Cloud Storage Mounting on Android OS

The goal is to mount a cloud storage directory to the internal storage of an Android device, at the Operating System level.

CSMAOS
Cloud Storage Mounting on Android OS

College of Engineering and Computer Science
Wright State University
State of Ohio
City of Fairborn

Hanen Alkhafaji
Wright State University
3640 Colonel Glenn Hwy.
Dayton, OH 45435 USA
Phone (937) 775-1000
alkhafaji.2@wright.edu
http://www.wright.edu

Summary

To be decided...

Preface

To be decided...

Acknowledgements

This template is based on the Laursen's DTU Thesis template. [Lyn]
Dr. Mateti has been my professor and mentor for this project.

Contents

List of Figures

CHAPTER 1

Introduction

To be decided...

Project Scope

2.1 Introduction

This document outlines the scope of the project that is being explored, researched, and implemented for CS 6970. It includes the description, scope, and deliverables of the project as well as details on what criteria is expected to be met in order for this project to be deemed accepted. Another important purpose for this document is so that all stakeholders involved in this project have a common understanding of the scope, expectations, and goals of this project. August 1, 2019 is the expected completion date and submission of deliverables for this project.

2.2 Project Purpose and Justification

The Cloud Storage Mounting on Android OS project has been approved for researching, planning, designing, building, and implementation. The project involves mounting a cloud storage directory at the Operating System level so that the directory can be accessed from only application running on the mobile device. The purpose is to make that cloud storage appear to be native to the device rather than hosted elsewhere. The successful implementation of this project will meet the requirements for the course CS 6970 as well as produce software that will greatly reduce the friction of cloud storage access for mobile device users.

2.3 Scope Description

The scope of the Cloud Storage Mounting on Android OS project will involve research, planning, designing, building, and implementing a solution for mounting a cloud storage at the Operating System level of an Android mobile device. Open source resources will be used as tools for implementation as well as starting points.

2.4   High Level Requirements

The only requirement specified for this particular project is to properly perform mounting of a cloud storage directory so that it can be accessed from anywhere on the mobile device.

2.5   Boundaries

The scope of the Cloud Storage Mounting on Android OS project includes all work involving research, planning, designing, building, and implementing a solution for mounting a cloud storage at the Operating System level of an Android mobile device. Tasks that will also be involved include gathering requirements, writing up requested documents and documentation as well as the technical report, deploying the solution, and testing the solution on an Android mobile device. The scope of this project will not include an implementation for Windows or Apple mobile devices or implementations for other cloud storage providers outside of Google Cloud Platform.

2.6   Strategy

The Cloud Storage Mounting on Android OS project will be implemented by one developer using a machine running Linux and has Android Studio installed. The developer will research open source options to use as a template to begin the implementation. The implementation will be tested locally on the Linux machine to debug and make sure functionality is operating as expected. Then, either by using the mobile device emulator or the physical Android device (Samsung Galaxy Tab A6), the implementation will be tested on a mobile device for completeness.

2.7   Deliverables

The Cloud Storage Mounting on Android OS project will yield an Android APK file that can be downloaded and installed on an Android mobile device. A technical report detailing the project will be delivered as well.

2.8   Acceptance Criteria

Acceptance criteria have been established for the PMD Project to ensure thorough vetting and successful completion of the project. The acceptance criteria are both qualitative and quantitative in nature. All acceptance criteria must be met in order to achieve success for this project:

Meet all deliverables within scheduled time and budget tolerances. Reduce schedule delays by at least 30Reduce budget overruns by at least 30Improve Acme Consulting's resource allocation ability. Accomplish an overall performance improvement in program metrics.

Read more: https://www.projectmanagementdocs.com/template/project-documents/scope-statement/ixzz5pnIgkmLx

2.9   Constraints

The project must be completed within the range of days between May 13, 2019 and August 1, 2019. That is about 12 weeks. Development will be performed on a personal Linux (Ubuntu 18.10) machine. Testing will be performed on the same machine and a Samsung Galaxy Tab A6 mobile device that has been rooted.

2.10   Assumptions

The application will be installed on only Android devices. The user on the device has root access. The directory being mounted is on Google Cloud provider, not any other provider.

CHAPTER 3

Development Plan

3.1 Scope

Please refer to the Project Scope section [2] for details.

3.2 Software Development Process

3.3 Project Planning and Oversight

This project will be planned and managed using a tool called Trello [Tre]. Columns are used to specify the status of each work item referred to as a 'card'. Each card describes the task that needs to be completed. Checklists can be added to a card in order to plan the work breakdown and check things off as they are completed. Dr. Mateti and I are both members of this board and can be assigned to different cards in order to specify who needs to work on certain items. Each card has a conversation functionality where we can both add comments in order to communicate about the work completed and provide updates. When a card is updated, each of us receives an email and mobile notification.

The columns are 'To Do', 'In Progress', 'Ready for Review', 'Done', and 'Postponed'.

I added cards into the 'To Do' column at the onset of this project in order to make sure that I had goals for myself to reach. I broke the project down into six milestones. These depicted progress made on the technical report. Every other week, an updated draft is due. I specified those due dates on each card. As Dr. Mateti discovered items he wanted me to research or perform, he would add cards to the 'To Do' column as well.

When I was ready to pick those items up and begin working on them, I would move the card into the 'In Progress' column. Once I completed the task, I would put a comment on the card and move it into 'Ready for Review' so that Dr. Mateti was aware that I was finished with the item and needed his feedback. If it was an item that did not need feedback, I would move it straight into the 'Done' column. If it

required feedback, Dr. Mateti would respond with a comment on the card and I would move the card into a column based on the feedback. If more work needed to be done, I would move the card back into 'In Progress'. If Dr. Mateti was happy with the work, I would move that card into 'Done' and move on to the next card.

Dr. Mateti added a column called 'Postponed' for items that we had planned to work on at the start of the project. However, we found that we were running out of time and wanted to prioritize other items above certain research labs. Those cards were moved into 'Postponed'. If time permits, we will revisit those items, but they are not high priority right now.

