

1.111.4 Automate system administration tasks by scheduling jobs to run Weight 4

Linux Professional Institute Certification — 102

Geoff Robertson <ge@ffrey.com>

Nick Urbanik <nicku@nicku.org>

2005 July

Description of Objective

Candidate should be able to use cron or anacron to run jobs at regular intervals and to use at to run jobs at a specific time. Task include managing cron and at jobs and configuring user access to cron and at services.

Key files, terms, and utilities include:

```
/etc/anacrontab
/etc/at.deny
/etc/at.allow
/etc/crontab
/etc/cron.allow
/etc/cron.deny
/var/spool/cron/*
at
atq
atrm
crontab
```

111.4 Scheduling jobs [4] Key files, terms and utilities

```
crontab          at
/etc/anacrontab  atq
/etc/crontab     /etc/at.deny
/etc/cron.allow  /etc/at.allow
/etc/cron.deny
/var/spool/cron/*
Basically
```

at – Run a command once

cron – Run a command periodically

The at command

at takes a time and a list of commands to run. Any output to STDOUT or STDERR will be mailed to the user running at.

```
$ at 2pm ← \
warning: commands will be executed using /bin/sh\
at> date ←\
at> ^D ←\
job 3 at 2002-05-08 14:00
```

The at command

The current umask, working directory and environment (except for TERM, DISPLAY and _) are saved and restored before running the job (unlike cron).

The commands to run will be read from STDIN or from a file given with **-f** .

Example at time specifications

at allows a *very* flexible time format.

17:36 Run at 5:36pm today or tomorrow.

9pm May 8 Run at 9pm on May 8th.

noon tomorrow Run at 12pm tomorrow.

now + 2 hours Run in 2 hours.

See [at\(1\)](#) for more details.

2 Queued jobs

`atq` lists a user's pending jobs.

```
$ atq ↔ \
A3      2002-05-08 14:00 a gus
```

\$ **atrm B3** ↔ removes the queued job.

\$ **at -c C3** ↔ dumps the job on STDOUT.

```
*2[linecolor=red,angleA=270,angleB=90]->AB [linecolor=red,angleA=270,angleB=90]-
>AC
 crontab
 cron is a daemon that reads everyone's crontab information, spawning new tasks at
 the appropriate times.
```

crontab file Replace your crontab file with *file*.

crontab -l List your crontab.

crontab -r Remove your crontab.

crontab -e Edit your crontab (with \$EDITOR).

6 crontab file format

A sample crontab file:

```
min0 hr7 dom1 monjan wild*\
echo "sleep in, you dont feel so good"\

comment# gratuitous noise\
0 17 * * dowmon,wed,fri      stdinwall%meeting in 5 minutes%\

0 9-18step/2 * * rangemon-fri    env$HOME/bin/cron.bihourly
```

*2commentXLine based, hash comments, ignored blank lines, etc [linecolor=red,angleA=270,angleB=90]->AB [linecolor=red,angleA=270,angleB=90]->AC

*3

minXMinute (0-59)

hrXHour (0-23) [linecolor=red,angleA=270,angleB=90]<-minminX[linecolor=red,angleA=270,angleB=90]

hrhrX

*4

domXDay of month (1-31)

monXMonth (1-12 or jan-dec)

dowXDay of week (0-7 or sun-sat) [linecolor=red,angleA=270,angleB=180]<-domdomX[linecolor=red,angleA=270,angleB=180]

monmonX[linecolor=red,angleA=270,angleB=0]<-dowdowX

*5

stepXStep

wildXWildcard

rangeXRanges

listXLists [linecolor=red,angleA=270,angleB=0]<-wildwildX[linecolor=red,angleA=270,angleB=0]

rangerangeX[linecolor=red,angleA=270,angleB=0]<-dowlistX[linecolor=red,angleA=270,angleB=0]

stepstepX

*6See [crontab\(5\)](#) for:

envXEnvironment variables

stdinXProviding STDIN [linecolor=red,angleA=270,angleB=0]<-envenvX[linecolor=red,angleA=270,angleB=0]

stdinstdinX

cron from root

A few extra issues arise when editing `/etc/crontab` (and similar "system" crontab files):

Don't use `crontab -e`, edit `/etc/crontab` directly.

Don't use `crontab -e`, edit `/etc/crontab` directly.

A new column (after `timespec`, before command) gives the user to the command run as.

Distributions often create directories for "common" frequencies. It usually makes much more sense to place a script in there, rather than adding your own crontab lines.

Debian (for example) runs any scripts in `/etc/cron.{daily,weekly,monthly}` – but these are triggered from normal entries in `/etc/crontab`, so there's no real mystery here.

(Debian specific?) `/etc/cron.d/*` is read in addition to `/etc/crontab` (they also have the extra user field).

anacron

Apparently some people turn their machines off.

If your computer is always turned off at night (for example), then daily jobs which are usually scheduled to run in the wee hours, will never be run. This is a problem.

anacron fixes this by running any missed jobs after a reboot (or other times, like AC-on for laptops).

anacron

Since anacron can't use the crontab files, it has its own simplified `/etc/anacrontab`.

If you only use the standard `/etc/cron.daily`, `monthly`, `weekly`, then no further configuration will be necessary. Otherwise, see [anacrontab\(5\)](#).

Note that the frequency of anacron jobs can only be specified in days.

License Of This Document

Copyright (C) 2005 Nick Urbanik <nicku@nicku.org>

You can redistribute modified or unmodified copies of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.