`cron` is used for repeated scheduled execution of commands.

If `/etc/cron.allow` exists, only users listed therein can access the service.
If `/etc/cron.deny` exists, all users except those listed therein can access the service.
If none of these files exist, all users can access the service.

It is not necessary to restart `crond` after the modification of a crontab file, as the changes will be reloaded automatically.

`crontab -e`	Edit your user crontab file
`crontab -l`	List the contents of your crontab file
`crontab -e -u jdoe`	Edit the crontab file of another user (command available only to the superuser)
`/etc/crontab`	System-wide crontab file; this is the list of commands to execute periodically
`/etc/cron.d/`	Directory containing commands to execute periodically, one command per file (which must have the same syntax as `/etc/crontab`)

`/etc/cron.hourly/`
`/etc/cron.daily/`
`/etc/cron.weekly/`
`/etc/cron.monthly/`
	Scripts placed in these directories will be automatically executed on the specified periods

`/var/spool/cron/`*`user`*	Crontab of *user*

/etc/crontab

`# m h dom mon dow user command`	
`25 6 * * 1 root foo.sh`	every Monday at 6:25 AM
`*/5 16 * * * root /opt/myscript.sh`	from 4:00 to 4:55 PM every 5 minutes everyday
`0,30 7 25 12 * jdoe /home/jdoe/bar.sh`	at 7:00 and 7:30 AM on 25th December
`3 17 * * 1-5 root baz.sh`	at 5:03 PM everyday, from Monday to Friday

m	minutes
h	hours
dom	day of month (1-31)
mon	month (1-12 or jan-dec)
dow	day of week (0-7 or sun-sat; 0=7=Sunday)
user	User as whom the command will be executed
command	Command that will be executed at the specified times

The `crond` daemon checks `/etc/crontab` every minute and runs the command as the specified user at the specified times. Each user may also set his own crontab scheduling, which will result in a file `/var/spool/cron/`*`user`*; this user's crontab file has the same format as the system-wide crontab file, except that the `user` field is not present.

/etc/anacrontab

`# period delay job-identifier command`	
`7 10 cron.weekly /opt/myscript.sh`	If the job has not been run in the last 7 days, wait 10 minutes and then execute the command

period	period, in days, during which the command was not executed
delay	delay to wait, in minutes, before execution of the command
job-identifier	job identifier in anacron messages; should be unique for each anacron job
command	command that will be executed

Anacron jobs are run by `crond`, and permit the execution of periodic jobs on a machine that is not always powered on, such as a laptop.
Only the superuser can schedule anacron jobs, which have a granularity of one day (vs one minute for cron jobs).
The file `/var/spool/anacron/`*`job_identifier`* contains the date of the last execution of the specified anacron job.