SUSE Linux Enterprise Real Time 12 SP5

Quick Start

SUSE Linux Enterprise Real Time 12 SP5

SUSE Linux Enterprise Real Time is an add-on to SUSE® Linux Enterprise. It allows you to run tasks which require deterministic real-time processing in a SUSE Linux Enterprise environment.

SUSE Linux Enterprise Real Time meets this requirement by offering several options for CPU and I/O scheduling, CPU shielding and for setting CPU affinities to processes.

Publication Date: 11/14/2024

Contents

1 Product Overview

If your business can respond more quickly to new information and changing market conditions, you have a distinct advantage over those that cannot. Running your time-sensitive mission-critical applications using SUSE Linux Enterprise Real Time reduces process dispatch latencies and gives you the time advantage you need to increase profits or avoid further financial losses, ahead of your competitors.

1.1 Key Features

Some of the key features for SUSE Linux Enterprise Real Time are:

- Pre-emptible real-time kernel.
- Ability to assign high-priority processes.
- Greater predictability to complete critical processes on time, every time.
  In comparison to normal Linux kernel, which is optimized for overall system performance regardless of individual process response time, SUSE Linux Enterprise Real Time kernel is tuned toward predictable process response time.
- Increased reliability.
- Lower infrastructure costs.
- Tracing and debugging tools that help you analyze and identify bottlenecks in mission-critical applications.

1.2 Specific Scenario

SUSE Linux Enterprise Real Time Service Pack 3 supports Virtualization and Docker usage. Refer to *Article "Virtualization Guide"* for reference.

2 Installing SUSE Linux Enterprise Real Time

To install SUSE Linux Enterprise Real Time 12 SP5, start a regular SUSE Linux Enterprise Server 12 SP5 installation. Select SUSE Linux Enterprise Real Time 12 SP5 as an add-on product during the installation. Alternately, if SUSE Linux Enterprise Server is already installed, you can start the Add-on Product installation from YaST or enable SUSE Linux Enterprise Real Time in the YaST SUSE Customer Center configuration. However, in the YaST Boot Loader configurator you need to manually select the `-rt` kernel flavor as the default.

SUSE Linux Enterprise Real Time always needs a SUSE Linux Enterprise Server 12 SP5 base, it cannot be installed in stand-alone mode. For information on how to install add-on products, see the SUSE Linux Enterprise 12 SP5 *Deployment Guide*, available at http://www.suse.com/doc. Refer to chapter *Installing Add-On Products*.

The following sections provide a brief introduction to the tools and possibilities of SUSE Linux Enterprise Real Time.

3 Managing CPU Sets with **cset**

In some circumstances, it is beneficial to be able to run specific tasks only on defined CPUs. For this reason, the Linux kernel provides a feature called *cpuset*. The `cpuset` feature provides the means to do a so called “soft partitioning” of the system. Dedicated CPUs, together with some predefined memory, work on several tasks.

`cset` consists of one “super command” called **`shield`** and the “regular commands” **`set`** and **`proc`**. The purpose of the super command **`shield`** is to create a common CPU shielding setup within one step by combining regular commands.

For more information about options and parameters of the **`shield`** subcommand, view the help by running:

```
cset help shield
```

3.1 Setting Up a CPU Shield for a Single CPU

The command **`cset`** provides the high level functionality to set up and manipulate CPU Sets. An example for setting up a CPU shield is:

```
cset shield --cpu=3
```

This will shield CPU3. On a machine with 4 cores CPU0-CPU2 are unshielded.

3.2 Setting up CPU Shields for Multiple CPUs

If you need to shield more than one CPUs, the argument of the `--cpu` option accepts comma separated lists of CPUs including range specifications:

```
cset shield --cpu=1,3,5-7
```

On a machine with 8 cores, this command will shield CPU1, CPU3, CPU5, CPU6, and CPU7. The CPUs CPU0, CPU2 and CPU4 will remain unshielded.

Already existing CPU shields can be extended by the same command. For example, to add CPU4 to the mentioned CPU set, use this command:

```
cset shield --cpu=1,3-7
```

CPU1, CPU3, CPU5 to CPU6 were already shielded and only CPU4 will additionally be shielded. Technically, the command is updating the current CPU shield schema. To reduce the number of shielded CPUs and to unshield CPU1, for example, use the following command:

```
cset shield --cpu=3-7
```

Now only CPU3, CPU4, CPU5, CPU6, and CPU7 are shielded. CPU0, CPU1, and CPU2 are available for system usage.

