NAME

anacron – runs commands periodically

SYNOPSIS

anacron [**-s**] [**-f**] [**-n**] [**-d**] [**-q**] [**-t anacrontab**] [**-S spooldir**] [*job*] ...
anacron [-S spooldir] -u [**-t anacrontab**] [*job*] ...
anacron [**-V**|**-h**]
anacron -T [-t anacrontab]

DESCRIPTION

Anacron can be used to execute commands periodically, with a frequency specified in days. Unlike **cron(8)**, it does not assume that the machine is running continuously. Hence, it can be used on machines that aren't running 24 hours a day, to control daily, weekly, and monthly jobs that are usually controlled by **cron**.

When executed, Anacron reads a list of jobs from a configuration file, normally */etc/anacrontab* (see **anacrontab(5)**). This file contains the list of jobs that Anacron controls. Each job entry specifies a period in days, a delay in minutes, a unique job identifier, and a shell command.

For each job, Anacron checks whether this job has been executed in the last n days, where n is the period specified for that job. If not, Anacron runs the job's shell command, after waiting for the number of minutes specified as the delay parameter.

After the command exits, Anacron records the date in a special timestamp file for that job, so it can know when to execute it again. Only the date is used for the time calculations. The hour is not used.

When there are no more jobs to be run, Anacron exits.

Anacron only considers jobs whose identifier, as specified in the *anacrontab* matches any of the *job* command-line arguments. The *job* arguments can be shell wildcard patterns (be sure to protect them from your shell with adequate quoting). Specifying no *job* arguments, is equivalent to specifying "*" (That is, all jobs will be considered).

Unless the **-d** option is given (see below), Anacron forks to the background when it starts, and the parent process exits immediately.

Unless the **-s** or **-n** options are given, Anacron starts jobs immediately when their delay is over. The execution of different jobs is completely independent.

If a job generates any output on its standard output or standard error, the output is mailed to the user running Anacron (usually root), or to the address contained by the MAILTO environment variable in the crontab, if such exists.

Informative messages about what Anacron is doing are sent to **syslogd(8)** under facility **cron**, priority **notice**. Error messages are sent at priority **error**.

"Active" jobs (i.e. jobs that Anacron already decided to run and now wait for their delay to pass, and jobs that are currently being executed by Anacron), are "locked", so that other copies of Anacron won't run them at the same time.

OPTIONS

-f Force execution of the jobs, ignoring the timestamps.

-u Only update the timestamps of the jobs, to the current date, but don't run anything.

-s Serialize execution of jobs. Anacron will not start a new job before the previous one finished.

-n Run jobs now. Ignore the delay specifications in the */etc/anacrontab* file. This options implies **-s**.

-d Don't fork to the background. In this mode, Anacron will output informational messages to standard error, as well as to syslog. The output of jobs is mailed as usual.

-q Suppress messages to standard error. Only applicable with **-d**.

-t anacrontab
Use specified anacrontab, rather than the default

-T Anacrontab testing. The configuration file will be tested for validity. If there is an error in the file, an error will be shown and anacron will return 1. Valid anacrontabs will return 0.

-S spooldir
Use the specified spooldir to store timestamps in. This option is required for users who wish to run anacron themselves.

-V Print version information, and exit.

-h Print short usage message, and exit.

SIGNALS

After receiving a **SIGUSR1** signal, Anacron waits for running jobs, if any, to finish and then exits. This can be used to stop Anacron cleanly.

NOTES

Make sure that the time-zone is set correctly before Anacron is started. (The time-zone affects the date). This is usually accomplished by setting the TZ environment variable, or by installing a */usr/lib/zoneinfo/localtime* file. See **tzset(3)** for more information.

Timestamp files are created in the spool directory for each job in anacrontab. These are never removed automatically by anacron, and should be removed by hand if a job is no longer being scheduled.

DEBIAN-SPECIFIC CONFIGURATION

On Debian-based systems, anacron will be activated hourly every day from 07:30 local time to 23:30 local time through cron job (on non-systemd systems where cron is installed and enabled) or systemd timer (on systemd-based systems). On activation, anacron will check if it missed some jobs. If yes, it will start those jobs after a short period of time.

By default, The hourly activation of anacron will not take place when the system is using battery and no AC power is connected to the computer. It is meant to reduce power usage and extend battery life, but such design might lead to unwanted results. Users may disable this feature and let anacron run regardless of power supply. Please read Debian-specific documentation in */usr/share/doc/anacron/README.Debian* file for detailed instruction in now to change such behaviour.

FILES

/etc/anacrontab
Contains specifications of jobs. See **anacrontab(5)** for a complete description.

/var/spool/anacron
This directory is used by Anacron for storing timestamp files.

/lib/systemd/system/anacron.service
This file provides systemd service for anacron.

/lib/systemd/system/anacron.timer
This file provides systemd timer for anacron. Currently the service is triggered hourly through systemd timer.

SEE ALSO

anacrontab(5), cron(8), tzset(3)

The Anacron *README* file.

For Debian-specific modifications, please read */usr/share/doc/anacron/README.Debian* file for detailed information.

BUGS

Anacron never removes timestamp files. Remove unused files manually.

Anacron uses up to two file descriptors for each active job. It may run out of descriptors if there are more than about 125 active jobs (on normal kernels).

Mail comments, suggestions and bug reports to Sean 'Shaleh' Perry <shaleh@(debian.org|valinux.com)>.

AUTHOR

Anacron was originally conceived and implemented by Christian Schwarz <schwarz@monet.m.isar.de>. The current implementation is a complete rewrite by Itai Tzur <itzur@actcom.co.il>.

The code base was maintained by Sean 'Shaleh' Perry <shaleh@(debian.org|valinux.com)>. During 2004-2006, it was maintained by Pascal Hakim <pasc@(debian.org|redellipse.net)>. During 2009-2014, it was maintained by Peter Eisentraut <petere@debian.org>.

Nowadays anacron in Debian is co-maintained by various developers from Debian Project.