3.4 Establishing a Software Development Environment

1. Setup Linux Operating System
2. Dr. Mateti provided a rooted Nexus 7 tablet
3. Install OCamlFUSE
4. Install Visual Studio Code

3.5 Software Requirements Analysis

3.6 System Design

3.7 Software Requirements Analysis

3.8 Software Design

3.9 Software Implementation

3.10 Software Quality Assurance

3.11 Schedules and Activities

- May 14: Decide on project topic
- May 14: On Campus Meeting
- May 17: Read about Fuse

- May 21: Read about SSHFS

- May 21: Install and Try ES File Explorer

- May 23: Install and Try X-plore

- May 23: Draft 1

- June 2: Learned LaTeX and Converted Report

- June 3: Project Scope Document

- June 4: On Campus Meeting

- June 6: Draft 2

- June 8: OCamlFUSE Mount on Linux

- June 11: Dr. Mateti provides rooted Nexus 7 tablet

- June 11: On Campus Meeting

- June 13: Cloud Storage Mounts in Android

- June 17: Lab Report: OCamlFUSE

- June 18: Make Changes to Sample Code

- June 20: Draft 3

- June 21: Create Outline to Guide Project

- June 23: Mount USB

- July 17: Draft 4

- July 25: Draft 5

- August 1: Final Technical Report

CHAPTER 4

OCamlFUSE

4.1 Explore OCamlFUSE Source Code

4.1.1 Summary

(https://github.com/astrada/google-drive-ocamlfuse)

1. Source code downloaded from GitHub. [Str]

2. ML versus MLI file The first thing I noticed was that there were duplicate files in the source folder that simply had different extensions. After some research, I found that the extension ML stands for Meta Language which is the umbrella programming language that contains OCaml. I suspect that the extension MLI stands for Meta Language Interface, but I cannot be certain of that since I was unable to find its expansion in my research. The reason I suspect it is Interface is because of what I observed when I looked at the files themselves. I opened bufferPool.ml and bufferPool.mli and compared them to each other. In the ML file, I found full function implementations while in the MLI file, I found a listing of function signatures. So, the MLI files must be interfaces that are used by other classes so as to hide the implementation in the ML from outsiders. Dr. Mateti clarified for me that MLI files are compiled from ML files.

3. SLOCCOUNT

```
hanen@hanen:~$ sloccount /home/hanen/Desktop/google-drive-ocamlfuse-beta/
Have a non-directory at the top, so creating directory top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//LICENSE to top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//Makefile to top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//README.md to top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//_config.yml to top_di
Creating filelist for bin
Creating filelist for doc
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//dune-project to top_d
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//google-drive-ocamlfus
Creating filelist for test
Creating filelist for tools
```

```
Have a non-directory at the top, so creating directory src_top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/appDir.ml to src_top
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/bufferPool.ml to src
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/bufferPool.mli to sr
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/buffering.ml to src_
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/buffering.mli to src
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/cache.ml to src_top_
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/cache.mli to src_top
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/cacheData.ml to src_
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/cacheData.mli to src
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/concurrentGlobal.ml
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/config.ml to src_top
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/context.ml to src_to
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/dbCache.ml to src_to
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/dbCache.mli to src_t
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/drive.ml to src_top_
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/dune to src_top_dir
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/gaeProxy.ml to src_t
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/keyValueStore.ml to
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/memoryCache.ml to sr
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/memoryCache.mli to s
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/mime.ml to src_top_d
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/oauth2.ml to src_top
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/state.ml to src_top_
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/threadPool.ml to src
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/threadPool.mli to sr
Adding /home/hanen/Desktop/google-drive-ocamlfuse-beta//src/utils.ml to src_top_
Categorizing files.
Finding a working MD5 command....
Found a working MD5 command.
Computing results.

SLOC     Directory        SLOC-by-Language (Sorted)
7834     src_top_dir      ml=7834
663      bin              ml=663
211      test             ml=211
76       tools            sh=76
0        doc              (none)
0        top_dir          (none)

Totals grouped by language (dominant language first):
ml:            8708 (99.13%)
sh:              76 (0.87%)
```

```
Total Physical Source Lines of Code (SLOC)              = 8,784
Development Effort Estimate, Person-Years (Person-Months) = 1.96 (23.50)
  (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                      = 0.69 (8.30)
  (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 2.83
Total Estimated Cost to Develop                        = $ 264,557
  (average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
SLOCCount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOCCount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOCCount'."
```

4. Quick Code Exploration

   a) There is a Make module in the concurrentGlobal file.

   b) The file drive.ml appears to have the bulk of logic behind this implementation.

   c) I picked a function from the buffer.mli interface file called `write_to_block` and did a project wide search for it. I found it referenced in the drive.ml file as
   `Buffering.MemoryBuffers.write_to_block` I do not understand why the case is different in the reference. Buffering when calling the function versus buffering in the definition of the file. I would not have made that connection before, but now am aware of it.

   d) Gapi shows up all over the code. It stands for Google API.

4.1.2 Pseudo-code

4.1.2.1 Class: appDir

1. Getters and Setters

   a) `config_path: string`

   b) `data_dir: string`

   c) `cache_dir: string`

   d) `log_dir: string`

   e) `state_path: string`

   f) `app_log_path: string`

     g) `curl_log_path: string`

2. `function xdg_data_home`
   Try to get the value of the environment variable `XDG_DATA_HOME` and return
   that value.
   If exception thrown trying to get this value, return that it was not found.

3. `function xdg_config_home`
   Try to get the value of the environment variable `XDG_CONFIG_HOME` and return
   that value.
   If exception thrown trying to get this value, return that it was not found.

4. `function xdg_cache_home`
   Try to get the value of the environment variable `XDG_CACHE_HOME` and return
   that value.
   If exception thrown trying to get this value, return that it was not found.