3.3 Showing CPU Shields

After the CPU shielding is set up you can display the current configuration by running **`cset shield`** without additional options:

```
cset shield
cset: --> shielding system active with
cset: "system" cpuset of: 0-2 cpu, with: 47
cset: "user" cpuset of:  3-7 cpu,  with: 0
```

By default, CPU shielding consists of at least of three `cpuset`s:

- `root` exists always and contains all available CPUs.
- `system` is the `cpuset` of unshielded CPUs.
- `user` is the `cpuset` of shielded CPUs

3.4 Shielding Processes

Certain processes or groups of processes can be assigned to a shielded `cpuset`, after the CPU set is created. To start a new process in the shielded CPU set use the `--exec` option:

```
cset shield --exec APPLICATION
```

To move already running processes to the shielded CPU set use the `--shield` and `--pid` options. The `--pid` option accepts a comma-separated list of PIDs and range specifications:

```
cset shield --shield --pid=1,2,600-700
```

This moves processes with PID 1, 2, and from 600 to 700 to the shielded CPU set. If there is a gap in the range from 600 to 700, then only those available process will be moved to the shield without warning. **`cset`** handles threads like processes and will also interpret TIDs and assign them to the required CPU set.

Warning

The `--shield` option does not check the processes you request to move into the shield. This means that the command will move *any* processes that are bound to specific CPUs—even kernel threads. You can cause a complete system lockup by indiscriminately specifying arbitrary PIDs to the `--shield` command.

3.5 Showing Shielded Processes

Use the **`cset shield`** command to show the number of currently shielded processes (the same command can be used to show the current CPU shield setup). To list shielded and unshielded processes, add the **`--verbose`** option:

```
cset shield --verbose
cset: --> shielding system active with
cset: "system" cpuset of: 0-2,4-15 cpu, with:
   USER       PID  PPID S TASK NAME
      -------- ----- ----- - ---------
         root         1     0 S init [3]
[...]

cset: "user" cpuset of:    3 cpu, with: 1
   USER       PID  PPID S TASK NAME
```

```
      -------- ----- ----- - ---------
         root     10202 10170 S application
```

3.6 Unshielding Processes

To remove a process (or group of processes) from the CPU shield use the `--unshield` option. The argument for `--unshield` is similar to the `--shield` option. This option accepts a comma-separated list of PIDs/TIDs and range specifications:

```
cset shield --unshield --pid=2,650-655
```

This command will unshield the process with the PID 2 and the processes in range of 650 and 655.

3.7 Resetting CPU Sets

To delete CPU sets use the **`cset`** option `--reset`. This will unshield all CPUs and migrate dedicated processes to all available CPUs again.

4 Managing Tree-like Structures with **cset**

More detailed configuration of cpusets can be done with the **`cset`** commands `set` and `proc`.

The subcommand `set` is used to create, modify and destroy `cpuset`s. Compared to the supercommand `shield`, the `set` subcommand can additionally assign memory nodes for NUMA machines.

Besides assigning memory nodes, the subcommand **`set`** creates cpusets in a tree-like structure, rooted at the `root` `cpuset`.

To create a `cpuset` with the subcommand **`set`** you need to specify the CPUs which should be used. Either use a comma-separated list or a range specification:

```
cset set --cpu=1-7 "/one"
```

This command will create a `cpuset` called `one` with assigned CPUs from CPU1 to CPU7. To specify a new `cpuset` called `two` that is a subset of `one`, proceed as follows:

```
cset set --cpu=6 "/one/two"
```

Cpusets follow certain rules. Children can only include CPUs that the parents already have. If you try to specify a different `cpuset`, the kernel `cpuset` subsystem will not let you create that `cpuset`. For example, if you create a `cpuset` that contains CPU3, and then attempt to create a child of that `cpuset` with a CPU other than 3, you will get an error, and the `cpuset` will not be created. The resulting error is somewhat cryptic and is usually “Permission denied”.

