

# Fli-o: The File I/O language



Manager: Matthew Chan (mac2474)

System Architect: Justin Gross (jg3544)

Tester: Gideon Cheruiyot (gkc2112)

Language Guru: Eyob Tefera (et2546)

# Motivation

- File manipulation in most languages leaves something desired
  - Wanted to make file manipulation easier
-

# What makes Fli-0 so special?

- Native types for Files and Directories
  - Built-in library for common file and directory operations
  - Simplified file interactions
-

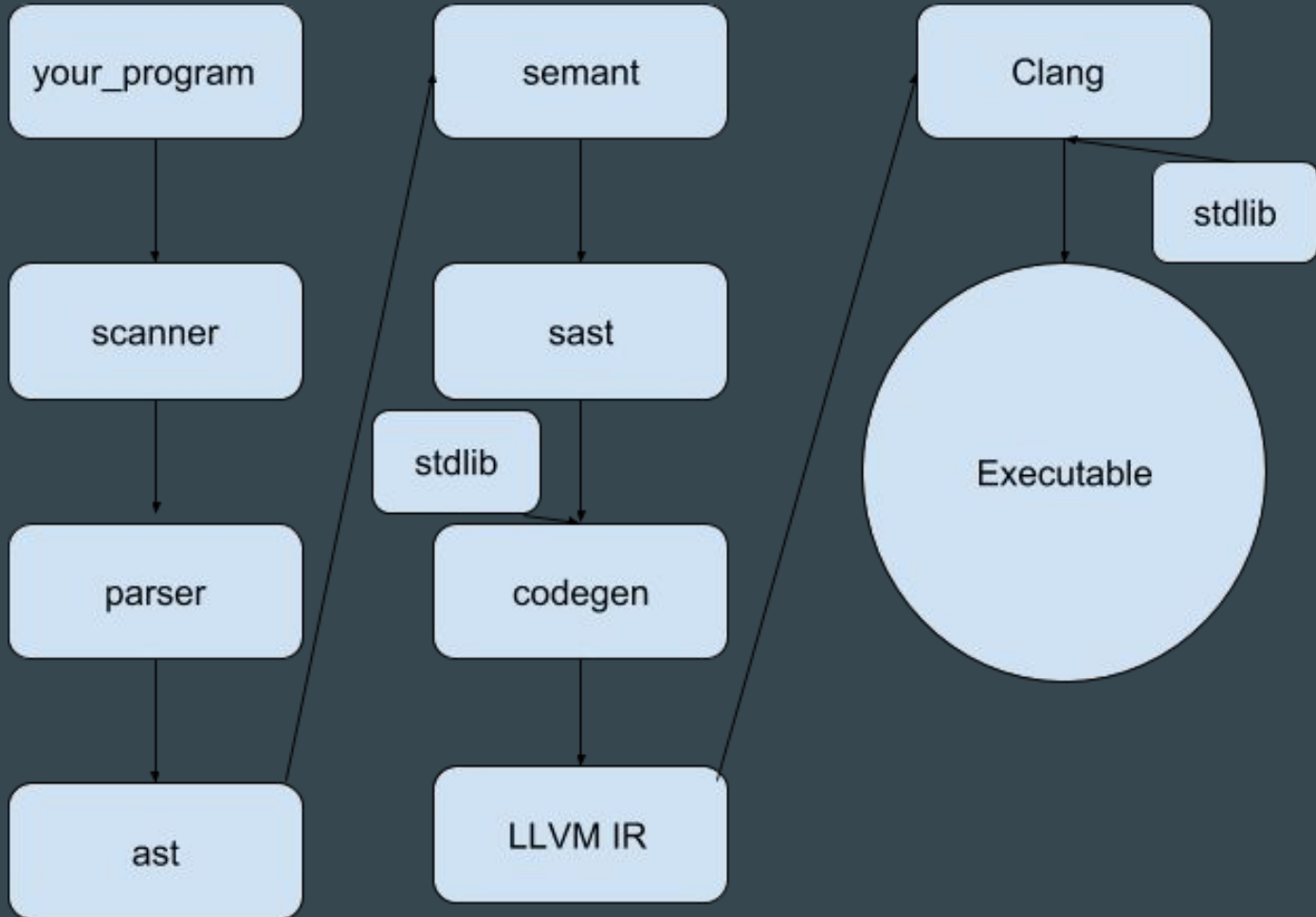
# Fli-o's Evolution

- Originally had two ideas
    - File Manipulation
    - String operations
  - Had project proposal reviewed
    - Figured out it was better to focus on one
  - File manipulation seemed more unique, focused on that
-