5. `function get_config_path`
   input: `config_path: string, xdg_base_directory: boolean,`
   `base_dir: string, fs_label: string`
   if `config_path` is not empty, then return `config_path` and false (not base
   directory)
   else if `xdg_base_directory` is true, then make directory `xdg_config_dir` and
   return `xdg_config_path` and true (is base directory)
   else if `xdg_config_path` exists, then return `xdg_config_path` and true (is base
   directory)
   else return `default_base_dir` + `fs_label` + "config" if `base_dir` is empty or
   `base_dir` + `fs_label` + "config"| if `base_dir` is not empty and false (not base
   directory)

6. `function create`
   input: `config: ConfigFileStore.data, config_path: string,`
   `base_dir: string, fs_label: string, xdg_base_directory: boolean`
   set `data_dir` to `config.Config.data_directory` if `config.Config.data_directory`
   is not empty
   set `data_dir` to `xdg_data_home` + "gdfuse" + `fs_label` if `xdg_base_directory`
   is true
   otherwise, set `data_dir` to `default_base_dir` + `fs_label` if `base_dir` is
   empty or set `data_dir` to `base_dir` + `fs_label` if `base_dir` is not empty
   set `cache_dir` to `config.Config.cache_directory` if `config.Config.cache_directory`
   is not empty
   set `cache_dir` to `xdg_cache_home` + "gdfuse" + `fs_label` if `xdg_base_directory`
   is true
   otherwise, set `cache_dir` to `data_dir` + "cache"
   set `log_dir` to `config.Config.log_directory` if `config.Config.log_directory`
   is not empty

     set `log_dir` to `cache_dir` + "log" if `xdg_base_directory` is true
     otherwise, set `log_dir` to `data_dir`
     set `state_path` to `data_dir` + "state"
     set `app_log_path` to `log_dir` + "gdfuse.log"
     set `curl_log_path` to `log_dir` + "curl.log"
     return `config_path`, `data_dir`, `cache_dir`, `log_dir`

7. `function create_directories`
   input: `app_dir`
   make directory for `data_dir` from `app_dir`
   make directory for `cache_dir` from `app_dir`
   make directory for `log_dir` from `app_dir`

4.1.2.2   Class: buffering

1. `function`
   pseudo-code

4.1.2.3   Class: bufferPool

1. `function create`
   input: `pool_size: integer`, `buffer_size: integer`
   set `max_buffers` to `pool_size / buffer_size` if `pool_size % buffer_size` = 0
   otherwise, set `max_buffers` to `pool_size / buffer_size + 1`
   return object containing `max_buffers`, `buffer_count`, `buffer_size`, `free_buffers`, `pending_requests`

2. `function acquire_buffer`
   input: `mutex: object`, `condition: object`, `buffer_pool: object`
   try to get a free buffer from the Queue.
   if the Queue is empty, then:
   if `buffer_count` < `max_buffers`, then increase `buffer_count` by 1 and return a new buffer with the buffer's id set to `buffer_count`, mutex set to Mutex.Create(), and condition set to Condition.Create()
   else, increase `pending_requests` by 1 and while `free_buffers` is equal to zero, wait. Once, waiting is done, decrease `pending_requests` by 1 and get buffer.

3. `function release_buffer`
   input: `buffer: object`, `condition: object`, `buffer_pool: object`
   add a buffer to the Queue
   wake a thread up

4.1.2.4   Class: cache

1. `function`
   pseudo-code

4.1.2.5   Class: cacheData

1. `function`
   pseudo-code

4.1.2.6   Class: concurrentGlobal

1. `function`
   pseudo-code

4.1.2.7   Class: config

1. `function`
   pseudo-code

4.1.2.8   Class: context

1. `function`
   pseudo-code

4.1.2.9   Class: dbCache

1. `function`
   pseudo-code

4.1.2.10   Class: drive

1. `function`
   pseudo-code

4.1.2.11   Class: gaeProxy

1. `function`
   pseudo-code

4.1.2.12   Class: keyValueStore

1. `function`
   pseudo-code

4.1.2.13   Class: memoryCache

1. `function`
   pseudo-code

4.1.2.14   Class: mime

1. `function`
   pseudo-code

4.1.2.15   Class: oauth2

1. `function`
   pseudo-code

4.1.2.16   Class: state

1. Getters and Setters

   a) `auth_request_id: string`
   b) `auth_request_date: GapiDate.t`
   c) `refresh_token: string`
   d) `last_access_token: string`
   e) `access_token_date: GapiDate.t`
   f) `saved_version: string`

2. `function empty`
   set `auth_request_id`, `refresh_token`, `last_access_token`, and `saved_version`
   to empty string.
   set `auth_request_date` and `access_token_date` to GapiDate.epoch.

3. `function of_table`
   input: `table: object`
   get each class variable from the table.

4. `function to_table`
   input: `data: object`
   add each class variable to the table.

4.1.2.17   Class: threadPool

1. `function`
   pseudo-code

4.1.2.18   Class: utils

1. function `get_thread_id`
   return id of current thread.

2. function `try_finally`
   input: `f: lambda`, `finally: lambda`
   try to run f and finally and return the result.
   if that is unsuccessful, raise an exception.

3. function `with_in_channel`
   input: `path: string`, `f: lambda`
   try to run the input f and `close_in` and return the result.
   if that is unsuccessful, raise an exception.

4. function `with_out_channel`

5. function `log_message`
   input: `format: object`
   if verbose is true, then use ifprintf to print full log message.
   if verbose is false, then use fprintf to print simple log message.

6. function `log_with_header`

7. function `log_exception`

8. function `with_lock`

9. function `try_with_m`
   input: `f: lambda`, `handle_exception: lambda`, `s: object`
   try to run the provided function f with s as a parameter. if this fails, run
   `handle_exception` with the exception e and the provided s argument as pa-
   rameters.

10. function `raise_m`
    input: `m: object`
    raise exception m that is passed in.

11. function `try_finally_m`

12. function `lock`

13. `function unlock`

14. `function with_lock_m`

15. `function safe_find`

16. `function get_from_string_table`

17. `function flags_to_string`
    input: `flags: list`
    map the list of flags provided to it's respective string representation.

18. `function xattr_flags_to_string`
    return "AUTO" if flag is Fuse.AUTO
    return "CREATE" if flag is Fuse.CREATE
    return "REPLACE" if flag is Fuse.REPLACE

19. `function start_browser`
    input: `browser: string`, `url: string`
    run Unix command to launch the browser and visit the provided url.
    if browser is not provided, then the defaults will be "xdg-open", "firefox", "google-chrome", "chromium-browser", and "open".

20. `function with_retry`
    input: `f: lambda`, `label: string`
    recursively try to run f.
    if that attempt is unsuccessful and the `max_retries` has been reached, then raise an exception.
    if that attempt is unsuccessful and the `max_retries` has not been reached, then increase n by 1 and continue to recursively attempt to run f.

21. `function safe_makedir`
    input: `dir: string`
    if directory does not exist, make the directory.

