# A Backup Program for Linux and Windows.

## File Backup

Backup the '/home/{user}' file system to the backup storage and to cloud storage. This is a program built to satisfy my own way of working. It may or may not be useful to others. The basic usage is either at the command line or run automatically at some set time. For Linux, this automatic run can be done with either a cron job or using System-D. For Windows, use the built in Task Scheduler program. See (<https://www.howtogeek.com/123393/how-to-automatically-run-programs-and-set-reminders-with-the-windows-task-scheduler/>) for example.

Logging of results is automatic using a database and includes start and stop times, elapsed time, and any errors occurring. Adding the verbose option also sets the logging to verbose mode adding more information on files and directories backed up and errors.

The program is written in Python and is tested on Fedora Linux and Windows 11 using Python versions 3.9, 3.10 and 3.11(beta4) with no errors and approximately 95% coverage.

### Part 1: Backup to external storage

Backup any new and changed files in the global home directory to the backup directory on an external disk drive. A new file is one that has been added or renamed since the last backup. A changed file is defined as one whose modification time stamp is greater than the last stored backup time stamp.

Any new directories needed are created in the backup store and new/changed files are copied to the store.

### Part 2 – Backup to Cloud Storage

*Not yet implemented*

After the external storage is updated, backup to cloud storage using the 'restic' program.

## Usage

At the command prompt, enter backup [args].

Arguments (all are optional, -b is default if none are given):

* -s, --setup: (*Partially Implemented)* Run the setup portion to configure the program. Currently sets the file source and destination directories to default values and enables backup to the external storage device.
* -b, --backup Run the backup portion. This is the default if no other option is included, required if backup is desired when other options are included. It will fail with a message to the console and to the log if no configuration is found.
* -r, --restore (*Not yet implemented) R*estore the previously saved cloud backup

* -t, --test (*Not yet implemented)* Run the backup portion showing what would be accomplished without actually saving anything.
* -v, --verbose Show information as program steps through the file backup process. When verbose mode is selected, the verbose messages are also added to the log.

**Examples**:

[[me@thebes](mailto:me@thebes)]$ backup

Will run the backup program with the previously stored configuration file.

[[me@thebes](mailto:me@thebes)]$ backup -s

Will run the backup program opening the configuration window. When the configuration window is closed, the program will be terminated.

[[me@thebes](mailto:me@thebes)]> backup -s -b

Will first open the configuration window then, when the window is closed, runs the backup.

[[me@thebes](mailto:me@thebes)]> backup -b -v

Will run the backup program with the previously stored configuration file. Additional progress messages will be sent to standard output along with any error messages. The logging subsystem will also save the additional messages.

---

Author: Lorn B Kerr (lornburtkerr at gmail dot com)

Copyright: (c) 2022 Lorn B Kerr

License: MIT License; see file LICENSE