -0500

```
Filter implemented in arrlib.c and codegen.ml
```

commit 6255565448e3f95f2242da760247dcb3891686bc Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Fri Dec 7 15:42:30 2018 -0500

```
Handles all compiler warnings for clean compilation (#74)
```

commit a8ddebfffdc334ef4484be0b9c7db739ad337153 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Dec 4 16:51:02 2018 -0500

```
Adds support for map() [tested] (#71)
```

commit 74a610b5e06befb4514a8a74d35ba4662fdacd63 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Dec 4 08:12:06 2018 -0500

```
Adds top level Makefile for automatic testing (#70)

At the top directory, run `make all` which will run the Makefile in the
`src/`, then run the `testall.sh` script.
```

commit 289e6380c16f9a043b2db796eb4fe1ce7b68a5cf Author: Ayer Chan
oc2237@columbia.edu Date: Tue Dec 4 07:18:18 2018 −0500

```
Arrlib.c (#66)

* get rid of strcmp, add type flg,keep track of  array in array size

* get rid of strcmp, add type flag, keep track of array in array size

* solve merge conflit

* comment ifdef

* Adds function calls to InitArray() at array vdecl

* Adds code generation for array assign [not finished]

* Adds specialized implementation of the array library

* Adds support for Arrays using int and string

Currently only support:
        Array[int, int]
        Array[int, string]
        Array[string, string]
        Array[string, int]

* Adds changes per PR feedback
```

commit 986b06e24dbd3de0b9fad5810945f792b0dad4e8 Author: Graham Patterson
gpp2109@columbia.edu Date: Sun Dec 2 08:03:31 2018 −0500

```
Update gitignore
```

commit 5124fee1250055de0d9d44ac8a86819708360e20 Author: Graham Patterson
gpp2109@columbia.edu Date: Sun Dec 2 08:02:25 2018 −0500

```
remove .o
```

commit c70025d9d2adf1c85399c772dadf258b9e2a4c45 Merge: 369507f a658f55 Author: Graham Patterson gpp2109@columbia.edu Date: Sun Dec 2 08:01:06 2018 −0500

```
Merge branch 'master' of github.com:pattersongp/plt-f18
```

commit 369507f7696bb1fa684e205cc79f8f323cc6e6b5 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Sat Dec 1 16:43:48 2018 −0500

```
regular expression fix and example (#68)

* Adds debug message

* adds code example for lexer
```

commit a658f55a55810cb8c2d1da40a43354e2817aa8cc Author: Graham Patterson pattersongp.usmc@gmail.com Date: Sat Dec 1 16:43:48 2018 −0500

```
regular expression fix and example (#68)

* Adds debug message

* adds code example for lexer
```

commit d1dffa71b619cb2414684ce4483833d07329884a Author: Christopher Thomas sethosayher@gmail.com Date: Sat Dec 1 16:43:11 2018 −0500

```
Partial LRM Update (#67)

* Fixed minor typos in opening paragraphs, added mention of  keyword re: array inital
izing.

* Added table of permissible types for Array Keys & Values.

* Added Other Code Requirements Section.

* Added documentation for string concat
```

commit 8c7ccd8e51eb5ce2effec8d77361e675d52bfa98 Author: Jason Konikow jkon1513@gmail.com Date: Sat Dec 1 15:04:37 2018 −0500

```
Testall (#65)

* started testall.sh, removed outdated array tests

* Starts refactoring test suite

* partially modified testall.sh for FIRE, need to add lib checks

* created checklist for test modifications

* wrote 3 new assignment tests and ref output files

* update testall.sh to check for required libs

* testall.sh succesfully runs tests

* added assign and binop fail cases to tests

* add fail cases for relational binops in tests

* added clean function with -c option to testall

* corrected typo in expected error mssgs

*  add fail cases tests for identifiers

* add fail case tests for formals into test suite

* added fail case tests for unop into test suite

*  added fail test cases for if stmt, for stmt, and while stmt. created test.md to tr
ack unchecked errors

* added fail cases for func, no main, print, and sprint into test suite

* corrected typos, completed fail cases

* fixed typo

* removed old test files
```

commit 9362fdb540b53f3f7e33774b40ae5c5ec1433ad4 Author: fspano118
fas2154@columbia.edu Date: Sat Dec 1 14:39:29 2018 -0500

> removed initarray from all places except Sast (#64)

commit 7bbc9225716ed090eea6b44a88efe636bc7312dc Author: fspano118 fas2154@columbia.edu Date: Sat Dec 1 14:26:50 2018 −0500

> Semant array (#63)
>
> * fixes to printing in sast, arrays in semant
>
> * non-nested arrays semants working

commit 87df010c04019d56278e34a70bb5e7f6107aa361 Merge: d3ffad4 b9ee5f8 Author: fspano118 fas2154@columbia.edu Date: Sat Dec 1 14:06:48 2018 −0500

> Merge pull request #62 from pattersongp/utilfix
>
> Fixes bug with null terminated string in strcat

commit b9ee5f8a69af6728f3a74da60e31a10dfaa45bc2 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Sat Dec 1 14:03:13 2018 −0500

> Fixes bug with null terminated string in strcat

commit d3ffad401c132a79a883aeb3a6ea88c21fb5bf22 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Sat Dec 1 13:53:57 2018 −0500

> Adds list reverses to blocks in conditionals [bug fix] (#61)

commit 6a75892f0c1688fc1ad3bcfc5dff9804bea4480e Author: Graham Patterson pattersongp.usmc@gmail.com Date: Fri Nov 30 14:24:03 2018 −0500

> Implements regex grab in fire (#60)
>
> Syntax for grabbing a slice of a string is:
> ```
>         str s = "token";
>         regx r = "ok";
>         str ret = s.grab(r);
>         /* ret is 'ok' */
> ```

commit d9a24fe6100595c6514e24a0ee82c4347e6c755d Merge: 80867cd 3f79f1f Author: Christopher Thomas sethosayher@gmail.com Date: Thu Nov 29 21:44:52 2018 −0500

> Merge pull request #59 from pattersongp/regex-grab
>
> Adds a library call to parse the matches string

commit 3f79f1f2d82332d6c1af45f1401742dc7b570827 Author: Graham Patterson gpp2109@columbia.edu Date: Thu Nov 29 21:42:17 2018 −0500

> Adds a library call to parse the matches string

commit 80867cd984ca85d9c9fb9643c7a36f9acd0c0e0c Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Nov 29 19:41:59 2018 −0500

