

Systems Software Report CA1

DT288

BSc in Computer Science

**Byron Farrell**

**C16503843**

School of Computer Science

TU Dublin – City Campus

**10/03/2020**

**Table of Contents**

*Functionality Checklist* 3

*Feature 1 - System Architecture including makefile* 4

*Feature 2 - Daemon (Setup/ Initialisation/ Management)* 4

*Feature 3 - Daemon (Implementation)* 4

*Feature 4 - Backup Functionality* 5

*Feature 5 - Transfer Functionality* 5

*Feature 6 - Lockdown folder for Backup / Transfer* 5

*Feature 7 - Reporting (IPC)* 5

*Feature 8 - Logging and Error Logging* 5

*Conclusion* 6

# *Functionality Checklist*

|  |  |  |
| --- | --- | --- |
| ***Feature*** | ***Description*** | ***Implemented*** |
| F1 | System Architecture including makefile | Yes or No |
| F2 | Daemon (Setup/Initialisation/Management) | Yes or No |
| F3 | Daemon (Implementation) | Yes or No |
| F4 | Backup Functionality | Yes or No |
| F5 | Transfer Functionality | Yes or No |
| F6 | Lockdown folder for Backup / Transfer | Yes or No |
| F7 | Reporting (IPC) | Yes or No |
| F8 | Logging and Error Logging | Yes or No |

Have you included a video demo as part of the assignment: Yes or No

Link to Video: please paste link here

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Byron Farrell

10/03/2020

# *Feature 1 - System Architecture including makefile*

Detailed description of the system architecture choices made.

How Separation of Concerns (SoC) and Single Responsibility Principle (SRP) was followed.

Architecture Diagram.

# *Feature 2 - Daemon (Setup/ Initialisation/ Management)*

Detailed description of the daemon setup:

Startup Script and init process

Daemon control options

To create the daemon process I first invoked the fork() system call which created a new child process from my current “main” process. I then killed the parent process which created an orphan process (a process with no parent). Next I made the process its own session leader this will detach the process from its current TTYL meaning I can no longer interact with the process from the terminal window i.e. view or send data through my current terminal session. I then changed the processes current directory to “/” also know as root. The I set the file creation mode of the process to 0777 (rwx) using the umask system call. Finally I close all open file descriptors.

# *Feature 3 - Daemon (Implementation)*

# *Feature 4 - Backup Functionality*

Detailed description of the backup implementation

# *Feature 5 - Transfer Functionality*

The transfer feature is achieved by running the manager executable I created for the interacting with the daemon and passing it the “transfer” argument. This will send a string with the value “transfer” to the daemon process using a message queue. The daemon will read the incoming message and send it to a message handler function. That function will then invoke the transfer function which will create a sub process using execlp() with the command cp. This will copy the contents from the internal site to the live site. Before the transfer function is invoked the lock\_file(char \*file\_path) function is called and the live site is locked from all users. When the process is finished the live site is unlocked. Also the transfer will not happen if there are any errors. If there is an error the transfer will fail and the error will be logged using syslog().

# *Feature 6 - Lockdown folder for Backup / Transfer*

Detailed description of the lockdown functionality/implementation

# *Feature 7 - Reporting (IPC)*

Detailed description of how child processes communicate success/failure of tack to be completed to parent process etc….

# *Feature 8 - Logging and Error Logging*

Detailed description of the error and logging functionality included in the code solution.

# *Conclusion*

Summary of the implementation and achievement