# YAROS

A RTOS for embedded systems

**by Olivier Dion**

# Table of Contents

# Prologue

A RTOS (Real-Time Operating System) is by definition an *OS* that must deal with real-time deadlines for its applications. Those deadlines can either be hard or soft. A hard deadline is one that will provoke a system failure if missed, therefore a RTOS that respects hard deadlines must guaranteed that all hard deadline are met. A soft deadline is one that will not provoke a system failure if missed, but the RTOS must try to meet its requirement.

YAROS (Yet Another Real-time Operating System) aims to be a soft RTOS for embedded systems for the AVR architecture. I don't personally like the letters OS in RTOS, because in the case of YAROS, it's simply not true. YAROS is a kernel for embedded systems. It provides abstraction over devices drivers, tasks scheduling, system interruptions, etc. It does not provides system applications or user applications, like an OS would do.

I created YAROS during one of my course, out of boredom. After reading a few books on OS design, I decided to test my skills and knowledges by making my own kernel. One can says that I was highly inspired by Linux, and it's entirely true. I had no idea how to start, and I though that following the step of the giant would gave me a better idea of where to go. I've used YAROS during the whole semester for every assignments and the final project. It was for me a way to test and improve YAROS. This is the past, but I want to continue this project to see where it can go and learn more from it.

I would like to thank **Robert Love** for his book *Linux Kernel Development*, that helped me understand the internal of Linux.

## How to Read this Manual

You have to read this manual's sections linearly. If you skip a section, you will not understand the next one. But, you can skip chapter without getting too much confuse. Although, it's recommended to not skip chapter.

## Improvement and Suggestion

This is my first RTOS, but also my first manual. If you have any suggestion or improvement for either YAROS or this manual, please feel free to send your comments to `olivier.dion@polymtl.ca`.

# 1 Concepts of YAROS

This chapter will explain to you the concept of YAROS, without going into the internal details. After reading this chapter, you should have a clear idea on what is the purpose of YAROS, and the concepts around it.

## 1.1 C Dialect

YAROS uses *GNU* extensions for C. Notable usage are function attributes, designated initializers and inline assembly. YAROS also use flexible array member, which is not compatible with C++. This was a design decision because YAROS needs everything to be compile time. Although it would be possible to accomplish the same thing using templates, it would also require to create a base type for the task structure. The best thing would be to implement a memory manager to avoid using those compile time constraints. Finally, here's a list of pros and cons for using C++.

- Pros
  - Template meta-programming
- Cons
  - Casting is not enough permissive
  - Debugging mangled names is a real pain
  - Operators overloading are useless
  - Namespace are useless
  - Default arguments are useless
  - Exceptions and virtual tables are banned
  - The whole C++ standard library and the STL are banned
  - Less portable.

As you can see, the only pros for using C++ is template for better meta-programming. Using C make it more portable and this is very important for YAROS. The cons outweigh the pros, in my opinion, and therefore, only C shall be use for YAROS.

## 1.2 Autogen

YAROS makes extensive use of **GNU autogen** to keep files synchronized. Most of them are use for developping, therefore you don't need to know them. There's one exeception to this. To generate your kernel configuration, you will need to generate a definition file that will be use by autogen to generate the header for your kernel. Chapter 2 [Configuring], page 9.

## 1.3 Definition of a Task

A task in YAROS is what we can refer to as a thread. There are 3 properties that you should consider when developing a task. Its priority, its niceness and its stack size.

### 1.3.1 Task's Priority

A task's priority range between 0 and 7 inclusive. The lower the value is, the higher is the priority. As for now, the priority only dictated the order of the task in the running queue. This behavior might change in the future.

### 1.3.2 Task's Niceness

A task's nice value range between 0 and 15 inclusive. It determines how much CPU time his allocated for the task before rescheduling. We said that a task is nice to other if its nice value is high. This means that the task will be running less time before scheduling to another task. On the opposite, a task that is not nice, *i.e.* with a low value of nice, will run longer.

### 1.3.3 Task's Stack Size

The stack size is probably the most important value to evaluate. Underestimate the value and you might overflow and corrupt your RAM, resulting in an undefined behavior. Overestimate and you will waste RAM space. A good way to start is to put a high safe value. Then, reduce the stack size until there's a problem. From there, just return to your last safe point and add some padding just to be extra careful. Stack overflow is the most common and dangerous problem, be very cautious.

Note that the size of your stack is added with the minimal stack value, which is 40 bytes. This minimal stack is to ensure there's enough space for context switch. 32 registers + SREG + return address + kill_self address = 37 bytes. There's a little of padding just to be extra cautious.