> Adds support for writing files. (#58)

commit 9a94d131542c13310fde0f49177346c4c26b0ebe Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Nov 29 12:47:59 2018 −0500

> Passes all tests cases in tests-gp/ except for Arrays (#57)

commit e1c93188cd98e221ed6016f213f32cf79f9b2268 Merge: fec36b0 c03f375 Author: fspano118 fas2154@columbia.edu Date: Thu Nov 29 11:01:04 2018 −0500

> Merge pull request #56 from pattersongp/quick-fix
>
> Fixes some array typ issues in semantic and sast/ast.

commit c03f375b1b208cb42f183a0eb831f050ce0e9209 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Nov 29 10:53:55 2018 −0500

> Fixes some array typ issues in semantic and sast/ast.

commit fec36b0f904dbe9642f009787911323524af628d Merge: e3185d7 27b0517 Author: Ayer Chan oc2237@columbia.edu Date: Thu Nov 29 10:06:29 2018 −0500

> Merge pull request #55 from pattersongp/arraylib-ayer
>
> Array Library additions

commit 27b0517a038092a33d903e475b80b19ab693929b Merge: a794ba4 e3185d7 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Nov 29 09:47:28 2018 −0500

> Merge branch 'master' into arraylib-ayer

commit a794ba47c7c55539aa9a586cc183b2884f4df087 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Thu Nov 29 09:45:08 2018 −0500

> Cleans up repository

commit ea9977384068746a9597d8f937f3dfaf32baf1ac Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Wed Nov 28 13:40:22 2018 −0500

> Adds util

commit e3185d7f69d7835877265e38d22e88bfdbf3c395 Merge: 9ba50e7 239ce02 Author:
fspano118 fas2154@columbia.edu Date: Thu Nov 29 09:35:36 2018 −0500

> Merge pull request #54 from pattersongp/integrate-semantic-1
>
> Integrate semantic 1

commit 239ce027269a4431296a8350af45883f3e9aebb4 Author: Graham Patterson
gpp2109@columbia.edu Date: Wed Nov 28 21:54:08 2018 −0500

> fix check assign

commit e7b4ba39a0ed828e03f0bde05feaac43080b0736 Author: frank spano
fspano@tripcents.co Date: Wed Nov 28 21:34:21 2018 −0500

> check assign

commit 51867e319017e8534f9a3848941ea51cd021f851 Author: frank spano
fspano@tripcents.co Date: Wed Nov 28 21:08:12 2018 −0500

> fixed not finding formals

commit 87ce3e7635c0bfcf231056aa72757aa86c07e1c7 Author: frank spano
fspano@tripcents.co Date: Wed Nov 28 20:57:30 2018 −0500

> strcat

commit 6e2b758b2d3c3676acf21340560572d7b4bcad18 Author: Graham Patterson

gpp2109@columbia.edu Date: Wed Nov 28 17:14:28 2018 −0500

```
Furthing semantic integration

Currently supporting most features of the language.
Still missing:
        + regex compare
        + strcat
        + sprint


There is also a bug with the built in functions and arguments being
decalred in the semantic instead of at runtime.
```

commit 10e57b59675fde6f5a66e1cfcbc5dcca94fcf240 Merge: 555df97 a777730 Author: Graham Patterson gpp2109@columbia.edu Date: Wed Nov 28 16:26:24 2018 −0500

```
Merge branch 'integrate-semant' into integrate-semant-f
```

commit 555df97f1f8610f133531cd45fbedc8a88720f3c Author: frank spano fspano@tripcents.co Date: Wed Nov 28 14:52:17 2018 −0500

```
compiling semant with sast Vdecl change
```

commit a7777309f7782c24f569ba99b23511095b0bf116 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Nov 28 14:37:28 2018 −0500

```
Fixes codegen for semant, untested
```

commit 457ee29a12e009271cd4dbb3b7f639c593f9c8ed Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Nov 28 13:40:22 2018 −0500

```
Adds util
```

commit fb0f47d5f7021ce01af6033e747c33cfa34da361 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Nov 28 13:01:46 2018 −0500

```
Starts integration of semant.ml into codebase
```

commit 9ba50e7206db42a26b0c413afdbf9a3a2683acd7 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Nov 27 21:49:38 2018 −0500

```
Frank tmp2 (#53)

* Added -s flag

* Added detection of void functions and duplicate name functions to semant.ml

* Added some comments re: void & dup detection in semant.ml

* Added beginnings of symbol table function.

* Revised declaration symbol table function; have to find way to express expr in orde
r for function to work in semant.ml.

* Revised semant.ml and built-in function code.

* change arraydecl to array start semant changes

* changed assign and array assign to stmnt in AST

* added vdecl to AST under stmnt

*   whelo

* adds fix that requires Block body

* added call scafolding to rec expr

* sast v1

* added sast

* fixed compilation errors in sast, corrected several pretty printing errors in sast

* modifed fire.ml to reflect new semant.check params

* modified semant after refactors, compiles up to check_function

* fixed incorrect param in symbols, semant compiling throught start of rec expr

* corrected ast.op token names in semant, compiles through binop/unop

* semant compiling through function call semantic check in expr

* finished rec expr and check_bool, compiles up to check_stmt
```

* added SStringLit to pretty printer

* replaced arg in check_binds

* partial merge

* vdecl

* compiling except for 1 line

* semant compiling

* changed sast and fire to build sast

* semant compiles pre check_stmnt refactor

* fucking around

* attempts to pass enviroment through check_stmt calls

* logically sound fixing syntax errors

* semant compiled, needs testing

commit 0910015f9f4af7b573974d0c6f53cbc3e651c72d Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Nov 27 18:21:27 2018 −0500

Adds function calls and modifies array lib.

