RhythmOS Reference Manual

Reference Documentation for RhythmOS
Edition 0.1, for `RhythmOS` Version 0.1.
05 October 2011

Dustin Dorroh (dustindorroh@gmail.com)

This text is a description of the features that are present in the `RhythmOS` kernel (version 0.1, 05 October 2011).

This is Edition 0.1, last updated 05 October 2011, of *RhythmOS Reference Manual*, for `RhythmOS`, Version 0.1.

Copyright © 2012 Dustin Dorroh (dustindorroh@gmail.com)

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

> 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, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table of Contents

1 Overview

This manual is for `RhythmOS` (version 0.1, 05 October 2011), which is a an attempt at a barebones UNIX-like kernel. This document intends to cover the how to build, use, and develop on Rhythmos as well as explain in depth some of the kernel internals. For up to date news check the project home which is located at http://code.google.com/p/rhythmos/

1.0.1 RhythmOS Features

- Small C library
- Kernel mode and user mode
- Basic Multitasking through the use of context switches and memory protection
- Dynamic memory allocation (using buddy allocation technique)
- Virtual memory
- Kernel Paging
- UNIX-like system calls:

    Process management: `fork`, `kill`, `execve`, `waitpid` ...
    - Interprocess Communication (IPC): pipe, dup
- Read only filesystem

1.0.2 Source Code

Use git to clone a copy of the source tree. Use the command below to clone the RhythmOS repository.

```
git clone https://code.google.com/p/rhythmos/
```

2 Toolchain Setup

Build a RhythmOS Toolchain

If your native architecture is x86 then you will most likely already have a toolchain that will work for RhythmOS. If your architecture is something else you will have to build an toolchain to cross compile to x86.

You can find your architecture in Linux from one of the following commands:

```
$ arch x86_64

$ lscpu | grep Arch
Architecture:          x86_64

$ uname -m
x86_64
```

2.1 Build Environment

Export build variables

```
$ export PREFIX=/usr/local/cross
$ export TARGET=i386-elf
```

Create build directories for gcc binutils and newlibc

```
$ mkdir -p build/gcc,binutils,newlib
$ sudo mkdir /usr/local/cross
```

2.2 Source Download

Download and extract Binutils and gcc and newlibc

```
$ wget http://ftp.gnu.org/gnu/binutils/binutils-2.21.1a.tar.bz2
$ wget http://ftp.gnu.org/gnu/gcc/gcc-4.6.2/gcc-4.6.2.tar.bz2
$ wget ftp://sources.redhat.com/pub/newlib/newlib-1.20.0.tar.gz
$ tar xjvf binutils-2.21.1a.tar.bz2
$ tar xjvf gcc-4.6.2.tar.bz2
$ tar xzvf newlib-1.20.0.tar.gz
```

Download gmp mpfr and mpc required by gcc

```
$ wget ftp://ftp.gmplib.org/pub/gmp-5.0.2/gmp-5.0.2.tar.bz2
$ wget http://www.mpfr.org/mpfr-current/mpfr-3.1.0.tar.bz2
$ wget http://www.multiprecision.org/mpc/download/mpc-0.9.tar.gz
```

Extract gmp mpfr and mpc into the gcc directory so that gcc will build them automatically

```
$ cd gcc-4.6.2
$ tar xjvf ../gmp-5.0.2.tar.bz2 && mv gmp-5.0.2/ gmp
$ tar xjvf ../mpfr-3.1.0.tar.bz2 && mv mpfr-3.1.0/ mpfr
$ tar xzvf ../mpc-0.9.tar.gz  && mv mpc-0.9/ mpc
$ cd ..
```

2.3 Binutils

Configure Binutils

```
 cd build/binutils/
$ ../../binutils-2.21.1/configure --target=$TARGET \
                                  --prefix=$PREFIX \
                                  --disable-nls
```

Make and Install Binutils

```
$ make all
$ sudo make install
$ cd ../..
```

2.4 GCC

Setup Environment

```
$ export PATH=$PATH:$PREFIX/bin
```

Configure GCC

```
$ cd build/gcc
$ ../../gcc-4.6.2/configure  --target=$TARGET \
                             --prefix=$PREFIX \
                             --disable-nls \
                             --enable-languages=c \
                             --without-headers
```

Make And Install GCC

```

make all-gcc
$ sudo make install-gcc
$ cd ../..
```

2.5 C Library

Configure Newlib

```
$ cd build/newlib
$ ../../newlib-1.20.0/configure --target=$TARGET \
                                --prefix=$PREFIX
```

Make and Install Newlib

```
$ make all
$ sudo make install
```

Run command below if `sudo make install` cant find utils in '`/usr/cross/bin`'

```
$ sudo ln -s /usr/local/cross/bin/i386-elf-* /usr/local/bin
$ sudo make install
```

3 Building RhythmOS

From the top level project directory execute the configure script.

```
$ ./configure
```

To build everything just type '`make`'. This should build the kernel image '`kernel.img`', the file system image '`filesystem.img`', and the grub boot image '`grub.img`'

```
$ make
```

To only build root file system type '`make fs`'. To only build the bootloader type '`make boot`'.

