

Architecture, Operating Systems & Networks Report CA2

TU259

**Borui Cai**

**D21127091**

School of Computer Science

TU Dublin – City Campus

**2022/3/27**

**Table of Contents**

*Functionality Checklist* 3

*Feature 1 - Script Architecture* 4

*Feature 2 – Backup Functionality* 4

*Feature 3 – Transfer Functionality* 4

*Feature 4 – Lockdown Folder* 5

*Feature 5 - System Health* 5

*Feature 6 – Logging and Auditing* 5

*Conclusion* 6

# *Functionality Checklist*

|  |  |  |
| --- | --- | --- |
| ***Feature*** | ***Description*** | ***Implemented*** |
| F1 | Script Architecture | Yes or No |
| F2 | Backup Functionality | Yes or No |
| F3 | Transfer Functionality | Yes or No |
| F4 | Lockdown folder for Backup / Transfer | Yes or No |
| F5 | System Health and Scheduled Task | Yes or No |
| F6 | Logging and Auditing | Yes or No |

Have you included a video demo as part of the assignment: 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:

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

Borui Cai

2022/3/27

# *Feature 1 - Script Architecture*

Detailed description of the script architecture choices made.

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

Architecture Diagram.

All functions are separated into different scripts, and bring together with UI in the main.sh script.

Coupling is reduced by providing config files.

File Structure:



# *Feature 2 – Backup Functionality*

Detailed description of the backup process and how it was implemented

Back up is done by zipping files and then named by type and date.

Key code is following:

sudo zip -r ${backup}/$1$(date +%F).zip ${source}

# *Feature 3 – Transfer Functionality*

Detailed description of the transfer process and how it was implemented

Key code:

sudo rsync -avz ${source}/ ${dest} --delete

Destination folder is synced with the source only by modifying and deleting files.

*Feature 4 – Lockdown Folder*

Detailed description of how the folder permissions were changed to lock the folders down during the backup/transfer process.

Key codes:

sudo chmod -R 444 $source

sudo chmod -R u=rwX,g=rwX,o=rX $source

Changing accessibility to read only before sync.

Changing back after according to whether filetype is directory or file.

# *Feature 5 - System Health*

Detailed description of the system health implementation and how the report was generated. The description of how the scheduled task was setup should be detailed in this section.

Append the output of top command to log file using:

top -n 1 >> config/top.log

Backup and health log are break into different scripts, thus can easily be output to the tasks config file.

Then, tasks in the config file are added using crontab command.

service cron start

echo -e "00 21 \* \* \* .$(pwd)/backup.sh live\n15 21 \* \* \* .$(pwd)/backup.sh\n30 21 \* \* \* .$(pwd)/healthLog.sh" > config/cron.txt

crontab config/cron.txt

# *Feature 6 – Logging and Auditing*

Describe all logging tasks performed in this section.

All logs output to AuditLog.txt, using following codes:

loc=$(cat config/config.txt | sed -n '/^intranet:.\*$/p' | tr -d '\r' | awk '{split($0,a,":"); print a[2]}')/

sudo auditctl -w $loc -p rwxa

sudo ausearch -f $loc | aureport -f -i > config/AuditLog.txt

# *Conclusion*

Summary of the implementation and achievement

All functionalities are broken down into minimal module scripts. All scripts can run standalone, thus allowing easy to reuse.

Config files are used to eliminate the changing of source codes for different user settings.

UI Menu is stored in config file thus allow easy editing.