4.1.3   Mount Google Drive on Linux Machine

4.1.3.1   Install OCamlFUSE

4.1.3.2   Allow Permissions through Browser

4.1.3.3   Make Directory and Mount Drive

1. Run commands to install OCamlFUSE

   ```
   sudo add-apt-repository ppa:alessandro-strada/ppa
   sudo apt-get-update
   sudo apt-get install google-drive-ocamlfuse
   sudo google-drive-ocamlfuse
   ```

2. Allow Permissions through Browser Allow gdfuse to see, edit, create, and delete all of your Google Drive files. Select Google account and allow access permissions.
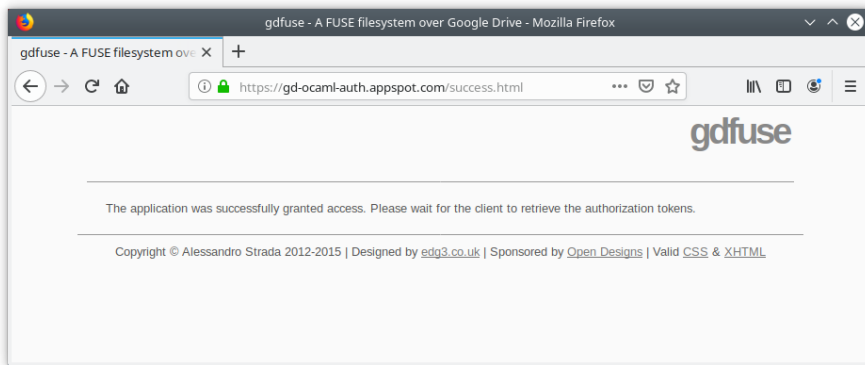

Figure 4.1: Screenshot ocaml7.png.

3. Make Directory and Mount Drive

   ```
   hanen@hanen:mkdir ~/GoogleDrive
   hanen@hanen:google-drive-ocamlfuse ~/GoogleDrive
   hanen@hanen:cd GoogleDrive
   hanen@hanen:ls
   '2019-Hanen-CS 6970'     Misc
   ```

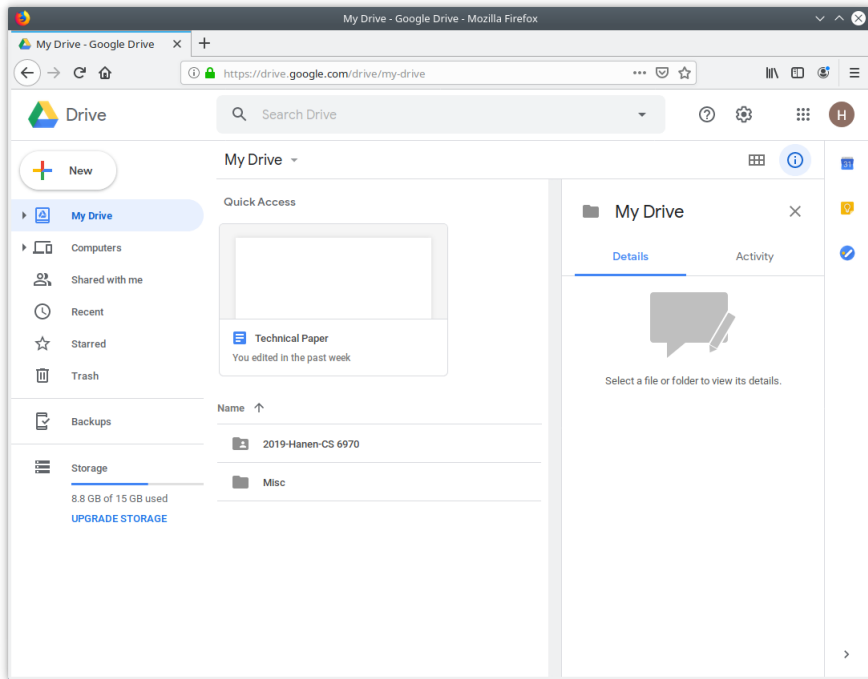   That final output matches the contents of that Google Drive in the browser as seen below. See Figure 4.2

Figure 4.2: Content of GD as seen through the browser.

4.1.4 Mount Google Drive on Android Tablet

CHAPTER 5

Concluding Remarks

To be decided...

APPENDIX A

OCaml Programming Language

A.0.1 OCaml Development Environment

1. OCaml development environment
   (https://github.com/janestreet/install-ocaml)

   a) Install opam
      ```
      sudo add-apt-repository ppa:avsm/ppa
      sudo apt update
      sudo apt install -y opam m4 This was a success!
      ```

   b) Initialize opam
      ```
      sudo opam init -y --compiler=4.07.1
      sudo opam update -uy
      sudo echo $(opam env)
      ```
      This was a success!

   c) Install libraries and tools
      ```
      sudo opam install -y async core js_of_ocaml js_of_ocaml-ppx
      merlin utop ocp-indent
      ```
      This installed so many things and took a while, but was a success!

   d) Test Installation

      ```
      hanen@hanen:~$ git clone
      https://github.com/janestreet/install-ocaml
      Cloning into 'install-ocaml'...
      remote: Enumerating objects: 20, done.
      remote: Counting objects: 100% (20/20), done.
      remote: Compressing objects: 100% (14/14), done.
      remote: Total 58 (delta 10), reused 14 (delta 6),
      pack-reused 38
      Unpacking objects: 100% (58/58), done.
      hanen@hanen:~$ cd install-ocaml/01-hello-world
      hanen@hanen:~/install-ocaml/01-hello-world$ dune build
      ```

```
hello_world.exe
hanen@hanen:~/install-ocaml/01-hello-world$ dune exec
./hello_world.exe
Hello, World
```

e) Run Tests

```
hanen@hanen:~/install-ocaml/01-hello-world$
cd ../02-expect-tests
hanen@hanen:~/install-ocaml/02-expect-tests$
dune runtest
Done: 17/19 (jobs: 1)File "expect_test_example.ml",
line 1, characters 0-0:
diff (internal) (exit 1)
(cd _build/default && /usr/bin/diff -u
expect_test_example.ml expect_test_example.ml.corrected)
--- expect_test_example.ml
2019-06-08 15:37:18.946700012
-0400
+++ expect_test_example.ml.corrected
2019-06-08 15:37:21.494597583
-0400
@@ -2,5 +2,5 @@

let%expect_test _ =
let () = printf "foo" in
-  [%expect {| bar |}]
  +  [%expect {| foo |}]
    ;;
```

