

File Backup

PYTHON PROJECT | CPRG-260

Amy Leung
SAIT

Table of Contents

PRE-PROCESS.....	2
Decision Making Process	2
Cite Resources	2
Analysis	2
Scope	2
Analysis	3
Functionality	3
Inputs	3
Outputs.....	3
Constants	3
Formulas	3
Algorithm	4
Test data	4
DURING PROCESS.....	5
POST-ANAYLSIS.....	9
1. SWOT	9
2. Document what part of the initial requirement analysis you missed, therefore needed to go back. Why did you miss that part of the problem? –.....	9
3. Document what you would do differently next time and why you would change it –.....	10
4. What was your post-understanding of the project –	10

PRE-PROCESS

Decision Making Process

Cite Resources

Backups
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/4/html/introduction_to_system_administration/s2-disaster-backups-types
https://searchdatabackup.techtarget.com/feature/Full-incremental-or-differential-How-to-choose-the-correct-backup-type
https://docs.python.org/3/library/shutil.html
https://docs.python.org/2/library/shutil.html
Log files
https://docs.python.org/2.6/library/logging.html
https://docs.python.org/3/howto/logging.html
https://docs.python.org/2/library/logging.html
Errors / Exceptions
https://docs.python.org/3/library/exceptions.html

Analysis

I first searched online to find modules that I thought would be useful and were needed to understand the project. I have spent a significant amount of time in class on python, c, and sql code making it somewhat easier to figure out what I thought was necessary.

Scope

Backups a chosen directory/file from input, to a location keeps a log file (one text, one sql) of the activity of the program.

Analysis

Functionality

The user will run the program, then a menu will appear with the options to backup and view log files(sql or code). If the user chooses to then input entered using absolute path will be used to backup the files to a destination. Then the program will concatenate the file in an outfile for logs. The program will also put the log into an entry for the sql database.

Inputs

menu_Choice (python - String)
backup (python - String)
viewLog (python - String)
counter (python - int)
max_Count (python - int)
date (python - int)
log_txt (python + bash + sql)

Outputs

Finish_program_run (Python - string)
Location of where to find the log files (Python - string)
Logfile (Python - String)

Constants

Finish_Program (Python)
outfile (python)
HOME (python - string)
date (python – using module)
logfile (python + bash + sql)

Formulas

-

Algorithm

```
Menu prompt: enter user choice
  If Choice == backup
    Directory/file User input
    Date
    For loop to copy file
    Log outfile
  Elseif Choice == view log file
    Choice == .txt
    Print txt log file
    Choice == .sql
    Print txt sql file
    Connector.
    For loop to split data.
  Else choice == exit
    Close program
```

Test data

User inputs choice

User inputs directory/file to be copied

Program displays the path

Program copies the file(s)/directory to a specific path

Program creates an outfile log text

Program creates an outfile log sql

User inputs view log file

User inputs which log file to view

Program prints the path where to find the log text file

Program prints Log text file

Program prints the path where to find the log sql file

Program prints the log sql file.

DURING PROCESS

Date: June 23, 2020

Start Time: ~9:00AM

Stop Time: 9:47AM

Agenda:

- Received the python project during class

Date: June 25, 2020

Start Time: 1:22PM

Stop Time: 1:45PM

Agenda:

- Asked about the technical parts about the project
- Was having trouble understanding and deciding

Date: July 28, 2020

Start Time: 4:15PM

Stop Time: 5:36PM

Agenda:

- Imported a few modules(sys, time, datetime)
- Started by making a date with the constant variable DATE and used .strftime() from a previous module
- Created a few different functions that I think I will need for my project:
 - menu_exit() – to exit menu
 - backup() – to do the backup
 - log_file() – to do the logging
 - main_menu() – creates a menu to access the options
- finished the skeleton of the project
- finished the choice for the main_menu and the menu_exit()function

Date: July 31, 2020

Start Time: 12:13PM

Stop Time: 12:15PM

Agenda:

- Asked kam about the project because I was having trouble with my USB connection.
- Turns out that there is a problem with the VM so my project cannot use a USB as I initially intended.
- Instead of a USB, I'm creating a directory named "backup" that will be the destination

Date: August 01, 2020

Start Time: 12:30 PM

Stop Time: 5:15 PM

Agenda:

- Read up about the shutil and logging library on docs.python.org – decided that using this library would be best to copy files, and create a log file.
- Imported 3 modules(shutil, logging, sys)
- Added more details to the backup() function created a source/destination and made a for loop to copy files
- Decided to add a try/except/finally block to the backup()
- I used shutil to do the copy stuff <https://docs.python.org/3/library/shutil.html>
- I did 2 tests. The first ended up blank. The second is fixed because I mixed shutil.copy(backup_path, _) it should have been shutil.copy(backup_path, backup_destination)
- I lengthened the log function to actually do the logging I read up about the library in the <https://docs.python.org/3/library/logging.html> and tested twice.
 - There was an error because I didn't use the proper format, so pycharm helped me to suggest fixing it, I needed to add the % and the () because it was incomplete.
- Overall: a lot of progress done, next time I need to create a sql database and create a table and implement it in my python. Also need to fix the logging

Date: August 03, 2020

Start Time: 9:03 PM

Stop Time: 11:35 PM

Agenda:

- Fixed the logging. I misspelled logging. I spelled logging instead changed it to make it correct
- Created a log_choice function – reason is because I wanted to return to the log menu instead of the main_menu.

