

Periodic Processes

or, cron-icles of a habitual user

David Morgan

Objective

- automated periodic execution
- of arbitrary specified commands
- at arbitrary specified intervals

Overview

- implemented by cron daemon
- following instructions in "cron table" files
- "crontab" file contents
 - commands to run
 - times to run them

cron daemon

- executed at startup
- then runs continuously

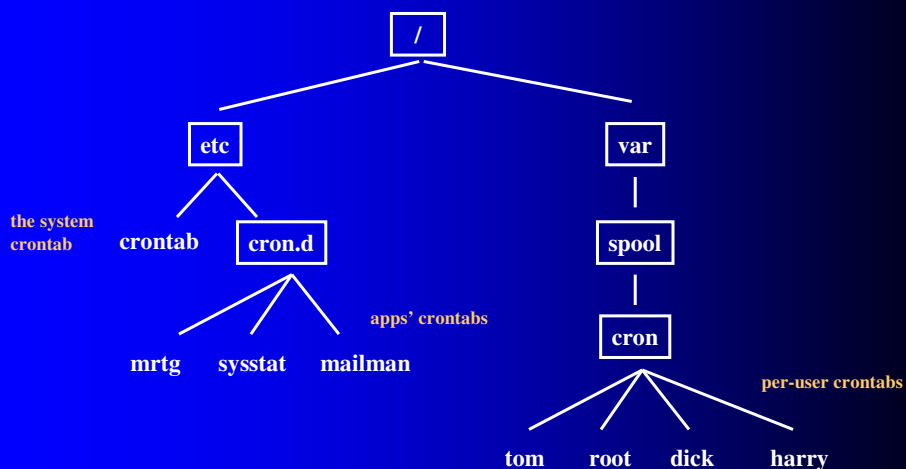
cron operation

- startup
 - reads all crontabs (cron tables)
 - stores them in memory
 - goes to sleep
- every minute
 - wakes up
 - re-stores any changed crontabs
 - executes any tasks scheduled for that minute

crontab locations

- /etc/crontab
 - maintained by system administrator
- /var/spool/cron/<username>
 - per-user crontabs
 - name of crontab file is user's name
 - maintained by individual users
- /etc/cron.d directory
 - auto-installed by apps that want cron service

cron table files



Format of crontab instructions

<when> [<who>] <what>

Example:

```
30 1 * * * root /usr/bin/find /tmp -atime +3 -exec rm -f { } \;
```

when who what

At 1:30 each morning, as root, run the command which removes all files in the /tmp directory that have not been accessed in 3 days

<what>

- anything executable from command line
 - aliases
 - keywords
 - functions
 - builtins
 - executable files (binary and script)

(tip: specify full path for executable files)

<when>

Time fields

minute	0 to 59
hour	0 to 23
day of month	1 to 31
month	1 to 12
day of week	0 to 6 (0 = Sunday)

Time field contents

- star - matches all values
- single integer - matches exactly
- hyphen-separated integer pair - matches the range
- comma-separated list of above - matches everything listed

Examples

10:45am Monday through Friday
45 10 * * * 1-5

every half-hour on Fridays and on the 13th
0,30 * 13 * 5

every half-hour on Mon Tues Wed and Fri
0,30 * * * 1-3,5

every 3 hours on the hour
0 0,3,6,9,12,15,18,21 * * *

every 3 hours on the hour
0 0-21/3 * * *

minute – hour – day of month – month – day of week

as whom are commands executed?

- per-user crontabs executed as "their" user
 - /var/log/cron/root executed as root
 - /var/log/cron/tom executed as tom
- other crontabs' instructions must specify user explicitly, per each command
 - <when> <who> <what>

editing cron tables

- per-user crontabs
 - crontab command
- /etc/crontab
 - any editor
- /etc/cron.d
 - auto-installed, not maintained
(except possibly upon package update)

crontab command

- `crontab <filename>`
installs named file as `/var/log/cron/<username>`
- `crontab -e`
edits user's crontab
- `crontab -l`
lists content of user's crontab
- `crontab -r`
removes user's crontab

Preinstalled directory-span cron jobs

- some distributions install a system crontab that
 - hourly, executes scripts in `/etc/crontab.hourly`
 - daily, executes scripts in `/etc/crontab.daily`
 - weekly, executes scripts in `/etc/crontab.weekly`
 - monthly, executes scripts in `/etc/crontab.monthly`
- affords alternate method to install periodic jobs
 - mere placement in above "path of execution" directories does the job
 - no crontab file to edit