To show a table containing useful information, like CPU list and memory list, use the `-r` parameter. The “-X” column shows the exclusive state of CPU or memory. The “path” column shows the real path in the virtual `cpuset` file system.

```
cset set -r
```

On NUMA machines memory nodes can be assigned to a `cpuset` similar to CPUs. The `--mem` option of the subcommand **`set`** allows a comma-separated and inclusive range specification of memory nodes. This example will assign MEM1, MEM3, MEM4, MEM5 and MEM6 to the `cpuset` `new_set`:

```
cset set --mem=1,3-6 new_set
```

Additionally, with the `--cpu_exclusive` and `--mem_exclusive` options (without any additional arguments) set the CPUs or memory nodes exclusive to a `cpuset`:

```
cset set --cpu_exclusive "/one"
```

The status of exclusive state of CPU or memory is shown in the `-X` column when running:

```
cset set -r
```

For more detailed information about options and parameters of the subcommand **`set`**, view the **`cset`** help:

```
cset help set
```

After the `cpuset` is initialized, the subcommand **`proc`** can start processes on certain `cpuset`s with the `--exec` option. The following will start the application `fastapp` within the `cpuset` `new_set`:

```
cset proc --exec --set new_set fastapp
```

To move an already running process inside an already existing `cpuset` use the option `--move`. It accepts a comma-separated list and range specifications of PIDs. The following command will move processes with PID 2442 and within range of 3000 to 3200 into the `cpuset` `new_set`:

```
cset proc --move 2442,3000-3200 new_set
```

Listing processes running within a specific `cpuset` can be done by using the option `--list`.

```
cset proc --list new_set
```

The subcommand **`proc`** can also move the entire list of processes within one cpuset to another `cpuset` by using the option `--fromset` and `--toset`. This will move all process assigned to `old_set` and assign them to `new_set`:

```
cset proc --move --fromset old_set \
   --toset new_set
```

For more detailed information about options and parameters of the subcommand **`proc`**, view the help:

```
cset help proc
```

5 Setting Real-time Attributes of a Process with **chrt**

Use the **`chrt`** command to manipulate the real-time attributes of an already running process (like scheduling policy and priority), or to execute a new process with specified real-time attributes.

It is highly recommended for applications which do not use real-time specific attributes by their own but should nevertheless experience the full advantages of real-time. To get full real-time experiences, call these applications with the **`chrt`** command and the right set of scheduler policy and priority parameters.

With the following command, all running processes with their real-time specific attributes are shown. The selection `class` shows the current scheduler policy and `rtprio` the real-time priority:

```
ps -eo pid,tid,class,rtprio,comm
...
 1437  1437 FF      40  fastapp
```

The truncated example above shows the `fastapp` process with PID 1437 running and with scheduler policy `SCHED_FIFO` and priority 40. Scheduler policy abbreviations are:

- `TS - SCHED_OTHER`
- `FF - SCHED_FIFO`
- `RR - SCHED_RR`

It is also possible to get the current scheduler policy and priority of single processes by specifying the PID of the process with the `-p` parameter. For example:

```
chrt -p 1437
```

Scheduler policies have different minimum and maximum priority values. Minimum and maximum values for each available scheduler policy can be retrieved with **`chrt`**:

```
chrt -m
```

To change the scheduler policy and the priority of a running process, **`chrt`** provides the options `--fifo` for `SCHED_FIFO`, `--rr` for `SCHED_RR` and `--other` for `SCHED_OTHER`. The following example will change the scheduler policy to `SCHED_FIFO` with priority 42 for PID 1437:

```
chrt --fifo -p 42 1437
```

Warning

Handle changing of real-time attributes of processes with care. Increasing the priority of certain processes can harm the entire system, depending on the behavior of the process. In some cases, this can lead to a complete system lockup or bad influence on certain devices.

For more information about chrt, see the **`chrt`** man page with **`man 1 chrt`**.

6 Specifying a CPU Affinity with **`taskset`**

The default behavior of the kernel is to keep a process running on the same CPU if the system load is balanced over the available CPUs. Otherwise, the kernel tries to improve the load balancing by moving processes to an idling CPU. In some situations, however, it is desirable to set a CPU affinity for a given process. In this case, the kernel will not move the process away from the selected CPUs. For example, if you use shielding, the shielded CPUs will not run any processes that do not have an affinity to the shielded CPUs. Another possibility to remove load from the other CPUs is to run all low priority tasks on a selected CPU.

If a task is running inside a specific `cpuset`, the affinity dialog must match at least one of the CPUs available in this set. The **`taskset`** command will not move a process outside the `cpuset` it is running in.

To set or retrieve the CPU affinity of a task a bitmask is used. It is represented by a hexadecimal number. If you count the bits of this bitmask, the lowest bit represents the first logical CPU as they are found in `/proc/cpuinfo`. For example:

`0x00000001`
: is processor #0.

`0x00000002`
: is processor #1.

