



---

Documentation Library

## **SRR Module**

**The srripc Linux Kernel Module, Version 1.4.43**

**Cogent Real-Time Systems, Inc.**

**August 11, 2010**

## **SRR Module: The srripc Linux Kernel Module, Version 1.4.43**

This library is an interface to the **srripc** synchronous message passing module for the Linux kernel written by Sam Roberts and Andrew Thomas of Cogent Real-Time Systems, Inc., with many helpful additions from the user community. It provides synchronous message passing between processes and/or threads for the Linux operating system. It also enables queued asynchronous event notification (proxies), timed delivery of proxies and signals, triggering of proxies between tasks, and user-space interrupt handling. Synchronous message passing is a fast, flexible, and robust IPC mechanism, particularly useful for building systems composed of multiple co-operating processes. This implementation has been inspired by the QNX™ operating system, and includes a QNX 4 compatible API layer.

Published August 11, 2010  
Cogent Real-Time Systems, Inc.  
162 Guelph Street, Suite 253  
Georgetown, Ontario  
Canada, L7G 5X7

Toll Free: 1 (888) 628-2028  
Tel: 1 (905) 702-7851  
Fax: 1 (905) 702-7850

Information Email: [info@cogent.ca](mailto:info@cogent.ca)  
Tech Support Email: [support@cogent.ca](mailto:support@cogent.ca)  
Web Site: [www.cogent.ca](http://www.cogent.ca)

Copyright © 1995-2011 by Cogent Real-Time Systems, Inc.

### Revision History

Revision 1.4.43-1 October 2006  
Replaced spin locks with mutexes, improved testing.  
Revision 1.4.21-1 February 2005  
Improved error handling, timers, and memory usage.  
Revision 1.4.18-1 June 2004  
Changes related to 2.6 kernel, bug fixes.  
Revision 1.4.16-1 June 2004  
Able to handle threads, added shared library.  
Revision 1.4.6-1 August 2002  
Incorporated interrupt handling.  
Revision 1.4.0-1 May 2002  
Corrected qnx\_\* functions, and others.  
Revision 1.3.12-1 May 2000  
Initial release of documentation.

## **Copyright, trademark, and software license information.**

### **Copyright Notice**

© 1995-2011 Cogent Real-Time Systems, Inc. All rights reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written consent of Cogent Real-Time Systems, Inc.

Cogent Real-Time Systems, Inc. assumes no responsibility for any errors or omissions, nor do we assume liability for damages resulting from the use of the information contained in this document.

### **Trademark Notice**

Cascade DataHub, Cascade Connect, Cascade DataSim, Connect Server, Cascade Historian, Cascade TextLogger, Cascade NameServer, Cascade QueueServer, RightSeat, SCADALisp and Gamma are trademarks of Cogent Real-Time Systems, Inc.

All other company and product names are trademarks or registered trademarks of their respective holders.

## **END-USER LICENSE AGREEMENT FOR COGENT SOFTWARE**

**IMPORTANT - READ CAREFULLY:** This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Cogent Real-Time Systems Inc. ("Cogent") of 162 Guelph Street, Suite 253, Georgetown, Ontario, L7G 5X7, Canada (Tel: 905-702-7851, Fax: 905-702-7850), from whom you acquired the Cogent software product(s) ("SOFTWARE PRODUCT" or "SOFTWARE"), either directly from Cogent or through one of Cogent's authorized resellers.

The SOFTWARE PRODUCT includes computer software, any associated media, any printed materials, and any "online" or electronic documentation. By installing, copying or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree with the terms of this EULA, Cogent is unwilling to license the SOFTWARE PRODUCT to you. In such event, you may not use or copy the SOFTWARE PRODUCT, and you should promptly contact Cogent for instructions on return of the unused product(s) for a refund.

### **SOFTWARE PRODUCT LICENSE**

The SOFTWARE PRODUCT is protected by copyright laws and copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. **EVALUATION USE:** This software is distributed as "Free for Evaluation", and with a per-use royalty for Commercial Use, where "Free for Evaluation" means to evaluate Cogent's software and to do exploratory development and "proof of concept" prototyping of software applications, and where "Free for Evaluation" specifically excludes without limitation:

- i. use of the SOFTWARE PRODUCT in a business setting or in support of a business activity,
- ii. development of a system to be used for commercial gain, whether to be sold or to be used within a company, partnership, organization or entity that transacts commercial business,
- iii. the use of the SOFTWARE PRODUCT in a commercial business for any reason other than exploratory development and "proof of concept" prototyping, even if the SOFTWARE PRODUCT is not incorporated into an application or product to be sold,
- iv. the use of the SOFTWARE PRODUCT to enable the use of another application that was developed with the SOFTWARE PRODUCT,
- v. inclusion of the SOFTWARE PRODUCT in a collection of software, whether that collection is sold, given away, or made part of a larger collection.
- vi. inclusion of the SOFTWARE PRODUCT in another product, whether or not that other product is sold, given away, or made part of a larger product.

2. **COMMERCIAL USE:** COMMERCIAL USE is any use that is not specifically defined in this license as EVALUATION USE.

3. **GRANT OF LICENSE:** This EULA covers both COMMERCIAL and EVALUATION USE of the SOFTWARE PRODUCT. Either clause (A) or (B) of this section will apply to you, depending on your actual use of the SOFTWARE PRODUCT. If you have not purchased a license of the SOFTWARE PRODUCT from Cogent or one of Cogent's authorized resellers, then you may not use the product for COMMERCIAL USE.

- A. **GRANT OF LICENSE (EVALUATION USE):** This EULA grants you the following non-exclusive rights when used for EVALUATION purposes:

Software: You may use the SOFTWARE PRODUCT on any number of computers, either stand-alone, or on a network, so long as every use of the SOFTWARE PRODUCT is for EVALUATION USE. You may reproduce the SOFTWARE PRODUCT, but only as reasonably required to install and use it in accordance with this LICENSE or to follow your normal back-up practices.

Subject to the license expressly granted above, you obtain no right, title or interest in or to the SOFTWARE PRODUCT or related documentation, including but not limited to any copyright, patent, trade secret or other proprietary rights therein. All whole or partial copies of the SOFTWARE PRODUCT remain property of Cogent and will be considered part of the SOFTWARE PRODUCT for the purpose of this EULA.