The test failed because there is a different in the actual results versus what was expected. The following commands will copy the results into what was expected and show that the tests will pass after that because there will no longer be a difference.

```
hanen@hanen:~/install-ocaml/02-expect-tests$ dune promote
Promoting _build/default/expect_test_example.ml.corrected to
expect_test_example.ml.
hanen@hanen:~/install-ocaml/02-expect-tests$ dune runtest
hanen@hanen:~/install-ocaml/02-expect-tests$ git diff
diff --git a/02-expect-tests/expect_test_example.ml b/
02-expect-tests/expect_test_example.ml
index 75a19d9..9bb1c70 100644
--- a/02-expect-tests/expect_test_example.ml
+++ b/02-expect-tests/expect_test_example.ml
@@ -2,5 +2,5 @@ open! Core
```

```
let%expect_test _ =
let () = printf "foo" in
-   [%expect {| bar |}]
  +   [%expect {| foo |}]
     ;;
```

f) Editor Setup: Installed Visual Studio Code. Install a plugin for OCaml through Visual Studio Code by opening Visual Studio Code, pressing Ctrl+P, and entering `ext install hackwaly.ocaml`[1] into the text field. Once enter is pressed, Visual Studio code will automatically install the OCaml plugin.

A.0.2   Online OCaml Lessons

A.0.2.1   Simple Expressions

A.0.2.2   Imperative Programming

A.0.2.3   Functions

A.0.2.4   New Examples by Hanen

A.0.2.5   Lessons Followed

1. Lesson 1 - Simple Expressions
   This covered computing numeric values, defining strings, working with arrays, string manipulation, and defining and operating on tuples. Tuples can be made up of different data types. There are some functions built-in for tuples that have two elements. The functions presented in the tutorial were `fst` for getting the first element and `snd` for getting the second.

2. Lesson 2 - Imperative Programming
   `let` is the keyword used to set the results of some computation to a named variable. However, once a variable is set to a particular value using the `let` keyword, it cannot be modified to a different value. A compilation error results. The way to get around this constraint is to use the keyword `ref` on the right side of the `let` statement. That reference can then be modified. (`let x = ref 42;;`) `printf` is used similar to C to print formatted text to the terminal. Looping syntax is very similar to many other program languages, except when looping through a series of numbers backwards the word `downto` is used as opposed to `to` in the ascending direction. For comparison of values, the output is a boolean of either `true` or `false`. These greater than, less than, equal, or not equal to comparisons are not limited to only numeric values. The equal

---

[1]Insert citation.

and not equal comparison symbols are similar to VB where a single equal sign represents an equivalence comparison while a less than sign followed by a greater than sign represents a non-equivalence comparison. The only limitation is that there isn't support for comparing values of different types. However, functions like `string_of_int` can be used to convert an integer to a string so that it can be safely compared to another string. `if then else` logic is very straight forward. `while` loops logic is also easy to understand and uses a `while do done` structure.

3. Lesson 3 - Functions
   Functions can be defined in one line using the `let` keyword very similar to defining a variable, but it takes arguments. One argument can be provided or several arguments in a tuple. Calling these functions is exactly the same as all other programming languages. Multiple values can be returned from a function by returning a tuple. Defined functions can also be called in a partial manner. This almost seems like extension methods from the C# world. A function that takes two arguments can be called with only one argument. However, it takes into account the value present at the time in which its called and uses that as the second parameter. `let mul x y = x * y`
   `let double = mul 2`
   `double 8` Anonymous functions are lambda expressions. They are functions defined without a name. These are useful for generating inline functions to be passed as a parameter to a function. In this case, they do not need to be assigned an identifier. Functions such as `List.map` and `List.fold_left` are useful for combining the power of anonymous functions and iterators to get a task done efficiently by iterating over a list and performing an operation on each value as the iterator visits each element.

4. This was a really great way for me to get my feet wet with the OCaml Programming Language. Before this course, I had not heard of this language before and had not tried to use it. I needed a tutorial like this one in order to understand the language better.

A.0.3   Adjust OCaml Example Projects

A.0.3.1   Go Fish

1. Original Source Code - I found the code for this game on Rosetta Code under the OCaml implementation. The game play is based on one player and an AI player who is automated through the back-end randomization functionality. The user chooses a card and asks the AI player if it has that card. If the AI player does, then it must give it. If the AI player does not, then the user must pick up a card from the deck. The player who loses their entire hand of cards first wins. As the code is currently written, the players are named "a" and "b".

There is not a way to change that. Also, the user picks the card to ask about by typing in a number between a provided range which corresponds to the cards left in the player's hand.

2. Allowing For Choosing Players' Names - After wrestling with the code and trying to better familiarize myself with OCaml, this ended up being a pretty straight forward task. It took much longer since I made several failed attempts along the way trying to understand how OCaml projects are structured. The following code ended up being the only piece I needed to change.

From this:

```
(try
  if Random.bool()
      then make_turn "a" "b" player_a player_b
      else make_turn "b" "a" player_b player_a;
with Exit -> ());
```

To this:

```
(try
  if Random.bool()
      then make_turn Sys.argv.(1) Sys.argv.(2) player_a player_b
      else make_turn Sys.argv.(2) Sys.argv.(1) player_b player_a;
with Exit -> ());
```

This is how it was executed through the terminal:

```
hanen@hanen:~/Desktop/gofish$ ocamlc -g -o gofish gofish.ml
File "gofish.ml", line 153, characters 21-36:
Warning 52: Code should not depend on the actual values of
this constructor's arguments. They are only for information
and may change in future versions. (See manual section 9.5)
```

This is a snippet of the game-play:

```
hanen@hanen:~/Desktop/gofish$ ./gofish "dr. mateti" "hanen"

player hanen asked for Sixs
player dr. mateti gives (Six-Clubs)

player hanen asked for Fours
player dr. mateti has no Fours
```

```
(Queen-Clubs), (Jack-Clubs), (Nine-Spades), (Eight-Diamonds),
(Nine-Hearts), (Nine-Clubs), (Queen-Spades), (Queen-Diamonds)
Ranks: Eight, Nine, Jack, Queen
choose from 1 to 4
```

A.0.3.2   Guess the Number

1. Original Source Code - The game-play on this Guess the Number game is very
   simple. A random number generator is used to "think of a number" and the
   player puts in numbers until they guess the number that was chosen at random.
   The user cannot specify the maximum of the range and they are not given any
   feedback on how far off they are from the selected number.