Compiles programs but with a type error:
llc: arr.test.ll:33:19: error: '@add' defined with type 'i32 (i8*, i32*, i32*)*

commit 485f6e4aef9aceddb1f3d52068700ee234a2735c Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Nov 27 17:36:02 2018 −0500

Removes warnings

commit d58af69da0ade4579b67211d1f7324188815df2f Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Nov 27 17:16:52 2018 −0500

Adds some forgotten changes

commit e0662560196335aae4b6289b7c7028b151dabfe9 Author: Oi I Chan oiichan@dyn−

129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 16:33:28 2018 −0500

> arrlib.c compiles

commit 3f2dd86848568ae9607f711f7c81652757bb2a7d Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Nov 27 16:10:06 2018 −0500

> Adds init(typ, typ) for array library

commit e7810b17474d9485a3a883ea6ebe2be752bcf1c3 Author: Oi I Chan oiichan@dyn-
129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 15:58:31 2018 −0500

> change initarray

commit bee421244acd237a17c8c9c7efd2d686a7a40520 Author: Oi I Chan oiichan@dyn-
129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 15:54:05 2018 −0500

> update arrlib.c initArray and add

commit b7187aaf09c87a05dbbfe95ac3d1d39379902529 Merge: 8402df4 7449f48 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Nov 27 15:13:06 2018 −0500

> Merge branch 'arraylib-ayer' of github.com:pattersongp/plt-f18 into arraylib-ayer

commit 8402df4de9a6293029d62317c3552f37f8670e06 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Nov 27 15:11:00 2018 −0500

> Adds init() and compiles the array init in codegen

commit 7449f48fad7e4b3c9fc687b0eb379a2986a78d9d Author: Oi I Chan oiichan@dyn-
129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 14:54:34 2018 −0500

> update arrlib.c changing set incorporating to add

commit 50636c6607eb5d266c6bb90c3338bb3f47b48255 Author: Oi I Chan oiichan@dyn-
129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 14:26:03 2018 −0500

> update codegen

commit 26ce632dbeb1f76d413ab5d1ccd2ba7c107a6be8 Merge: 1fbc485 6c7241b Author: Oi

I Chan oiichan@dyn-129-236-227-4.dyn.columbia.edu Date: Tue Nov 27 14:08:20 2018 -0500

> Merge branch 'arraylib-ayer' of https://github.com/pattersongp/plt-f18 into arraylib-ayer
>
> get updates

commit 2405568d4d8bd0ad9eea6a6030ad123fff3a3fc8 Merge: f3fd4af 6dc800c Author: Christopher Thomas sethosayher@gmail.com Date: Sun Nov 25 16:06:32 2018 -0500

> Merge pull request #52 from pattersongp/strcat
>
> Adds string concatenation operator to codegen

commit 1fbc485322d1db8dc1b9a612e539198cdf9cde91 Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sun Nov 25 12:02:33 2018 -0500

> delete some files

commit 65c7d9046bfc41fc6be8c88927faee9d33e8f021 Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sun Nov 25 11:58:08 2018 -0500

> add get() and not to do remove element

commit e87c19f35434f8ffbad8476920f067eb7778d83e Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sun Nov 25 01:12:01 2018 -0500

> confirm using c instead of c++ and add get method

commit 5109dd98d35c2833b5d28e40d59ff869e603012b Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sat Nov 24 14:37:52 2018 -0500

> no error message, create a c++ base for array

commit 75f1ad4a418a0336bb3fb09ae7c27259074b0748 Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Fri Nov 23 15:45:25 2018 -0500

> trying c++

commit cf4fa62c1d12eaf12a5731d9c7f42b321b91e0fc Author: Oi I Chan oiichan@dyn-129-236-227-169.dyn.columbia.edu Date: Tue Nov 20 20:27:14 2018 -0500

> array initialization and adding to array works

commit 30154fdd1298c92896cecd9a47b2a8af0951c154 Author: Oi I Chan oiichan@dyn-129-236-227-169.dyn.columbia.edu Date: Tue Nov 20 15:57:17 2018 -0500

> create cpp array template that do absolutely nothing and create c array template as w
> ell

commit 6c7241b9d791779028cb351122a0ad47912fe547 Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sun Nov 25 11:58:08 2018 -0500

> add get() and not to do remove element

commit 12dd8b153e05ab016e8d90093ecabcaf0288adfa Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sun Nov 25 01:12:01 2018 -0500

> confirm using c instead of c++ and add get method

commit c5e79b6f0ee7c95742c95a29cc35278077c10a80 Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Sat Nov 24 14:37:52 2018 -0500

> no error message, create a c++ base for array

commit 32aa9ac81bb7c6b8d08e2119571ccc6e9660d78e Author: Oi I Chan oiichan@0is-MacBook-Pro.local Date: Fri Nov 23 15:45:25 2018 -0500

> trying c++

commit 6dc800c39f9927101649eaea9f48ce34e61b66b4 Author: Graham Patterson gpp2109@columbia.edu Date: Tue Nov 20 20:51:23 2018 -0500

```
Adds string concatenation operator to codegen


Sample Program:
        func int main = () => {
                str s1 = "hello ";
                str s2 = "world!";
                str s3 = s1 ^ s2;

                /* Single strcat */
                sprint(s3);

                /* grouped strcat */
                sprint((s1 ^ s2) ^ s3);

                return 0;
        }
```

commit aba6a0acc2bab902f1b4a94115f72303cada6ba3 Author: Oi I Chan oiichan@dyn-129-236-227-169.dyn.columbia.edu Date: Tue Nov 20 20:27:14 2018 -0500

```
array initialization and adding to array works
```

commit f3fd4af3a9015b4360e629d41514a08872cc803d Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Nov 20 18:27:34 2018 -0500

```
Adds file support (#51)

* Starts working on the Open() and file interaction.

Supports:
func void main = () => {
        file[w] f;
        f = open("helloWorld.txt", ",");
}

* Adds working read() function in Fire and readFire() in lib.

* Adds support for reading text files.

Features:
        read() : Reads in a chunk of delimited by delim
        open() : Opens filename and sets delim
```

commit c41bd0cd27c9d6b829df3427d196db0bee161ad5 Author: Oi I Chan oiichan@dyn-129-236-227-169.dyn.columbia.edu Date: Tue Nov 20 15:57:17 2018 -0500

> create cpp array template that do absolutely nothing and create c array template as well

commit 24fb3d0d0466ea6b4ff94a30e1562f52f784db79 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Mon Nov 19 17:35:44 2018 -0500

> Adds support for matching on regular expressions and strings (#50)

commit 3e5399b343d488bd587f40fbb3fcb27c30c42711 Merge: a61604a 656a699 Author: Christopher Thomas sethosayher@gmail.com Date: Mon Nov 19 11:13:02 2018 -0500

> Merge pull request #49 from pattersongp/codegen-regx
>
> Fixes bug with vdecls

commit 656a699e0340984c8453b1b0088c5c177012b2d0 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Mon Nov 19 11:09:44 2018 -0500

