REDUCE 3.7 Installation and Rebuilding

Codemist Ltd

March 24, 1999

1 Introduction

This document explains how to install, build or rebuild REDUCE 3.7 using the CSL Lisp system. It covers the cases where your computer runs Microsoft Windows 95, 98 or NT, or Linux or various brands of Unix. Much of the description here will only be relevant to those who have obtained the "professional" version of REDUCE, which comes complete will all its source code. If only the "personal" version is available all material here that relates to source code may be ignored.

2 Installation

2.1 Windows

REDUCE is installed by running the `setup` program that forms part of the distribution. This prompts to discover which components of REDUCE you want to install and where you would like the files placed.

Note that under Windows two versions of the REDUCE executable are provided. The one called `r37.exe` runs in a window in the usual way. The one called `r37c.exe` runs as an old-fashioned command-line program. This version may prove more useful if REDUCE is to be run from a script. If you often use a command-window it may be useful to create a simple one-line batch file called `r37.bat` and contents

```
x:\r37\lisp\csl\win32\r37 %*
```

(where you should replace `x:` with the path to where the REDUCE files were installed) and place this file in a directory that is on your search path. You can the launch REDUCE by just issuing the command `r37`. As an alternative you could add the REDUCE executable directory to your path. These steps have not been automated as part of the installatiom procedure since many users will be content launching REDUCE by clicking on the relevant icon.

2.2 Unix and Linux

REDUCE is supplied as a collection of compressed archives. They are called

`r37exec.tar.gz`	REDUCE executables plus a few support files
`r37xmpl.tar.gz`	Example files nd matching logs
`r37doc.tar.gz`	Documentation
`r37src.tar.gz`	Source code of REDUCE
`r37lisp.tar.gz`	Source code of the CSL Lisp system

You will first need to access the files on the CDROM. This may involve mounting it: you will use whatever procedure you usually do for accessing a CDROM. I will suppose that this makes the files available with paths such as `/cdrom/r37exec.tar.gz` but the exact path will depend on how your Linux is configured.

Select a directory within which you wish to install REDUCE. Select it is the current directory. Unpacking the tar files will create a sub-directory called "r37" and everything that is unpacked will be within this directory. If you are the system manager you might reasonably select `/usr/local`, and if you are an individual user you might just start in your home directory. In each case unpack such of the archives as you will want using a command like:

```
cd /usr/local
tar xvfz /cdrom/r37exec.tar.gz
```

It is suggested that to start with you unpack the executables, the examples and the documentation. Between them these will use up between 25 and 30 Mbytes of disc space. The directory structure created will be as shown below. Names in bold fixed-pitch type are represent directories, while those in italic are files.

- `r37`
  - `doc`
    - `html`
      - ...
      - *r37.html*
    - *install.pdf*
    - *r37.pdf*
    - *uguide.pdf*
    - `util`
  - `gnuplot`
  - `lisp`
    - `csl`
      - `csl-c`
      - `cslbase`
      - `linux(etc)`
        - `log`
        - *Makefile*
        - *r37*
        - *r37.img*
      - `util`
        - *setexec*
  - `log`
  - `packages`
    - ...
    - `support`
      - *patches.red*
    - ...

If you have unpacked the REDUCE executables you will find a file `r37/lisp/csl/linux/r37`, and to make it easier to use the system you might like to install a link to it from some convenient place, eg using a command similar to one of the two following ones:

```
ln -s /usr/local/r37/lisp/csl/linux/r37 /usr/local/bin/r37
ln -s ~/r37/lisp/csl/linux/r37 ~/bin/r37
```

For these to make sense either `/usr/local/bin` or the `bin` sub-directory of your home directory should be on your usual search path. If you set up the a symbolic link as shown then reduce should be able to find the `r37.img` file it needs without you having to do anything more.

If you are running on some version of Unix it may be that file-permissions were not set up when the REDUCE files moved to your machine. Select the `r37/lisp/csl/util` directory as current and run the script "setexec"

```
cd r37/lisp/csl/util
source setexec
```

to correct this if you need to (and you can tell that if you get a complaint when you try to execute one of the scripts in the `util` directory).

Documentation is placed in `r37/doc`, with the manual in a form suitable for use with Adobe's Acrobat reader as `r37/doc/r37.pdf` or some more browsable HTML help accessed starting from `r37/doc/html/r37.html`.