Unless expressly permitted under this EULA or otherwise by Cogent, you will not:

- i. use, reproduce, modify, adapt, translate or otherwise transmit the SOFTWARE PRODUCT or related components, in whole or in part;
- ii. rent, lease, license, transfer or otherwise provide access to the SOFTWARE PRODUCT or related components;
- iii. alter, remove or cover proprietary notices in or on the SOFTWARE PRODUCT, related documentation or storage media;
- iv. export the SOFTWARE PRODUCT from the country in which it was provided to you by Cogent or its authorized reseller;
- v. use a multi-processor version of the SOFTWARE PRODUCT in a network larger than that for which you have paid the corresponding multi-processor fees;
- vi. decompile, disassemble or otherwise attempt or assist others to reverse engineer the SOFTWARE PRODUCT;
- vii. circumvent, disable or otherwise render ineffective any demonstration time-outs, locks on functionality or any other restrictions on use in the SOFTWARE PRODUCT;
- viii. circumvent, disable or otherwise render ineffective any license verification mechanisms used by the SOFTWARE PRODUCT;
- ix. use the SOFTWARE PRODUCT in any application that is intended to create or could, in the event of malfunction or failure, cause personal injury or property damage; or
- x. make use of the SOFTWARE PRODUCT for commercial gain, whether directly, indirectly or incidentally.

**B. GRANT OF LICENSE (COMMERCIAL USE):** This EULA grants you the following non-exclusive rights when used for COMMERCIAL purposes:

Software: You may use the SOFTWARE PRODUCT on one computer, or if the SOFTWARE PRODUCT is a multi-processor version - on one node of a network, either: (i) as a development systems for the purpose of creating value-added software applications in accordance with related Cogent documentation; or (ii) as a single run-time copy for use as an integral part of such an application. This includes reproduction and configuration of the SOFTWARE PRODUCT, but only as reasonably required to install and use it in association with your licensed processor or to follow your normal back-up practices.

Storage/Network Use: You may also store or install a copy of the SOFTWARE PRODUCT on one computer to allow your other computers to use the SOFTWARE PRODUCT over an internal network, and distribute the SOFTWARE PRODUCT to your other computers over an internal network. However, you must acquire and dedicate a license for the SOFTWARE PRODUCT for each computer on which the SOFTWARE PRODUCT is used or to which it is distributed. A license for the SOFTWARE PRODUCT may not be shared or used concurrently on different computers.

Subject to the license expressly granted above, you obtain no right, title or interest in or to the SOFTWARE PRODUCT or related documentation, including but not limited to any copyright, patent, trade secret or other proprietary rights therein. All whole or partial copies of the SOFTWARE PRODUCT remain property of Cogent and will be considered part of the SOFTWARE PRODUCT for the purpose of this EULA.

Unless expressly permitted under this EULA or otherwise by Cogent, you will not:

- i. use, reproduce, modify, adapt, translate or otherwise transmit the SOFTWARE PRODUCT or related components, in whole or in part;

- ii. rent, lease, license, transfer or otherwise provide access to the SOFTWARE PRODUCT or related components;
- iii. alter, remove or cover proprietary notices in or on the SOFTWARE PRODUCT, related documentation or storage media;
- iv. export the SOFTWARE PRODUCT from the country in which it was provided to you by Cogent or its authorized reseller;
- v. use a multi-processor version of the SOFTWARE PRODUCT in a network larger than that for which you have paid the corresponding multi-processor fees;
- vi. decompile, disassemble or otherwise attempt or assist others to reverse engineer the SOFTWARE PRODUCT;
- vii. circumvent, disable or otherwise render ineffective any demonstration time-outs, locks on functionality or any other restrictions on use in the SOFTWARE PRODUCT;
- viii. circumvent, disable or otherwise render ineffective any license verification mechanisms used by the SOFTWARE PRODUCT, or
- ix. use the SOFTWARE PRODUCT in any application that is intended to create or could, in the event of malfunction or failure, cause personal injury or property damage.

4. **WARRANTY:** Cogent cannot warrant that the SOFTWARE PRODUCT will function in accordance with related documentation in every combination of hardware platform, software environment and SOFTWARE PRODUCT configuration. You acknowledge that software bugs are likely to be identified when the SOFTWARE PRODUCT is used in your particular application. You therefore accept the responsibility of satisfying yourself that the SOFTWARE PRODUCT is suitable for your intended use. This includes conducting exhaustive testing of your application prior to its initial release and prior to the release of any related hardware or software modifications or enhancements.

Subject to documentation errors, Cogent warrants to you for a period of ninety (90) days from acceptance of this EULA (as provided above) that the SOFTWARE PRODUCT as delivered by Cogent is capable of performing the functions described in related Cogent user documentation when used on appropriate hardware. Cogent also warrants that any enclosed disk(s) will be free from defects in material and workmanship under normal use for a period of ninety (90) days from acceptance of this EULA. Cogent is not responsible for disk defects that result from accident or abuse. Your sole remedy for any breach of warranty will be either: i) terminate this EULA and receive a refund of any amount paid to Cogent for the SOFTWARE PRODUCT, or ii) to receive a replacement disk.

5. **LIMITATIONS:** Except as expressly warranted above, the SOFTWARE PRODUCT, any related documentation and disks are provided "as is" without other warranties or conditions of any kind, including but not limited to implied warranties of merchantability, fitness for a particular purpose and non-infringement. You assume the entire risk as to the results and performance of the SOFTWARE PRODUCT. Nothing stated in this EULA will imply that the operation of the SOFTWARE PRODUCT will be uninterrupted or error free or that any errors will be corrected. Other written or oral statements by Cogent, its representatives or others do not constitute warranties or conditions of Cogent.

In no event will Cogent (or its officers, employees, suppliers, distributors, or licensors: collectively "Its Representatives") be liable to you for any indirect, incidental, special or consequential damages whatsoever, including but not limited to loss of revenue, lost or damaged data or other commercial or economic loss, arising out of any breach of this EULA, any use or inability to use the SOFTWARE PRODUCT or any claim made by a third party, even if Cogent (or Its Representatives) have been advised of the possibility of such damage or claim. In no event will the aggregate liability of Cogent (or that of Its Representatives) for any damages or claim, whether in contract, tort or otherwise, exceed the amount paid by you for the SOFTWARE PRODUCT.

These limitations shall apply whether or not the alleged breach or default is a breach of a fundamental condition or term, or a fundamental breach. Some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, or certain limitations of implied warranties. Therefore the above limitation may not apply to you.

#### 6. **DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS:**

Separation of Components. The SOFTWARE PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer.

Termination. Without prejudice to any other rights, Cogent may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such an event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

7. **UPGRADES:** If the SOFTWARE PRODUCT is an upgrade from another product, whether from Cogent or another supplier, you may use or transfer the SOFTWARE PRODUCT only in conjunction with that upgrade product, unless you destroy the upgraded product. If the SOFTWARE PRODUCT is an upgrade of a Cogent product, you now may use that upgraded product only in accordance with this EULA. If the SOFTWARE PRODUCT is an upgrade of a component of a package of software programs which you licensed as a single product, the SOFTWARE PRODUCT may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.
8. **COPYRIGHT:** All title and copyrights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, text and 'applets', incorporated into the SOFTWARE PRODUCT), any accompanying printed material, and any copies of the SOFTWARE PRODUCT, are owned by Cogent or its suppliers. You may not copy the printed materials accompanying the SOFTWARE PRODUCT. All rights not specifically granted under this EULA are reserved by Cogent.
9. **PRODUCT SUPPORT:** Cogent has no obligation under this EULA to provide maintenance, support or training.
10. **RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a)(1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as appropriate. Manufacturer is Cogent Real-Time Systems Inc. 162 Guelph Street, Suite 253, Georgetown, Ontario, L7G 5X7, Canada.
11. **GOVERNING LAW:** This Software License Agreement is governed by the laws of the Province of Ontario, Canada. You irrevocably attorn to the jurisdiction of the courts of the Province of Ontario and agree to commence any litigation that may arise hereunder in the courts located in the Judicial District of Peel, Province of Ontario.

