

CRONTAB(5)

CRONTAB(5)

NAME

crontab - tables for driving cron

DESCRIPTION

A crontab file contains instructions to the cron(8) daemon of the general form: ``run this command at this time on this date''. Each user has their own crontab, and commands in any given crontab will be executed as the user who owns the crontab. Uucp and News will usually have their own crontabs, eliminating the need for explicitly running su(1) as part of a cron command.

Blank lines and leading spaces and tabs are ignored. Lines whose first non-space character is a hash-sign (#) are comments, and are ignored. Note that comments are not allowed on the same line as cron commands, since they will be taken to be part of the command. Similarly, comments are not allowed on the same line as environment variable settings.

An active line in a crontab will be either an environment setting or a cron command. An environment setting is of the form,

```
name = value
```

where the spaces around the equal-sign (=) are optional, and any subsequent non-leading spaces in value will be part of the value assigned to name. The value string may be placed in quotes (single or double, but matching) to preserve leading or trailing blanks. The value string is not parsed for environmental substitutions, thus lines like

```
PATH = $HOME/bin:$PATH
```

will not work as you might expect.

Several environment variables are set up automatically by the cron(8) daemon. SHELL is set to /bin/sh, and LOGNAME and HOME are set from the /etc/passwd line of the crontab's owner. PATH is set to "/usr/bin:/bin". HOME, SHELL, and PATH may be overridden by settings in the crontab; LOGNAME is the user that the job is running from, and may not be changed.

(Another note: the LOGNAME variable is sometimes called USER on BSD systems... on these systems, USER will be set also.)

In addition to LOGNAME, HOME, and SHELL, cron(8) will look at MAILTO if it has any reason to send mail as a result of running commands in ``this'' crontab. If MAILTO is defined (and non-empty), mail is sent to the user so named. If MAILTO is defined but empty (MAILTO=""), no mail will be sent. Otherwise mail is sent to the owner of the crontab.

On the Debian GNU/Linux system, cron supports the pam_env module, and loads the environment specified by /etc/security/pam_env.conf. However, the PAM setting do NOT override the settings described above nor any settings in the crontab file itself. Note in particular that if you want a PATH other than "/usr/bin:/bin", you will need to set it in the crontab file.

The format of a cron command is very much the V7 standard, with a number of upward-compatible extensions. Each line has five time and date fields, followed by a command, followed by a newline character ('\n'). The system crontab (/etc/crontab) uses the same format, except that the username for the command is specified after the time and date fields and before the command. The fields may be separated by spaces or tabs.

Commands are executed by cron(8) when the minute, hour, and month of year fields match the current time, and when at least one of the two day fields (day of month, or day of week) match the current time (see ``Note'' below). cron(8) examines cron entries once every minute. The time and date fields are:

field	allowed values
-----	-----
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

A field may be an asterisk (*), which always stands for ``first-last''.

Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive. For example, 8-11 for an ``hours'' entry specifies execution at hours 8, 9, 10 and 11.

Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: ``1,2,5,9'', ``0-4,8-12''.

Step values can be used in conjunction with ranges. Following a range with ``/<number>'` specifies skips of the number's value through the range. For example, ``0-23/2'' can be used in the hours field to specify command execution every other hour (the alternative in the V7 standard is ``0,2,4,6,8,10,12,14,16,18,20,22''). Steps are also permitted after an asterisk, so if you want to say ``every two hours'', just use ``*/2''.

Names can also be used for the ``month'' and ``day of week'' fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.

The ``sixth'' field (the rest of the line) specifies the command to be run. The entire command portion of the line, up to a newline or % character, will be executed by /bin/sh or by the shell specified in the SHELL variable of the crontab file. Percent-signs (%) in the command, unless escaped with backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input. There is no way to split a single command line onto multiple lines, ala the shell's trailing "\".

Note: The day of a command's execution can be specified by two fields -- day of month, and day of week. If both fields are restricted (i.e., aren't *), the command will be run when either field matches the current time. For example, ``30 4 1,15 * 5'' would cause a command to be run at 4:30 am on the 1st and 15th of each month, plus every Friday.

Instead of the first five fields, one of eight special strings may appear:

string	meaning
-----	-----
@reboot	Run once, at startup.
@yearly	Run once a year, "0 0 1 1 *".
@annually	(same as @yearly)
@monthly	Run once a month, "0 0 1 * *".
@weekly	Run once a week, "0 0 * * 0".
@daily	Run once a day, "0 0 * * *".
@midnight	(same as @daily)
@hourly	Run once an hour, "0 * * * *".

EXAMPLE CRON FILE

```
# use /bin/bash to run commands, instead of the default /bin/sh
SHELL=/bin/bash
# mail any output to 'paul', no matter whose crontab this is
MAILTO=paul
#
# run five minutes after midnight, every day
5 0 * * *      $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * *    $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5   mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun    echo "run at 5 after 4 every sunday"
```

EXAMPLE SYSTEM CRON FILE

```
This has the username field, as used by /etc/crontab.
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs do.
```

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# m h dom mon dow user  command
42 6 * * *      root    run-parts --report /etc/cron.daily
47 6 * * 7      root    run-parts --report /etc/cron.weekly
52 6 1 * *      root    run-parts --report /etc/cron.monthly
#
# Removed invocation of anacron, as this is now handled by a
# /etc/cron.d file
```

SEE ALSO

```
cron(8), crontab(1)
```

EXTENSIONS

When specifying day of week, both day 0 and day 7 will be considered Sunday. BSD and ATT seem to disagree about this.

Lists and ranges are allowed to co-exist in the same field. "1-3,7-9" would be rejected by ATT or BSD cron -- they want to see "1-3" or "7,8,9" ONLY.

Ranges can include "steps", so "1-9/2" is the same as "1,3,5,7,9".

Months or days of the week can be specified by name.

Environment variables can be set in the crontab. In BSD or ATT, the environment handed to child processes is basically the one from /etc/rc.

Command output is mailed to the crontab owner (BSD can't do this), can be mailed to a person other than the crontab owner (SysV can't do this), or the feature can be turned off and no mail will be sent at all (SysV can't do this either).

All of the '@' commands that can appear in place of the first five fields are extensions.

AUTHOR

Paul Vixie <paul@vix.com>