SUSE

Working with `systemd` Timers

WHAT?

From running a backup script at regular intervals to starting a specific process as soon as the machine boots, there are plenty of tasks that require scheduling on a Linux system. `systemd` timers provide a flexible mechanism for scheduling and managing jobs and services.

WHY?

This article is intended to provide a complete overview of `systemd` timers covering creating, maintaining, testing, troubleshooting and migrating from cron.

EFFORT

It takes 10 minutes to create an example `systemd` timer. You need up to 30 minutes to fully understand how `systemd` timers work.

REQUIREMENTS

- Basic understanding of `systemd`.
- `root` or sudo privileges. To use `systemd` timers as a regular user, refer to *Section 7, "Using timers as a regular user"* first.

Publication Date: 08 Oct 2026

Contents

1 The `systemd` timer concept 3

2 Creating a timer 3

3 Managing timers 5

4 Timer types 8

5 Testing calendar entries 11

6 Getting e-mail notifications when a timer fails 12

7 Using timers as a regular user 14

8 Migrating from cron to `systemd` timers 15

9 Troubleshooting and FAQs 17

10 For more information 20

11 Legal Notice 20

A GNU Free Documentation License 21

1 The systemd timer concept

systemd timer units provide a mechanism for scheduling jobs on Linux. The execution time of these jobs can be based on the time and date or on events.

systemd timer units are identified by the .timer file name extension. Each timer file requires a corresponding service file it controls. In other words, a timer file activates and manages the corresponding service file. systemd timers support the following features:

- Jobs scheduled using a timer unit can depend on other systemd services. Timer units are treated as regular systemd services, so can be managed with **systemctl**.
- Timers can be real-time (being triggered on calendar events) or monotonic (being triggered at a specified time elapsed from a certain starting point).
- Time units are logged to the system journal, which makes it easier to monitor and troubleshoot them.
- Timers use the centralized systemd management services.
- If the system is off during the expected execution time, the timer is executed once the system is running again.

2 Creating a timer

The following example shows how to set up a timer that triggers the helloworld.sh shell script after boot time and repeats its execution every 24 hours relative to its activation time. It also runs Monday to Friday at 10 a.m.

2.1 *Hello World* example

1. Create the file /etc/systemd/system/helloworld.service with the following content:

```
[Unit]
Description="Hello World script"

[Service]
ExecStart=/usr/local/bin/helloworld.sh
```

This is a systemd service file that tells systemd which application to run.

2. Create the file `/etc/systemd/system/helloworld.timer` with the following content:

```
[Unit]
Description="Run helloworld.service 5min after boot and every 24 hours relative to
 activation time"

[Timer]
OnBootSec=5min
OnUnitActiveSec=24h
OnCalendar=Mon..Fri *-*-* 10:00:00
Unit=helloworld.service

[Install]
WantedBy=multi-user.target
```

   This is the timer file that controls the activation of the respective service file.

3. Verify that the files you created above contain no errors:

```
> systemd-analyze verify /etc/systemd/system/helloworld.*
```

   If the command returns no output, the files have passed the verification successfully.

4. Start the timer:

```
> sudo systemctl start helloworld.timer
```

   Activates the timer for the current session only.

5. Enable the timer to make sure that it is activated on boot:

```
> sudo systemctl enable helloworld.timer
```

2.2 The example explained

EXAMPLE 1: THE SERVICE FILE

```
[Unit]
Description="Hello World script" (1)

[Service]
ExecStart=/usr/local/bin/helloworld.sh (2)
```

(1) A brief description explaining the service file's purpose.

(2) The application to execute.

> The `[Unit]` and `[Service]` sections are the minimum sections required for a service file to work. `systemd` service files normally contain an `[Install]` section that determines one or more targets for a service to load. This section is not required in service files for timers, since this information is provided with the timer file. For advanced configuration, refer to Managing `systemd` targets with systemctl (https://documentation.suse.com/smart/systems-management/html/reference-managing-systemd-targets-systemctl/reference-systemctl-managing-targets.html).

