1

TR-94033

RWC-1 Linker Manual
Release 0.2

Massively Parallel Software Laboratory

Received February 1995

Tsukuba Research Center, Real World Computing Partnership
Tsukuba Mitsui Building 16F, 1-6-1 Takezono
Tsukuba-shi, Ibaraki 305, JAPAN

*This document was prepared by Tadashi Sato, Mitsubishi Electric Computer Systems (Tokyo) Co.Ltd, as the part of the contract with Massively Parallel Software Laboratory

Contents

I RWC-1 Cross Linker User's Manual 4

1 rld 6
1.1 synopsis . . . 6
1.2 description . . . 6
1.3 options . . . 6
1.4 see also . . . 9

II RWC-1 Binary Utilities User's Manual 10

1 rar 12
1.1 synopsis . . . 12
1.2 description . . . 12
1.3 options . . . 12
1.3.1 modifiers . . . 14
1.4 see also . . . 14

2 rranlib 15
2.1 synopsis . . . 15
2.2 options . . . 15
2.3 see also . . . 15

3 rnm 16
3.1 synopsis . . . 16
3.2 description . . . 16
3.3 options . . . 16
3.4 see also . . . 18

4 robjdump 19
4.1 synopsis . . . 19
4.2 description . . . 19
4.3 options . . . 19
4.4 see also . . . 20

5 rsize 21
5.1 synopsis . . . 21
5.2 description . . . 21
5.3 options . . . 21
5.4 see also . . . 22

6 rstrings 23
6.1 synopsis . . . 23
6.2 description . . . 23

6.3 options 23
6.4 see also 24

7 rstrip **25**
7.1 synopsis 25
7.2 description 25
7.3 options 25

Part I
RWC-1 Cross Linker User's Manual

RWC-1 Cross Linker User's Manual
Edited by Tadashi Sato

Copyright @copyright 1994, Real World Computing Partnership. This software is derived from the gnu ld developed by Free Software Foundation, Inc. The conditions of copyright follow the terms of the GNU General Public Licence.

Copyright @copyright 1991, 1992, 1993 Free Software Foundation, Inc.

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

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

1 rld

rld is the RWC-1 cross-linker.

1.1 synopsis

rld [**-o** *output*] *objfile ...*
[**-d**|**-dc**|**-dp**] [**-defsym** *symbol=expression*]
[**-e** *entry*] [**-g**] [**--help**] [**-i**]
[**-l***ar*] [**-L***searchdir*] [**-M**] [**-Map** *mapfile*]
[**-noinhibit-exec**] [**-R** *filename*]
[**-r**|**-Ur**] [**-S**] [**-s**] [**-sort-common**]
[**-Ttext** *textorg*] [**-Tdata** *dataorg*] [**-Tbss** *bssorg*]
[**-t**] [**-u** *sym*] [**-V**] [**-v**] [**--version**]
[**-warn-common**] [**-y***sym*] [**-X**] [**-x**]
[**-align_sec** *section=pagesize*] [**-abort_alignment**]

1.2 description

rld is the GNU linker[1] ported to the RWC-1 for cross-linking. **rld** combines a number of object and archive files, relocates their data and ties up symbol references. Often the last step in building a newly compiled executable program is a call to **rld**. Many linkers abandon execution immediately upon encountering an error. Whenever possible, **rld** continues executing, allowing you to identify other errors (or, in some cases, to get an output file despite the error).

The RWC-1 cross-linker **rld** is meant to cover a broad range of situations. As a result, you have many command line choices to control its behavior.

1.3 options

The plethora of command-line options may seem intimidating, but in actual practice few of them are used in any particular context. For instance, **rld** is frequently used to link standard RWC-1 object files supported by the RWC-1 system. On such a system (Suppose there are the standard RWC-1 object files in /lib), to link the file **hello.o**:

```
$ rld -o output /lib/crt0.o hello.o -L/lib -lmpc
```

This tells **rld** to create a file called **output** by linking the file **/lib/crt0.o** with **hello.o** and the library **libmpc.a** which is in the search directories specified by -L option. The -l option needs the -L option, because rld has no standard search directories.