   The following is the current state of the code:

   ```
   #!/usr/bin/env ocaml

   let () =
     Random.self_init();
     let n =
       if Random.bool () then
         let n = 2 + Random.int 8 in
         print_endline "Please guess a number between 1 and 10 excluded";
         (n)
       else
         let n = 1 + Random.int 10 in
         print_endline "Please guess a number between 1 and 10 included";
         (n)
     in
     while read_int () <> n do
       print_endline "The guess was wrong! Please try again!"
     done;
     print_endline "Well guessed!"
   ```

   The following is the current game-play in the terminal:

   ```
   hanen@hanen:~/Desktop/gofish$ ocamlc -g -o guessnum guessnum.ml
   hanen@hanen:~/Desktop/gofish$ ./guessnum
   Please guess a number between 1 and 10 excluded
   1
   The guess was wrong! Please try again!
   2
   The guess was wrong! Please try again!
   3
   ```

```
The guess was wrong! Please try again!
4
The guess was wrong! Please try again!
5
The guess was wrong! Please try again!
6
Well guessed!
```

2. Let User Specify Max of Range - I wanted to give the user the ability to specify what the max of the range should be for the number that is selected randomly for guessing.

The following is the new state of the code after the adjustment:

```
let () =
  Random.self_init ();
  let n =
    if Random.bool () then
      let n = 2 + Random.int ((int_of_string Sys.argv.(1)) - 2) in
      Printf.printf "Please guess a number between 1 and %s excluded
      \n" Sys.argv.(1);
      (n)
    else
      let n = 1 + Random.int (int_of_string Sys.argv.(1)) in
      Printf.printf "Please guess a number between 1 and %s included
      \n" Sys.argv.(1);
      (n)
  in
  while read_int () <> n do
    print_endline "The guess was wrong! Please try again!"
  done;
  print_endline "Well guessed!"
```

The following is the new game-play in the terminal:

```
hanen@hanen:~/Desktop/gofish$ ocamlc -g -o guessnum guessnum.ml
hanen@hanen:~/Desktop/gofish$ ./guessnum 5
Please guess a number between 1 and 5 excluded
2
The guess was wrong! Please try again!
3
Well guessed!
```

3. Warm/Cold Indicator For Guess - I wanted to be able to implement in some logic that would let the user know if they are close or far off with their guess.

Now that I'm allowing the user to choose any number for the max of the range, this is a nice-to-have feature. If they choose their max at 100, it would be helpful to know if they are close or not with their guess.

Lab Reports

B.1   Mount USB Through OTG Using ADB

First, the USB device needs to be authorized based on it's IP address.

```
hanen@hanen:~$ adb tcpip 5555
error: device unauthorized.
This adb server's $ADB_VENDOR_KEYS is not set
Try 'adb kill-server' if that seems wrong.
Otherwise check for a confirmation dialog on your device.
hanen@hanen:~$ adb connect 192.168.1.105:5555
connected to 192.168.1.105:5555
```

However, sometimes that isn't enough, especially once the computer running adb is disconnected from being attached to the tablet over USB cable.

```
hanen@hanen:~$ adb devices
List of devices attached
192.168.1.105:5555      unauthorized
0a655f09        device
```

The tablet was restarted. When prompted to allow for enabling USB debugging, a checkbox was selected to always allow that permission. This fixed the issue once the tablet was unplugged.

```
hanen@hanen:~$ adb kill-server
hanen@hanen:~$ adb start-server
* daemon not running; starting now at tcp:5037
* daemon started successfully
hanen@hanen:~$ adb connect 192.168.1.105:5555
connected to 192.168.1.105:5555
hanen@hanen:~$ adb devices
List of devices attached
192.168.1.105:5555      device
0a655f09        device
```

Now, it is time to partition the drive in order to mount.

```
hanen@hanen:~$ adb -s 192.168.1.105:5555 shell
1|shell@flo:/ $ sm list-disks
disk:8,0
shell@flo:/ $ sm partition disk:8,0 private
shell@flo:/ $
```

Under Settings -> Storage USB, it will show that the portable device (the flash drive) was removed/no longer exists. This is the point in which the tablet restarts. Once it turns back on, the USB Drive has moved out of portable devices and into the collection of internal storage.