# Table of Contents

<b>I. Programmers Manual</b>	<b>ix</b>
1. Description	1
1.1. Implementation Background	1
1.2. System Requirements	1
1.3. OS-Specific Behavior	2
1.3.1. Address Space	??
1.3.2. Forks in pre-2.2 Linux Kernels	??
1.3.3. Error Numbers	??
2. Installation	3
2.1. Unpacking the Archive	3
2.2. Testing the SRR Module	3
2.3. Installing the SRR Module	4
2.4. Enabling Auto-Loading for the SRR Module	5
2.5. Example Code	5
3. Copyrights and Disclaimers	8
A. GNU General Public License	9
B. GNU Lesser General Public License	15
<b>II. Reference</b>	<b>23</b>
I. Send/Receive/Reply API Reference	24
SrrSend	25
SrrReceive	26
SrrRelay	27
SrrReply	28
SrrProxyAttach	29
SrrProxyDetach	31
SrrProxyTrigger	32
SrrNameAttach	33
SrrNameDetach	34
SrrNameLocate	35
II. QNX Compatibility API Reference	36
Send	37
Receive	38
Relay	39
Reply	40
qnx_proxy_attach	41
qnx_proxy_detach	42
Trigger	43
qnx_name_attach	44
qnx_name_detach	45
qnx_name_locate	46
qnx_vc_attach	47
III. Undocumented functions	48
SrrClockGettime	49
SrrClockSettime	50
SrrCreceive	51
SrrCreceivemx	52
SrrDebug	53
SrrFd	54
SrrKill	55

SrrPflags.....	56
SrrProxyRemDetach.....	57
SrrReadmsg.....	58
SrrReadmsgmx.....	59
SrrReceivemx.....	60
SrrReg.....	61
SrrReplymx.....	62
SrrSendmx.....	63
SrrTimerDelete.....	64
SrrTimerGettime.....	65
SrrTimerSettime.....	66
SrrUnload.....	67
SrrUnreg.....	68
SrrVcDetach.....	69
SrrWritemsg.....	70
SrrWritemsgmx.....	71
<b>Index.....</b>	<b>??</b>
<b>Colophon.....</b>	<b>73</b>



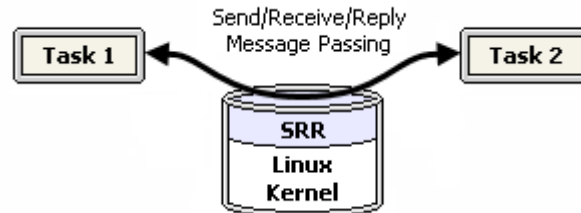
# I. Programmers Manual

## Table of Contents

<b>1. Description</b> .....	<b>1</b>
<b>2. Installation</b> .....	<b>3</b>
<b>3. Copyrights and Disclaimers</b> .....	<b>8</b>
<b>A. GNU General Public License</b> .....	<b>9</b>
<b>B. GNU Lesser General Public License</b> .....	<b>15</b>

# Chapter 1. Description

The SRR Module (**srripc**) is an implementation of synchronous Send/Receive/Reply (S/R/R) message passing as a Linux kernel module. This module provides synchronous message passing among processes and threads, queued asynchronous event notification (proxies), timed delivery of proxies and signals, triggering of proxies between tasks, and user-space interrupt handling.



Synchronous messaging is a very fast means of communication and synchronization between two processes or threads, allowing a system developer to break a problem into separate threads of execution without the need for semaphores, mutexes or shared memory. User-space interrupt handling is implemented using a simple byte code interpreter that runs at interrupt time in the kernel, followed by the delivery of a proxy (scheduled asynchronous notification) to the user program after each interrupt. The user program transmits the byte code to the module at run time, and performs all non-time-critical processing in user space when the proxy is delivered. This makes it possible to write device drivers that service interrupts with no kernel programming. This implementation has been inspired by the QNX™ operating system, and includes a QNX 4 compatible API layer.

This is a beta release of the software. It has been used successfully to run a fairly large set of QNX applications under Linux, but it hasn't been used extensively. We welcome any comments, and hope it will be useful for anybody porting code from QNX to Linux, or wanting to experiment with S/R/R under Linux, perhaps even with the intention of eventually porting to QNX. To view the software version history, please refer to the **CHANGES** file in the distribution.

## 1.1. Implementation Background

The ideas behind this implementation of S/R/R under Linux are simple:

1. Extend the kernel by writing a loadable kernel module.
2. Register with that module by opening its dev file `/dev/srr`, either explicitly with `SrrReg()`, or implicitly by making one of the library calls.
3. Map all (pseudo-) system calls to an `ioctl()` to the driver -- like other system calls the `ioctl()` can block, not block, perform arbitrary reads/writes into the process' address space, etc. It does everything that the QNX S/R/R API appears to do.

The `ioctl()`/module implementation should look familiar to a QNX systems programmer, since it is essentially an inversion of a QNX I/O resource manager. In QNX, S/R/R is used to pass structures holding Posix system call arguments (`read`, `write`, `ioctl`) to the I/O manager. In Linux we use an `ioctl()` to pass structures holding the arguments to QNX system calls to the kernel module.

## 1.2. System Requirements

To run the SRR Module you need to be running a 32-bit Linux kernel of version 2.4.18 or later. Earlier versions of linux 2.4.x series may work. There is no SRR Module support for Linux 2.2.x kernels.

You will also need a C compiler installed on your Linux computer so that you can compile the SRR module into your Linux kernel.

## 1.3. OS-Specific Behavior

As the libraries cover different operating systems, there are a few OS-specific differences in behavior to note.

### 1.3.1. Address Space

It is difficult to access the address space of anything but the currently running process in Linux. In fact, it is not generally possible: the process's memory may be swapped to disk. This means buffering all messages in kernel space before causing a context switch to the process for which the message is destined. Unfortunately this requires all data to be copied from sender to kernel to receiver, whereas under QNX, message passing results in a single copy directly from the sender's address space to the receiver's.