EXAMPLE 2: THE TIMER FILE

```
[Unit]
Description="Run helloworld.service 5min after boot and every 24 hours relative to
 activation time" ❶

[Timer]
OnBootSec=5min ❷
OnUnitActiveSec=24h ❸
OnCalendar=Mon..Fri *-*-* 10:00:00 ❹
Unit=helloworld.service ❺

[Install]
WantedBy=multi-user.target ❻
```

❶ A brief description explaining the timer file's purpose.

❷ Specifies a timer that triggers the service five minutes after the system boot. See *Monotonic timers* for details.

❸ Specifies a timer that triggers the service 24 hours after the service has been activated (that is, the timer triggers the service once a day). See *Real-time timer* for details.

❹ Specifies a timer that triggers the service at fixed points in time (in this example, Monday to Friday at 10 a.m.). See *Real-time timer* for details.

❺ The service file to execute.

❻ The `systemd` target in which the timer gets activated. For more information on `systemd` targets, refer to Managing `systemd` targets with systemctl (https://documentation.suse.com/smart/systems-management/html/reference-managing-systemd-targets-systemctl/reference-systemctl-managing-targets.html).

3 Managing timers

You can manage timers using the **`systemctl`** command.

Starting and stopping timers

```
> sudo systemctl start TIMER.timer
> sudo systemctl restart TIMER.timer
> sudo systemctl stop TIMER.timer
```

Enabling and disabling timers

```
> sudo systemctl enable TIMER.timer
> sudo systemctl disable TIMER.timer
```

Showing the timer file contents

```
> sudo systemctl cat TIMER.timer
```

Checking on a specific timer

```
> sudo systemctl status TIMER.timer
```

EXAMPLE 3: TIMER STATUS

```
> sudo systemctl status helloworld.timer
● helloworld.timer - "Run helloworld.service 5min after boot and every 24 hours
relative to activation time" ❶
Loaded: loaded (/etc/systemd/system/helloworld.timer; disabled; vendor preset:
 disabled) ❷
Active: active (waiting) since Tue 2022-10-26 18:35:41 CEST; 6s ago ❸
Trigger: Wed 2022-10-27 18:35:41 CEST; 23h left ❹
Triggers: ● helloworld.service ❺
❻
Oct 26 18:35:41 neo systemd[1]: Started "Run helloworld.service 5min after boot and
 every 24 hours relative to activation time". ❼
```

❶ The timer's file name and description.

❷ Lists whether a timer has been successfully parsed and is kept in memory (loaded), shows the full path to the timer file, and shows whether the timer is being started at boot time (enabled) or not (disabled). The first value shows the current system configuration, the second value the vendor preset.

❸ Indicates whether the timer is active (waiting to trigger events) or inactive. If active, it also shows the time that has passed since the last activation (6 seconds in this example).

❹ Date and time the timer is triggered next.

❺ Name of the service file the timer triggers.

❻ Optional line pointing to documentation (for example, man pages). If not available, an empty line is shown (as in this example).

❼ Latest journal entry created by the timer.

To list all timers available on the system, use **`systemctl list-timers`**. The following options are available:

List all active timers:

```
> sudo systemctl list-timers
```

List all timers including inactive ones:

```
> sudo systemctl list-timers --all
```

List all timers matching a pattern:

```
> sudo systemctl list-timers PATTERN
> sudo systemctl list-timers --allPATTERN
```

`PATTERN` must be a name or a shell globbing expression. The operators `*`, `?`, and `[]` may be used. Refer to **`man 7 glob`** for more information on globbing patterns.

List timers matching a certain state:

```
> sudo systemctl list-timers --state=STATE
```

`STATE` takes the following values: `active`, `failed`, `load`, `sub`. See **`man systemctl`** for details.

EXAMPLE 4: LISTING TIMERS

Running any **`systemctl list-timers`** results in a table similar to the one below. In this example, all active timers matching the pattern `snapper*` are listed:

```
> sudo systemctl list-timers snapper*
NEXT❶                         LEFT❷       LAST❸                          PASSED❹
 UNIT❺                      ACTIVATES❻

----------------------------------------------------------------------------------------------------
Tue 2022-10-26 19:00:00 CEST 39min left Tue 2022-10-26 18:00:29 CEST 19min ago
 snapper-timeline.timer snapper-timeline.service
Wed 2022-10-27 08:33:04 CEST 14h   left Tue 2022-10-26 08:33:04 CEST 9h ago
 snapper-cleanup.timer  snapper-cleanup.service
```

❶ The point in time when the timer runs next.

