Patrick Swett

10/19/2018

CSS 490

Program 3: Backup

**Application Describe/Usage:**

This application will take the current directory that the “backup.py” file has been placed in and copy everything in that directory and all sub directories into the amazon s3 storage bucket of the user’s choice. The application will only copy over files that are not currently in that storage block or have been modified in some way. If the file is already in the storage and has not been modified since being put in storage the program will not upload that same file again. The program will then terminate, and the user may rerun it at any time. The user can also place the “backup.py” file in another directory and backup that directory too.

**Application Design:**

Layout: This program is split into 3 main blocks of code

***The Connection:*** This first block of the script is in charge of connecting to the amazon s3 storage using the users Access key, Secret key, and Bucket name. This block of code will throw an exception if any of these 3 user placed fields and incorrect and prompt the user to check and re-enter their information.

***The S3 Check:*** This block of code is used to check the current S3 storage bucket to see if a currently selected file is already in the S3 bucket and will also compare the modification date of the currently selected file to the modification date of the same file within the bucket. If the file exists in the bucket and has been recently modified, then it will be updated in the S3 bucket. However, if a filename is found and it has not been modified, then it will not be uploaded.

***The Directory Recursion:*** This block of code recurses into the current directory that the “backup.py” file is in and will visit every file in and under the current directory. This code block will also put these files into the S3 storage after calling the S3 Check block to see if the file already exists in the bucket or if it meets the modification requirements. This block is also responsible for keeping the integrity of the file structure and will make sure that while it is putting files into the bucket, it is keeping the correct local file path and structure.

**Application Testing:**

This program was tested using multiple different methods. The first testing method was just a simple upload to a specified S3 bucket. This method was checked off quickly once it was confirmed that every file was copied over to the S3 bucket and that the file structure was maintained and correct within the S3 bucket. The second test that was conducted on the program was the modification of files. We would attempt to upload a file that already has a copy of itself in the S3 storage and we also confirmed that the file would not be uploaded unless the local directory modification date was after the modification date that was saved to the S3 bucket. There are also a few other less obvious tests that were conducted. The first of these was placing the “backup.py” file into another directory after already uploading files to the S3 storage. We confirmed that this still works and will simply treat the new current directory as an extension of the old current directory. Lastly, we did some simple exception testing with invalid Access keys, Secret keys, and bucket names and created errors to prompt the user when there was an error, or they entered their information incorrectly.