Scheduling Repeated Jobs with Cron

If you need to repeat a task on a schedule, you can use the cron service. Every minute the cron service checks to see if there are any scheduled jobs to run and if so runs them. Cron jobs are often used to automate a process or perform routine maintenance. You can schedule cron jobs by using the `crontab` command.

`cron` - A time based job scheduling service. This service is typically started when the system boots.

`crontab` - A program to create, read, update, and delete your job schedules.

A crontab (cron table) is a configuration file that specifies when commands are to be executed by cron. Each line in a crontab represents a job and contains two pieces of information: 1) when to run and 2) what to run. The time specification consists of five fields. They are minutes, hour, day of the month, month, and day of the week. After the time specification you provide the command to be executed.

Crontab Format

```
* * * * * command
| | | | |
| | | | +-- Day of the Week   (0-6)
| | | +---- Month of the Year (1-12)
| | +------ Day of the Month  (1-31)
| +-------- Hour              (0-23)
+---------- Minute            (0-59)
```

The command will only be executed when all of the time specification fields match the current date and time. You can specify that a command be run only once, but this is not the typical use case for cron. Typically, one or more of the time specification fields will contain an asterisk (`*`) which matches any time or date for that field. Here is an example crontab.

```
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
```

Here is a graphical representation of the above crontab entry.

```
0 7 * * 1 /opt/sales/bin/weekly-report
| | | | |
| | | | +-- Day of the Week   (0-6)
| | | +---- Month of the Year (1-12)
| | +------ Day of the Month  (1-31)
| +-------- Hour              (0-23)
+---------- Minute            (0-59)
```

This job will run only when the minute is 0, the hour is 7, and the day of the week is 1. In the day of the week field 0 represents Sunday, 1 Monday, etc. This job will run on any day and during any month since the asterisk was used for those two fields.

If any output is generated by the command it is mailed to you. You can check your local mail with the `mail` command. If you would prefer not to get email you can redirect the output of the command as in this example.

```
# Run at 02:00 every day and send output to a log file.
0 2 * * * /opt/acme/bin/backup-db > /var/opt/acme/backup-db.log 2>&1
```

You can provide multiple values for each of the fields. If you would like to run a command every half-hour, you could do this.

```
# Run every 30 minutes.
0,30 * * * * /opt/acme/bin/half-hour-check

# Another way to do the same thing.
*/2 * * * * /opt/acme/bin/half-hour-check
```

Instead of using `0,30` for the minute field you could have used `*/2`. You can even use ranges with a dash. If you want to run a job every minute for the first four minutes of the hour you can use this time specification: `0-4 * * * * command`.

There are several implementations of the cron scheduler and some allow you to use shortcuts and keywords in your crontabs. Common keywords have been provided below, but refer to the documentation for cron on your system to ensure these will work.

Keyword	Description	Equivalent
@yearly	Run once a year at midnight in the morning of January 1	`0 0 1 1 *`
@annually	Same as @yearly	`0 0 1 1 *`
@monthly	Run once a month at midnight in the morning of the first day of the month	`0 0 1 * *`
@weekly	Run once a week at midnight in the morning of Sunday	`0 0 * * 0`
@daily	Run once a day at midnight	`0 0 * * *`
@midnight	Same as @daily	`0 0 * * *`
@hourly	Run once an hour at the beginning of the hour	`0 * * * *`
@reboot	Run at startup	N/A

Using the Crontab Command

Use the `crontab` command to manipulate cron jobs.

`crontab file` - Install a new crontab from file.

`crontab -l` - List your cron jobs.

`crontab -e` - Edit your cron jobs.

`crontab -r` - Remove all of your cron jobs.

```
$ crontab -l
no crontab for bob
$ cat my-cron
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
$ crontab my-cron
$ crontab -l
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
$ crontab -e
# $EDITOR is invoked.
$ crontab -r
$ crontab -l
no crontab for bob
$
```

CronMaker is a software to make cron jobs and after that you can put these cron jobs in crontab configuration file.

Link: http://www.cronmaker.com