cron scheduler

Dr Peadar Grant

March 15, 2024

Contents

1 Unattended execution

There are many scenarios where a program needs to run unattended:

1. A Python script should be run every hour. It retrieves weather information and stores it in a database management system.
2. A bash script e-mails a report at 5PM each day using the `msmtp` program.
3. A Git repository needs to be `git pull`ed once a day at 6PM.
4. A backup routine should be run every evening at 9PM.
5. A rogue employee wishes to send a mail at 6PM this evening with their resignation and post a tweet disparaging their employer!

1.1 Requirements

We require that our command can run:

1. Without direct intervention at the shell
2. Without any user even logged in at all
3. Also we require that the program(s) will be run as required even if the system is restarted.

The command in question might be:

1. A shell script (in Bash or another language)
2. A program installed from the package manager
3. A program we've written ourselves
4. Any combination thereof that we can write at the shell prompt!

1.2 Program requirements

If a command is going to be run unattended, we need to make sure that:

1. Employs only **standard text-based** input / output
    - Rules out “full screen” terminal-based programs.
2. Runs from start to finish in a relatively **linear path**
3. It **does not prompt** for or require interactive user input. Can sometimes use command-argur to:
    - Supply information that’s normally prompted for
    - Bypass confirmation questions
4. It doesn’t hang (get stuck) in the event of errors.
    - Can cause stuck runs to “pile up” consuming memory.

1.3 Options

Scheduled execution generally falls into two distinct but similar scenarios:

Periodic execution according to a repeating rule.

- Use the `cron` scheduler

One shot execution at some point in the future.

- Use the `at` scheduler
- May also employ `cron` if a rule can be written.

2 cron scheduler

The **cron** scheduler is a standard part of Linux and UNIX.

- It is controlled by means of the `crontab` file.
- Can schedule execution based on:
  - Date(s)
  - Day of week
  - Time (to 1 minute precision)

2.1 cron job

A scheduled task is colloquially referred to as a “cron job”.

We need to know:

1. The command to run
2. The working directory required
3. When the command is to be scheduled for execution at.

Cron jobs normally execute as the user who creates them.

- Some cron jobs will need to run as root.