## 1.4 The Scheduler

YAROS is a time-sharing system. It follows a simple [round-robin], page 15, policy and is therefore preemptive. It's athought possible for task to be cooperative by forcing scheduling.

Because of the scheduling policy, you should put the task with the smallest time slice at the begining of the running queue.

## 1.5 Synchronizing Tasks

To synchronize tasks between them, you will have to use a mutex. Mutex in YAROS are a single byte wrap in a structure. Because you only need 1 bit to lock and unlock, every mutex has 8 keys. Therefore, every time you want to use a mutex, you will need to provide a pointer to the structure and the key that you want to acquired/released.

## 1.6 Functions

This section will describe to you the interface that you can use with YAROS. Functions that are tagged as *Atomic* are guaranteed to be executed without rescheduling. Functions that are tagged as *Macro* are composite of regular or atomic function.

### 1.6.1 Kernel Control

These functions manipulate the kernel.

`void init_kernel` (*void*)

> Initiliaze the kernel. This should be the first function you call in your main.
>
> **Warning:** Using YAROS without initializing the kernel first result in undefined behavior.

`void panic_kernel` (*void*)

> Stop all system activities and print ⊣ KERNEL PANIC.
>
> **Warning:** The kernel can not recover from a panic.

`void run_kernel` (*void*)

> Launch the kernel scheduler. This should be the last thing in your main.
>
> **Warning:** If no task is scheduled, the kernel will spin a loop until a task is scheduled.

### 1.6.2 Time Control

These functions help you with getting the time.

`U64 clk` (*void*)

> Return the time in milliseconds since the kernel was launched.

`jiffy_t tick` (*void*) [Atomic]

> Return the number of jiffies ellapsed since the kernel was launched.

### 1.6.3 Task and Scheduler

These functions give you control over the tasks and the scheduler

`void init_task` (*struct task* *`task`, *taskfunc* `func`, *void* *`data`)

> Initialize the task pointed by `task` with function entry point pointed by `func` and parameter passed to the entry point, pointed by `data`.
>
> **Warning:** `task` and `func` can not be `NULL`.

`void kill_self` (*void*) [Atomic]

> Kill the calling task and reschedule.
>
> **Warning:** Killing the last running task will triggered a kernel panic.

`void kill_task` (*struct task* *`task`) [Atomic]

> Kill the task pointed by `task`.
>
> **Note:** If `task` is `NULL`, kill the calling task.
>
> **Warning:** Killing the last running task will triggered a kernel panic.

`void reschedule` (*void*) [Atomic]

> Force rescheduling, *i.e.* do not wait for the kernel to switch the context.
>
> **Note:** Rescheduling when there's only one task running has no effect.

`void resume_task` (*struct task* *`task`) [Atomic]

> Put the task pointed by `task` in the running queue. If `NULL` is passed, the calling task is put in the running queue.
>
> **Note:** Resuming a task that is already running has no effect.

`run_task` (`TASK`, `FUNC`, `DATA`) [Macro]

Initialize and resume a task.

$\mapsto$ ({ init_task(`TASK`, `FUNC`, `DATA`); resume_task(`TASK`); })

`void suspend_task` (*struct task* *`task`) [Atomic]

Put the task pointed by `task` in the sleeping queue. If `NULL` is passed, the calling task is put in the sleeping queue.

**Note:** Suspending a task that is already sleeping has no effect.

**Warning:** As for now, if the last running task is suspending itself, the kernel panic. This might change in the future by putting the system in low energy mode.

`void wait` (*jiffy_t* `delay`)

Reschedule the calling task until `delay` has passed.

### 1.6.4 Mutex

These functions help you synchronize task between them.

`void spinlock` (*volatile struct mutex *mutex, U8 key*)

Try to acquire a mutex pointed by `mutex` with key `key`.
Return `OK` on success, `-EBUSY` if the mutex is already acquired.

**Warning:** This should never be call in a ISR.

`void spinlock` (*volatile struct mutex *mutex, U8 key*)

Loop until the mutex pointed by `mutex` with key `key` is acquired.

**Warning:** Those type of lock are not recursive! To avoid deadlock, always release them before locking them again in the same task.

`void unlock` (*volatile struct mutex *mutex, U8 key*)

Unlock the mutex pointed by `mutex` with key `key`.

**Note:** The task is then reschedule. This is a design decision so that other task that are waiting for the mutex have a chance to lock it.