> Fixes bug with vdecls

commit a61604aea3fa58a9ce40b308107c6d3f7549e0e2 Author: Ayer Chan oc2237@columbia.edu Date: Sun Nov 18 17:53:05 2018 -0500

> Lrmupdate (#48)
>
> * starts updating WRT john's comments
>
> * Made updates with John's comment
>
> * Add sprint()
>
> * Update LRM with John's comment

commit c6910ed9ed77c22b83cc35bdd9e2b565e7151ac2 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Sun Nov 18 08:32:02 2018 -0500

> Adds support for printing and storing string (#47)

commit 322707490a1ca3efcc86e5bfd5eae6ee220bb333 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Sat Nov 17 09:06:08 2018 -0500

```
Refac codegen (#46)

* progress on codegen-- currently compiling

* Modifies codegen.ml to remove vdecls and support mixed decls
```

commit 33e0c20b4058c182c2898e8f9ce475011e718358 Author: fspano118
fas2154@columbia.edu Date: Thu Nov 15 16:33:19 2018 -0500

```
removes vdecls and expr option (#45)

Removed vdecl and expr from parser and ast. We decided it did not make
sense to have global variable declarations, so now vdecls are part of
stmt. Also, declarations with/without assignments will be distinguished
at the parser level. We will not be assigning default values to
variables.
```

commit 24aef33f3958d41dff634042d0abbf5dd10e496c Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Wed Nov 14 14:17:54 2018 -0500

```
Codegen111018 (#44)

* Adds to ignore (#42)

* Parses user functions without body

* Adds batteries package to Dockerfile and linking in Makfile

* Parsing return statements

* Compiles helloworld.fire

* Handles recursion, while, for, fib, gcd
```

commit 72ebec7c07dddd20668e811eb3db41afbb873508 Merge: 37096e3 38124e5 Author:
Christopher Thomas sethosayher@gmail.com Date: Tue Nov 13 14:25:40 2018 -0500

> Merge pull request #43 from pattersongp/batteries-included
>
> Adds batteries package to Dockerfile and linking in Makfile

commit 37096e3b6227433d97c643162406c9d79b6b6fd4 Author: Ayer Chan
oc2237@columbia.edu Date: Mon Nov 12 17:45:57 2018 −0500

> Update Test files to reflect current changes of our language (#40)
>
> * add file and print test
>
> * change function to without semis
>
> * change all func to main and add void as nothing to return
>
> * delete _build
>
> * puts arrlib.c back

commit 38124e58d0e57ff75d5e19e0455b743d815c9d46 Author: Graham Patterson
gpp2109@columbia.edu Date: Sat Nov 10 15:20:16 2018 −0500

> Adds batteries package to Dockerfile and linking in Makfile

commit df62bc7cb375dcd8927a6e172f56fc02afb35d53 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Sat Nov 10 12:14:20 2018 −0500

> Adds to ignore (#42)

commit b8f60dffd7585446d21d8843924c5da68f28e2ab Merge: fd4dad0 2c832ca Author:
Christopher Thomas sethosayher@gmail.com Date: Thu Nov 8 17:53:29 2018 −0500

> Merge pull request #41 from pattersongp/pattersongp-patch-1
>
> Update .gitignore

commit 2c832ca76343773a04700db7e9b43d8da1c7d97c Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Thu Nov 8 17:51:45 2018 −0500

> Update .gitignore

commit fd4dad00d6865126eeb1b6eef2d86d5cba596c9d Merge: 5229286 d90ea0e Author: fspano118 fas2154@columbia.edu Date: Thu Nov 8 15:30:28 2018 −0500

```
Merge pull request #39 from pattersongp/revert-38-semi

Revert "added semi colon to func assignment in parser"
```

commit d90ea0e6c6ba882787f284d1c3b656ac9cf184a9 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Nov 8 12:07:54 2018 −0500

```
Revert "added semi colon to func assignment in parser"
```

commit 52292864658602340cf5683e4eb6bf76bd9265dd Merge: 8a8e63b 14d2df9 Author: Christopher Thomas sethosayher@gmail.com Date: Wed Nov 7 17:18:25 2018 −0500

```
Merge pull request #38 from pattersongp/semi

added semi colon to func assignment in parser
```

commit 14d2df9502eef2f270a9f32c5e3524f39a8caa0b Author: Jkon1513 jkon1513@gmail.com Date: Wed Nov 7 17:15:14 2018 −0500

```
added semi colon to func assignment in parser
```

commit 8a8e63b969eeeff45b74c7945ffe7d7c8f87d11e Merge: 3f73c97 8947b5d Author: Christopher Thomas sethosayher@gmail.com Date: Wed Nov 7 17:10:47 2018 −0500

```
Merge pull request #37 from pattersongp/ArrayFix

Changed ArrayDecl to Array
```

commit 8947b5dbb77a691f279e84f23971d30e005c273c Author: frank spano fspano@tripcents.co Date: Wed Nov 7 14:14:23 2018 −0500

```
Changed ArrayDecl to Array
```

commit 3f73c97d9756beeb09ad4b5eaeeead5b11ddc5fd Merge: 610c0fd 099961f Author: Christopher Thomas sethosayher@gmail.com Date: Sat Nov 3 01:45:01 2018 −0400

```
Merge pull request #36 from pattersongp/sprint1101

Adds code generation for bool, int, initialization
```

commit 099961f15066b09e6fe1e1bd4760b4596b48fd82 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Thu Nov 1 18:36:10 2018 −0400

```
Adds code generation for bool, int, initialization
```

commit 610c0fd66b788ae031c4cbbf74413a35ee5b36e1 Merge: 5de2dad f4037ea Author:
Christopher Thomas sethosayher@gmail.com Date: Thu Nov 1 13:45:44 2018 −0400

```
Merge pull request #35 from pattersongp/codegen

Adds basic code generation
```

commit 5de2dad64ed2c41fc0c8d29d0ae359f5cb4ff67c Merge: 11f1ba4 1e87666 Author:
Christopher Thomas sethosayher@gmail.com Date: Thu Nov 1 13:45:30 2018 −0400

```
Merge pull request #34 from pattersongp/ast-finish

Finishes AST
```

commit f4037ea9be5401bb8d21370201ee5333f2b7abdd Author: Graham Patterson
gpp2109@columbia.edu Date: Thu Nov 1 09:14:15 2018 −0400

```
Adds basic code generation
```