❷ The time left till the next timer run.

❸ The point in time when the timer ran last.

❹ Time elapsed since the last timer run.

❺ The name of the timer unit.

❻ The name of the service the timer activates.

4 Timer types

`systemd` supports two types of timers: real-time (based on calendar) and monotonic (based on events). Although timers are normally persistent, `systemd` also allows to set up transient timers that are only valid for the current session.

Real-time timer

Real-time timers are triggered by calendar events. They are defined using the option `OnCalendar`. You can specify when to trigger an event based on date and time. Use the following template:

```
OnCalendar=DayOfWeek❶ Year-Month-Day❷ Hour:Minute:Second❸
```

❶ Day of week. Possible values are `Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, `Sat`. Leave out to ignore the day of the week.

❷ Date. Specify month and day by two digits, year by four digits. Each value can be replaced by the wildcard `*` to match every occurrence.

❸ Time. Specify each value by two digits. Each value can be replaced by the wildcard `*` to match every occurrence.

Applies to all values: Use two dots to define a continuous range (`Mon..Fri`). Use a comma to delimit a list of separate values (`Mon,Wed,Fri`).

EXAMPLE 5: REAL-TIME TIMER EXAMPLES

- 6 p.m. every Friday:

```
OnCalendar=Fri *-*-* 18:00:00
```

- 5 a.m. every day:

```
OnCalendar=Mon..Sun *-*-* 5:00:00
```

- 1 a.m. and 3 a.m. on Sundays and Tuesdays:

```
OnCalendar=Tue,Sun *-*-* 01,03:00:00
```

- Single date:

```
OnCalendar=Mo..Sun 2023-09-23 00:00:01
```

- To specify triggers at different times, you can create more than one OnCalendar entry in a single timer file:

```
OnCalendar=Mon..Fri *-*-* 10:00
OnCalendar=Sat,Sun *-*-* 22:00
```

For a full list of available features and options, refer to **`man 7 systemd.time`** that offers additional information on the following topics:

- shorten the syntax and use abbreviations
- specify repetitions
- find specific days in a month (last day of month, last Sunday, etc.)
- apply time zones

Monotonic timers

Monotonic timers are triggered at a specified time elapsed from a certain event, such as a system boot or system unit activation event. Values are defined as time units (minutes, hours, days, months, years, etc.). The following units are supported: `usec`, `msec`, `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, `years`. There are several options for defining monotonic timers:

- `OnActiveSec`: time after unit activation

```
OnActiveSec=50minutes
```

- `OnBootSec`: time after system boot

```
OnBootSec=10hours
```

- `OnStartupSec`: time after the service manager is started. For system services, this is almost equal to `OnActiveSec`. Use this for user services where the service manager is started at user login.

```
OnStartupSec=5minutes 20seconds
```

- `OnUnitActiveSec`: time after the corresponding service was last activated

```
OnUnitActiveSec=10seconds
```

- `OnUnitInactiveSec`: time after the corresponding service was last deactivated

```
OnUnitInactiveSec=2hours 15minutes 18 seconds
```

Transient timers

Transient timers are temporary timers that are only valid for the current session. Using these timers, you can either use an existing service file or start a program directly. Transient timers are invoked by running **`systemd-run`**.

The following example runs the `helloworld.service` unit every two hours:

```
> sudo systemd-run --on-active="2hours" --unit="helloworld.service"
```

To run a command directly, use the following syntax. In this example, the script `/usr/local/bin/helloworld.sh` is called directly:

```
> sudo systemd-run --on-active="2hours" /usr/local/bin/helloworld.sh
```

If the command takes parameters, add them separated by space:

```
> sudo systemd-run --on-active="2hours" /usr/local/bin/helloworld.sh --
language=pt_BR
```

Transient timers can be monotonic or real-time. The following switches are supported and work as described in *Monotonic timers*:

- `--on-active`
- `--on-startup`
- `--on-unit-active`
- `--on-unit-inactive`
- `--on-calendar`

For more information, refer to **`man 1 systemd-run`**.