# 2 Kernel Configuration

CONFIG_JIFFY [Config Option]

# 3 Examples

## 3.1 Defining the Tasks

```
#include "YAROS.h"

/* foo has priority level 0,
   nice value of 5 and a stack of 50 bytes */
struct task foo = TASK(TASK_P0, TASK_N5, 50);

/* foo has priority level 4,
   nice value of 9 and a stack of 75 bytes */
struct task bar = TASK(TASK_P4, TASK_N9, 75);

OS_TASK void
do_foo(void *task);

OS_TASK void
do_bar(void *nil);
```

## 3.2 A Basic Main

```
int
main(int argc, char *argv[])
{
    /* Initialize the Kernel */
    init_kernel();

    /* Foo is scheduled */
    run_task(&foo,
             do_foo,
             &bar);

    /* Bar is suspended */
    init_task(&bar,
              do_bar,
              NULL);

    /* krun never return */
    run_kernel();
}
```

## 3.3 The Tasks Implementation

```
OS_TASK void
do_foo(void *bar_task)
{
    static const char foo_str[] = {'f', 'o', 'o', '\n'};

    while (1) {

        /* Write "foo_str\n" to USART 0 */
        write_usart(0, foo_str, sizeof(foo_str));

        /* Resume bar */
        resume_task(&bar);

        /* Wait 1 second */
        wait(HZ);

        /* Suspend bar */
        suspend_task(&bar);
    }
}

OS_TASK void
do_bar(void *nil)
{
    static const char bar_str[] = {'b', 'a', 'r', '\n' };

    while (1) {

        /* Write "bar\n" to USART 0 */
        write_usart(0, bar_str, sizeof(bar_str));

        /* Wait 1/2 second */
        wait(HZ / 2);
    }
}
```

# 4 The Internal of YAROS

This chapter will explain to you how YAROS work internally. After reading this chapter, you should be able to understand how the scheduler work with tasks, how interupts and devices are manage and how mutexes work.

## 4.1 Structures

`dlist` [struct]

```
{
    struct dlist *next, *prev;
};
```

Doubly linked list intrusive.

`task` [struct]

```
{
    struct dlist self;
    U16 * volatile stack_pointer;
    U8 running:1;
    U8 priority:3;
    U8 nice:4;
    sstack_t size;
    U8 stack[ ];
};
```

A task is the central structure in YAROS. The member `self` denote a node in a queue. The node has 3 states. It can either be attached to itself, therefore YAROS has no knowledge about the task. It can be attached to the running queue. Or it can be attached to the sleeping queue.

**Note:** The member `self` was called `this` before. But because C++ use it as a keyword, it was changed to `self`.

**Note:** The type sstack_t means **size stack type**. It is an horrible name, so if you have a better idea for a typename, please submit it.

## 4.2 Global Variales

This section will enumerate all global variables used by YAROS. All global variables are defined in `autogen/def/global.def` . As an user you should never try to access those variables directly.

`jiffy_t jiffies` [Global]

Just like a clock has a finite precision, YAROS has it own limit when it come to counting time, and this limit is call a jiffy. We define a jiffy to be the smallest unit of time for the system. The variable `jiffies` hold the amount of jiffies ellapsed since the system had started. This variable is increment `HZ` times per second, `volatile U8` *time_slice* [Global]

This variable represent the number of jiffies left for the current running task before rescheduling is done by the system.

`struct dlist` *sleeping_queue* [Global]

Represent the queue of task that are sleeping.

`struct dlist` *running_queue* [Global]

Represent the queue of task that are ready for scheduling.

`struct dlist` ** volatile current_task* [Global]

A pointer to the current task running.

`U16 *` *volatile stack_pointer* [Global]

Represent the address of the top of the stack of the current running task.

# Appendix A Glossary

Round-Robin Policy
: In a round-robing scheduling policy, every task as a time slice. When this time slice is expired, the task is putted at the tail of the queue and the next task is scheduled.

RTOS
: A RTOS is a real-time operating system. It must be able to meet soft and hard deadline. See [Prologue], page 1.

  It has its own context (stack of execution) and is independent of other tasks. Of course just like thread, tasks can be synchronize between them via mutexes or semaphores. See Section 1.3 [Task], page 3.

Xlist
: A Xlist is a list of terms wrapped in a X macro individually. Xlist are useful to synchronize list of elements between files that use them but expand the X macro differently. See ⟨undefined⟩ [Xlist], page ⟨undefined⟩.

YAROS
: YAROS is a RTOS for embedded system for the AVR family. It's the subject of this manual. See [Prologue], page 1.

# Appendix B Copying This Manual



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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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/licenses/`.

    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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
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 1.3
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.

# Function and Macros Index

# List of Data Structures

## D



## T

# List of Global Variables

## *



## C



## D



## J



## U