taskrunner.py

Purpose:

This application is used for running commands dispatched to it via the internal os libraries provided in Python. This application allows you to provide threshold for attempting to re-run of the application in case of failure scenarios. This application also ensures singleton process behavior by creating a lock file, such that for every unique dispatch request, the lock file is considered unique and it's existence governs the process execution.

Command-line:

Command line arguments are,

- '--command', help = 'command to run'
- '--lockfile', help = 'path to lock file'; could be in /tmp/
- '--threshold', help = 'failure threshold'
- '--counter-file', help = 'counter filer to keep track of failed attempts'

Execution Workflow:

- Verify the following exist in command options, *Else Exit showing Help.*
    - (ア) Command,
    - (イ) Lockfile
    - (ウ) Threshold & Failure Counter File
- Creating Lock File, Returns STATUS = True or False; *On False Goto Step 7.*
    - (ア) Create a filesystem node (file, device special file or named pipe) at .tmp, *Return False if node already exists. (OSError: [Errno 17] File exists.*

    - (イ) Hard Link (copy) the lock file path to Tmp Node created above. *Return False if Lock File already exists. (OSError: [Errno 17] File exists.*
    - (ウ) Remove the Temp node file created in 2.(ア)
    - (エ) Return True.
- Once Execution is allowed, now remove the Failure counter file.
    - (ア) Check if Threshold is Not 0 and Counter File exists, remove counter file.
- Run the command via os.system dispatch. Log Success.
- Remove the Lock File, *Raises Exception & Exits if the remove is requested on a directory. Corresponds to errno EISDIR.*
- Exit neatly with status 0.
- It seems, lock file creation was not successful, Why :(!!?. Probably because lock file already exists, that means the command is already executing for that particular lock file.
    - (ア) Check Threshold not 0 and counter file exist; for True log error and return back, For false just return
        - If Threshold is lesser than Failure counter file's numeric content, Then rewrite the content of failure counter as 0, return True
    - Else increment the counter in failure counter file and return False.

Failure Counter

Failure Counter is used to check against a threshold, if execution requests for a particular command associated to a particular lock file and counter file, have exceeded more than the threshold provided with failures due to an already existing lockfile (already executing command).

Failure counter is incremented for every failed execution request, and it's not reset to 0 for a successful request.

The purpose of the failure counter is to find out the commands or execution requests which fail for more than a threshold time in the lifetime of a system execution cycle, and such commands or execution requests must be reported.

The most common failure scenario is an already existing lock file, and if the lock file is existing across multiple execution requests, then the command has been executing for a long time or is in a deadlock condition.

It should not be confused and used with commands which fail rarely, and to catch them when they fail **continuously** due to some system failure etc...