

# Computer Science Tripos – Part II – Progress Report

## A strongly consistent index for email using git and MirageOS.

O. Hope, Jesus College

Originator: Dr A. Madhavapeddy

28th January 2019

**Project Supervisor:** Mr D. Allsopp

**Director of Studies:** Prof C. Mascolo

**Project Overseers:** Prof G. Winskel & Dr R. Mortier

### Progress compared to the schedule

Currently the project is on track compared to the original schedule. According to the proposal, at this point the core of the project was meant to be complete in OCaml which is the case. Furthermore, this was intended to be the stage where implementation of extensions starts to take place alongside the building of test frameworks to gather data for evaluation. These are both currently in the early stages but have begun. This shows that while the project is not ahead it is not behind, so should be finished in good time.

### Difficulties encountered

The project has run relatively smoothly. However, I did have to modify the schedule slightly, moving from prototyping to implementation more quickly than originally expected as there was more that needed re-implementing in OCaml compared to the commandline tool than I was originally expecting. Also, I had some difficulty at first working out how to use the necessary libraries but this has all been resolved at this point.

Something else that should be noted is that while one of the original success criteria was “Implementation of a working git overlay on top of the *filesystem in MirageOS*” what I have actually created is a tool with Mirage libraries that functions on UNIX-like systems. This is after discussion with my supervisor as we deemed it a more interesting problem. However, due to the nature of the libraries that I am using and OCaml functors, the majority of my code will function in exactly the same manner on MirageOS, there will just be less interesting things for it to do.

### What has been accomplished

Thus far I have created a library in OCaml which, with the help of an ocaml-git library, allows me to perform git plumbing commands in OCaml. Using this I have then implemented the necessary functions for basic interaction with a maildir in a git store through

OCaml. Finally, I have created a commandline interface that wraps this, allowing for interaction from users and other programs. This tool also syncs a standard maildir with the git store and handles any issues of concurrency upon modification attempts from multiple concurrent processes.