For Linux the provisions of the GNU Public Library License mean that you have to be provided with object files for all the executables that together form REDUCE. The reason for this is that you may want to (or indeed need to) re-link them with a newer version of the Linux system libraries. Because different Linux installations may have been installed with different generations of these libraries and some versions are not compatible with others it *could* be that you will find that when first unpacked the r37 executable will not load and run at all. To re-build it from object code you should use the script called `relink` that is in the `r37/lisp/csl/util` directory:

```
cd r37/lisp/csl/linux
sh ../util/relink
```

To use this script you will have to have a set of Linux development tools available, specifically `gcc` and the usual C libraries.

If you intend to use the REDUCE interface to `gnuplot` you should ensure that that package is installed. Gnuplot itself is not part of REDUCE and it may be easiest for you to fetch and install it in some quite independent way — for instance many Linux CDROMS or web mirrors provide it as a standard option for your installation. The directory `r37/gnuplot` contains copies of `gnuplot` files as they would be found on a typical software archive, and if necessary you can unpack and install from there. For further information about `gnuplot`, its installation and use, you should check the documentation files that accompany it and the web sites that they reference.

3 Testing an installation

On first installing REDUCE it may make sense to run all the REDUCE test scripts that have been provided. This should make it possible to verify that that installation was correct, and it also as a side-effect produces a log file that compares the speed REDUCE has on your machine with that observed on a reference one at Codemist. For the initial release of REDUCE 3.7 this reference system is based on an Intel Pentium II running at 400 MHz, and the tests were run under Windows NT 4.0.

To run the complete tests you need to select the correct part of the REDUCE tree as your current directory, and in the case of Unix systems you need to make sure that the executable status of various files are properly set. The `setexec` script in `r37/lisp/csl/util` arranges this. Then `util/testall` runs all the tests and `util/checkall` compares the results with a set of reference logs. At the end you will find your own logs from individual packages in `r37/log`, a file showing any differences between your results and the reference set in `r37/csl/lisp/<system>/log/checkall.log` and a summary of timing in `r37/csl/lisp/<system>/log/times.log`.

Windows	Unix

`cd r37\lisp\csl\win32`
`..\util\testall`
`..\util\checkall`
	`cd r37/lisp/csl/util`
	`sh ./setexec`
	`cd ../linux`
	`../util/testall`
	`../util/checkall`

On the reference computer running the full set of tests takes between 20 and 30 minutes on an otherwise unused machine. If you interrupt the tests part way through the script `util/testrest` (used just as `util/testall` is) will continue running tests from wherever you broke off. Some workstations may be substantially slower (certainly by up to a factor of ten) than the reference machine so you may need some patience here.

The `log/checkall.log` file will contain a section for each test that was run. The differences reported will certainly include every line that reports how long anything took. If you have installed any patches in your version of REDUCE those too might cause changes in the test output (and these changes will not then represent errors). Also on different platforms the exact results from some numeric calculations done in machine-arithmetic will differ.

The file `log/times.log` has a line in it for each of the test files, and this records the time that this test tool on both the local and the reference computer. It ends up with giving a speed ratio between the two machines based on all these results. Each test has two times associated with it. One includes and the other excludes "garbage collection" time. High garbage collection overheads are generally an indication that you are short of memory. The time excluding garbage collection is expected to be tolerably consistent, but even then it can vary to 5 to 10 percent even on a single computer. Such issues as the exact position and layout of files on the disc or other background activity that the computer runs at the same time have such effects.

4 Removal

For Unix you can remove REDUCE by just removing the `r37` directory and all its contents, and any symbolic links you made into it. For Windows de-installation is via the "add/remove programs" item on the control panel as usual.

5 Applying patches

From time to time minor updates and corrections to REDUCE will be published: these can be located via the REDUCE home page, which is `http://www.rrz.uni-koeln.de/REDUCE/`. The corrections will be present as a downloadable file called `patches.red`. Comments in this file should explain what changes are being made, but the bulk of the material there is not intended for the casual reader. To install the patches you should first locate the patches file in the REDUCE file structure. It should be `r37/packages/support/patches.red`. Make a safe backup copy of this file, and replace it with the new copy that you have downloaded. Read the comments in the new patches file to see what has changed. Note that whenever you install new patches you can expect some changes in the REDUCE test logs, at least in formatting and layout. Now go

```
cd r37/lisp/csl/linux  (or win32, or whatever)
# back up r37.img for safety here, please
sh ../util/patchup     (or ..\util\patchup)
```

REDUCE should run for a short time and update `r37.img` with the new patches. When you next launch REDUCE its startup banner should reflect the new date associated with the version of the patches file you have just incorporated. Note that there is no way to remove a set of patches short of re-building the whole of REDUCE from source: just putting back an older `patches.red` file and re-applying that is not guaranteed to undo all effects of an intermediate patch. So keep a copy of your original `r37.img` so you can re-instate that if you have any trouble. That should be the only file changed by the `patchup` job.