The command-line options to **rld** may be specified in any order and may be repeated. For the most part, repeating an option with a different argument will either have no further effect, or override prior occurrences (those further to the left on the command line) of the option.

The exceptions–commands which may meaningfully be used more than once–are **-defsym**, **-L**, **-l**, **-R**, and **-u**.

The list of object files to be linked together, shown as *objfile*, may follow, precede, or be mixed in with command-line options, except that an *objfile* argument may not be placed between an option flag and its argument.

Usually the linker is invoked with at least one object file, but other forms of binary input files can also be specified with **-l**, **-R**. If *no* binary input files at all are specified, the linker issues the message '**No input files**' and does not produce any output.

Option arguments must either follow the option letter without intervening whitespace, or be given as separate arguments immediately following the option that requires them.

-d

-dc

-dp These three options are equivalent. The multiple forms are supported for compatibility with other linkers. Use any of them to make **rld** assign space to common symbols, even if a relocatable output file is specified (**-r**).

-defsym *symbol=expression* Create a global symbol in the output file containing the absolute address given by *expression*. You may use this option as many times as necessary to define multiple symbols in the command line. A limited form of arithmetic is supported for the *expression* in this context: you may give a hexadecimal constant or the name of an existing symbol, or use + and - to add or subtract hexadecimal constants or symbols.

-e *entry* Use *entry* as the explicit symbol for beginning execution of your program, rather than the default entry point.

-g Accepted, but ignored. Provided for compatibility with other tools.

--help Print a summary of the command-line options to standard output and exit. For compatibility with GNU programs, this option and **--version** begin with two dashes instead of one. For compatibility with other linkers, the other options start with only one dash.

-i Perform an incremental link (same as option **-r**).

-l*ar* Add an archive file *ar* to the list of files to be linked. This option may be used any number of times. For every *ar* specified, **rld** will search its path-list for occurrences of **lib***ar*.a.

-L*searchdir* This command adds path *searchdir* to the list of paths that **rld** will search for archive libraries. You may use this option any number of times. rld does not have a default search path.

-M Print (to the standard output file) link map–diagnostic information about where symbols are mapped by **rld**, and information on global common storage allocation.

-Map *mapfile* Print to the file *mapfile* link map–diagnostic information about where symbols are mapped by **rld**, and information on global common storage allocation.

-noinhibit-exec Normally, the linker will not produce an output file if it encounters errors during the link process. With this flag, you can specify that you wish the output file produced even after non-fatal errors.

-o *output* Output *output* that is a name for the program produced by **rld**. If this option is not specified, the name '**a.out**' is used by default.

-R *filename* Read symbol names and their addresses from *filename*, but do not relocate the file or include it in the output. This allows your output file to refer symbolically to absolute locations of memory defined in other programs.

-r Generates relocatable output–i.e., generate an output file that can in turn serve as **rld** input. This is often called *partial* linking. If this option is not specified, an absolute file is produced. When linking C++ programs, this option *will not* resolve references to constructors. **-Ur** may be used as an alternative.

This option functions the same as **-i**.

-S Omits debugger symbol information (but not all symbols) from the output file.

-s Omits all symbol information from the output file.

-sort-common Normally, when **rld** places the global common symbols in the appropriate output sections, it sorts them by size. First come all the one byte symbols, then all the two bytes, then all the four bytes, and then everything else. This is to prevent gaps between symbols due to alignment constraints. The -sort-common option disables sorting by size.

-Tbss *org*

-Tdata *org*

-Ttext *org* Use *org* as the starting address for–respectively–the **bss**, **data** or the **text** segment of the output file. *textorg* must be a hexadecimal integer.

-t Prints names of input files as **rld** processes them.

-u *sym* Forces *sym* to be entered in the output file as an undefined symbol. This may, for example, trigger linking of additional modules from standard libraries. **-u** may be repeated with different option arguments to enter additional undefined symbols.

-Ur For anything other than C++ programs, this option is equivalent to **-r**: it generates relocatable output–i.e., an output file that can in turn serve as input to **rld**. When linking C++ programs, **-Ur** *will* resolve references to constructors, unlike **-r**.

