NAME

cron - daemon to execute scheduled commands (Vixie Cron)

SYNOPSIS

cron [**-f**] [**-l**] [**-L** *loglevel*]

DESCRIPTION

cron is started automatically from /etc/init.d on entering multi-user runlevels.

OPTIONS

-f   Stay in foreground mode, don't daemonize.

-l   Enable LSB compliant names for /etc/cron.d files. This setting, however, does not affect the parsing of files under /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly or /etc/cron.monthly.

-n   Include the FQDN in the subject when sending mails. By default, cron will abbreviate the hostname.

-L loglevel

Tell cron what to log about **jobs** (errors are logged regardless of this value) as the sum of the following values:

1   will log the start of all cron jobs

2   will log the end of all cron jobs

4   will log all failed jobs (exit status != 0)

8   will log the process number of all cron jobs

The default is to log the start of all jobs (1). Logging will be disabled if *levels* is set to zero (0). A value of fifteen (15) will select all options.

NOTES

cron searches its spool area (/var/spool/cron/crontabs) for crontab files (which are named after accounts in /etc/passwd); crontabs found are loaded into memory. Note that crontabs in this directory should not be accessed directly - the *crontab* command should be used to access and update them.

cron also reads /etc/crontab, which is in a slightly different format (see crontab(5)). In Debian, the content of /etc/crontab is predefined to run programs under /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly. This configuration is specific to Debian, see the note under **DEBIAN SPECIFIC** below.

Additionally, in Debian, *cron* reads the files in the /etc/cron.d directory. *cron* treats the files in /etc/cron.d as in the same way as the /etc/crontab file (they follow the special format of that file, i.e. they include the *user* field). However, they are independent of /etc/crontab: they do not, for example, inherit environment variable settings from it. This change is specific to Debian see the note under **DEBIAN SPECIFIC** below.

Like /etc/crontab, the files in the /etc/cron.d directory are monitored for changes. In general, the system administrator should not use /etc/cron.d/, but use the standard system crontab /etc/crontab.

/etc/crontab and the files in /etc/cron.d must be owned by root, and must not be group- or other-writable. In contrast to the spool area, the files under /etc/cron.d or the files under /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly may also be symlinks, provided that both the symlink and the file it points to are owned by root. The files under /etc/cron.d do not need to be executable, while the files under /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly do, as they are run by *run-parts* (see run-parts(8) for more information).

cron then wakes up every minute, examining all stored crontabs, checking each command to see if

it should be run in the current minute. When executing commands, any output is mailed to the owner of the crontab (or to the user named in the MAILTO environment variable in the crontab, if such exists). The children copies of cron running these processes have their name coerced to uppercase, as will be seen in the syslog and ps output.

Additionally, *cron* checks each minute to see if its spool directory's modtime (or the modtime on the */etc/crontab* file) has changed, and if it has, *cron* will then examine the modtime on all crontabs files and reload those which have changed. Thus *cron* need not be restarted whenever a crontab file is modified. Note that the crontab(1) command updates the modtime of the spool directory whenever it changes a crontab.

Special considerations exist when the clock is changed by less than 3 hours, for example at the beginning and end of daylight savings time. If the time has moved forwards, those jobs which would have run in the time that was skipped will be run soon after the change. Conversely, if the time has moved backwards by less than 3 hours, those jobs that fall into the repeated time will not be re-run.

Only jobs that run at a particular time (not specified as @hourly, nor with '*' in the hour or minute specifier) are affected. Jobs which are specified with wildcards are run based on the new time immediately.

Clock changes of more than 3 hours are considered to be corrections to the clock, and the new time is used immediately.

cron logs its action to the syslog facility 'cron', and logging may be controlled using the standard *syslogd(8)* facility.

ENVIRONMENT

If configured in */etc/default/cron* in Debian systems, the *cron* daemon localisation settings environment can be managed through the use of */etc/environment* or through the use of */etc/default/locale* with values from the latter overriding values from the former. These files are read and they will be used to setup the LANG, LC_ALL, and LC_CTYPE environment variables. These variables are then used to set the charset of mails, which defaults to 'C'.

This does **NOT** affect the environment of tasks running under cron. For more information on how to modify the environment of tasks, consult crontab(5)

The daemon will use, if present, the definition from */etc/timezone* for the timezone.

The environment can be redefined in user's crontab definitions but *cron* will only handle tasks in a single timezone.

DEBIAN SPECIFIC

Debian introduces some changes to *cron* that were not originally available upstream. The most significant changes introduced are:

—   Support for /etc/cron.{hourly,daily,weekly,monthly} via /etc/crontab,

—   Support for /etc/cron.d (drop-in dir for package crontabs),

—   PAM support,

—   SELinux support,

—   auditlog support,

—   DST and other time-related changes/fixes,

—   SGID crontab(1) instead of SUID root,

—   Debian-specific file locations and commands,

—   Debian-specific configuration (/etc/default/cron),

—   numerous other smaller features and fixes.

Support for /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly is provided in Debian through the default setting of the /etc/crontab file (see the system-wide example in crontab(5)). The default system-wide crontab contains four tasks: run every hour, every day, every week and every month. Each of these tasks will execute **run-parts** providing each one of the directories as an argument. These tasks are disabled if **anacron** is installed (except for the hourly task) to prevent conflicts between both daemons.

As described above, the files under these directories have to be pass some sanity checks including the following: be executable, be owned by root, not be writable by group or other and, if symlinks, point to files owned by root. Additionally, the file names must conform to the filename requirements of **run-parts**: they must be entirely made up of letters, digits and can only contain the special signs underscores ('_') and hyphens ('-'). Any file that does not conform to these requirements will not be executed by **run-parts**. For example, any file containing dots will be ignored. This is done to prevent cron from running any of the files that are left by the Debian package management system when handling files in /etc/cron.d/ as configuration files (i.e. files ending in .dpkg-dist, .dpkg-orig, and .dpkg-new).

This feature can be used by system administrators and packages to include tasks that will be run at defined intervals. Files created by packages in these directories should be named after the package that supplies them.

Support for /etc/cron.d is included in the *cron* daemon itself, which handles this location as the system-wide crontab spool. This directory can contain any file defining tasks following the format used in /etc/crontab, i.e. unlike the user cron spool, these files must provide the username to run the task as in the task definition.

Files in this directory have to be owned by root, do not need to be executable (they are configuration files, just like /etc/crontab) and must conform to the same naming convention as used by run-parts(8) : they must consist solely of upper- and lower-case letters, digits, underscores, and hyphens. This means that they **cannot** contain any dots. If the **-l** option is specified to *cron* (this option can be setup through /etc/default/cron, see below), then they must conform to the LSB namespace specification, exactly as in the **--lsbsysinit** option in *run-parts*.

The intended purpose of this feature is to allow packages that require finer control of their scheduling than the /etc/cron.{hourly,daily,weekly,monthly} directories to add a crontab file to /etc/cron.d. Such files should be named after the package that supplies them.

Also, the default configuration of *cron* is controlled by */etc/default/cron* which is read by the init.d script that launches the *cron* daemon. This file determines whether *cron* will read the system's environment variables and makes it possible to add additional options to the *cron* program before it is executed, either to configure its logging or to define how it will treat the files under /etc/cron.d.

SEE ALSO
crontab(1), crontab(5), run-parts(8)

AUTHOR
Paul Vixie <paul@vix.com> is the author of *cron* and original creator of this manual page. This page has also been modified for Debian by Steve Greenland, Javier Fernandez-Sanguino and Christian Kastner.