commit 1e8766652413d96c9ac46d42faa19cfa5bc893e5 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 30 17:33:38 2018 −0400

```
Finishes AST
```

commit 11f1ba4ee305569850807a2711c9ae88dc4d6bbd Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 30 02:01:13 2018 −0400

```
Ast work2 (#33)

* adds function calls

* Adds function calls and finishes AST, minus array decl

* modify makefile and remove warnings
```

commit edb2826d5cd9b9f152f3f019adce11d0004fb6c6 Author: Ayer Chan
oc2237@columbia.edu Date: Mon Oct 29 15:41:21 2018 -0400

```
Test (#31)

* Create test.md

* add local files to remote

* delete testfiles

* Create test.md

* update testfiles according to vdels

* addtest for while if map filter regex

* update filter and regx
```

commit 72c20ca93b71684d2495cf8f1295d52c2a732e31 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Mon Oct 29 15:05:01 2018 -0400

```
Ast work (#30)

* Adds for, conditionals

* finishes stmt
```

commit 43a724d96aaa63bd81027fc1ae223bf90446ffc6 Author: fspano118
fas2154@columbia.edu Date: Sun Oct 28 12:56:01 2018 -0400

```
ast properly parsing function declarations (#28)
```

commit bd2bc808bcfa7e8eedaeefd16723db39cf20713c Merge: 7eb2c75 877f84e Author:

fspano118 fas2154@columbia.edu Date: Fri Oct 26 15:05:33 2018 −0400

> Merge pull request #27 from pattersongp/sprint1025
>
> Adds more types to the parser

commit 877f84e16a1018c0e38b06b1cccac5433383b16a Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Oct 25 17:40:12 2018 −0400

> Adds retrieve and adds underscore, numbers to ID name

commit eabe7c513d2d5cf03841e5f9b404cedf634446a9 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Oct 25 17:32:22 2018 −0400

> Adds unop

commit 3477fd61cc43ae92bade33cd8343eb9129df6f05 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Thu Oct 25 16:58:38 2018 −0400

> Adds binop to parser

commit 7eb2c75e6dcc062fddee1412c7562aec6e249cd8 Merge: de01b90 3271e39 Author: Christopher Thomas sethosayher@gmail.com Date: Thu Oct 25 14:47:13 2018 −0400

> Merge pull request #26 from pattersongp/pretty-print1
>
> Starting pretty printing and parse support

commit 3271e39f9c72a0fa6cd56c3400e2ed837d75e34a Author: Graham Patterson pattersongp.usmc@gmail.com Date: Wed Oct 24 12:04:17 2018 −0400

> Starting pretty printing and parse support
>
> Currently parses declarations and initializations of bool, str, int, and prints out the program correctly.

commit de01b904d1aec3edb86738e9d889c6edf50e24b6 Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Oct 23 16:43:25 2018 −0400

> Compiles with warnings, but is a place to start dev. (#25)

commit d97ac942cb0803106a8e7202756cd3f2807c8106 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 23 11:20:50 2018 −0400

```
move makefile
```

commit d14f3e9cae2b00587b86fc1a165bd9e1cbb313c2 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 23 10:51:23 2018 −0400

```
Adds skeleton of a Makefile (#23)
```

commit 25a9889976154c0a368bc6d019303fcfad3b0d11 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 23 10:50:55 2018 −0400

```
Add docker (#22)

* Adds Dockerfile and documentation

* Adds Dockerfile and documentation

* mend

* Update README.md
```

commit fbc2ecfc57c8b99a33e79d1fdb6287f803f603db Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Mon Oct 22 20:09:04 2018 −0400

```
adds skeleton for ast and top level entry point to the compiler (#24)
```

commit b836ed50eec0273e631e0fd506e9c513f0878ddf Author: Jason Konikow
jkon1513@gmail.com Date: Tue Oct 16 08:40:11 2018 −0400

```
LRM update (#21)

* created LRM stub

* created LRM structure

* completed lexical analysis section, syntax notation and identifiers

* added expressions section

* finished expressions, started statements, added intro text, added semantics
```

* fixed some stylistic inconsitencies

* fix typo

* Fix typos

* Modifies LRM -- gp

* Added Code Sample to ToC

* Added Code Sample Section

Also added 'filter' to reserved keywords. TO DO: add filter and map to Expressions, void to Semantics

* Created scaffolding for boolean type

Should bool utilize true or false, or 1 or 0?

* Added map and filter in expressions

* Update Sample Code + null+ update TableContent

* Void + small stylistic tweaks

Added void to reserved keywords.

* Update LRM.md

* Update LRM.md

* Updated function semantics and code sample

* Added void to expressions.

May be better suited for semantics/declarations?

* Changed location of void, made stylistic changes

* stabdardized FIRE naming, adjusted code sample to conform to LRM

* modified regex declaration in code example

commit 4d081ecc4a85563598db5cc2e73e79497dc10bab Author: Christopher Thomas

sethosayher@gmail.com Date: Mon Oct 15 20:54:40 2018 −0400

```
LRM merge (#20)

* created LRM stub

* created LRM structure

* completed lexical analysis section, syntax notation and identifiers

* added expressions section

* finished expressions, started statements, added intro text, added semantics

* fixed some stylistic inconsitencies

* fix typo

* Fix typos

* Modifies LRM -- gp

* Added Code Sample to ToC

* Added Code Sample Section

Also added 'filter' to reserved keywords. TO DO: add filter and map to Expressions, v
oid to Semantics

* Created scaffolding for boolean type

Should bool utilize true or false, or 1 or 0?

* Added map and filter in expressions

* Update Sample Code + null+ update TableContent

* Void + small stylistic tweaks

Added void to reserved keywords.

* Update LRM.md

* Update LRM.md
```

```
* Updated function semantics and code sample

* Added void to expressions.

May be better suited for semantics/declarations?

* Changed location of void, made stylistic changes
```

commit 74092a5feb1b3aef659ef71514d1766b612ea9af Author: Ayer Chan
oc2237@columbia.edu Date: Mon Oct 15 18:48:43 2018 −0400

```
Create 10152018 (#19)
```

commit e620f5bb1358e93bc4bf6ed39f282ad8c2e5530d Author: Jason Konikow
jkon1513@gmail.com Date: Mon Oct 15 17:56:36 2018 −0400

```
updating func md (#17)

* Update func.md

* Update func.md
```