-V Display the version number for **rld** and list the supported emulations. Display which input files can or cannot be opened.

-v Display the version number for **rld**.

--version Display the version number for **rld** and exit.

-warn-common Warn when a common symbol is combined with another common symbol or with a symbol definition. rld allows this somewhat sloppy practice, and this option allows you to find potential problems from combining global symbols.

-X If **-s** or **-S** is also specified, delete only local symbols beginning with '**L**'.

-x If **-s** or **-S** is also specified, delete all local symbols, not just those beginning with '**L**'.

-y*symbol* Print the name of each linked file in which the *symbol* appears. This option may be given any number of times. On many systems it is necessary to prepend an underscore. This option is useful when you have an undefined symbol in your link but don't know where the reference is coming from.

-align_sec *section=pagesize* Align the *section* by *pagesize*. For the *section*, you may use *text, data, data1 or bss*. For the *pagesize*, you may use hexadecimal.

-abort_alignment Exit, if errors are encountered during *page-align*. Without this option, rld issues warnings.

1.4 see also

robjdump.

Part II
RWC-1 Binary Utilities User's Manual

RWC-1 Binary Utilities User's Manual
Edited by Tadashi Sato

Copyright @copyright 1994, Real World Computing Partnership. This software is derived from the gnu ld developed by Free Software Foundation, Inc. The conditions of copyright follow the terms of the GNU General Public Licence.

Copyright @copyright 1991, 1992, 1993 Free Software Foundation, Inc.

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

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

1 rar

Create, modify, and extract from archives.

1.1 synopsis

rar [**-**]{*dmpqrtx*}*[abcilosuvV]* [*membername*] *archive files ...*

1.2 description

rar is one of the GNU binary utilities[2] ported to the RWC-1. The **rar** program creates, modifies, and extracts from archives. An *archive* is a single file holding a collection of other files in a structure that makes it possible to retrieve the original individual files (called archive *members*).

The original files' contents, mode (permissions), timestamp, owner, and group are preserved in the archive, and may be reconstituted on extraction.

rar can maintain archives whose members have unlimited-length names.

rar is considered a binary utility because archives of this sort are most often used as *libraries* holding commonly needed subroutines.

If you specify the modifier '**s**', **rar** will create an index to the symbols defined in the archive's relocatable object modules. Once created, this index is updated in the archive whenever **rar** makes a change to its contents (except for the '**q**' update operation). An archive with this index speeds up linking to the library, and allows library routines to call each other without regard to their placement in the archive.

You may use '**rnm -s**' or '**rnm --print-armap**' to list this index table. If an archive does not have this table, another form of **rar** called **rranlib** can be used to add the table.

rar requires at least two arguments for execution: the keyletter specifying the *operation* (optionally accompanied by other keyletters specifying *modifiers*), and the archive name.

Most operations can also accept further *file* arguments which specify the particular files to be operated on.

1.3 options

rar allows you to mix the order of the operation code *p* and modifier flags *mod*, within the first command-line argument.

If you wish, you may begin the first command-line argument with a dash.

The *p* keyletter specifies the operation to be executed; it may be any of the following, but you must specify only one.

d *Delete* modules from the archive. Specify the names of modules to be deleted as *files*. The archive is untouched if you specify no files for deletion.

If you specify the '**v**' modifier, **rar** will list each module as it is deleted.

m Use this operation to *move* archive members.

The ordering of members in an archive can make a difference in how programs are linked using the library, if a symbol is defined in more than one member.

If no modifiers are used with **m**, any members you name in the *file* arguments are moved to the *end* of the archive. If desired, you can use the '**a**', '**b**' or '**i**' modifiers to move them to a specified place.

p *Print* the specified members of the archive, to the standard output file. To show member names before their contents are copied to standard output, specify the '**v**' modifier.

If you specify no *files*, all the archive files are printed.

q *Quick append*: add *files* to the end of *the archive*, without checking for replacement.

The modifiers '**a**', '**b**' and '**i**' do *not* affect this operation; new members are always placed at the end of the archive.

The modifier '**v**' makes **rar** list each file as it is appended.

