Fortress Boot Camp Plan

Sukyoung Ryu

June 7, 2008

1 Interns and Their Tasks

- Static system: Nels Beckman (CMU) and Justin Hilbert (UT Austin)
- Code generation: Angelina Lee (MIT) and Michael Spiegel (University of Virginia)
- Syntax abstraction: Jon Rafkind (University of Utah) and Ryan Culpepper (Northeastern University)

2 Prerequisites

Before the Fortress boot camp starts, each intern is asked to do the following:

- Install fortress.
- Read the Overview chapter of the Fortress Language Specification Version 1.0.
- Write and run HelloWorld.fss.
- Familiarize with Trac: SCA form, Timeline, Tickets

3 Logistics

- The Fortress boot camp will take place in Freedom Trail (Rm. 3806).
- All PLRG members will stay in the room serving as TAs for the speakers.
- Each talk should involve hands-on work with code by all the interns. So, for example, when the speaker talks about traits and objects, the speaker may walk them through an exercise of writing some traits and objects.
- Each talk is strongly encouraged to interleave lecture with hands-on work from the beginning emphasizing the importance of test-driven design. For example, first, write this function. Now test it. Now write this object definition. Now test it. Now add this trait definition and modify your object definition to extend it. Now test it. Like that.
- Each talk is strongly encouraged to present library code.
- Each hour consists of 50-minute talk and 10-minute break.

4 Schedule

Date	Time	Lecturer	Content
05/21	9:00AM		Breakfast at the cafeteria (optional)
(Wed.)	9:30AM	Sukyoung Ryu	Welcome and overview of the Fortress boot camp (1)
	9:40AM	Eric Allen	Introduction to PLRG (2)
	10:30AM	Sukyoung Ryu	Language overview by example (I) (3)
	11:30AM	Sukyoung Ryu	Language overview by example (II) (3)
	12:30PM		Lunch
	1:30PM	Guy Steele	Traits, objects, and static parameters (4)
	2:30PM	Jan-Willem Maessen	Parallelism and transaction (5)
	3:30PM	Everyone	Homework (by Guy) (6)
	5:30PM		Dinner
05/22	9:00AM		Breakfast at the cafeteria (optional)
(Thu.)	9:30AM	Jan-Willem Maessen	Introduction to Fortress libraries (7)
	10:30AM	Victor Luchangco	Overloading and operators (8)
	11:30AM	Guy Steele	Arrays, ranges, comprehensions, and reductions
	12:30PM		Lunch
	1:30PM	Guy Steele	Collection types and Generators
	2:30PM	Jan-Willem Maessen	Defining a Generator, a hands-on tutorial (9)
	3:00PM		Tea
	4:00PM	Everyone	Homework (by Jan) (6)
	5:30PM		
05/23	9:00AM		Breakfast at the cafeteria (optional)
(Fri.)	9:30AM	David Chase	Introduction to the Fortress interpreter (10)
	10:30AM	Sukyoung Ryu	Fortress parser and AST (11)
	11:00AM	Eric Allen	Fortress static end (12)
	11:30AM	Jan-Willem Maessen	Fortress desugarer
	12:00PM		Lunch
	1:00PM	Christine Flood	Fortress evaluator
	1:30PM	Christine Flood	Fortress parallelism and transaction implementation (13)
	2:00PM	David Chase	Data structures and compiler design/development plan (14)
	3:00PM	Eric Allen	Fortress tools and Fortress summer tasks
	3:30PM	Everyone	Homework (by Eric) (6)
	5:30PM		

(1) Each intern will have two projects:

1. writing libraries or useful applications in Fortress
2. implementing some parts of the compiler

Each intern should be able to start working on their projects almost right after the boot camp.

(2) Gives a high-level introduction to our team

- working environment, culture, etc.
- our coding conventions and commit rules
- pair programming

(3) Language constructs to cover:

variables and functions (top level and local), literals (`()` , numbers, chars, strings), variable references, function calls, operator applications, function expressions, object expressions, assignments, `do` , `label` and `exit` , `for` , `while` , `if` , `case` , extremum expressions, `typecase` , `throw` , `try` , `as` , `asif` , components, APIs, imports, and exports

(4) Language constructs to cover:

traits, objects, fields, methods, functional methods, `excludes`, `comprises`, static parameters

(5) Language constructs to cover:

tuples, `atomic`, `spawn`, parallel do, generated expressions

(6) Homework in class pairing with PLRG members

(7) Gives an overview of the Fortress libraries and types defined in libraries

(8) How to define overloading and operators and how to use them

(9) Present a worked example of:

- How to add a new primitive type
- How to add a type that supports Generator
- What all the plumbing entails.

(10) Gives an overview of the Fortress interpreter: its overall architecture, the driver, a description of each package and what's contained in it, other important directories, various Ant targets, what libraries we rely on (dstm2, etc.)

`http://projectfortress.sun.com/Projects/Community/wiki/InterpreterGuide`

(11) Gives an overview of the Fortress parser, AST, how to revise them with principles, and facilities for transforming ASTs.

(12) Gives an overview of the Fortress static end: disambiguator and type checker.

(13) Gives an overview of how parallelism and transactions are implemented, particularly our use of dstm2 and Doug Lea's work stealing code.

(14) Gives an overview of the interpreter data structures including `FValue` and `FType` and the compiler design/development plan.