### 1.3.2. Forks in pre-2.2 Linux Kernels

There's a problem with pre-2.2 Linux kernels involving the use of `fork()` when the parent exits, and the child never makes an SRR Module library call. This problem does not exist in the 2.2.0 and later kernels.

In Unix, when a process forks, its file descriptors are duplicated in the child. In Linux this `dup()` is done without the co-operation of the driver to which the `fd` points, which is usually a good thing. However, the only way a given module can detect the death of a process is when that process closes the `fd` it uses to communicate with the module. That module needs to know about process death to release any processes that may be blocked on it. A call to `close()` on process exit/ death is guaranteed when a single process has a copy of the `fd`.

However, when a process forks, both the parent and the child have the `fd` open, and it *will not close until both processes close it*. This can happen explicitly via the `close()` function, or on exit. This can lead to scenarios where the parent exits, but the child still holds the `fd` open, so the kernel module never realizes a process has exited. We try to compensate for this by detecting when a process uses any of the APIs through a `fd` that it did not originally open, then forcing that process to open its own `fd`. Unfortunately, if the child never makes any API calls, we never know the parent died.

In Linux 2.2 closing a `fd` is indicated by **flush** being delivered to the module. If it is the last close, the **flush** is followed by a **release**. This is sufficient information for the module to operate correctly under all conditions. See `tfork.c` in the distribution for an example demonstrating the bug. The example also demonstrates that the bug does not exist when using the module under a 2.2 or later kernel.

### 1.3.3. Error Numbers

The error numbers for certain irrational errors, such as sending a message to oneself or attaching a proxy to a proxy are not covered in the QNX documentation. The system will throw an error, but the `errno` may not be correct. Also, contrary to what the QNX documentation says, it is possible to call `qnx_proxy_detach()` on *any* proxy, assuming you have sufficient privileges, not just on proxies attached to you.

# Chapter 2. Installation

The library installs as a dynamically loadable kernel module. It has been allocated an official minor number from the kernel's miscellaneous minor-number space.



The SRR Module will not compile on a 64-bit Linux kernel.

## 2.1. Unpacking the Archive

The SRR Module is shipped as a gzipped tar file.



If you download the archive using a Windows based browser you may find that the browser saves the archive as a `.gz` file instead of a `.tgz` file. If this happens, simply modify the file extension to be `.tgz` before proceeding.

To unpack the archive in preparation for installation, follow this procedure:

1. Log in as root, or **su** to root.
2. Download or move the `srr-x.x.x.tgz` file to a temporary directory and then **cd** to that directory.
3. Issue the **tar** command:

```
# tar xzvf srr-x.x.x.tgz
```

This will create the directory `srr-x.x.x/` and its sub-directories.

4. Change to the `srr-x.x.x/` directory:

```
# cd srr-x.x.x/
```

5. Issue the **make** command:

```
# make
```

## 2.2. Testing the SRR Module

We recommend testing the SRR Module before installing it, by following this procedure:

1. Ensure you are root, and change to the `srr-x.x.x/mod/` subdirectory.

```
# cd mod/
```

2. Issue the **make** command:

```
# make load
```



If this fails (sometimes in non-RedHat installations), you can load the module manually by issuing the **insmod** command:

```
/sbin/insmod srripc.o
```



If the manual load or the **make load** command fail with undefined symbols, go back to the `srr-x.x.x/` directory and issue the **make** commands:

```
# cd ../
```

```
# make clean
```

```
# make FORCESYMBOLS=n
```

3. Once the module is loaded, change to the `srr-x.x.x/test/` subdirectory and issue the following **make** command:

```
# cd ../test/
```

```
# make linux
```

4. The test could take as long as 3 minutes to complete, and the results should look something like this:

```
Day Month Date hh:mm:ss EST yyyy
Linux yourdirectory 2.4 #1 Day Month Date hh:mm:ss EDT yyyy i686 unknown
srr: 4096 bytes 453866 times 10 sec 45386 Hz 371802112 bytes/sec
srr: 1024 bytes 872556 times 10 sec 87255 Hz 178698240 bytes/sec
srr: 256 bytes 991484 times 10 sec 99148 Hz 50763776 bytes/sec
srr: 64 bytes 1185034 times 10 sec 118503 Hz 15168384 bytes/sec
srr: 16 bytes 1202528 times 10 sec 120252 Hz 3848064 bytes/sec
srr: 4 bytes 1214180 times 10 sec 121418 Hz 971344 bytes/sec
srr: 1 bytes 1212815 times 10 sec 121281 Hz 242562 bytes/sec
srr: 0 bytes 1206153 times 10 sec 120615 Hz 0 bytes/sec
proxy: 2047430 times 10 sec 204743 Hz
poll: 4096 bytes 324745 times 10 sec 32474 Hz 266027008 bytes/sec
poll: 1024 bytes 505171 times 10 sec 50517 Hz 103458816 bytes/sec
poll: 256 bytes 544045 times 10 sec 54404 Hz 27854848 bytes/sec
poll: 64 bytes 590454 times 10 sec 59045 Hz 7557760 bytes/sec
poll: 16 bytes 603719 times 10 sec 60371 Hz 1931872 bytes/sec
poll: 4 bytes 600985 times 10 sec 60098 Hz 480784 bytes/sec
poll: 1 bytes 601949 times 10 sec 60194 Hz 120388 bytes/sec
poll: 0 bytes 605903 times 10 sec 60590 Hz 0 bytes/sec
timer w/proxy: 1001 times, 10 sec, 100 Hz, period 9990.01 us
timer w/signal: 1000 times, 10 sec, 100 Hz, period 10000 us
```

## 2.3. Installing the SRR Module

1. Return to the `srr-x.x.x/` directory:

```
# cd ../
```

2. Issue the **make** command:

```
# make install
```

The install path defaults to `/usr/local/`. The installation adds both libraries and include files to the `lib/` and `include/` sub-directories. You can specify a different install path by issuing the **make** command like this:

```
# make install prefix=/my/favourite/path
```

3. The kernel module itself is placed in: `/lib/modules/kernel_version/misc/srripc.o`. This is non-negotiable, though you can move it from there yourself if you want to set your own `MODPATH` for **modprobe**.
4. The **make install** command should have put the following line into either `/etc/conf.modules` or `/etc/modules.conf`:

```
alias char-major-10-171 srripc
```

This line is necessary for auto-loading. If it isn't present, you should hand-edit the file to include it.

If the installation was successful, attempting to open `/dev/srripc` should now cause the SRR Module to load automatically. If your system is not set up for auto-loading, please refer to the next section.



If the permissions on the `/dev/srripc` module are not correct, you will encounter error messages when starting programs that use the SRR module such as:

```
Cannot count names: Bad file descriptor
Queue manager could not attach name 'qserve': Device or resource busy

Could not initialize IPC
```

In some Linux distributions, it may be necessary to manually change the permissions on the SRR module's mount point. To change the permissions, you can issue this command, as root:

```
chmod a+rw /dev/srripc
```

If your kernel uses the udev file system, usually meaning that `/dev` is a temporary file system, the above command should be executed at every system reboot through a system initialization script.