5 Testing calendar entries

`systemd` provides a tool for testing and creating calendar timer entries for real-time timers: **`systemd-analyze calendar`**. It accepts the same argument as the `OnCalendar` entry required to set up real-time timers.

You can concatenate several arguments separated by space. If the term to test is correct, the output shows you when the timer is triggered next (in local time and UTC). It also shows the string in `Normalized form`, and it is recommended to use that string in the timer file. Consider the following examples:

```
> systemd-analyze calendar "Tue,Sun *-*-* 01,03:00:00"
Normalized form: Tue,Sun *-*-* 01,03:00:00
Next elapse: Sun 2021-10-31 01:00:00 CEST
(in UTC): Sat 2021-10-30 23:00:00 UTC
From now: 3 days left

> systemd-analyze calendar "Mon..Fri *-*-* 10:00" "Sat,Sun *-*-* 22:00"
Original form: Mon..Fri *-*-* 10:00
Normalized form: Mon..Fri *-*-* 10:00:00
Next elapse: Thu 2021-10-28 10:00:00 CEST
(in UTC): Thu 2021-10-28 08:00:00 UTC
From now: 19h left

Original form: Sat,Sun *-*-* 22:00
Normalized form: Sat,Sun *-*-* 22:00:00
Next elapse: Sat 2021-10-30 22:00:00 CEST
(in UTC): Sat 2021-10-30 20:00:00 UTC
From now: 3 days left
```

For recurring timers, use the `–iterations N` switch to list trigger times, then test whether they work as expected. The argument *N* specifies the number of iterations you would like to test. The following example string triggers every 8 hours (starting at 00:00:00) on Sundays:

```
> systemd-analyze calendar --iterations 5 "Sun *-*-* 0/08:00:00"
Original form: Sun *-*-* 0/08:00:00
Normalized form: Sun *-*-* 00/8:00:00
Next elapse: Sun 2021-10-31 00:00:00 CEST
(in UTC): Sat 2021-10-30 22:00:00 UTC
From now: 3 days left
Iter. #2: Sun 2021-10-31 08:00:00 CET
(in UTC): Sun 2021-10-31 07:00:00 UTC
From now: 3 days left
Iter. #3: Sun 2021-10-31 16:00:00 CET
(in UTC): Sun 2021-10-31 15:00:00 UTC
From now: 4 days left
```

```
Iter. #4: Sun 2021-11-07 00:00:00 CET
(in UTC): Sat 2021-11-06 23:00:00 UTC
From now: 1 week 3 days left
Iter. #5: Sun 2021-11-07 08:00:00 CET
(in UTC): Sun 2021-11-07 07:00:00 UTC
From now: 1 week 3 days left
```

6 Getting e-mail notifications when a timer fails

`systemd` does not offer a feature similar to cron's MAILTO. The procedure below describes a workaround to enable e-mail notifications when a timer fails.

The procedure consists of the following steps:

1. Create a script that sends an e-mail.
2. Create a `systemd` service file running the e-mail script.
3. Test the e-mail service file.
4. From the service that the timer controls, call the created e-mail service file via `OnFailure`.

> In the following example, we are using the **`mailx`** command from package `mailx`. It requires the Postfix e-mail server to be installed and correctly configured.
>
> 1. Create the script `/usr/local/bin/send_systemd_email`.
>
>   a. The script requires two parameters: `$1`, the e-mail address, and `$2`, the name of the service file for which the failure notification is received. Both parameters are supplied by the unit file running the mail script.
>
>    ```
>    #!/bin/sh
>    systemctl status --full "$2" | mailx -S sendwait\
>     -s "Service failure for $2" -r root@$HOSTNAME $1
>    ```
>
>   b. Make sure the script is executable:
>
>    ```
>    > sudo chmod 755 /usr/local/bin/send_systemd_email
>    ```
>
> 2. Create the file `/etc/systemd/system/send_email_to_`*`USER`*`@.service`.

```
[Unit]
Description=Send systemd status information by email for %i to USER

[Service]
Type=oneshot
ExecStart=/usr/local/bin/send_systemd_email EMAIL_ADDRESS %i
User=root
Group=systemd-journal
```

Replace *USER* and *EMAIL_ADDRESS* in the file with the login and e-mail address of the user that should receive the e-mail. `%i` is the name of the service that has failed (it is passed on to the e-mail service by the `%n` parameter).