# How Fli-o was created

- Code written in Ocaml, C, and LLVM
    - LLVM via Ocaml bindings
  - Communication through Facebook messenger
  - Collaboration through Github
-

# System Architecture



# Testing

We tested Fli-o in three ways:

- Unit Test - short to quickly find errors.
  - Manual Testing - building an exe for each .ll file. We used this to develop the test-suite one by one.
  - Test Script/Regression Testing
-

# Language Overview

## General Attributes:

- Statically typed
- Static scoping
- Applicative order evaluation
- Main function in global scope
  - Ex: OCaml, Python, bash

## Unique Features:

- Built-in functions
    - Operations on files and strings
-



# Built-in Functions

stdlib.c

## Operations on files

- file fopen(string fname)
  - int create(string fname)
  - int move(string f1, string f2)
  - int copy(string f1, string f2)
  - int delete(string fname)
  - int write(file f, string buf)
  - string read(file f, int len)
  - string readLine(file f)
  - int appendString(file f, string buf)
-

# Built-in Functions

stdlib.c

## Operations on strings

- void prints(string s)
- string concat(string s1, string s2)
- int strcmp(string s1, string s2)

## Operations on integers

- void print(int i)
- string intToStr(int i)

## Operations on directories

- dir dopen(string path)
- int rmdir(string path)

**DEMO**

# Demo Program

## Source Code

```
def addLineNumbers(string filename)
{
    file f = fopen(filename);
    // Create a copy of the current file
    string copyName = concat('lined_', filename);
    copy(filename, copyName);
    file newFile = fopen(copyName);
    string line = readLine(f);

    // Keep track of which line we are on
    int lineNo = 0;
    string prefix;

    // Loop through all of the lines in file f
    for(; strcmp(line, '') != 0;;) {
        prefix = concat('[', intToStr(lineNo));
        prefix = concat(prefix, ' ] ');
        // Write the lined version to the new file
        write(newFile, concat(prefix, line));
        line = readLine(f);
        lineNo = lineNo + 1;
    }
}
addLineNumbers('sample.txt');
```

# BEFORE

1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus condimentum sagittis lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamcorper accumsan velit vel vehicula. Proin tempor lacus arcu. Nunc at elit condimentum, semper nisi et, condimentum mi. In venenatis blandit nibh at sollicitudin. Vestibulum dapibus mauris at orci maximus pellentesque. Nullam id elementum ipsum. Suspendisse cursus lobortis viverra. Proin et erat at mauris tincidunt porttitor vitae ac dui.

2

3 Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent dictum luctus massa, non euismod lacus. Pellentesque condimentum dolor est, ut dapibus lectus luctus ac. Ut sagittis commodo arcu. Integer nisi nulla, facilisis sit amet nulla quis, eleifend suscipit purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam euismod ultrices lorem, sit amet imperdiet est tincidunt vel. Phasellus dictum justo sit amet ligula varius aliquet auctor et metus. Fusce vitae tortor et nisi pulvinar vestibulum eget in risus. Donec ante ex, placerat a lorem eget, ultricies bibendum purus. Nam sit amet neque non ante laoreet rutrum. Nullam aliquet commodo urna, sed ullamcorper odio feugiat id. Mauris nisi sapien, porttitor in condimentum nec, venenatis eu urna. Pellentesque feugiat diam est, at rhoncus orci porttitor non.