3.1 Build Internals

3.1.1 Linking

The linker combines input files into a single output file. The output file and each input file are in a special data format known as an object file format. Each file is called an object file. The output file is often called an executable, but for our purposes we will also call it an object file. Each object file has, among other things, a list of sections. We sometimes refer to a section in an input file as an input section; similarly, a section in the output file is an output section.

You can see the sections in an object file by using the `objdump` program with the '`-h`' option. Every object file also has a list of symbols, known as the symbol table. A symbol may be defined or undefined. Each symbol has a name, and each defined symbol has an address.

Linker Invocation

```
# Link using 'link.ld'
$(KERNEL_IMG): $(KERNEL_OBJECTS)
    $(LD) '-T' 'link.ld' '-o' $(KERNEL_IMG) $(KERNEL_OBJECTS)
```

The '`-T`' option instructs the linker `ld` to use the commands in the script file '`link.ld`'. It specifies the various segments used by the program. In the file we specify 3 segments:

- `.text` - Code segment
- `.data` - Data segment
- `.bss` - Stack segment

link.ld

```
OUTPUT_FORMAT("binary")
ENTRY(start)
phys = 0x00100000;
SECTIONS
{
  .text phys : AT(phys) {
    code = .;
```

```
        *(.text)
        *(.rodata)
        . = ALIGN(4096);
      }
      .data : AT(phys + (data - code))
      {
        data = .;
        *(.data)
        . = ALIGN(4096);
      }
      .bss : AT(phys + (bss - code))
      {
        bss = .;
        *(.bss)
        . = ALIGN(4096);
      }
     . = ALIGN(4096);
      end = .;
    }
```

In order for `RhythmOS` to be able to boot properly, three different binary images are needed (see Chapter 3 [Building RhythmOS], page 4). The *images* also have to be made in a particular order due to the build dependencies, thus the build process naturally falls into a three step process:

1. Kernel Image (see [Kernel Image], page 5)
2. Filesystem Image (see [Filesystem Image], page 5)
3. Bootloader Image (see [Bootloader Image], page 6)

Kernel Image

Executing `make` compiles individual source (`.c`) into object files (`.o`). The linker (`ld`) is then used next to link the object files into one binary '`src/kernel.img`'. See Section 5.1 [Assembly], page 9.

Filesystem Image

The `make fs` target executes the build script '`mk_filesystem_image.sh`' located in '`src`', the project's source directory.

The srcipt '`mk_filesystem_image.sh`' creates empty directories in the project tree, which will be a staging place for the root files that will be used by RhythmOS. Next, the programs that will run on the kernel (ie `sh`, `ls`, `cat`, etc ...) will be copied to the appropiate staging directory[1]. An example of how rootfs might be structured.

```
$ tree rootfs/
rootfs/
|-- bin
|   |-- cat
```

[1] For example binaries will placed in '`rootfs/bin`' directory on the host machine

```
|   |-- find
|   |-- ls
|   |-- pwd
|   `-- sh
|-- etc
`-- usr
```

An image, based off of the staging filesystem, is then built up in memory[2]. When complete the filesystem image will be located '`src/filesystem.img`'.

Bootloader Image

In order to run the kernel, its necessary to use a boot loader. This is the first thing that runs when a computer starts, and is responsible for loading the kernel file into memory and instructing the processor to start executing it.

The '`grub.img`' is the GRUB image that we will as are bootloader. This file contains our kernel image and our file system image.

```
$ mkdir floppy
$ sudo mount src/grub.img floppy
$ tree floppy/
floppy/
|-- boot
|   `-- grub
|       |-- fat_stage1_5
|       |-- menu.lst
|       |-- stage1
|       `-- stage2
|-- filesystem.img
`-- kernel.img

2 directories, 6 files
```

The '`menu.lst`' contains the menu settings for GRUB along with the boot commands for each entry.

```
$ cat floppy/boot/grub/menu.lst
timeout 5
title RhythmOS
root (fd0)
kernel /kernel.img
modulenounzip (fd0)/filesystem.img

$ sudo mount src/grub.img floppy
```

If you can't mount '`grub.img`' to edit it this is OK because the '`grub.img`' that comes with the source is already formated and for updating the image we use Mtools which doesn't require special permissions. See below for details...

[2] *See* '`fstool.c`' in the project source to see how the image is created

Using '`mk_boot_image.sh`' makes sure to delete the old kernel image and filesystem image from the GRUB image. It then writes the kernel image ('`kernel.img`') and filesystem image ('`kernel.img`') data to the GRUB image '`grub.img`'.

```
# Delete old images
$ mdel -i grub.img ::kernel.img || true
$ mdel -i grub.img ::filesystem.img || true

# Copy updated images
$ mcopy -i grub.img kernel.img :: || exit 1
$ mcopy -i grub.img filesystem.img :: || exit 1
```

The '`make boot`' target executes '`mk_boot_image.sh`' for you.