3. Verify the service file and fix the reported issues:

```
> systemd-analyze verify /etc/systemd/system/send_email_to_USER@.service
```

If the command returns no output, the file has passed the verification successfully.

4. To verify the complete procedure, start the service using the `dbus` instance for testing. (You can use any other service that is currently running. dbus is used in this example because the service is guaranteed to run on any installation.)

```
> sudo systemctl start send_email_to_USER@dbus.service
```

If successful, *EMAIL_ADDRESS* receives an e-mail with the subject `Service failure for dbus` containing dbus status messages in the body. (This is just a test, there is no problem with the dbus service. You can safely delete the e-mail, no action is required).
If the test e-mail has been successfully sent, proceed by integrating it into your service file.

5. To add an e-mail notification to the service, add an `OnFailure` option to the `Unit` section of the service file for which you would like to get notified in the event of failure:

```
[Unit]
Description="Hello World script"
OnFailure❶=send_email_to_USER❷@%n❸.service

[Service]
ExecStart=/usr/local/bin/helloworld.sh
```

❶ The `OnFailure` option takes a service as an argument.

❷ Replace the part of the service unit file name with the login name.

❸ Specifies the name of the service (`helloworld` in this example). This name is available in the e-mail service file as %i.

You have successfully set up the failure notification for `systemd` services.

Tip: Sending e-mail notifications to multiple users

The e-mail service file has the recipient's e-mail address hard-coded. To send notification e-mails to a different user, copy the e-mail service file, and replace the user login in the file name and the e-mail address within the copy.

To send a failure notification to several recipients simultaneously, add the respective service files to the service file (use spaces as a separator):

```
OnFailure=send_email_to_tux@%n.service send_email_to_wilber@%n.service
```

7 Using timers as a regular user

`systemd` timers can also be used by regular users. It helps you to automate recurring tasks like backups, processing images, or moving data to the cloud.

The same procedures and tasks as for system-wide timers are valid. However, the following differences apply:

- Timer and service files must be placed in `~/.config/systemd/user/`.
- All **`systemctl`** and **`journalctl`** commands must be run with the `--user` switch. **`systemd-analyze`** does *not* require this option.
  As a regular user, you must provide the path to the unit files, as in the examples below. Otherwise, if a system-wide timer with the same name exists, it would be executed or listed instead.

```
> systemctl --user start ~/.config/systemd/user/helloworld.timer
> systemctl --user enable ~/.config/systemd/user/helloworld.timer
> systemctl --user list-timers
> journalctl --user -u helloworld.*
> systemd-analyze verify ~/.config/systemd/user/helloworld.timer
```

Important: User timers only run during an active session

As with other `systemd` services started as a regular user, user timers only run when the user is logged in. Instead, to start user timers at boot time and keep them running after logout, enable *lingering* for each affected user:

```
sudo loginctl enable-linger USER
```

For more information, refer to **`man 1 loginctl`**.

Important: Environment variables are not inherited

The `systemd` user instance does not inherit environment variables set by scripts like `~/.profile` or `~/.bashrc`. To check the `systemd` environment, run **`systemctl --user show-environment`**.

To import any variables missing in the `systemd` environment, specify the following command at the end of your `~/.bashrc`:

```
systemctl --user import-environment VARIABLE1 VARIABLE2
```

8 Migrating from cron to `systemd` timers

All cron jobs can be migrated to `systemd` timers. Find instructions and an example here.