4

5 Nulla luctus sem sit amet nisi consequat, id ornare ipsum dignissim. Sed elementum elit nibh, eu condimentum orci viverra quis. Aenean suscipit vitae felis non suscipit. Suspendisse pharetra turpis non eros semper dictum. Etiam tincidunt venenatis venenatis. Praesent eget gravida lorem, ut congue diam. Etiam facilisis elit at porttitor egestas. Praesent consequat, velit non vulputate convallis, ligula diam sagittis urna, in venenatis nisi justo ut mauris. Vestibulum posuere sollicitudin mi, et vulputate nisl fringilla non. Nulla ornare pretium velit a euismod. Nunc sagittis venenatis vestibulum. Nunc sodales libero a est ornare ultricies. Sed sed leo sed orci pellentesque ultrices. Mauris sollicitudin, sem quis placerat ornare, velit arcu convallis ligula, pretium finibus nisl sapien vel sem. Vivamus sit amet tortor id lorem consequat hendrerit. Nullam at dui risus.

6

7 Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat semper velit consequat facilisis. Etiam facilisis justo non iaculis dictum. Fusce turpis neque, pharetra ut odio eu, hendrerit rhoncus lacus. Nunc orci felis, imperdiet vel interdum quis, porta eu ipsum. Pellentesque dictum sem lacinia, auctor dui in, malesuada nunc. Maecenas sit amet mollis eros. Proin fringilla viverra ligula. sollicitudin viverra ante sollicitudin

# AFTER

1 [0] Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus condimentum sagittis lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamcorper accumsan velit vel vehicula. Proin tempor lacus arcu. Nunc at elit condimentum, semper nisi et, condimentum mi. In venenatis blandit nibh at sollicitudin. Vestibulum dapibus mauris at orci maximus pellentesque. Nullam id elementum ipsum. Suspendisse cursus lobortis viverra. Proin et erat at mauris tincidunt porttitor vitae ac dui.

2 [1]

3 [2] Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent dictum luctus massa, non euismod lacus. Pellentesque condimentum dolor est, ut dapibus lectus luctus ac. Ut sagittis commodo arcu. Integer nisi nulla, facilisis sit amet nulla quis, eleifend suscipit purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam euismod ultrices lorem, sit amet imperdiet est tincidunt vel. Phasellus dictum justo sit amet ligula varius aliquet auctor et metus. Fusce vitae tortor et nisi pulvinar vestibulum eget in risus. Donec ante ex, placerat a lorem eget, ultricies bibendum purus. Nam sit amet neque non ante laoreet rutrum. Nullam aliquet commodo urna, sed ullamcorper odio feugiat id. Mauris nisi sapien, porttitor in condimentum nec, venenatis eu urna. Pellentesque feugiat diam est, at rhoncus orci porttitor non.

4 [3]

5 [4] Nulla luctus sem sit amet nisi consequat, id ornare ipsum dignissim. Sed elementum elit nibh, eu condimentum orci viverra quis. Aenean suscipit vitae felis non suscipit. Suspendisse pharetra turpis non eros semper dictum. Etiam tincidunt venenatis venenatis. Praesent eget gravida lorem, ut congue diam. Etiam facilisis elit at porttitor egestas. Praesent consequat, velit non vulputate convallis, ligula diam sagittis urna, in venenatis nisi justo ut mauris. Vestibulum posuere sollicitudin mi, et vulputate nisl fringilla non. Nulla ornare pretium velit a euismod. Nunc sagittis venenatis vestibulum. Nunc sodales libero a est ornare ultricies. Sed sed leo sed orci pellentesque ultrices. Mauris sollicitudin, sem quis placerat ornare, velit arcu convallis ligula, pretium finibus nisl sapien vel sem. Vivamus sit amet tortor id lorem consequat hendrerit. Nullam at dui risus.

6 [5]

7 [6] Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat semper velit consequat facilisis. Etiam facilisis justo non iaculis dictum. Fusce turpis neque, pharetra ut odio eu, hendrerit rhoncus lacus. Nunc orci felis, imperdiet vel interdum quis, porta eu ipsum. Pellentesque dictum sem lacinia, auctor dui in, malesuada nunc. Maecenas sit amet mollis eros. Proin fringilla viverra ligula. sollicitudin viverra ante sollicitudin congue.

# Next Steps

## Additional Features:

1. Pipes & exec
    - a. dup2 syscall
    - b. Issues: LLVM pointer indirection
  2. Arrays
  3. Built-in functions
-

# Lessons Learned

- Don't try to do too much
- Be more flexible on work allocation
- OCaml has a steep learning curve
- Keep track of your lets and ins

---

# Individual Thoughts

- Eyob
- Gideon
- Matthew
- Justin

---