REDUCE 3.7 User Guide

Codemist Ltd

March 24, 1999

1 Introduction

This document provides information that may help REDUCE users. It documents all the command-line options that REDUCE supports and explains about images, checkpointing and user-written initialisation files. For information on how to test or re-build all or parts of REDUCE see the installation guide.

2 Command-line options

The options here are can be used when REDUCE is run from a command prompt or of the windows interface provides a start-up dialog box that allows you to specify some of them.

`--` ***filename*** arranges that output is sent to the indicated file. It is intended to behave a little like "`>` *filename*" as command-line output redirection, but is for use in in windowed environments (in particular Windows NT) where this would not work. Under Windows if `--` is used REDUCE starts off with its main window minimised. If you launch REDUCE from a script and want to capture all its output it is suggested that you use this rather than shell re-direction of the standard output to get behaviour which is maximally portable across all the platforms that REDUCE runs on.

`-b` is a curious option, not intended for general or casual use. If given it causes the (batchp) function to return the opposite result from normal! Without "-b" (batchp) returns T either if at least one file was specified on the command line, or if the standard input is "not a tty" (under some operating systems this makes sense – for instance the standard input might not be a "tty" if it is provided via file redirection). Otherwise (ie primary input is directly from a keyboard) (batchp) returns nil. Sometimes this judgement about how "batch" the current run is will be wrong or unhelpful, so `-b` allows the user to coax the system into better behaviour.

`-c` just prints a dull and unimaginative copyright notice – having this option in there will tend to ensure that a copyright message is embedded in the object code somehow, while with luck nobody will be bothered too much by the fact that there is a stray option to get it displayed. Note that on some systems there is a proper character for the Copyright symbol… but there is little agreement about what that code is! Thus in some cases the message displayed may appear to have a junk character in it.

`-d` ***name=value*** defines a symbol at the start of a run. You may leave a blank after `-d` or start the symbol name immediately. The value specified will be set up as a Lisp string. If the equals sign and following value are omitted the name concerned gets set to the value `t`, the regular Lisp representation of "true". This mechanism makes it possible to pass information from the command line down into REDUCE and is used to control several of the testing and re-compilation jobs that are distributed as part of REDUCE.

`-e` is reserved by Codemist to enable "experimental" features in its code. The option may well be ignored, but if it is not its behaviour will not necessarily be consistent across versions or time. For instance it may at times have been used to enable special debugging modes within the Lisp system.

`-f` ***number*** is not supported on all versions of REDUCE. It instructs the system to start up and listen on a socket with the specified number (the request `-f-` uses a default socket, which is number 1206). Requests on this socket cause a new copy of REDUCE to be forked (with a limit to how many such tasks can be active at once) and requests passed down the socket are then processed much as if they came directly from the keyboard. This facility is intended to be of use when running REDUCE on a central system to support remote clients. In this release of the software it is not guaranteed and end-users of REDUCE are not given any further information about how to try to use it.

`-g` sets some options that may be usefule when debugging code. Specifically it is a short-hand for `-d*backtrace`, and has an effect similar to saying "`on backtrace;`" at the start of a REDUCE run.

`-i` ***image-file*** specifies an image file that should be made readable by REDUCE when it start up. See the section on image files later in this guide. The request `-i-` specifies the default image, which will normally by called `r37.img` and will be found in the directory where the REDUCE executable lives.

`-k` ***number*** sets the size of the heap that REDUCE should use in units of Kilobytes. Thus `-k12000` suggests use of (around) 12 Mbytes. REDUCE normally expands its memory when it needs to and most users should not need to use this option. An extended form of the request such as `-k8000/2` uses a second number (2 in this example) to control the amount of space used for one of the system's internal stacks. The default value for this second number is 1: if you experience stack overflow problems it may be worth increasing this number to 2 or even 3, but unless you do see stack overflow crashes there is no benefit at all in trying to tune this parameter. REDUCE should (on a large enough computer) be able to cope with heap sizes up to most of 2000 Mbytes. `-k0` tells REDUCE to use its default storage allocation, which is to start off with a few megabytes allocate but then to expand its heap as need-be. A directive to use a specified amount of memory disables this ability to increase the heap size at run-time.