## 2.4. Enabling Auto-Loading for the SRR Module



Most Linux distributions ship with auto-loading enabled. So, unless you are running a modified or trimmed down version of the kernel, you can almost certainly ignore this section.

You can test to see if your kernel is auto-loading as follows:

1. Unload the SRR Module using the **rmmod** command:

```
# /sbin/rmmod srripc
```

2. Run the SRR Module test program (explained above) from the `srr-x.x.x/test/` subdirectory:

```
# cd srr-x.x.x/test/
# make linux
```

If the test runs, then your kernel has successfully auto-loaded the SRR Module. If the test did not run, then you will have to rebuild your kernel with kernel auto-loading turned on.

1. In the 2.2 and later kernel's configuration section "Loadable Module Support", answer **YES** to both:

```
Enable loadable module support
```

and

```
Kernel module loader
```

2. Recompile your Linux kernel and reboot.

## 2.5. Example Code

There are several example programs included in the `srr-x.x.x/exe/` directory. At the time of this writing, the following programs are included, both as source code and compiled:

- `irqtest.c`
- `pthread.c`
- `recv.c`
- `reply.c`
- `rwmttest.c`
- `send.c`
- `srrctl.c`
- `tfork.c`
- `tserver.c`

- ttimers.c
- wait.c

Shown below is the contents of file `srr-x.x.x/exe/send.c`:



This code is shown as an example only, and may be slightly out of date for your version of the SRR Module. Please use the sample programs in your distribution to ensure full compatibility.

```
/*
 * send.c: a simple program that sends a message, and prints the reply
 *
 * $Id: srr_man.sgml,v 1.1.1.1 2010-08-11 21:47:08 robert Exp $
 */

#include <assert.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include "srr.h"

int main(int argc, char* argv[])
{
    int          opt, len;
    int          pid = 0;
    char*        name = 0;
    char         smsg[BUFSIZ] = "";
    char         rmsg[BUFSIZ] = "";

    sprintf(smsg, "msg from %d", getpid());

    while((opt = getopt(argc, argv, "p:n:")) != -1)
    {
        switch(opt)
        {
            {
            case 'p':
                pid = atoi(optarg);
                break;

            case 'n':
                name = optarg;
                break;

            default:
                exit(1);
            }
        }

    if(SrrReg() == -1)
    {
        printf("send %d: InitSrr() failed: [%d] %s\n",
            getpid(), errno, strerror(errno));

        exit(1);
    }

    if(name)
    {
        if((pid = qnx_name_locate(0, name, 0, 0)) == -1)
        {
            printf("send %d: qnx_name_locate() failed: [%d] %s\n",
                getpid(), errno, strerror(errno));
            exit(1);
        }
    }
}
```

```

printf("send %d: sending: \"%s\"\n", getpid(), msg);

if((len = SrrSend(pid, msg, rmsg, strlen(msg)+1, BUFSIZ)) == -1)
{
    printf("send %d: Send() failed: [%d] %s\n",
        getpid(), errno, strerror(errno));
    exit(1);
}

printf("send %d: received len %d: \"%s\"\n", getpid(), len, rmsg);

return 0;
}

```



## Chapter 3. Copyrights and Disclaimers

The module is copyright © 2000 by Cogent Real-Time Systems, Inc.. It is distributed under the same conditions as the Linux Kernel, the [GNU General Public License](#) (GPL). The API library (`srr_lib.c`, `srr.h`) is also distributed under the GPL. See the `COPYING` file included in the distribution for the details of this licence.

In the past, the API library has been distributed under the [GNU Lesser General Public License](#) (LGPL), which allowed you to use the library in commercial software. Due to the rising cost and effort of maintaining this library, we have modified the library licensing starting with version 1.4.1 of the module to the GPL. This makes the license inappropriate for use in commercial software, since the GPL requires that the commercial software must therefore also be covered by the GPL. If you wish to use the SRR Module in a commercial product, please contact Cogent for alternate licensing terms.

Please do not interpret this change in license as a "money grab". The purpose of this change is to allow us to devote the time and resources to the SRR Module that it deserves. If you use this module for non-commercial reasons, or make the source code available to anything developed with this module, then it remains free. If you are making money from this module, then you already understand the nature of the commercial food chain, and understand the cost of maintaining a high-quality product. We ask only that you license the fruit of our labour as you ask others to license yours.

**Limited Liability.** The SRR Module software and documentation is offered "as is," without warranty of any kind. Neither Cogent Real-Time Systems, Inc. nor any related third parties shall be liable for the cost of any direct, indirect, incidental, or consequential damages whatsoever, such as, but not limited to, loss of anticipated benefits or profits resulting from the use of this software. The entire risk as to the results and performance of this software is assumed by the user.

# Appendix A. GNU General Public License

## GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 by Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

*\* Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.*

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

#### Section 0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program",

below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

### *Section 1*

You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

### *Section 2*

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

### **Section 3**

You may copy and distribute the Program (or a work based on it, under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

### **Section 4**

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

### **Section 5**

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

### **Section 6**

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

### **Section 7**

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

### *Section 8*

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

### *Section 9*

The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

### *Section 10*

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### *NO WARRANTY Section 11*

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

### Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type “show w”. This is free software, and you are welcome to redistribute it under certain conditions; type “show c” for details.

The hypothetical commands “show w” and “show c” should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than “show w” and “show c”; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program “Gnomovision” (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# Appendix B. GNU Lesser General Public License

## GNU Lesser General Public License

Version 2.1, February 1999

Copyright © 1991, 1999 by Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

*\* Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.*

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method:

1. we copyright the library, and
2. we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.



Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the *Lesser* General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

### **Section 0**

This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

## Section 1

You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

## Section 2

You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. The modified work must itself be a software library.
- b. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

### **Section 3**

You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

### **Section 4**

You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

### **Section 5**

A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

### **Section 6**

As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a. Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

## Section 7

You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## Section 8

You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library 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.

## **Section 9**

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

## **Section 10**

Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

## **Section 11**

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

## **Section 12**

If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

## **Section 13**

The Free Software Foundation may publish revised and/or new versions of the Lesser General Public 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.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

## **Section 14**

If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## **NO WARRANTY Section 15**

BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

## **Section 16**

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## **How to Apply These Terms to Your New Libraries**

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the library’s name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library ‘Frob’ (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That’s all there is to it!