`0x00000003`
: is processor #0 and processor #1.

`0xFFFFFFFE`
: all but the first CPU.

If a given dialog does not contain any valid CPU on the system, an error is returned. If taskset returns without an error, the given program has been scheduled to the specified list of CPUs.

The command **`taskset`** starts a new process with a given CPU affinity, or to redefine the CPU affinity of an already running process.

EXAMPLES

`taskset -p` *`PID`*
: Retrieves the current CPU affinity of the process with PID *`pid`*.

`taskset -p` *`maskPID`*
: Sets the CPU affinity of the process with the *`PID`* to *`mask`*.

`taskset` *`maskcommand`*
: Runs *`command`* with a CPU affinity of `mask`.

For more detailed information about **`taskset`**, see the man page **`man 1 taskset`**.

7 Changing I/O Priorities with **`ionice`**

Handling I/O is one of the critical issues for all high-performance systems. If a task has lots of CPU power available, but must wait for the disk, it will not work as efficiently as it could. The Linux kernel provides three different scheduling classes to determine the I/O handling for a process. All of these classes can be fine-tuned with a nice level.

The *Best Effort* Scheduler

The *Best Effort* scheduler is the default I/O scheduler, and is used for all processes that do not specify a different I/O scheduler class. By default, this scheduler sets its nice level according to the nice value of the running process.

There are eight different nice levels available for this scheduler. The lowest priority is represented by a nice level of seven, the highest priority is zero.

This scheduler has the scheduling class number 2.

The *Real Time* Scheduler

The real-time I/O class always gets the highest priority for disk access. The other schedulers will only be served if no real-time request is present. This scheduling class may easily lock up the system if not implemented with care.

The real-time scheduler defines nice levels (similar to the *Best Effort* scheduler).

This scheduler has the scheduling class number 1.

The *Idle* Scheduler

The *Idle* scheduler does not define any nice levels. I/O is only done in this class if no other scheduler runs an I/O request. This scheduler has the lowest available priority and can be used for processes that are not time-critical.

This scheduler has the scheduling class number 3.

To change I/O schedulers and nice values, use the **`ionice`** command. This provides a means to tune the scheduler of already running processes, or to start new processes with specific I/O settings.

EXAMPLES

`ionice -c3 -p$$`

Sets the scheduler of the current shell to `Idle`.

`ionice`

Without additional parameters, this prints the I/O scheduler settings of the current shell.

`ionice -c1 -p42 -n2`

Sets the scheduler of the process with process id 42 to `Real Time`, and its nice value to 2.

`ionice -c3 /bin/bash`

Starts the Bash shell with the `Idle` I/O scheduler.

For more detailed information about **`ionice`**, see the **`ionice`** man page with **`man 1 ionice`**

8 Changing the I/O Scheduler for Block Devices

The Linux kernel provides several block device schedulers that can be selected individually for each block device. All but the `noop` scheduler perform a kind of ordering of requested blocks to reduce head movements on the hard disk. If you use an external storage system that has its own scheduler, you should disable the Linux internal reordering by selecting the `noop` scheduler.

THE LINUX I/O SCHEDULERS

noop

The *noop* scheduler is a very simple scheduler that performs basic merging and sorting on I/O requests. This scheduler is mainly used for specialized environments that run their own schedulers optimized for the used hardware, such as storage systems or hardware RAID controllers.

deadline

The main point of *deadline* scheduling is to try hard to answer a request before a given deadline. This results in very good I/O for a random single I/O in real-time environments. In principle, the *deadline* scheduler uses two lists with all requests. One is sorted by block sequences to reduce seeking latencies, the other is sorted by expire times for each request. Normally, requests are served according to the block sequence, but if a request reaches its deadline, the scheduler starts to work on this request.

cfq

The *Completely Fair Queuing* scheduler uses a separate I/O queue for each process. All of these queues get a similar time slice for disk access. With this procedure, the *CFQ* tries to divide the bandwidth evenly between all requesting processes. This scheduler has a similar throughput as the *anticipatory* scheduler, but the maximum latency is much shorter.
For the average system this scheduler yields the best results, and thus is the default I/O scheduler on SUSE Linux Enterprise systems.

To print the current scheduler of a block device like `/dev/sda`, use the following command:

```
cat /sys/block/sda/queue/scheduler
noop deadline [cfq]
```

In this case, the scheduler for `/dev/sda` is set to `cfq`, the `Completely Fair Queuing` scheduler. This is the default scheduler on SUSE Linux Enterprise Real Time.