commit 05dedf81083c03f5e527604fa978e8c625b0cdbe Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Mon Oct 15 17:49:17 2018 −0400

```
parser (#18)

* starts parser

* added fdecls, actuals, implementing array

* reducing errors

* fixes no reductions

* compiles

* Finishes parser
```

commit b49ab4904a452b7a0832ad8eb9967dca3c5c097a Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Mon Oct 15 08:56:54 2018 −0400

```
Update func.md (#16)
```

commit 4779a59ffb7830b53cfe8985ab3ee50ef6c2f4d4 Author: Ayer Chan
oc2237@columbia.edu Date: Sun Oct 14 20:52:38 2018 −0400

```
Ochan4 patch 2 (#14)

* First update to array

* Update array.md

...

* Update array.md

* Update Function add delete element
```

commit a0a1dcadde722289ae92c1947a01555391c8d24a Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Fri Oct 12 13:40:04 2018 −0400

```
Starts scanner (#15)
```

commit 06443835d8013d298bd5d6b2b1691117dbf76c1e Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 9 14:24:48 2018 −0400

```
move files
```

commit d1f3b700dd041ae02293077722240e5e4cfbf6fd Author: Jason Konikow
jkon1513@gmail.com Date: Tue Oct 9 14:23:17 2018 −0400

```
func semantic  (#12)

* completed first draft of semantic

* minor formatting changes
```

commit 4a15850f955a08bf88ac889a8df1d9f572c55dd3 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 9 14:21:58 2018 −0400

```
Regex semantics (#13)

* Add skeleton to the regex semantic

* add order of precedence

* finish
```

commit ceb4f8baa7d5bde3958fc792397bee9f168910d6 Author: Christopher Thomas sethosayher@gmail.com Date: Tue Oct 9 14:13:29 2018 −0400

```
String semantics (#11)

* Update string.md ...

* Updated example and a section on characters

* Updated default parameters
```

commit cad1b8433839546ce8533915a752ae41b5353a65 Merge: 17d4d2a 6b49e37 Author: Christopher Thomas sethosayher@gmail.com Date: Mon Oct 8 23:50:55 2018 −0400

```
Merge pull request #10 from pattersongp/file-semantics

Merging File Semantics update by Frank after 2 reviews.
```

commit 6b49e37c6d4d5e6283c7741956805ce63a343fe0 Author: frank spano fspano@tripcents.co Date: Mon Oct 8 09:41:39 2018 −0400

```
file semantics
```

commit 17d4d2a47954c8e1cd930f0d9e1c98b84ac48fdc Author: Graham Patterson pattersongp.usmc@gmail.com Date: Tue Oct 2 15:22:33 2018 −0700

```
Adds skeleton of parser and scanner (#7)
```

commit 51f3e9eb41d77fe32d7b4c7f00ec078781ea3a74 Merge: dff6663 faf7b10 Author: Christopher Thomas sethosayher@gmail.com Date: Tue Oct 2 18:22:06 2018 −0400

> Merge pull request #9 from pattersongp/semantics
>
> Comprehensive and accurate meeting notes for our Oct 2nd PLT Meeting.

commit faf7b10b67ef0ee10be19cb4d1dd3ef2ba779b39 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 2 16:31:10 2018 −0400

> Add the meeting notes from 10022018

commit 479ce1abbef3652c24d204a246d279667fb1de6f Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Oct 2 16:30:50 2018 −0400

> Add the stubbed out markedown files for semantics.

commit dff666359b78573f3883bc27e20fe6c1b0b67f8b Author: Ayer Chan
oc2237@columbia.edu Date: Tue Oct 2 16:16:18 2018 −0400

> Update proposal.md (#8)
>
> * Update proposal.md
>
> * Update proposal.md

commit fd871274b0129f90d0c108225900f45d69b5c827 Author: Graham Patterson
gpp2109@columbia.edu Date: Mon Oct 1 21:00:37 2018 −0400

> Update README.md
>
> adds john to team members

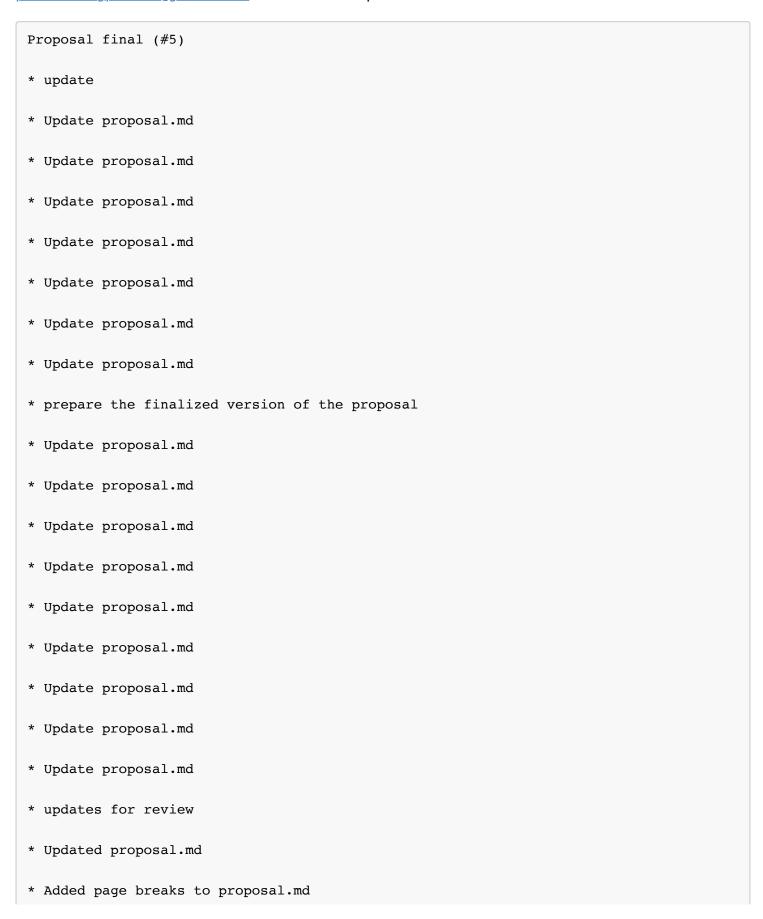commit 01f7591893246be550b6508b811858ec040dc551 Author: Graham Patterson
gpp2109@columbia.edu Date: Mon Oct 1 20:59:41 2018 −0400