Since the point of this operation is speed, the archive's symbol table index is not updated, whether it already existed or not. You can use '**rar s**' or **rranlib** to explicitly update the symbol table index.

r Insert *files* into *archive* (with *replacement*). This operation differs from '**q**' in that any previously existing members are deleted if their names match those of the files being added.

If one of the named *files* does not exist, **rar** displays an error message, and does not disturb any existing archive members with that name.

By default, new members are added at the end of the file, but you may use one of the modifiers '**a**', '**b**' or '**i**' to request placement relative to some existing member.

The modifier '**v**' used with this operation elicits a line of output for each file inserted, along with either '**a**' or '**r**' to indicate whether the file was appended (no old member deleted) or replaced.

t Display a *table* listing the contents of *archive*, or those of the files listed in *files* that are present in the archive. Normally only the member name is shown. If you also want to see the modes (permissions), timestamp, owner, group, and size, you can request a specific combination using the '**v**' modifier.

If you do not specify any *files*, all files in the archive are listed.

If there is more than one file with the same name (say, for example, '**fie**') in an archive (say, '**b.a**'), '**rar t b.a fie**' will list only the first instance. To see them all, you must ask for a complete listing–for our example use, '**rar t b.a**'.

x *Extract* members (named *files*) from the archive. You can use the '**v**' modifier with this operation, to request that **rar** list each name as it extracts it.

If you do not specify any *files*, all files in the archive are extracted.

1.3.1 modifiers

A number of modifiers (*mod*) may immediately follow the *p* keyletter, to specify variations on an operation's behavior:

a Add new files *after* an existing member of the archive. If you use the modifier **a**, the name of an existing archive member must be present as the *membername* argument, before the *archive* specification.

b Add new files *before* an existing member of the archive. If you use the modifier **b**, the name of an existing archive member must be present as the *membername* argument, before the *archive* specification. (same as '**i**').

c *Create* the archive. When you request an update, the specified *archive* is always created if it did not exist. Unless you specify this modifier, a warning is issued when the archive file is created.

i Insert new files *before* an existing member of the archive. If you use the modifier **i**, the name of an existing archive member must be present as the *membername* argument, before the *archive* specification. (same as '**b**').

l This modifier is accepted but not used.

o Preserve the *original* dates of members when extracting them. If you do not specify this modifier, files extracted from the archive will be stamped with the time of extraction.

s Write an object-file index into the archive, or update an existing one, even if no other change is made to the archive. You may use this modifier flag either with any operation, or alone. Running '**rar s**' on an archive is equivalent to running '**rranlib**' on it.

u Normally, **rar r...** inserts all files listed into the archive. Use this modifier if you would like to insert *only* those of the files you list that are newer than existing members with the same names. The '**u**' modifier is allowed only for the '**r**' (replace) operation. In particular, the combination '**qu**' is not allowed, since checking the timestamps would cancel any speed advantage expectes from the '**q**' operation.

v This modifier requests the *verbose* version of an operation. Many operations display additional information such as names of files processed, when the '**v**' modifier was appended.

V This modifier shows the **rar** version number.

1.4 see also

rnm, **rranlib**.

2 rranlib

Generate index to archive.

2.1 synopsis

rranlib [**-v**|**-V**] *archive*

rranlib is one of the GNU binary utilities[2] ported to the RWC-1. **rranlib** generates an index to the contents of an archive, and stores it in the archive. The index lists each symbol defined by archive members that are relocatable object files.

You may use '**rnm -s**' or '**rnm --print-armap**' to list this index.

An archive with such an index speeds up linking to the library, and allows routines in the library to call each other without regard to their placement in the archive.

The **rranlib** program is another form of **rar**. Running **rranlib** is completely equivalent to executing '**rar -s**'.

2.2 options

-v

-V Print the **rranlib** version number and exit.

2.3 see also

rar, **rnm**.

3 rnm

List symbols from object files.

3.1 synopsis

rnm [**-a**|**--debug-syms**] [**-g**|**--extern-only**] [**-B**]
[**-C**|**--demangle**] [**-s**|**--print-armap**]
[**-A**|**-o**|**--print-file-name**] [**-n**|**-v**|**--numeric-sort**]
[**-p**|**--no-sort**] [**-r**|**--reverse-sort**]
[**-u**|**--undefined-only**] [**--help**] [**-V**|**--version**]
[**-t** *radix*|**--radix**=*radix*] [**-P**|**--portability**]
[**-f** *format*|**--format**=*format*] [*objfile* ...]

