NAME

cron − clock daemon

SYNOPSIS

cron [**−d**[*level*]]

DESCRIPTION

The **cron** daemon executes tasks that must be repeated every now and then (cron jobs), and tasks that must be run just once (at jobs). It is normally used to run daily or weekly system maintenance scripts. What it needs to run and when is specified in a number of "cron tables", or crontab files for short. These tables are:

/usr/lib/crontab
/usr/local/lib/crontab
/var/lib/crontab
/var/opt/*name***/lib/crontab** (Minix-vmd only)
/usr/spool/crontabs/*user*

These files follow the usual pattern: One for the standard things, one for local tasks per organization, one for tasks per system, and one crontab per installed package. (Cron reads **/usr/lib/packages** to find names of installed packages, it doesn't just grab everything in **/var/opt**.) The last set of files fall outside the normal pattern, they are per user crontabs that one can create with the **crontab**(1) command. The file names in **/usr/spool/crontabs/** are login names of the file owners.

The format of a crontab file is described in **crontab**(5).

AT jobs

Cron also takes care of the execution of jobs issued by **at**(1) that are found in **/usr/spool/at/**. Cron simply runs the AT job as if there were an "sh at-job" as a cron job at the appropriate time under the user-id of the owner of the script. The script takes care of the rest. See **at**(1) for the details.

Job I/O

Standard input, output and error are the same as cron's if the job is started by the system crontabs or from package crontabs. This means that output from system jobs usually ends up on the console and in the log file. Output from personal cron jobs or at jobs are mailed to the owner of the job. No mail is sent if the job is silent.

OPTIONS

[**−d**[*level*]]
Set the debug level, by default 1. Makes **cron** print info on what it happens to be doing. Level 1 just tells about sleep times and what job is executed, level 2 also shows the internal crontab data on a table load. (With time fields translated to match those of **struct tm**, see **ctime**(3).)

SIGNALS

Cron takes the following actions when sent one of the following signals:

SIGHUP Reload the crontab tables if they changed since the last time they were loaded, and reexamine the AT job spool. Used by **at**(1) and **crontab**(1).

SIGUSR1 Increase the debug level by 1.

SIGUSR2 Turn debugging off.

ENVIRONMENT

Cron sets the environment variables **USER**, **LOGNAME**, **HOME**, and **SHELL** to the user's login name (2x), home directory, and shell if a job is executed for a given user. The working directory is set to the user's home directory. Everything else is inherited from **cron**, exactly as **cron** got it when it started. Note that commands are always passed to **/bin/sh**, not to the user's shell.

System cron jobs are in principle executed with **cron**'s environment, use **−u root** or the crontab file **/usr/spool/crontabs/root** if you want to give root the same treatment as ordinary users.

FILES

/usr/lib/crontab	Main Minix crontab file.
/usr/local/lib/crontab	Local jobs for all systems in an organization.
/var/lib/crontab	System specific jobs.
/var/opt/*name***/lib/crontab**	Per package jobs for Minix-vmd.
/usr/lib/packages	List of installed packages.
/usr/spool/crontabs/*user*	Per user jobs.
/usr/spool/at/*	Jobs issued by **at**(1).
/usr/run/cron.pid	Process id of cron when cron is running. Used by **at**(1) and **crontab**(1) to send cron a hangup signal.

SEE ALSO

at(1), **crontab**(1).

NOTES

A job is not reissued until a previous instance of it has exited. The next time to execute is computed from the previous time it ran. If job issuing lags behind on the system time then the next time to run it is computed from the current system time.

BUGS

Cron doesn't like it if the system time is changed. If set forward then cron will react when it next wakes up by running all jobs within the skipped time once or twice before it catches up. Setting the clock backwards makes cron play dead until the system time passes the old time. (Changing the system time is bad idea anyway, and not just because of cron.)

Jobs that fall in the missing hour in a change to Daylight Saving Time are skipped. Nothing is done in the extra hour on the change out of DST.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)