# II. Reference

## Table of Contents

<a href="#">I. Send/Receive/Reply API Reference .....</a>	<a href="#">24</a>
<a href="#">II. QNX Compatibility API Reference.....</a>	<a href="#">36</a>
<a href="#">III. Undocumented functions.....</a>	<a href="#">48</a>



# I. Send/Receive/Reply API Reference

## Table of Contents

<a href="#">SrrSend</a> .....	25
<a href="#">SrrReceive</a> .....	26
<a href="#">SrrRelay</a> .....	27
<a href="#">SrrReply</a> .....	28
<a href="#">SrrProxyAttach</a> .....	29
<a href="#">SrrProxyDetach</a> .....	31
<a href="#">SrrProxyTrigger</a> .....	32
<a href="#">SrrNameAttach</a> .....	33
<a href="#">SrrNameDetach</a> .....	34
<a href="#">SrrNameLocate</a> .....	35

# SrrSend

SrrSend — sends a message, blocks until replied.

## Syntax

```
#include <srripc.h>
int SrrSend (
    pid_t pid,
    void* msg,
    void* rmsg,
    size_t ssize,
    size_t rsize
);
```

## Arguments

*pid*

The process to whom the message is sent.

*msg*

The message to be sent.

*rmsg*

The buffer for the reply message.

*ssize*

The size of the message to be sent.

*rsize*

The size of the buffer for the reply message.

## Returns

The amount of data placed into the reply buffer, or -1 on failure.

## Description

This function emulates the QNX Send function. The reply is returned as a character string.

## Errors

- **EFAULT** invalid message
- **EHOSTUNREACH** Destination node not in the netmap, or physical I/O error has occurred
- **EINTR** The function was interrupted by a signal
- **EINVAL** invalid message length
- **ENOMEM** not enough memory available for operation
- **ESRCH** the process ID does not exist

## See Also

[SrrReceive](#), [SrrReply](#)

# SrrReceive

SrrReceive — receives a message.

## Syntax

```
#include <srripc.h>
pid_t SrrReceive (
    pid_t pid,
    void* msg,
    size_t* size
);
```

## Arguments

*pid*

The process to receive a message from; 0 means from any process.

*msg*

The buffer for the received message.

*size*

On call, the size of the buffer for the received message; on return, the size of the message in the buffer.

## Returns

On success, the process ID of the sender, or -1 on failure.

## Description

This function performs a synchronous Receive. Once a message has been received the sending task is blocked until the recipient issues a reply message using the SrrReply function. Unlike the QNX version of this function ([Receive](#)), the *size* parameter indicates how much data has been received.

## Errors

- **EFAULT** invalid buffer size
- **EINTR** the function call was interrupted by a signal
- **ESRCH** the process ID does not exist

## See Also

[SrrSend](#), [SrrReply](#)

# SrrRelay

SrrRelay — not yet documented.

## Syntax

```
#include <srripc.h>
(  
);
```

# SrrReply

SrrReply — replies to a previously received message.

## Syntax

```
#include <srripc.h>
int SrrReply (
    pid_t pid,
    void* msg,
    size_t size
);
```

## Arguments

*pid*

The process to reply to.

*data*

The message to reply with.

*size*

The size of the reply message.

## Returns

Returns zero on success and -1 on failure. Errno is set on error.

## Description

The SrrReply function is a wrap of the C function Reply. Any message that you receive using the SrrReceive function should be replied to. This function should only be used if you are also using SrrReceive.

## Errors

- **EFAULT** Invalid message
- **EINTR** Function call interrupted by signal
- **EINVAL** invalid virtual circuit buffer
- **ENOMEM** not enough memory for operation
- **ESRCH** the process ID does not exist

## See Also

[SrrSend](#), [SrrReceive](#),

# SrrProxyAttach

SrrProxyAttach — creates a proxy message for a process.

## Syntax

```
#include <srripc.h>
pid_t SrrProxyAttach (
    pid_t pid,
    char* data,
    int nbytes,
    int priority
);
```

## Arguments

*pid*

The process ID of the task to receive the message. If set to 0, the proxy is attached to the calling process.

*data*

The message data.

*nbytes*

The message size.

*priority*

The priority of the proxy. If set to -1, the proxy priority is the same as the calling process.

## Returns

The process ID of the proxy, else -1.

## Description

This function creates a message proxy that will deliver a standard message to a task whenever it is called. The proxy will accept any message from any other task, but will always send its one message to the task identified with *pid*. The sending process will not block, and its message data will be discarded by the proxy.

Proxies are used to wake up receive-blocked processes that are waiting for messages. They can also be used to send non-blocking messages between processes. Proxy messages are queued, so if a proxy is triggered 95 times, its receiving task will get 95 identical messages. A proxy may have at most 65535 messages pending.

## Errors

- **EAGAIN** No process entries free to make a proxy.
- **EINVAL** The proxy message exceeds the maximum size.
- **ENOMEM** The kernel doesn't have enough memory to hold the message.
- **ESRCH** The process ID (*pid*) does not exist.

## **See Also**

[SrrProxyDetach](#)

# SrrProxyDetach

SrrProxyDetach — detaches a proxy from its target process.

## Syntax

```
#include <srripc.h>
int SrrProxyDetach (
    pid_t pid
);
```

## Arguments

*pid*

The proxy ID number as returned by SrrProxyAttach.

## Returns

0 on success, else -1.

## Description

This function releases a proxy from its associated task. When a task dies all proxies attached to it are automatically removed.

## Errors

- **EPERM** This task has insufficient privileges to do this.
- **ESRCH** The proxy does not exist.

## See Also

[SrrProxyAttach](#) [SrrProxyTrigger](#)



# SrrProxyTrigger

`SrrProxyTrigger` — triggers a proxy into sending its message.

## Syntax

```
#include <srripc.h>
pid_t SrrProxyTrigger (
    pid_t pid
);
```

## Arguments

*pid*

The proxy ID number as returned by `SrrProxyAttach`.

## Returns

0 on success, else -1.

## Description

This function triggers a proxy to send its message to the receiving task. Its calling process does not block, and if more than one `SrrProxyTrigger` call is sent while the task is busy, up to 65535 proxy messages will be queued for later delivery.

## Errors

- **EPERM** This task has insufficient privileges to do this.
- **ESRCH** The proxy does not exist.

## See Also

[SrrProxyAttach](#) [SrrProxyDetach](#)

# SrrNameAttach

`SrrNameAttach` — attaches a name to a process.

## Syntax

```
#include <srripc.h>
int SrrNameAttach (
    nid_t n,
    const char* name
);
```

## Arguments

*n*

The applicable node. If 0 the local node is used.

*name*

The name to be attached.

## Returns

On success, returns a name ID, used as the argument to `SrrNameDetach`. On failure, returns -1 and sets `errno`.

## Description

This function attaches a name to this process. If a name starts with a slash '/' then it is considered to be a global name. Global names are registered with optionally run global process name servers.

## Errors

- `EAGAIN` Name space used up on this node
- `EBUSY` Specified name already exists on that node
- `Resource Temporarily Unavailable` This error may occur when you have exceeded the maximum limit for names. Please refer to the Increasing the Maximum Number of Named Processes section of the Troubleshooting Guide for instructions on how to increase this limit.

