Michael Blunt

Startup.sh writeup

Startup.sh is intended to be a vessel for launching all required programs and starting a restart clock. Before this project, I did not know how to use crontabs to launch programs, so for my simplicity and understanding, I only launched one program. This program is not strictly necessary for the program design, but honestly it is less complex to edit than a crontab anyways, so I prefer this method. Here is the basic flow of this our startup sequence.

(System booting up) 🡪 root crontab (/etc/crontab) 🡪 startup.sh 🡪 everything else

Line by Line explanation:

* Changes directories to /home/pi/Programs, not required since we are using absolute paths
  + Obsolete feature left over from the first versions of this program. I have since learned how about absolute versus relative paths, which is a much more reliable method that is not reliant on the user that runs the program.
  + Non-technical explanation:
    - Obsolete feature, it has no impact on current functionality.
    - Was used to specify the “working directory” or the area in which the program ran.
    - Has been replaced by using absolute paths, or using the full system path of a folder from the root directory to execute a program, create a file, etc.
* Launches programs:
  + IMPORTANT: the ampersand after launch commands specifies the programs are to be backgrounded, this is what allows startup.sh to launch multiple programs and not be stuck after launching the first program. DO NOT REMOVE THE AMPERSANDS!
  + Transfer
    - Shellscript, launched with “sh” command
  + Monitor
    - Shellscript, launched with “sh” command
  + Sensor
    - Python program, runs on python3
  + UAH
    - Shellscript, launched with “sh” command
  + Note: any of the programs can be disabled at next reboot by “commenting” them out with a pound character in front of the line (“#”)
* Log:
  + After the programs are launched, the startup.sh program writes to a log file with the current time. This occurs before the restart clock is started, so if the clock fails to start for some unknown reason, no error will be thrown. I am aware of this possibility and have added a failsafe in the monitor program (see startup check writeup in monitor program writeup).
* Restart timer:
  + Very simple, sleeps for 720 minutes (12 hours) and then triggers a restart.