4 Running RhythmOS

Once the kernel has been copied to the disk image, you can then run it under qemu using one of the following commands:

```
$ qemu -fda src/grub.img
$ qemu -daemonize -fda src/grub.img       # daemonize qemu
$ qemu -monitor stdio -fda src/grub.img   # output qemu debug on stdio
```

This same command can also be run through the `make` target `run`;

```
$make run
```

To run without using the '`grub.img`' you can pass the kernel image and filesystem image seperatly to QEMU.

```
$ qemu -kernel src/kernel.img -initrd src/filesystem.img
```

5 Programming the kernel

5.1 Assembly

5.2 C

5.2.1 The Run-Time Library

A major part of writing code for your a kernel is getting suckered into rewriting the *run-time library*, also known as `libc`. This is because the *RTL* is the most OS *dependent* part of the compiler package: the C RTL provides enough functionality to allow you to write *portable* programs, but the inner workings of the RTL are *dependent* on the OS in use.

The aim is to replicate the library defined by the *ISO*[1] C standard because this will make porting programs to my kernel easier. Writing a non-standard library will only result in having to re-write any application your trying to port; not to mention many open source projects that do conform to these standards will be open for me to use.

[1] International Standardization Organization

6 Multitasking

7 Memory

8 Dynamic Memory Allocation

9 System calls

> Source for system calls are located in 'syscall.c', 'fscalls.c', 'filedesc.c', 'unixproc.c', 'pipe.c'.

When the interrupt occurs, the CPU switches to kernel mode, and executes the interrupt handler, which is actually implemented in assembler within '`start.s`'. This pushes the register values onto the stack, and then calls the `interrupt_handler` function, which is implemented in `C`. The registers that were put onto the stack are available as a parameter to this function, as a regs object. The interrupt handler then calls the `syscall` function, which inspects the registers to determine which system call was requested, and then dispatches to the appropriate handler function.

For example the system call `write` is implemented by the handler function `syscall_write`. The parameters to the system call may be accessed by looking at the process's stack. One of the saved registers is the stack pointer, which the syscall function uses to determine the location in memory of the parameters, which it then passes to the handler function. The handler for the system call then performs whatever actions are necessary, which can include privileged operations such as writing to any area of memory, since this code runs in kernel mode. Once the handler function returns, control is passed backwards along the same path, until it returns to the process that was previously executing. In some cases, such as the read or exit system calls, the process may have been killed or suspended. If this is the case, syscall performs a context switch, so that a different process will be executed when the interrupt handler returns.

The `exit` system call is used by a process to terminate itself. When this call is made, the `kill_process` function is called, and the handler function returns the special value `-ESUSPEND`, indicating to the syscall dispatching function that the current process is no longer running, and it should perform a context switch. Processes should always call `exit` as their last action, to indicate the kernel that they have no more instructions to be executed.

List of all the syscalls RhythmOS has available:

```
KERNEL              USER         LOCATION
-----------------------------------------
syscall_pipe        pipe         pipe.c
syscall_getpid      getpid       syscall.c
syscall_exit        exit         syscall.c
syscall_write       write        syscall.c
syscall_read        read         syscall.c
syscall_geterrno    geterrno     syscall.c
syscall_brk         brk          syscall.c
syscall_send        send         syscall.c
syscall_receive     receive      syscall.c
syscall_kill        kill         syscall.c
syscall_close       close        filedesc.c
syscall_dup2        dup2         filedesc.c
syscall_fork        fork         unixproc.c
syscall_execve      execve       unixproc.c
syscall_waitpid     waitpid      unixproc.c
syscall_stat        stat         fscalls.c
syscall_open        open         fscalls.c
syscall_chdir       chdir        fscalls.c
syscall_getcwd      getcwd       fscalls.c
syscall_getdent     getdent      fscalls.c
```

10 I/O

11 Filesystems

The contents of the root filesystem must be adequate to *boot*, *restore*, and/or *recover* the system.

- To *boot* a system, enough must be present on the root partition to mount other filesystems. This includes utilities, configuration, boot loader information, and other essential start-up data.
- For the *recovery* of a system, utilities are needed by the user to diagnose and reconstruct the damaged system; these utilities need be present on the root filesystem.
- To *restore* a system, those utilities needed to restore from system backups (on floppy, tape, etc.) must be present on the root filesystem.

See The Root Filesystem Standards for more a more indepth analysis.

Below is an example of a root filesystem '`/`'. Ones marked with an asterisk (`*`) are currently beind used by `RhythmOS`.

`Rootfs file system structure`

'`/bin`'	* Essential command binaries
'`/boot`'	Static files of the boot loader
'`/dev`'	Device files
'`/etc`'	* Host-specific system configuration
'`/lib`'	Essential shared libraries and kernel modules
'`/media`'	Mount point for removeable media
'`/mnt`'	Mount point for mounting a filesystem temporarily
'`/opt`'	Add-on application software packages
'`/sbin`'	Essential system binaries
'`/srv`'	Data for services provided by this system
'`/tmp`'	Temporary files
'`/usr`'	* Secondary hierarchy
'`/var`'	Variable data

12 Reporting bugs

To report bugs or suggest enhancements for RhythmOS, please send electronic mail to `dustindorroh@gmail.com`.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
http://fsf.org/

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 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 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. 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.

Concept index