## See Also

[SrrNameDetach](#), [SrrNameLocate](#),

# SrrNameDetach

SrrNameDetach — detaches a name.

## Syntax

```
#include <srripc.h>
int SrrNameDetach (
    nid_t n,
    int name_id
);
```

## Arguments

*n*

The applicable node. If 0 the local node is used.

*name\_id*

The name to be detached.

## Returns

0 on success, else -1, and errno is set.

## Description

Detaches a name attached by SrrNameAttach.

## Errors

- **EINVAL** nameid does not exist, or if it does, you do not own it

## See Also

[SrrNameAttach](#), [SrrNameLocate](#)

# SrrNameLocate

SrrNameLocate — locates a process by name.

## Syntax

```
#include <srripc.h>
pid_t SrrNameLocate (
    nid_t n,
    const char* name,
    unsigned* sz,
    unsigned* copies
);
```

## Arguments

*n*

The node to search. If set to 0 the local node is searched.

*name*

The name of the process to locate.

*sz*

An initial value for the size of the virtual circuit buffer, if the *name* is on a different node. This size will generally be the size of the largest message. A value of 0 is possible, since virtual circuits expand as needed.

*copies*

If any value other than NULL is passed, this gets set to 1 if the name has been located, and 0 if not.

## Returns

On success, the ID of the process that attached the name. On failure, -1 and errno is set.

## Description

## Errors

- **ESRCH** The named process does not exist.

## See Also

[SrrNameAttach](#), [SrrNameLocate](#)

## II. QNX Compatibility API Reference

### Table of Contents

<a href="#">Send</a>	<a href="#">37</a>
<a href="#">Receive</a>	<a href="#">38</a>
<a href="#">Relay</a>	<a href="#">39</a>
<a href="#">Reply</a>	<a href="#">40</a>
<a href="#">qnx_proxy_attach</a>	<a href="#">41</a>
<a href="#">qnx_proxy_detach</a>	<a href="#">42</a>
<a href="#">Trigger</a>	<a href="#">43</a>
<a href="#">qnx_name_attach</a>	<a href="#">44</a>
<a href="#">qnx_name_detach</a>	<a href="#">45</a>
<a href="#">qnx_name_locate</a>	<a href="#">46</a>
<a href="#">qnx_vc_attach</a>	<a href="#">47</a>

# Send

Send — QNX-style wrapper on SrrSend.

## Syntax

```
#include <qnxipc.h>
int Send (
    pid_t pid,
    void* smsg,
    void* rmsg,
    unsigned snbytes,
    unsigned rnbytes
);
```

## See

[SrrSend](#),

# Receive

Receive — QNX-style wrapper on SrrReceive.

## Syntax

```
#include <qnxipc.h>
pid_t Receive (
    pid_t pid,
    void* msg,
    unsigned* nbytes
);
```

## Description

Unlike SrrReceive, this function gives no indication of how much data has been received.

## See

[SrrReceive](#)

# Relay

Relay — not yet documented.

## Syntax

```
#include <srripc.h>
(  
);
```



# Reply

Reply — QNX-style wrapper on SrrReply.

## Syntax

```
#include <qnxipc.h>
int Reply (
    pid_t pid,
    void* msg,
    unsigned nbytes
);
```

## See

[SrrReply](#),

## qnx\_proxy\_attach

qnx\_proxy\_attach — QNX-style wrapper on SrrProxyAttach.

### Syntax

```
#include <qnxipc.h>
pid_t qnx_proxy_attach (
    pid_t pid,
    char* data,
    int nbytes,
    int priority
);
```

### See

[SrrProxyAttach](#)

## qnx\_proxy\_detach

qnx\_proxy\_detach — QNX-style wrapper on SrrProxyDetach.

### Syntax

```
#include <qnxipc.h>
int qnx_proxy_detach (
    pid_t pid
);
```

### See

[SrrProxyDetach](#)

# Trigger

Trigger — QNX-style wrapper on SrrProxyTrigger.

## Syntax

```
#include <qnxipc.h>
pid_t Trigger (
    pid_t pid
);
```

## See

[SrrProxyTrigger](#)

## **qnx\_name\_attach**

`qnx_name_attach` — QNX-style wrapper on `SrrNameAttach`.

### **Syntax**

```
#include <qnxipc.h>
int qnx_name_attach (
    nid_t n,
    const char* name
);
```

### **See**

[SrrNameAttach](#),

## qnx\_name\_detach

qnx\_name\_detach — QNX-style wrapper on SrrNameDetach.

### Syntax

```
#include <qnxipc.h>
int qnx_name_detach (
    nid_t n,
    int name_id
);
```

### See

[SrrNameDetach](#)

## qnx\_name\_locate

qnx\_name\_locate — QNX-style wrapper on SrrNameLocate.

### Syntax

```
#include <qnxipc.h>
pid_t qnx_name_locate (
    nid_t n,
    const char* name,
    unsigned* sz,
    unsigned* copies
);
```

### See

[SrrNameLocate](#)

## **qnx\_vc\_attach**

`qnx_vc_attach` — stub function.

### **Syntax**

```
qnx_vc_attach (node, taskid, max-msg-length, flags);
```

### **Returns**

Returns its PID argument.



# III. Undocumented functions

## Table of Contents

<a href="#">SrrClockGettime</a>	<a href="#">49</a>
<a href="#">SrrClockSettime</a>	<a href="#">50</a>
<a href="#">SrrCreceive</a>	<a href="#">51</a>
<a href="#">SrrCreceivemx</a>	<a href="#">52</a>
<a href="#">SrrDebug</a>	<a href="#">53</a>
<a href="#">SrrFd</a>	<a href="#">54</a>
<a href="#">SrrKill</a>	<a href="#">55</a>
<a href="#">SrrPflags</a>	<a href="#">56</a>
<a href="#">SrrProxyRemDetach</a>	<a href="#">57</a>
<a href="#">SrrReadmsg</a>	<a href="#">58</a>
<a href="#">SrrReadmsgmx</a>	<a href="#">59</a>
<a href="#">SrrReceive</a>	<a href="#">60</a>
<a href="#">SrrReg</a>	<a href="#">61</a>
<a href="#">SrrReplymx</a>	<a href="#">62</a>
<a href="#">SrrSendmx</a>	<a href="#">63</a>
<a href="#">SrrTimerDelete</a>	<a href="#">64</a>
<a href="#">SrrTimerGettime</a>	<a href="#">65</a>
<a href="#">SrrTimerSettime</a>	<a href="#">66</a>
<a href="#">SrrUnload</a>	<a href="#">67</a>
<a href="#">SrrUnreg</a>	<a href="#">68</a>
<a href="#">SrrVcDetach</a>	<a href="#">69</a>
<a href="#">SrrWritemsg</a>	<a href="#">70</a>
<a href="#">SrrWritemsgmx</a>	<a href="#">71</a>

## SrrClockGettime

SrrClockGettime — undocumented.