B.2   RClone

B.2.1   Linux Machine

Following the tutorial on Rclone, the commands were executed below. [Rcl]

```
hanen@hanen:~$ sudo apt install rclone
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no
longer required:
  libncursesw5 libtinfo5 linux-modules-4.18.0-22-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  rclone
0 upgraded, 1 newly installed, 0 to remove and 9 not upgraded.
Need to get 4,743 kB of archives.
After this operation, 19.7 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu cosmic/universe amd64
rclone amd64 1.41-1 [4,743 kB]
Fetched 4,743 kB in 2s (2,381 kB/s)
Selecting previously unselected package rclone.
(Reading database ... 232149 files and directories currently installed.)
Preparing to unpack .../rclone_1.41-1_amd64.deb ...
Unpacking rclone (1.41-1) ...
Setting up rclone (1.41-1) ...
Processing triggers for man-db (2.8.4-2) ...
hanen@hanen:~$ rclone config
2019/07/12 23:21:03 NOTICE: Config file
"/home/hanen/.config/rclone/rclone.conf" not found - using defaults
No remotes found - make a new one
```

```
   n) New remote
   s) Set configuration password
   q) Quit config
   n/s/q> n
   name> remote
   Type of storage to configure.
   Choose a number from below, or type in your own value
    1 / Alias for a existing remote
      \ "alias"
    2 / Amazon Drive
      \ "amazon cloud drive"
    3 / Amazon S3 Compliant Storage Providers (AWS, Ceph,
   Dreamhost, IBM COS, Minio)
      \ "s3"
    4 / Backblaze B2
      \ "b2"
    5 / Box
      \ "box"
    6 / Cache a remote
      \ "cache"
    7 / Dropbox
      \ "dropbox"
    8 / Encrypt/Decrypt a remote
      \ "crypt"
    9 / FTP Connection
      \ "ftp"
   10 / Google Cloud Storage (this is not Google Drive)
      \ "google cloud storage"
   11 / Google Drive
      \ "drive"
   12 / Hubic
      \ "hubic"
   13 / Local Disk
      \ "local"
   14 / Microsoft Azure Blob Storage
      \ "azureblob"
   15 / Microsoft OneDrive
      \ "onedrive"
   16 / Openstack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
      \ "swift"
   17 / Pcloud
      \ "pcloud"
   18 / SSH/SFTP Connection
      \ "sftp"
```

```
19 / Webdav
   \ "webdav"
20 / Yandex Disk
   \ "yandex"
21 / http Connection
   \ "http"
Storage> drive
Google Application Client Id - leave blank normally.
client_id>
Google Application Client Secret - leave blank normally.
client_secret>
Scope that rclone should use when requesting access from drive.
Choose a number from below, or type in your own value
 1 / Full access all files, excluding Application Data Folder.
   \ "drive"
 2 / Read-only access to file metadata and file contents.
   \ "drive.readonly"
   / Access to files created by rclone only.
 3 | These are visible in the drive website.
   | File authorization is revoked when the user deauthorizes
   the app.
   \ "drive.file"
   / Allows read and write access to the Application Data folder.
 4 | This is not visible in the drive website.
   \ "drive.appfolder"
   / Allows read-only access to file metadata but
 5 | does not allow any access to read or download file content.
   \ "drive.metadata.readonly"
scope> 1
ID of the root folder - leave blank normally.  Fill in to access
"Computers" folders. (see docs).
root_folder_id>
Service Account Credentials JSON file path  - leave blank normally.
Needed only if you want use SA instead of interactive login.
service_account_file>
Remote config
Use auto config?
 * Say Y if not sure
 * Say N if you are working on a remote or headless machine or
 Y didn't work
y) Yes
n) No
y/n> y
```

This will pop up an internet browser which will ask for permissions as illustrated in Figure B.1. Once the Allow button is clicked, a Success screen will display as illustrated in Figure B.2.
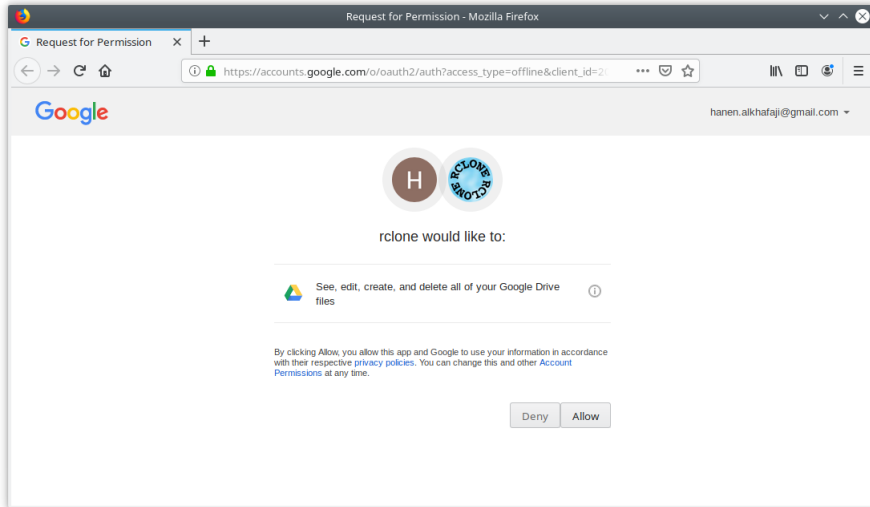

Figure B.1: First RClone Screenshot.

```
If your browser doesn't open automatically go to the following
link: http://127.0.0.1:53682/auth
Log in and authorize rclone for access
Waiting for code...
Got code
Configure this as a team drive?
y) Yes
n) No
y/n> n
--------------------
[remote]
client_id =
client_secret =
scope = drive
root_folder_id =
service_account_file =
token = {"access_token":"ya29.GltEB013ECs9MfyyGMcUfoSt_zNOt5
jhU0bd7iON_VHep0cqyIVXC211TkJ1hdyKFEPb98rBKznYaGP1aoF4xIkHHBi
5O8Gre6t4VpwVX06wGZkEELEr3sxzeu2L","token_type":"Bearer",
```
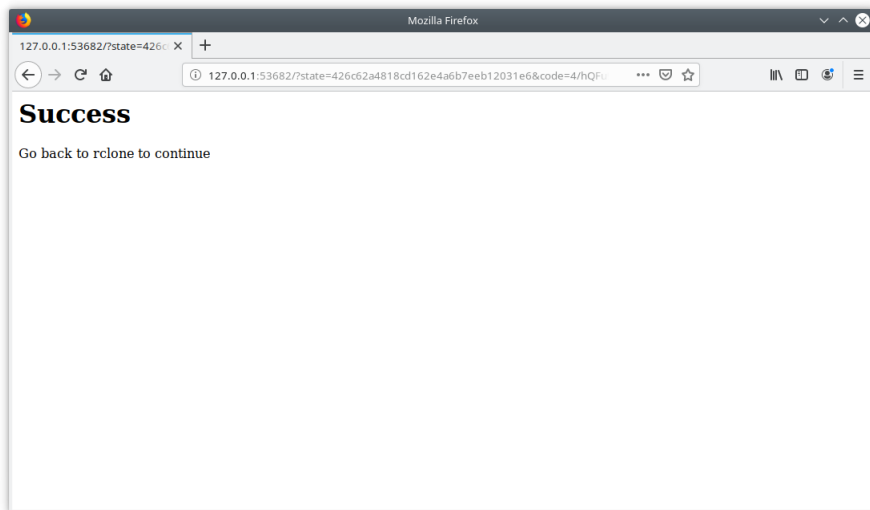
Figure B.2: Second RClone Screenshot.

```
"refresh_token":"1/TQ9fv-iDU4NqZkM5hngLF01tDhRfrtm_XKbDRdfarR8",
"expiry":"2019-07-13T00:24:12.763120806-04:00"}
--------------------
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
Current remotes:

Name                    Type
====                    ====
remote                  drive

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q
hanen@hanen:~$ rclone lsd remote:
        -1 2019-06-19 22:53:37        -1 2019-Hanen-CS-6970
```

```
            -1 2019-06-04 10:46:41        -1 Misc
```

B.2.2 Android Tablet

As in the Linux Machine section above, the same commands are used on the Android device.