> Update README.md

commit 0af6b799f2843b45d0d3e795fa9c520034fc6b05 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Sat Sep 29 11:15:23 2018 −0700

> uploads proposal notes from John Hui; changes filename (#6)

commit 96988865c7a58050c55c43c030a6d7820c4d2047 Author: Graham Patterson

pattersongp.usmc@gmail.com Date: Thu Sep 20 09:07:33 2018 −0400

```
Proposal final (#5)

* update

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* prepare the finalized version of the proposal

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* Update proposal.md

* updates for review

* Updated proposal.md

* Added page breaks to proposal.md
```

```
* Renamed proposal.md to FIRE.md
```

commit a9f06ef300d32ae946f4f72b02ef454da6b1b570 Author: Ayer Chan
oc2237@columbia.edu Date: Tue Sep 18 16:16:19 2018 −0400

```
Create 09182018.md (#3)
```

commit 337c2cac87bacc2e7ccf360599ee282bec5b4124 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Sep 11 13:48:30 2018 −0400

```
add 9-11-18 meeting notes (#2)
```

commit 4c8dcddf196b2b53b24fefdf836aec75b128dc4e Merge: b94f627 6b7ddb1 Author:
jkon1513 jkon1513@gmail.com Date: Tue Sep 11 13:48:18 2018 −0400

```
Merge pull request #1 from pattersongp/proposal-stub

Proposal stub
```

commit 6b7ddb188ba250861a438366e4e5fc1545e85419 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Sep 11 12:01:48 2018 −0400

```
Update proposal.md
```

commit b94f62794b52880172a9873eebe125bada92d517 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Sep 11 11:41:46 2018 −0400

```
update
```

commit 2c1c1b8786faba0a59e9adc57aafaf7af080c158 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Sep 11 11:41:05 2018 −0400

```
update
```

commit da1ea91dde86e1ec08ed76e16b4e729ca0a115cd Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Tue Sep 11 11:39:18 2018 −0400
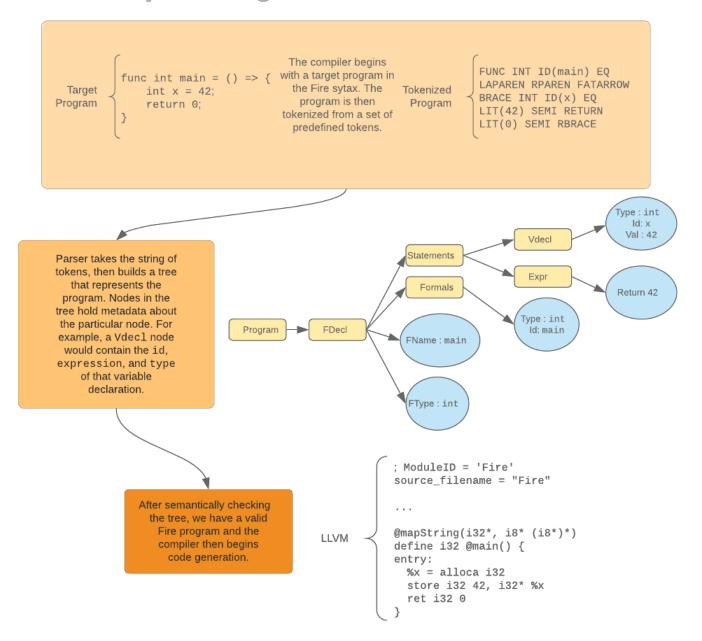
```
Adds stubbed out proposal
```

commit d3ae9e9c307fb191543e00a6b969506c7fac0839 Author: Graham Patterson
pattersongp.usmc@gmail.com Date: Thu Sep 6 09:19:02 2018 −0400

> This will be the last direct commit to master, everything else through branches

5. Architectural Design

The diagram below describes both the architecture and the interface between each component. The interfaces between each component are data structures defined in various components. For example, a tree representing a program is given to the code generator to traverse the data structure, generating appropriate LLVM in order to compile the program for the native distribution.

Fire Compiler Diagram

Target Program

```
func int main = () => {
    int x = 42;
    return 0;
}
```

The compiler begins with a target program in the Fire sytax. The program is then tokenized from a set of predefined tokens.

Tokenized Program

```
FUNC INT ID(main) EQ
LAPAREN RPAREN FATARROW
BRACE INT ID(x) EQ
LIT(42) SEMI RETURN
LIT(0) SEMI RBRACE
```

Parser takes the string of tokens, then builds a tree that represents the program. Nodes in the tree hold metadata about the particular node. For example, a Vdecl node would contain the id, expression, and type of that variable declaration.

Program → FDecl

Statements

Vdecl → Type : int, Id : x, Val : 42

Expr → Return 42

Formals

FName : main

Type : int, Id : main

FType : int

After semantically checking the tree, we have a valid Fire program and the compiler then begins code generation.

LLVM

```
; ModuleID = 'Fire'
source_filename = "Fire"

...

@mapString(i32*, i8* (i8*)*)
define i32 @main() {
entry:
  %x = alloca i32
  store i32 42, i32* %x
  ret i32 0
}
```

- **Scanner, Parser, AST** Frank and Graham
- **Semantic Checker** Frank, Jason, and Chris
- **Semantically Checked AST** Frank, Jason, and Chris
- **Code Generation** Graham and Ayer
- **Array Library** Ayer and Graham
- **File Library** Graham and Frank
- **Regular Expression Library** Graham
- **Utilities Library** Graham

6. Testing Plan

- *Show two or three representative source language programs along with the target language program generated for each*
- *Show the test suites used to test your translator*
- *Explain why and how these test cases were chosen*
- *What kind of automation was used in testing*
- *State who did what*

Testing was a pivotal part of our development process. As FIRE matured as a language, we rapidly realized that as our language grew more complex, we would have to develop automated means of testing new and existing functionality. The following section details our test plan.

6.1 Testing Workflow

The structure of a FIRE test is to generate a test case for each salient feature (ideally as the feature is developed) store the expected output of that program in a file titled `test_name.out`, and then run a script that compiles the program and compares the generated output to the expected output. The script utilizes UNIX's `diff` tools to perform this comparison — if a difference is detected, the test fails, but if not, it passes. Our Workflow was inspired by MicroC's test suite.