6 Re-building from the REDUCE sources

If you have a copy of the Professional Version of REDUCE it comes complete with all source files. Users of the Personal system do not have the files needed to do this and can ignore the rest of this document. You may wish to recompile either just one REDUCE module or the whole system. This will mainly be the case if you are developing new packages for REDUCE. The recipe is

```
cd r37/lisp/csl/linux  (or win32, or whatever)
../util/full37
```

This runs for a two or three minutes on the reference system, and generates a log file in `log/full37.log`. The version of `r37.img` that it re-creates should have all current patches installed.

To re-compile just a single REDUCE package, for instance `groebner`, use the sequence

```
cd r37/lisp/csl/linux  (or win32, or whatever)
../util/package groebner
```

7 Re-compiling the CSL Lisp system

REDUCE is built on top of a Lisp system: in this place that Lisp is called CSL and all its sources are included in the Professional Version. When distributed in this way the CSL Lisp system is intended for use just in support of REDUCE so detailed information about its capabilities and support for it (apart for as a component of REDUCE) is not provided. The core parts of CSL are coded in the C language. This makes it (fairly) easy to move CSL and hence REDUCE to new computer architectures provided they have reliable C compilers and can give enough of an illusion that they support 32-bit code. Image files such as `r37.img` can be created using a CSL that runs on one computer architecture and re-loaded on another. Thus to mount REDUCE on another type of computer you just need to compile this C code. Create a new directory, calling it `r37/lisp/csl/my-machine` and select it as the current directory. Select one of the ready-made version of `Makefile` from `../util` and copy it of link it into this new directory. If your local configuration is not exactly the same as one of the ones already catered for you will need to edit the `Makefile` by hand. If you are attempting this sort of re-compilation it is assumed that you already know enough to sort out the details of that for yourself. In particular you may well find that your C compiler needs some special flags setting or (even more probable) that custom directives are needed to get all relevant libraries scanned.

For use on Windows you will find `Makefile.w32` is set up to use the Watcom C compiler (tested using version 11), and `Makefile.vc` to use Microsoft's Visual C++ (tested using version 5). `Makefile.gcc` is a good starting place for a generic Unix port using the widely available free GNU C compiler.

It should then be the case that just saying `make r37` (or `make r37.exe` in the Windows case) should build the relevant executable. There is no guarantee that the source code will compile either first time or correctly on any system other than the one you specified when you originally obtained it, and porting to new architectures can call for changes in various system-dependent parts of the code. Thus this level of re-compilation is intended to provide flexibility for the expert who can cope with such issues for themselves rather than being a fully-supported and guaranteed recipe for use by novices. In particular in the past newer versions of C compilers have sometimes been incompatible with older versions from the same vendor and Codemist does not guarantee to keep the CSL C sources updated to cope with all such possible oddities.

As previously mentioned, when you have a new `r37` executable you can copy and existing `r37.img` to the directory it lives in (REDUCE looks for this image file in the directory it finds its executable in) and test the system.

Of course even though you have the capability to re-compile all of REDUCE in this way that does not mean you have permission to use it beyond the terms of your license, and in particular you may not distribute versions of the software that you have compiled for existing or new machines.

8 Creating new profile information

Parts of the REDUCE source code are translated into C and incorporated as part of the CSL Lisp system. Doing this helps with performance. After major changes to the REDUCE sources it may be useful to be able to review which parts of REDUCE deserve this optimisation and to re-create the C. Doing this is a fairly costly business and only a very few users are liable to want to attempt it. The recipe (shows as done on Windows this time) is

```
cd r37\lisp\csl\win32
make slowr37.exe
..\util\boot37
```

This makes a special version of REDUCE that has full functionality but which is much larger and slower than the final version. It does not have any parts of it compiled into C.

```
..\util\profile
```

The job that profiles REDUCE runs for over an hour on the reference machine. It creates a file `profile.dat` in the current directory. This file lists the most heavily used functions as revealed by running all the REDUCE test scripts. You then need to copy this file to the place where the standard copy of it lives and use it to guide selective compilation into C:

```
copy profile.dat ..\csl-c
..\util\c-code37
```

Finally you must re-compile `r37.exe` and re-build `r37.img` to match. If you do one of these but not the other the system can be in an incoherent state and may crash arbitrarily:

```
make r37.exe
..\util\full37
```

After such a major re-build it would be prudent to run all the tests again:

```
del ..\..\..\log\*.rlg
..\util\testall
..\util\checkall
```

and inspect `checkall.log` to see that all is well.