To change the schedulers, echo one of the names `noop`, `deadline`, or `cfq` into `/sys/block/<device>/scheduler`. For example, if you want to set the I/O scheduler of the device `/dev/sda` to `noop`, use the following command:

```
echo "noop" > /sys/block/sda/queue/scheduler
```

To set other variables in the `/sys` file system, use a similar approach.

9 Tuning the Block Device I/O Scheduler

All schedulers, except for the *noop* scheduler, have several common parameters that may be tuned for each block device. You can access these parameters with `sysfs` in the `/sys/block/<device>/queue/iosched/` directory. The following parameters are tuneable for the respective scheduler:

Anticipatory Scheduler

`read_batch_expire`

If write requests are scheduled, this is the time in milliseconds that reads are served before pending writes get a time slice. If writes are more important than reads, set this value lower than `read_expire`.

`write_batch_expire`

Similar to `read_batch_expire` for write requests.

Deadline Scheduler

`read_expire`

The main focus of this scheduler is to limit the start latency for a request to a given time. Therefore, for each request, a deadline is calculated from the current time plus the value of `read_expire` in milliseconds.

`write_expire`

Similar to `read_expire` for write requests.

`fifo_batch`

If a request hits its deadline, it is necessary to move the request from the sorted I/O scheduler list to the dispatch queue. The variable `fifo_batch` controls how many requests are moved, depending on the cost of each request.

`front_merges`

The scheduler normally tries to find contiguous I/O requests and merges them. There are two kinds of merges: The new I/O request may be in front of the existing I/O request (front merge), or it may follow behind the existing request (back merge). Most merges are back merges. Therefore, you can disable the front merge functionality by setting `front_merges` to `0`.

`write_starved`

In case some read or write requests hit their deadline, the scheduler prefers the read requests by default. To prevent write requests from being postponed forever, the variable `write_starved` controls how often read requests are preferred until write requests are preferred over read requests.

CFQ Scheduler

`back_seek_max` **and** `back_seek_penalty`

The *CFQ* scheduler normally uses a strict ascending elevator. When needed, it also allows small backward seeks, but it puts some penalty on them. The maximum backward sector seek is defined with `back_seek_max`, and the multiplier for the penalty is set by `back_seek_penalty`.

`fifo_expire_async` **and** `fifo_expire_sync`

The `fifo_expire_*` variables define the timeout in milliseconds for asynchronous and synchronous I/O requests. To prefer synchronous operations over asynchronous ones, `fifo_expire_sync` value should be lower than fifo_expire_async.

`quantum`

Defines number of I/O requests to be dispatched at once by the block device. This parameter is used for synchronous requests.

`slice_async`, `slice_async_rq`, `slice_sync`, **and** `slice_idle`

These variables define the time slices a block device gets for synchronous or asynchronous operations.

- `slice_async` and `slice_sync` serve as a base value in milliseconds for asynchronous or synchronous disk slice length calculations.
- `slice_async_rq` for how many requests can an asynchronous disk slice accommodate.
- `slice_idle` defines how long I/O scheduler idles before servicing next thread.

The system default Block Device I/O Scheduler can be also set by the kernel parameter `elevator=`. For example, `elevator=deadline` changes the I/O Scheduler to `deadline`.

10 For More Information

A lot of information about real-time implementations and administration can be found on the Internet. The following list contains several selected links:

- More detailed information about the real-time Linux development and an introduction how to write a real-time application can be found in the real-time Linux community Wiki. http://rt.wiki.kernel.org, http://rt.wiki.kernel.org/index.php/HOWTO:_Build_an_RT-application
- The `cpuset` feature of the kernel is explained in `/usr/src/linux/Documentation/cgroups/cpusets.txt`. More detailed documentation is available from http://lwn.net/Articles/127936/. -->
- For more information about the deadline I/O scheduler, refer to https://en.wikipedia.org/wiki/Deadline_scheduler. In your installed system, find further information in `/usr/src/linux/Documentation/block/deadline-iosched.txt`.
- The CFQ I/O scheduler is covered in detail in http://en.wikipedia.org/wiki/CFQ and `/usr/src/linux/Documentation/block/cfq-iosched.txt`.

A GNU Licenses

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

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 noncommercially. 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 in order 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 formats 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 noncommercially, 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 http://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 &ldquo;GNU
Free Documentation License&rdquo;.
```

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.