In order to run commands, the Termux app must be installed from the Google Play Store.

I struggled for awhile to get past some permission issues until I discovered a very helpful YouTube video [Kal], which helped me overcome this obstacle.

I used the steps from a GitHub page [Cra] to setup rclone on the Android tablet using Termux. It behaved exactly the same way with the exception of a few additional prompts that I was expected to interact with compared to the Linux machine. I responded to these additional prompts by simply pressing Enter without providing any input. This results in the application using the defaults. The following are the prompts I got before the browser popped up asking for permissions:

```
service_account_credentials
auth_owner_only
use_trash
skip_gdocs
skip_checksum_gphotos
shared_with_me
trashed_only
formats
export_formats
import_formats
allow_import_name_change
use_created_date
list_chunk
impersonate
alternate_export
upload_cutoff
chunk_size
acknowledge_abuse
keep_revision_forever
size_as_quote
v2_download_min_size
pacer_min_sleep
pacer_burst
server_side_across_configs
```

B.3   ODrive

B.4   InSync

B.5   X-Plore

B.6   Write FileSystem with FUSE

B.7   ES File Explorer

B.8   Mounting using SSHFS

APPENDIX C

Meeting Minutes

C.1 5/14/2019

This initial meeting of the summer involved deciding on what project would be worked on for this independent study course. Dr. Mateti presented two topics we had discussed in meetings prior to this after we had discussed other options and narrowed them down to a project dealing with Cloud Computing or a project dealing with Cloud Storage. Both projects would involve Android. The Cloud Computing topic would be one in which I would research different cloud computing approaches. The Cloud Storage topic would involve implementing a solution for mounting different cloud storage folders onto an Android device at the operating system level. There are many apps available that will do this very thing on the application level, where a user interacts with their cloud storage through the application itself, but cannot access that mounted storage from any other app on the device. Dr. Mateti showed me some apps like that on his phone which included ES File Explorer and X-plore. The goal would be mounting the storage at the OS level so that the user can access it from any app as if it were simply just another folder on their Android file system.

I decided to select the topic of Cloud Storage, because I really wanted to implement something this summer and this sounded both interesting and personally useful. Once I selected the topic I wanted to focus on, Dr. Mateti elaborated further on expectations and next steps. We will meet in his office at 6pm on Tuesdays or Thursdays on an as-needed basis. In the meantime, I will be uploading my meeting minutes and technical document drafts to a shared Google Drive folder that Dr. Mateti has been given access. I will, also, be organizing the tasks needed to accomplish this project using an application called Trello. Dr. Mateti has been given access to this board on Trello. The technical document will be around 50 pages once it is complete and will use Software Engineering principles. I was tasked to explore open source options for mounting Google Cloud and/or Firebase. Other things to read about and look into included Fuse and SSHFS. The plan will be to implement any solution on a Linux PC first. Then, when it is working to our satisfaction, it will be ported over to Android.

Dr. Mateti also provided me with an Android tablet to be used for this course. I will be bringing back the other tablet I borrowed in our next meeting.

C.2   6/4/2019

Dr. Mateti and I met to primarily discuss the development plan for my project for the summer. The goal for the summer is to be able to demonstrate mounting Google Drive at the operating system level on an Android device. The secondary goal is to try the same for another cloud storage provider, but we will determine later if we have time for that.

On KDE, I need to install GDrive (KIO). There is an authentication error that appears that can be resolved by going into System Settings and adding an Online Account for Google Drive. There was also another error that could be resolved by going into Personal Settings and enabling the KDE Wallet. If all else fails, remove the account under System Settings and add it back.

Dr. Mateti provided some helpful hints on using Latex such as for using a typewriter script to distinguish a certain chunk of text from the remaining report. beginverbatim terminal output endverbatim was another helpful tip.

Dr. Mateti also provided me with a usb input cable that I will use to create a DIY OTG cable, so that I can plug a USB into my tablet. I will plug the USB into my tablet and mount it using the code I develop.

Some action items that Dr. Mateti sent me away with were:

1. Put together a Development Plan document.

2. Revisit the init lab from the Android Security and Internals course from last semester.

3. Browse the OcamlFUSE code and determine comfort level with using it. Do lab report.

4. Rclone might be able to do mounting at the OS level and can handle more cloud storage providers. Do lab report. Include SLOC count of source code.

5. Odrive. Do lab report.

6. InSync. Try the free trial. Do lab report. Do lab report on mounting a USB drive.

C.3   6/11/2019

Dr. Mateti provided me with a tablet that has already been rooted. This will help me make progress without the risk of bricking the device like I did before. Dr. Mateti reviewed his feedback to my OCamlFUSE lab report. He shared a Latex format he would like me to use for my final report. He asked me to mount a usb drive using an OTG cable. He, also, asked that I make some changes to an existing OCaml project. The plan is to focus on these two items. The feedback given by Dr. Mateti was uploaded on Google Drive. I will need to upload the Latex files he provided and fill things in based on his suggestions.

APPENDIX D

Acronyms

OS   operating system

Bibliography

[Cra]   Nick Craig-Wood. *rclone on Android with Termux*. URL: `https://github.com/ncw/rclone/wiki/rclone-on-Android-with-Termux`.

[Kal]   KaliTut. *how to Install Termux On Android and fix permission denied*. URL: `https://www.youtube.com/watch?v=jGtbeWCwus0`.

[Lyn]   Kongens Lyngby. *Laursen's XeLaTeX thesis template*. [Online; accessed 30-June-2019]. URL: `https://www.overleaf.com/project/5d2403bf45f04e303b8ff383`.

[Rcl]   Rclone. *Google Drive*. URL: `https://rclone.org/drive/`.

[Str]   Alessandro Strada. *FUSE filesystem over Google Drive*. URL: `https://github.com/astrada/google-drive-ocamlfuse`.

[Tre]   Trello. *Android Project Trello Board*. URL: `https://trello.com/b/Svn8i351/android-project`.