Scheduling by placement

```
CRONTABS(4)                                Crontabs users' Manual                                CRONTABS(4)
NAME
  crontabs - configuration and scripts for running periodical jobs
SYNOPSIS
  run-parts [--list|--test]<directory>
DESCRIPTION
  Crontabs is a historical name for the run-parts script and the system crontab. The run-parts command runs all executable files in the specified directory. The execution of files can be allowed or denied by creating file jobs.allow or jobs.deny which worked similar as other allow/deny config files. The file must be created in the specified
```

```
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
                                excerpt from /etc/crontab
```

Execution schedule, consequently, for an executable in any of these directories:

- /etc/cron.hourly - 1 minute past every hour
- /etc/cron.daily - every 4:02am
- /etc/cron.weekly - every 4:22am on day 0 (Sunday)
- /etc/cron.monthly - every 4:24am on 1st-of-the-month

anacron - makeup for missed jobs

- lists commands with periodicities in /etc/anacrontab
 - "run.parts /etc/cron.daily" with periodicity one day
 - "run.parts /etc/cron.weekly" with periodicity one week
 - "run.parts /etc/cron.monthly" with periodicity one month
- records commands' actual run dates in /var/spool/anacron/
- /etc/cron.whateverly can contain a script for anacron
 - script runs anacron conditionally
 - provided it hasn't been run within its periodicity interval
 - provided the machine is not on battery
 - records the new run date (today) for future reference


at 1-time, non-recurring jobs

```
root@instructor:~# tty;date;at now + 1 minute /dev/pts/2
Sat 07 Aug 2021 09:06:15 AM PDT
warning: commands will be executed using /bin/sh
at> echo "It is getting late." > /dev/pts/2
at> <EOT>
job 52 at Sat Aug 7 09:07:00 2021
[root@instructor ~]#
[root@instructor ~]# printf "echo 'It is later than you think.' > /dev/pts/2\n" | at now + 2 minute
warning: commands will be executed using /bin/sh
job 53 at Sat Aug 7 09:08:00 2021
[root@instructor ~]#
[root@instructor ~]# atq
52      Sat Aug 7 09:07:00 2021 a root
53      Sat Aug 7 09:08:00 2021 a root
[root@instructor ~]#
```

job 52 entered via stdin, interactively

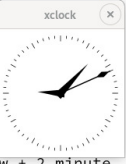
then, job 53 entered via stdin, through a pipe

job queue just before first deadline



at 1-time, non-recurring jobs

```
root@instructor:~# tty;date;at now + 1 minute /dev/pts/2
Sat 07 Aug 2021 09:06:15 AM PDT
warning: commands will be executed using /bin/sh
at> echo "It is getting late." > /dev/pts/2
at> <EOT>
job 52 at Sat Aug 7 09:07:00 2021
[root@instructor ~]#
[root@instructor ~]# printf "echo 'It is later than you think.' > /dev/pts/2\n" | at now + 2 minute
warning: commands will be executed using /bin/sh
job 53 at Sat Aug 7 09:08:00 2021
[root@instructor ~]#
[root@instructor ~]# atq
52      Sat Aug 7 09:07:00 2021 a root
53      Sat Aug 7 09:08:00 2021 a root
[root@instructor ~]# It is getting late. ← following first deadline
                                         first job has executed
```



at 1-time, non-recurring jobs

```
root@instructor:~# tty;date;at now + 1 minute
/dev/pts/2
Sat 07 Aug 2021 09:06:15 AM PDT
warning: commands will be executed using /bin/sh
at> echo "It is getting late." > /dev/pts/2
at> <EOT>
job 52 at Sat Aug 7 09:07:00 2021
[root@instructor ~]#
[root@instructor ~]# printf "echo 'It is later than you think.' > /dev/pts/2\n" | at now + 2 minute
warning: commands will be executed using /bin/sh
job 53 at Sat Aug 7 09:08:00 2021
[root@instructor ~]#
[root@instructor ~]# atq
52      Sat Aug 7 09:07:00 2021 a root
53      Sat Aug 7 09:08:00 2021 a root
[root@instructor ~]# It is getting late.
It is later than you think. ← following second deadline
                                second job has executed
```