### Syntax

```
#include <srripc.h>
int SrrClockGettime (
    clockid_t clock_id,
    struct timespec* tp
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrClockSettime

SrrClockSettime — undocumented.

### Syntax

```
#include <srripc.h>
int SrrClockSettime (
    clockid_t clock_id,
    struct timespec* tp
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrCreceive

SrrCreceive — undocumented.

## Syntax

```
#include <srripc.h>
int SrrCreceive (
    pid_t pid,
    void* msg,
    size_t* size
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrCreceivemx

SrrCreceivemx — undocumented.

## Syntax

```
#include <srripc.h>
int SrrCreceivemx (
    pid_t pid,
    size_t parts,
    struct _mxfer_entry* msgmx
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrDebug

SrrDebug — undocumented.

### Syntax

```
#include <srripc.h>
int SrrDebug (
    int level
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrFd

SrrFd — undocumented.

## Syntax

```
#include <srripc.h>
int SrrFd (
    void
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrKill

SrrKill — undocumented.

## Syntax

```
#include <srripc.h>
int SrrKill (
    pid_t pid,
    int signo
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.



# SrrPflags

SrrPflags — undocumented.

## Syntax

```
#include <srripc.h>
int SrrPflags (
    unsigned short flags,
    unsigned short mask,
    void* dum1,
    void* dum2
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrProxyRemDetach

SrrProxyRemDetach — undocumented.

## Syntax

```
#include <srripc.h>
int SrrProxyRemDetach (
    nid_t nid,
    pid_t pid
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrReadmsg

SrrReadmsg — undocumented.

### Syntax

```
#include <srripc.h>
int SrrReadmsg (
    pid_t pid,
    size_t offset,
    void* msg,
    size_t size
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrReadmsgmx

SrrReadmsgmx — undocumented.

### Syntax

```
#include <srripc.h>
int SrrReadmsgmx (
    pid_t pid,
    size_t offset,
    size_t parts,
    struct _mxfer_entry* msgmx
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrReceivevx

SrrReceivevx — undocumented.

### Syntax

```
#include <srripc.h>
int SrrReceivevx (
    pid_t pid,
    size_t parts,
    struct _mxfer_entry* msgmx
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrReg

SrrReg — undocumented.

## Syntax

```
#include <srripc.h>
int SrrReg (
    void
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrReplymx

SrrReplymx — undocumented.

## Syntax

```
#include <srripc.h>
int SrrReplymx (
    pid_t pid,
    size_t parts,
    struct _mxfer_entry* msgmx
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrSendmx

SrrSendmx — undocumented.

## Syntax

```
#include <srripc.h>
int SrrSendmx (
    pid_t pid,
    size_t sparts,
    size_t rparts,
    struct _mxfer_entry* smsg,
    struct _mxfer_entry* rmsg
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.



# SrrTimerDelete

SrrTimerDelete — undocumented.

## Syntax

```
#include <srripc.h>
int SrrTimerDelete (
    timer_t tid
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrTimerGettime

SrrTimerGettime — undocumented.

### Syntax

```
#include <srripc.h>
int SrrTimerGettime (
    timer_t tid,
    struct itimerspec* value
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrTimerSettime

SrrTimerSettime — undocumented.

### Syntax

```
#include <srripc.h>
int SrrTimerSettime (
    timer_t tid,
    int flags,
    const struct itimerspec* value,
    struct itimerspec* ovalue
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrUnload

SrrUnload — undocumented.

## Syntax

```
#include <srripc.h>
int SrrUnload (
    );
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

## SrrUnreg

SrrUnreg — undocumented.

### Syntax

```
#include <srripc.h>
int SrrUnreg (
    void
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrVcDetach

SrrVcDetach — undocumented.

## Syntax

```
#include <srripc.h>
int SrrVcDetach (
    pid_t pid
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrWritemsg

SrrWritemsg — undocumented.

## Syntax

```
#include <srripc.h>
int SrrWritemsg (
    pid_t pid,
    size_t offset,
    void* msg,
    size_t size
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.

# SrrWritemsgmx

SrrWritemsgmx — undocumented.

## Syntax

```
#include <srripc.h>
int SrrWritemsgmx (
    pid_t pid,
    size_t offset,
    size_t parts,
    struct _mxfer_entry* msgmx
);
```

This function is not normally accessed by users, and is therefore not documented beyond showing its syntax.



# Index

SrrWritemsg, [70](#)  
SrrWritemsgmx, [71](#)

## Q

qnx\_name\_attach, [44](#)  
qnx\_name\_detach, [45](#)  
qnx\_name\_locate, [46](#)  
qnx\_proxy\_attach, [41](#)  
qnx\_proxy\_detach, [42](#)  
qnx\_vc\_attach, [47](#)

## R

Receive, [38](#)  
Relay, [39](#)  
Reply, [40](#)

## S

Send, [37](#)  
SrrClockGettime, [49](#)  
SrrClockSettime, [50](#)  
SrrCreceive, [51](#)  
SrrCreceivemx, [52](#)  
SrrDebug, [53](#)  
SrrFd, [54](#)  
SrrKill, [55](#)  
SrrNameAttach, [33](#)  
SrrNameDetach, [34](#)  
SrrNameLocate, [35](#)  
SrrPflags, [56](#)  
SrrProxyAttach, [29](#)  
SrrProxyDetach, [31](#)  
SrrProxyRemDetach, [57](#)  
SrrProxyTrigger, [32](#)  
SrrReadmsg, [58](#)  
SrrReadmsgmx, [59](#)  
SrrReceive, [26](#)  
SrrReceivemx, [60](#)  
SrrReg, [61](#)  
SrrRelay, [27](#)  
SrrReply, [28](#)  
SrrReplymx, [62](#)  
SrrSend, [25](#)  
SrrSendmx, [63](#)  
SrrTimerDelete, [64](#)  
SrrTimerGettime, [65](#)  
SrrTimerSettime, [66](#)  
SrrUnload, [67](#)  
SrrUnreg, [68](#)  
SrrVcDetach, [69](#)

## T

Trigger, [43](#)

# Colophon

This book was produced by Cogent Real-Time Systems, Inc. from a single-source group of SGML files. Gnu Emacs was used to edit the SGML files. The DocBook DTD and related DSSSL stylesheets were used to transform the SGML source into HTML, PDF, and QNX Helpviewer output formats. This processing was accomplished with the help of OpenJade, JadeTeX, Tex, and various scripts and makefiles. Details of the process are described in our book: *Preparing Cogent Documentation*, which is published on-line at

<http://developers.cogentrts.com/cogent/prepdoc/book1.html>.

Text written by Sam Roberts, Andrew Thomas, and Bob McIlvride.