1. Create a service file executing the script. See *Example 1, "The service file"* for details.
2. Create a timer file executing the service file. See *Example 2, "The timer file"* for general instructions.
   a. Convert calendar entries. Time is specified differently in cron and `systemd`. Use the patterns below as a conversion template:

   ```
   Cron:              Minute Hour Day Month DayOfWeek
   systemd: OnCalendar=DayOfWeek Year-Month-Day Hour:Minute:Second
   ```

   To test the converted calendar entry, follow the instructions in *Section 5, "Testing calendar entries"*.

b. Convert cron nicknames (`@NICK`):

```
Cron     : systemd timer
-------- : --------------------------
@reboot  : OnBootSec=1s
@yearly  : OnCalendar=*-01-01 00:00:00
@annually: OnCalendar=*-01-01 00:00:00
@monthly : OnCalendar=*-*-01 00:00:00
@weekly  : OnCalendar=Sun *-*-* 00:00:00
@daily   : OnCalendar=*-*-* 00:00:00
@hourly  : OnCalendar=*-*-* *:00:00
```

c. Convert variable assignments. The `systemd` variable assignment must go into the `[Service]` section. You cannot convert `MAILTO` this way—refer to the next step for this.

```
cron: VARIABLE=VALUE
systemd: Environment="VARIABLE=VALUE"
```

d. Set up e-mail notifications to replace cron's MAILTO feature by following the instructions in *Section 6, "Getting e-mail notifications when a timer fails"*.

EXAMPLE 6: CRON TO `systemd` TIMER MIGRATION

Here are the crontab entries which call the script `helloworld.sh` 5 minutes after booting and at 10 o'clock each Monday to Friday:

```
@reboot sleep 300 && /usr/local/bin/helloworld.sh
0 10 * * * 1-5 /usr/local/bin/helloworld.sh
```

The `systemd` service file (`helloworld.service`) calling the script looks like this:

```
[Unit]
Description="Hello World script"
[Service]
ExecStart=/usr/local/bin/helloworld.sh
```

The timer file (`helloworld.timer`) looks like this:

```
[Unit]
Description="Run helloworld.service 5min after boot and at 10am every Mon-Fri"
[Timer]
OnBootSec=5min
OnCalendar=Mon..Fri *-*-* 10:00:00
Unit=helloworld.service
[Install]
```

```
WantedBy=multi-user.target
```

9 Troubleshooting and FAQs

Learn how to debug and troubleshoot `systemd` timers that have failed. Find answers to frequently asked questions on `systemd` timers.

9.1 Avoiding errors

To avoid errors with `systemd` timers, make sure to follow these best practices:

- Verify that the executable you specify in the service with `ExecStart` runs correctly.
- Check the syntax of the service and timer files by running **`systemd-analyze verify`** *`FILE`*.
- Check execution times of calendar entries by running **`systemd-analyze calendar`** *`CALENDER_ENTRY`*.

9.2 Event is not triggered

When you activate a timer that contains non-critical errors, `systemd` silently ignores them. For example:

EXAMPLE 7: `systemd` TIMER FILE CUTOUT CONTAINING A NON-FATAL ERROR

```
[Timer]
OnBootSec=5min
OnClendar=Mon..Fri 10:00
Unit=helloworld.service
```

Line 3 contains a syntax error (`OnClendar` instead of `OnCalendar`). Since the `[Timer]` section contains a second timer entry (OnBoot), the error is not critical and is silently ignored. As a consequence, the Monday to Friday trigger is not executed. The only way to detect the error is to use the command **`systemd-analyze verify`**:

```
#  systemd-analyze verify /etc/systemd/system/helloworld.timer
/etc/systemd/system/helloworld.timer:7: Unknown key name 'OnClendar' in section
 'Timer', ignoring.
```

9.3 Checking the system journal for errors

As with every `systemd` service, events and actions triggered by timers are logged with the system journal. If a trigger does not behave as expected, check the log messages with **`journalctl`**. To filter the journal for relevant information, use the `-u` switch to specify the `systemd` timers and service files. Use this option to show the log entries for the timer *and* the corresponding service file:

```
sudo journalctl -u  helloworld.timer -u helloworld.service
```

or shorter (if applicable):

```
sudo journalctl -u  helloworld.*
```

`journalctl` is a tool that supports many options and filters. Please refer to **`man 1 journalctl`** for in-depth information. The following options are useful for troubleshooting timers:

- `-b`: Only show entries for the current boot.
- `-S today`: Only show entries from today.
- `-x`: Show help texts alongside the log entry.
- `-f`: Start with the most recent entries and continuously print the log as new entries get added. Useful to check triggers that occur in short intervals. Exit with **`Ctrl`**–**`C`**.