- Created a few variables to print the returning to menu and used .upper() transliteration
- Imported the sql connector in python and the error from the sql.connector
- Created the sql database(FinalProject) created a table called(Backupsqlllog) populated with 2 data types, date and VARCHAR
- Made 2 different functions both are sql. 1. View the sql log file 2. Inserts a query into the table if backup Is successful
- Used a dictionary to make a connection to the database
- Ran the program once on the pycharm and once on terminal to make sure it is how I wanted it to work.
- FINISHED THE SCRIPT

Date: August 05, 2020

Start Time: 9:03 AM

Stop Time: 10:36 AM

Agenda:

- Tried using both the shutil.copy and the shutil.copytree. so I could backup directories and files.
- Added a else statement to my menu so that if any other input that is not a number or one of the input strings(exit/backup/log etc.) then it will repeat the menu
- I used the import errno from lab 5 and used this document to import an OSError the oserror will check if it is a directory if it is not then it will copy a file instead using the shutil.copy <https://docs.python.org/3/library/exceptions.html> notice the notADirectoryError is mainly what im focused on.
- Added the insert_sql in the backup file so that if a file is backed up there is a query in the sql table.
- Finetuned project made sure variables are correct and is working.
- Did 1 more test to check if the file copy will work.

Date: August 05, 2020

Start Time: 11:00 AM

Stop Time: 11:22 AM

Agenda:

- Started on the post-analysis for the project.

Date: August 09, 2020
Start Time: 2:21 PM
Stop Time: 3:30 PM

Agenda:

- Finished the post analysis and the PowerPoint presentation.

Date: August 09, 2020
Start Time: 5:40 PM
Stop Time: 11:10 PM

Agenda:

- Continued with the presentation powerpoint.
- Ran one more time to ensure everything worked.
- Made a video recording for mt project. Posted on youtube.

Date: August 10, 2020
Start Time: 12:00 AM
Stop Time: 12:15 AM

Agenda:

- Had to reupload youtube video because for some reason audio got distorted.
LINK:
https://youtu.be/X3sQ_iGUgV0

POST-ANAYLSIS

1. SWOT

STRENGTHS	WEAKNESSES
<ul style="list-style-type: none">• Can have multiple files of the same program – good if program rollback is needed• User can view the sql log to see when was the last backup• Log file is helpful when you want to see the backup of the file destination and the errors if they occur.	<ul style="list-style-type: none">• Does not compare directories if they are modified or not• Multiple files of the same program – takes up a lot of space
OPPORTUNITIES	THREATS
<ul style="list-style-type: none">• Improve aesthetics• Better timings for the menus• Implement more log entries for failures and errors• Implement a gzip or tar for better convenience• Chance to create a mount for the USB or external drive.	<ul style="list-style-type: none">• Cannot exit from the program if you select backup and then not want to backup• Need to specify backup_destination exactly if not then it will create that path – makes it messy to handle

2. Document what part of the initial requirement analysis you missed, therefore needed to go back. Why did you miss that part of the problem? –

From the beginning I think I missed a few variables like an extra variable to input the log choice. Rather instead of a counter I used an evaluative method to loop through the program. For the output in the middle of doing my project realized it was unnecessary to print the path of the logfile because you could just view it in the project, which I think eliminated redundancy. I definitely think that I did not expand enough on the algorithm part. I should have planned it out more nicely and ended up just writing a part of the program. In the end I split the algorithm part into different functions that would perform the tasks, my planning for algorithm did not take in account method calls. I initially wanted to place the backup destination (USB) in the code itself. However, discussion with Kam confirmed that I would not be able to. The backup to external storage device (HDD, USB) initial planning fell through. however instead I decided that the user (AKA me) could create a directory (“Backup”) to be the backup destination. And rather than print the location of the paths of the logs file I think in the during process realized it would be slightly redundant, so I removed that.

3. Document what you would do differently next time and why you would change it –

I think there should be a way to make another function to do a task in sql instead of recreating the connection and cursor. The function blocks of code could possibly be reduced however I had no idea how to. If I spent a few extra hours researching and reading up on the python documentation I think it would make the program simpler. I think the backup would be nicer to implement if there was an option to tar.gz files so that it can be easier for the user to export, in the future if a usb exists. I would also implement a compare modified file to reduce the amount of overwriting.

4. What was your post-understanding of the project –

Making and creating functions is incredibly helpful. Making functions like `def menu_exit()` is much better than placing those lines in the actual menu. Especially `def backup()`, because it has so many lines of code and comments implemented it is much nicer and better creating the function instead of placing those lines of code into the block menu. Making log files with a basic config, is much simpler than I thought it would be just a little bit of playing around, and I got it to work the way I wanted it to. Researching and spending an hour or two on the python documentation really will help figuring out ways to carry out the functions of the project. For example, `shutil` ended up being indispensable and perfect for copying files. The `logging.basicConfig` was perfect for my needs and I could actually format the document. The last example was that my experience with java and C was perfect for understanding errors, exceptions and if there was one raised, I knew that something I could implement to create an event/activity.