3.2 description

rnm is one of the GNU binary utilities[2] ported to the RWC-1. **rnm** lists the symbols from object files *objfile*. If no object files are given as arguments, **rnm** assumes '**a.out**'.

3.3 options

The long and short forms of options, shown here as alternatives, are equivalent.

-A

-o

--print-file-name Precede each symbol by the name of the input file where it was found, rather than identifying the input file once only before the symbol list.

-a

--debug-syms Display debugger-only symbols. Normally, these are not listed.

-B The same as **--format=bsd** (included for compatibility with the MIPS **nm**).

-C

--demangle Decode (*demangle*) low-level symbol names into user-level names. Besides removing any initial underscore pretended by the system, this makes C++ function names readable.

-f *format*

--format=*format* Use the output format *format*, which can be "bsd", "sysv" or "posix". The default is "bsd". Only the first character of *format* is significant. It can be either upper or lower case.

-g

--extern-only Display only external symbols.

-n

-v

--numeric-sort Sort symbols numerically by their addresses, not alphabetically by their names.

-p

--no-sort Do not bother to sort the symbols, just print them as they are encountered.

-P

--portability Use the POSIX.2 standard output format instead of the default format. Equivalent to "-f posix".

-s

--print-armap When listing symbols from archive members, include the index: a mapping (stored in the archive by **rar** or **rranlib**) of what modules contain definitions for what names.

-r

--reverse-sort Reverse the sense of the sort (whether numeric or alphabetic), such that last comes first.

-t *radix*

--radix=*radix* Use *radix* as the radix for printing the symbol values. It must be "d" for decimal, "o" for octal, or "x" for hexadecimal.

-u

--undefined-only Only display undefined symbols (those external to each object file).

-V

--version Show the **rnm** version number and exit.

--help Show a summary of the **rnm** options and exit.

3.4 see also

rar, **robjdump**, **rranlib**.

4 robjdump

Display information from object files.

4.1 synopsis

robjdump [**-a**] [**-f**] [**-h**|**--header**] [**-j** *section*] [**-l**]
[**-r**|**--reloc**] [**-s**] [**--stabs**] [**-t**|**--syms**] [**-x**]
[**--version**] [**--help**] *objfile ...*

4.2 description

robjdump is one of the GNU binary utilities[2] ported to the RWC-1. **robjdump** displays information about one or more object files. The options control what particular information to display. This information is mostly useful to programmers who are working on compilation tools, as opposed to programmers who simply want to compile their programs.

objfile ... are the object files to be examined. When you specify archives, **robjdump** shows information on each of the member object files.

4.3 options

Where long and short forms of an option are shown together, they are equivalent. At least one option must be given besides **-l**.

-a If any files from *objfile* are archives, display the archive header information (in a format similar to '**ls -l**'). In addition to the information you can list with '**rar tv**', '**robjdump -a**' shows the object file format of each archive member.

-f File header. Display summary information from the overall header of each file in *objfile*.

-h

--header Header. Display summary information from the object file section headers.

--help Print a summary of the **robjdump** options and exit.

-j *name* Only display section *name* information.

-l Label the display (using debugging information) with the source filename and line numbers corresponding to the object code shown.

-r

--reloc Relocation. Print the file's relocation entries.

-s Display the full contents of any sections requested.

--stabs Display the contents of the .stab, .stab.index, and .stab.excl sections from an ELF file. This is only useful on systems in which .stab debugging symbol-table entries are carried in an ELF section.

-t

--syms Symbol Table. Print the symbol table entries of the file. This is similar to the information provided by the '**rnm**' program.

--version Print the **robjdump** version number and exit.

-x Display all available header information, including the symbol table and relocation entries. Using '**-x**' is equivalent to specifying all of '**-a -f -h -r -t**'.

4.4 see also

rnm.

5 rsize

List section sizes and total size.