9.4 `systemd` timer: catching up on missed runs

If a `systemd` timer was inactive or the system was off during the expected execution time, missed events can optionally be triggered immediately when the timer is activated again. To enable this, add the configuration option `Persistent=true` to the `[Timer]` section:

```
[Timer]
OnCalendar=Mon..Fri 10:00
Persistent=true
Unit=helloworld.service
```

9.5 How to migrate from cron to `systemd` timers?

All cron jobs can be migrated to `systemd` timers. Here are general instructions on migrating a cron job:

1. Create a service file executing the script. See *Example 1, "The service file"* for details.

2. Create a timer file executing the service file. See *Example 2, "The timer file"* for general instructions.

   a. Convert calendar entries. Time is specified differently in cron and `systemd`. Use the patterns below as a conversion template:

   ```
   Cron:               Minute Hour Day Month DayOfWeek
   systemd: OnCalendar=DayOfWeek Year-Month-Day Hour:Minute:Second
   ```

   To test the converted calendar entry, follow the instructions in *Section 5, "Testing calendar entries"*.

   b. Convert cron nicknames (`@NICK`):

   ```
   Cron     : systemd timer
   -------- : --------------------------
   @reboot  : OnBootSec=1s
   @yearly  : OnCalendar=*-01-01 00:00:00
   @annually: OnCalendar=*-01-01 00:00:00
   @monthly : OnCalendar=*-*-01 00:00:00
   @weekly  : OnCalendar=Sun *-*-* 00:00:00
   @daily   : OnCalendar=*-*-* 00:00:00
   @hourly  : OnCalendar=*-*-* *:00:00
   ```

   c. Convert variable assignments. The `systemd` variable assignment must go into the `[Service]` section. You cannot convert `MAILTO` this way—refer to the next step for this.

   ```
   cron: VARIABLE=VALUE
   systemd: Environment="VARIABLE=VALUE"
   ```

   d. Set up e-mail notifications to replace cron's MAILTO feature by following the instructions in *Section 6, "Getting e-mail notifications when a timer fails"*.

EXAMPLE 8: CRON TO `systemd` TIMER MIGRATION

Here are the crontab entries which call the script `helloworld.sh` 5 minutes after booting and at 10 o'clock each Monday to Friday:

```
@reboot sleep 300 && /usr/local/bin/helloworld.sh
0 10 * * * 1-5 /usr/local/bin/helloworld.sh
```

The `systemd` service file (`helloworld.service`) calling the script looks like this:

```
[Unit]
Description="Hello World script"
[Service]
ExecStart=/usr/local/bin/helloworld.sh
```

The timer file (`helloworld.timer`) looks like this:

```
[Unit]
Description="Run helloworld.service 5min after boot and at 10am every Mon-Fri"
[Timer]
OnBootSec=5min
OnCalendar=Mon..Fri *-*-* 10:00:00
Unit=helloworld.service
[Install]
WantedBy=multi-user.target
```

10 For more information

- For a full reference on `systemd` timers including advanced configuration options (like delays or handling clock or time zone changes), refer to **`man 5 systemd.timer`**.
- Basic `systemd` concepts (https://documentation.suse.com/smart/systems-management/html/concept-systemd/concept-systemd.html)
- Starting and stopping `systemd` services (https://documentation.suse.com/smart/systems-management/html/reference-systemctl-start-stop-services/reference-systemctl-start-stop-services.html)
- Enabling and disabling `systemd` services (https://documentation.suse.com/smart/systems-management/html/reference-systemctl-enable-disable-services/reference-systemctl-enable-disable-services.html)
- Debugging failed `systemd` services (https://documentation.suse.com/smart/systems-management/html/task-debug-failed-systemd-services/index.html)
- Sending termination signals to `systemd` services (https://documentation.suse.com/smart/systems-management/html/task-send-termination-signals-systemd/task-send-termination-signals-systemd.html)

11 Legal Notice

Copyright© 2006– 2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see https://www.suse.com/company/legal/. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

A GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image for-

mats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally

prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.