Implementing UNIX with Effects Handlers

Ramsay Carslaw

October 4, 2023

Contents

Chapter 1

Introduction

Chapter 2

Background

2.1 Algebraic Effects and Effect Handlers

Algebraic effects and their corresponding handlers [1] [2] are a programming paradigm that when paired together offers a novel way to compose programs. When programs are written that are 'black boxes', that is to say their outputs are defined entirely by their inputs and all functions are pure computation [3], it is safe to make assumptions about the inputs. Assumptions like an age will always be given as an integer or all strings will not exceed the length allocated for them. When programs interact with the real world it is no longer safe to make these assumptions. To handle these 'side effects' most modern languages introduce the concept of an exception like the below example:

```
1 firstName = input(``Enter your first name: '')
2 try {
3   print(``Hello, '' + firstName)
4 } catch InvalidCharacter {
5   error(``Invalid Character in first name!'')
6 }
```

2.2 UNIX

Unix [4] is an operating system designed and implemented by Dennis M. Ritchie and Ken Thompson at AT&T's Bell Labs in 1974. It provides a file system (directories, file protection etc.), a shell, processes (pipe, fork etc) and a userspace. Since it's first release it has been reimplemented for a variety of systems.

Chapter 3

Methods

Chapter 4

Results

Chapter 5

Conclusion

Bibliography

[1] Gordon Plotkin and Matija Pretnar. Handlers of algebraic effects. In *European Symposium on Programming*, pages 80–94. Springer, 2009.

[2] Matija Pretnar. An introduction to algebraic effects and handlers. invited tutorial paper. *Electronic notes in theoretical computer science*, 319:19–35, 2015.

[3] John Hughes. Why functional programming matters. *The computer journal*, 32(2):98–107, 1989.

[4] Dennis M Ritchie and Ken Thompson. The unix time-sharing system. *Bell System Technical Journal*, 57(6):1905–1929, 1978.