Our workflow also contained a large number of erroneous test cases. For these tests, we noted the expected error message the erroneous test case should generate, and performed a similar comparison as our valid test cases when we attempted to compile the invalid FIRE programs. This helped us avoid prohibited behavior as FIRE matured.

We also had a separate workflow earlier in our design process for testing parsing errors. Prior to building up our automated test suite, we created a set of tests to check syntax as we designed our AST. This was pivotal in detecting issues with how we parsed and tokenized components of our language.

6.2 Regression Testing

Every time our compiler was built via our Makefile, the `make` command would automatically run our test suite, the logic for which resided in our `testall.sh` shell script. Written in Bash, this script resided in the root of our repository

and outputted whether a test succeeded or failed. This allowed us to ensure that older functionality was not adversely impacted by the development of new features.

6.3 testall.sh

Below is the code for `testall.sh`.

```sh
#!/bin/sh

# Testing script for FIRE
#
#  Compile, run, and check the output of each test against expected result
#  Expected failures run diff against the expected error, expected passes check again
st the expected output of the file

# Path to the LLVM interpreter
LLI="lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="gcc"

# Path to the Fire compiler
FIRE="src/Fire.native"

# Set time limit for all operations
ulimit -t 30

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
    echo "Usage: testall.sh [options] [.fire files]"
    echo "-h    Print this help"
    echo "-c Clear all the .diff and .err files from root generated by a failure in t
estall.sh"
    exit 1
```

```bash
}

SignalError() {
    if [ $error -eq 0 ] ; then
    echo "FAILED"
    error=1
    fi
    echo "  $1"
}

# Compare <outfile> <reffile> <difffile>
# compares the output file against the gold standard reffile, if any differences exis
t then writes to diff file
Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
    SignalError "$1 differs"
    echo "FAILED $1 differs from $2" 1>&2
    }
}

# Run <args>
# Report the command, run it, and report any errors
Run() {
    echo $* 1>&2
    eval $* || {
    SignalError "$1 failed on $*"
    return 1
    }
}

# RunFail <args>
# Report the command, run it, and expect an error
RunFail() {
    echo $* 1>&2
    eval $* && {
    SignalError "failed: $* did not report an error"
    return 1
    }
    return 0
}

Clean() {
    rm -f test-*
```

```
    rm -f fail-*
    echo "cleaning all diff and err files from root"
    exit 1
}

Check() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                            s/.fire//'`
    reffile=`echo $1 | sed 's/.fire$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe ${ba
sename}.out" &&
    Run "$FIRE" "$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
    Run "$CC" "-o" "${basename}.exe" "${basename}.s" "arrlib.o" "filelib.o" "regexlib
.o" "printlib.o" &&
    Run "./${basename}.exe" > "${basename}.out" &&
    Compare ${basename}.out ${reffile}.out ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
    if [ $keep -eq 0 ] ; then
        rm -f $generatedfiles
    fi
    echo "OK"
    echo "###### SUCCESS" 1>&2
    else
    echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
```

```
                                s/.fire//'`
    reffile=`echo $1 | sed 's/.fire$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$FIRE" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
    if [ $keep -eq 0 ] ; then
        rm -f $generatedfiles
    fi
    echo "OK"
    echo "###### SUCCESS" 1>&2
    else
    echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

while getopts ckdpsh opt; do
    case $opt in
    k) # Keep intermediate files
        keep=1
        ;;
    h) # Help
        Usage
        ;;
    c) # clear all .diff and .err files generated in root by testall
        Clean
        ;;
    esac
done

shift `expr $OPTIND - 1`
```

```
LLIFail() {
  echo "Could not find the LLVM interpreter \"$LLI\"."
  echo "Check your LLVM installation and/or modify the LLI variable in testall.sh"
  exit 1
}

which "$LLI" >> $globallog || LLIFail

if [ ! -f src/arrlib.o ]
then
    echo "Could not find arrlib.o"
    exit 1
fi

if [ ! -f src/regexlib.o ]
then
    echo "Could not find regexlib.o"
    exit 1
fi

if [ ! -f src/filelib.o ]
then
    echo "Could not find filelib.o"
    exit 1
fi

if [ ! -f src/printlib.o ]
then
    echo "Could not find printlib.o"
    exit 1
fi

if [ $# -ge 1 ]
then
    files=$@
else
    files="test/test-*.fire test/fail-*.fire"
fi

for file in $files
do
    case $file in
    *test-*)
        Check $file 2>> $globallog
        ;;
```

```
    *fail-*)
        CheckFail $file 2>> $globallog
        ;;
    *)
        echo "unknown file type $file"
        globalerror=1
        ;;
    esac
done


exit $globalerror
```

6.4 Sample Tests

6.5 Testing Credits

- **Parser Testing** Ayer
- `testall.sh` **design/rewrite** Chris and Jason
- **Invalid Test Cases** Jason
- **Valid Test Cases** Chris
- **Misc Tests** Jason and Graham

7. Lessons Learned

- *Each team member should explain his or her most important learning Include any advice the team has for future teams*

Graham

> The most important experience I had during this course was the demystification of compilers, interpreters,
> and translators. Working on various components of the compiler exposed the various choices of production
> compilers like gcc, clang, and Javascript interpreter. Engaging with various problems like code generation
> and intermediate representation of programs exposes problems that compiler writers have experienced. For
> example, when implementing the lexer code example, I needed to implement a token choice algorithm. When I
> researched the algorithm I usually found its shortcomings described in the context of the C++ compiler
> and the issues with <template<template>> and the >> operator. Engaging with algorithms that are used in
> tools that I use everyday was incredibly rewarding. This course made me a more well rounded programmer
> and software developer.

Christopher

> I am continually fascinated by language and the chasm that can exist between what we
> say and we mean to say. This chasm becomes especially apparent in the design of programming languages. In natural language, it is perilously easy to say something that is
> far removed from the concept you are attempting to convey – how many fights between
> friends, colleagues or couples stem from a poor or ambiguous choice of words. This same difficulty can plague the art of programming, and so much of the task of becoming
> proficient in a language is learning how to tell a computer exactly what it is you intend it to do. Programming languages can play a huge role in how successful these attempts can be. I've come to realize that the way a language is structured shapes the way we approach or even think about a problem. We are transmuted by the very tools we
> employ. PLT will rank as one of my favorite classes because I didn't just solve a problem, I had a say in shaping the inner life and mental models employed by the coders
> who use our language.

8. Appendix

- *Attach a complete code listing of your translator with each module signed by its author*