

22 Automated Tasks

In Linux, tasks can be configured to run automatically within a given period of time and on given dates. Red Hat Linux comes preconfigured to run certain system tasks to keep your system updated. For example, the `slocate` database is updated daily. A system administrator can use automated tasks to perform periodic backups, monitor the system, run custom scripts, and more.

22.1 Cron

Cron is a daemon that can be used to execute scheduled tasks according to a combination of the time, day of the month, month, day of the week, and week.

Cron assumes that the system is on continuously. If the system is not on when a task is scheduled it is not executed. To configure tasks based on periods instead of exact times, see Section 22.3, *Anacron*.

To use the cron service, you must have the `vixie-cron` RPM package installed. To determine if the package is installed, use the command `rpm -q vixie-cron`.

22.2 Configuring a Cron Task

The main configuration file for cron, `/etc/crontab`, contains the following lines:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

The first four lines are variables used to configure the environment in which the cron tasks are run. The value of the `SHELL` variable tells the system which shell environment to use (in this example the bash shell), and the `PATH` variable defines the path used to execute commands. The output of the cron tasks are emailed to the username defined with the `MAILTO` variable. If the `MAILTO` variable is defined as an empty string (`MAILTO=""`), email will not be sent. The `HOME` variable can be used to set the home directory to use when executing commands or scripts.

Each line in the `/etc/crontab` file has the format:

```
minute hour day month dayofweek command
```

- `minute` — any integer from 0 to 59
- `hour` — any integer from 0 to 23
- `day` — any integer from 1 to 31 (must be a valid day if a month is specified)
- `month` — any integer from 1 to 12 (or the short name of the month such as `jan`, `feb`, and so on)
- `dayofweek` — any integer from 0 to 7 where 0 or 7 represents Sunday (or the short name of the week such as `sun`, `mon`, and so on)
- `command` — the command to execute. The command can either be a command such as `ls /proc >> /tmp/proc` or the command to execute a custom script that you wrote.

For any of the above values, an asterisk (*) can be used to specify all valid values. For example, an asterisk for the month value means execute the command every month within the constraints of the other values.

A hyphen (-) between integers specifies a range of integers. For example, **1-4** means the integers 1, 2, 3, and 4.

A list of values separated by commas (,) specifies a list. For example, **3, 4, 6, 8** indicates those four specific integers.

The forward slash (/) can be used to specify step values. The value of an integer can be skipped within a range by following the range with **<integer>**. For example, **0-59/2** can be used to define every other minute in the minute field. Step values can also be used with an asterisk. For instance, the value ***/3** can be used in the month field to skip every third month.

Any lines that begin with a hash mark (#) are comments and are not processed.

Example 22-1 Examples of crontabs

```
# record the memory usage of the system every monday
# at 3:30AM in the file /tmp/meminfo
30 3 * * mon cat /proc/meminfo >> /tmp/meminfo
# run custom script the first day of every month at 4:10AM
10 4 1 * * /root/scripts/backup.sh
```

As you can see from the `/etc/crontab` file, it uses the `run-parts` script to execute the scripts in the `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, and `/etc/cron.monthly` files on an hourly, daily, weekly, or monthly basis respectively. The files in these directory should be shell scripts.

If a cron tasks needs to be executed on a schedule other than hourly, daily, weekly, or monthly, it can be added to the `/etc/cron.d` directory. All files in this directory use the same syntax as `/etc/crontab`.

The cron daemon checks the `etc/crontab` file, the `etc/cron.d/` directory, and the `/var/spool/cron` directory every minute for any changes. If any changes are found, they are loaded into memory. Thus, the daemon does not need to be restarted if a crontab file is changed.

Users other than root can configure cron tasks by using the `crontab` utility. All user-defined crontabs are stored in the `/var/spool/cron` directory and are executed using the usernames of the users that created them. To create a crontab as a user, login as that user and type the command `crontab -e` to edit the user's crontab using the editor specified by the `VISUAL` or `EDITOR` environment variable. The file uses the same format as `/etc/crontab`. When the changes to the crontab are saved, the crontab is stored according to username and written to the file `/var/spool/cron/username`.

22.2.1 Starting and Stopping the Service

To start the cron service, use the command `/sbin/service crond start`. To stop the service, use the command `/sbin/service crond stop`. It is recommended that you start the service at boot time. Refer to Chapter 8, *Controlling Access to Services* for details on starting the cron service automatically at boot time.

22.3 Anacron

Anacron is a task scheduler similar to cron except that it does not require the system to run continuously. It can be used to run the daily, weekly, and monthly jobs usually run by cron.

To use the Anacron service, you must have the `anacron` RPM package installed. To determine if the package is installed, use the command `rpm -q anacron`.

22.3.1 Configuration File

Anacron tasks are listed in the configuration file `/etc/anacron`. Each line in the configuration file corresponds to a task and has the format:

```
period delay job-identifier command
```

- `period` — frequency (in days) to execute the command
- `delay` — delay time in minutes
- `job-identifier` — description of the task, used in Anacron messages and as the name of the job's timestamp file, can contain any non-blank characters (except slashes).
- `command` — command to execute

For each task, Anacron determines if the task has been executed within the period specified in the `period` field of the configuration file. If it has not been executed within the given period, Anacron executes the command specified in the `command` field after waiting the number of minutes specified in the `delay` field.

After the task is completed, Anacron records the date in a timestamp file in the `/var/spool/anacron` directory. Only the date is used (not the time), and the value of the `job-identifier` is used as the filename for the timestamp file.

Environment variables such as `SHELL` and `PATH` can be defined at the top of `/etc/anacron` as with the cron configuration file.

The default configuration file looks similar to the following:

Figure 22–1 Default anacrontab

```
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

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

# These entries are useful for a Red Hat Linux system.
1      5      cron.daily          run-parts /etc/cron.daily
7      10     cron.weekly         run-parts /etc/cron.weekly
30     15     cron.monthly        run-parts /etc/cron.monthly
```

As you can see in Figure 22–1, *Default anacrontab*, the anacrontab for Red Hat Linux is configured to make sure the daily, weekly, and monthly cron tasks are run.

22.3.2 Starting and Stopping the Service

To start the anacron service, use the command `/sbin/service anacron start`. To stop the service, use the command `/sbin/service anacron stop`. It is recommended that you start the service at boot time. Refer to Chapter 8, *Controlling Access to Services* for details on starting the anacron service automatically at boot time.

22.4 Additional Resources

To learn more about configuring automated tasks, refer to the following resources.

22.4.1 Installed Documentation

- `cron` man page — overview of cron
- `crontab` man pages in sections 1 and 5 — The man page in section 1 contains an overview of the `crontab` file. The man page in section 5 contains the format for the file and some example entries.