5.1 synopsis

rsize [**-A**|**-B**|**--format**=*compatibility*] [**--help**]
[**-d**|**-o**|**-x**|**--radix**=*number*] [**-V**|**--version**] *objfile...*

5.2 description

rsize is one of the GNU binary utilities[2] ported to the RWC-1. The **rsize** utility lists the section sizes–and the total size–for each of the object files *objfile* in its argument list. By default, one line of output is generated for each object file or archive module.

5.3 options

-A

-B

--format *compatibility* Using one of these options, you can choose whether the output from **rsize** resembles output from System V **size** (using '**-A**', or '**--format sysv**'), or Berkeley **size** (using '**-B**', or '**--format berkeley**'). The default is a one-line format similar to Berkeley's.

--help Show a summary of acceptable arguments and options.

-d

-o

-x

--radix *number* Using one of these options, you can control whether the size of each section is given in decimal ('**-d**', or '**--radix 10**'); octal ('**-o**', or '**--radix 8**'); or hexadecimal ('**-x**', or '**--radix 16**'). Only these three values (8, 10, 16) are supported in '**--radix** *number*'. The total size is always given in two radices: decimal and hexadecimal for '**-d**' or '**-x**' output, or octal and hexadecimal if you're using '**-o**'.

-V

--version Display **rsize** version number.

5.4 see also

rar, **robjdump**.

6 rstrings

Print the strings of printable characters in files.

6.1 synopsis

rstrings [**-a**|**-**|**--all**] [**-f**|**--print-file-name**] [**-o**] [**--help**]
[**-v**|**--version**] [**-n** *min-len*|*-min-len*|**--bytes**=*min-len*]
[**-t** {*o,x,d*}|**--radix**={*o,x,d*}] *file*

6.2 description

rstrings is one of the GNU binary utilities[2] ported to the RWC-1. For each *file* given, **rstrings** prints the printable character sequences that are at least 4 characters long (or the number given with the options below) and are followed by a NUL or newline character. By default, it only prints the strings from the initialized data sections of object files; for other types of files, it prints all of the file's strings.

rstrings is mainly useful for determining the contents of non-text files.

6.3 options

The long and short forms of options, shown here as alternatives, are equivalent.

-a

--all

- Scan all of the object files, rather than just the initialized data sections.

-f

--print-file-name Print the name of the file before each string.

--help Print a summary of the **rstrings** options to standard output and exit.

-v

--version Print the **rstrings** version number to standard output and exit.

-n *min-len*

-min-len

-bytes=*min-len* Print sequences of characters that are at least *min-len* characters long, instead of the default length of 4.

-t {*o,x,d*}

--radix={*o,x,d*} Print the offset within the file before each string. The single character argument specifies the radix of the offset–octal, hexadecimal, or decimal.

-o Like **-t** *o*.

6.4 see also

rar, **rnm**, **robjdump**, **rranlib**.

7 rstrip

Discard symbols from object files.

7.1 synopsis

rstrip **[-s|--strip-all]** **[-S|-g|--strip-debug]**
[-x|--discard-all] **[-X|--discard-locals]**
[-v|--verbose] **[-V|--version]** **[--help]** *objfile...*

7.2 description

rstrip is one of the GNU binary utilities[2] ported to the RWC-1. **rstrip** discards all symbols from the object files *objfile*. The list of object files may include archives. At least one object file must be given.

rstrip modifies the files named in its argument, rather than writing modified copies under different names.

7.3 options

--help Show a summary of the **rstrip** options and exit.

-s

--strip-all Remove all symbols.

-S

-g

--strip-debug Remove debugging symbols only.

-x

--discard-all Remove non-global symbols.

-X

--discard-locals Remove compiler-generated local symbols. (These usually start with "L" or ".".)

-v

--verbose Verbose output: list all object files modified. In the case of archives, **rstrip -v** lists all members of the archive.

-V

--version Show the **rstrip** version number and exit.

References

[1] Steve Chamberlain and Roland Pesch, Edited by Jeffrey Osier., *Using LD, the GNU linker*, March 1993.

[2] Roland H. Pesch., *The GNU Binary Utilities, Version 2.2*, May 1993.