`-l` ***filename*** arranges to send a transcript of a REDUCE session to the named file. This option differs from `--` in that it displays all output on your screen in the normal way as well as keeping the transcript. Once REDUCE is running the Lisp-level function `spool` can be used to set up exactly the same sort of logging, eg using a command such as

```
lisp spool "logfile.log";
```

and on some versions of REDUCE there may be a menu item for the same purpose. These all use the same mechanism so use of one overrides and previous use of one of the others. Which you choose to find is a matter of personal preference and convenience.

`-m` ***n:l:h*** If `MEMORY_TRACE` was defined when REDUCE was compiled then this option makes it possible to provoke an interrupt after n memory probes when a reference in the (inclusive) range l to h is next made. The release version of REDUCE is not compiled with this option active, and indeed use of the option slows REDUCE down dramatically but makes it possible for Codemist to collect detailed traces of how it accesses memory. These traces have been used in the past to guide optimisation of the code. In normal cases this option will not be accepted and would not be useful or relevant to end users.

`-n` is sometimes useful to developers if they have created a new image file that doe snot restart properly. To be more specific, an image file contains copies of all the REDUCE code, and an indication of which part ot it should be run when REDUCE is started. If a new image had a restart function that was damaged but all other parts of it were useful it may be useful to launch REDUCE with the `-n` option which tells the system to ignore the restart function specified in the image and run a simple Lisp read-eval-print loop instead. This may allow an expert to debug or possibly correct the problem they had introduced. It should be clear from this explanation that ordinary users are not likely to want to use this switch, and if they do they will be faced with a Lisp-like interface that will tend to confuse them.

`-o` ***image-file*** is similar to `-i` but specifies an image file that is to be opened for output. There shoudl only be one such file and it will be the one that `faslout` and `preserve` write information to. See the later section on image files.

`-p` is reserved for system profiling options, and will normally have no effect on a release version of REDUCE.

`-q` is a specific converse of `-v` but represents the default behaviour for REDUCE and so is not generally useful.

`-r` ***n** [,n]* sets a seed for a random number generator. The default REDUCE-level generator is deterministic: it yields the same sequence each time REDUCE is run. For cases where this is not what is wanted an alternative Lisp-level function `random-number` exists. Its default behavior (which can be asked for specifically by saying `-r0`) is to seed its sequence based on the time of day and such other hard to repeat things. On some computers it may include timing information from user keystrokes and mouse activity and only freeze the seed when the user first asks for a random value, and in such cases best unpredictability will be achieved by performing several other calculations before doing anything that asks for a random number. This Lisp-level generator can be forced into a defined state by giving a command line option `-r` followed by one or two integers. If two integers are used up to 64 bits of seed can be specified.

`-s` is a short-hand for `-s*plap` and thus causes the Lisp compiler to display the bytecodes that it generates when compiling any REDUCE code. Some users may be interested to see this code and to judge how compact it is, but probably not many!

`-t` **-slshape module-name** was implemented to support some `perl` scripts that were being written to keep a version of REDUCE up to date. When invoked with this command-line option REDUCE does not run at all. It just prints a message to its standard output reporting the size and date associated with a module with the given name and it then stops. The idea here was that `perl` could capture and parse this information to test if the compiled module contained within the REDUCE image was up to date relative to the corresponding source file. This release version of REDUCE uses other schemes to re-build modules but the option is left enabled in case it is useful to others who wish to embed REDUCE within some larger software support structure. Note that when running with Windows there is no "standard output" so on that platform you either need to use the `r37c` command-line version of REDUCE or use `--` to capture the information to a file.

`-u` ***name*** undefines the given symbol, and is thus a converse of `-d`. There are probably not many circumstances where this is useful, but the inclusion of this option is motivated by completeness and by analogy with the options usually provided by C compilers.

`-v` causes more messages to be displayed when REDUCE starts up, and also a message when it stops (reporting the total run-time used). To see garbage collection messages from REDUCE it is necessary to issue a request

```
lisp verbos 3;
```

one REDUCE has started running.

`-w` is not used on current versions of REDUCE but is reserved for command-line control over whether REDUCE should run as a windowed or a command-line program.

`-x` is a debugging option only intended for use by Codemist. Normally if REDUCE detects in internal exception it attempts to recover. If `-x` is specified on the command line it allows the trap to take effect and this will typically generate a `core` file on Unix or attempt to enter a debugger on Windows. Anybody who attempts to modify the C code of the REDUCE/Lisp kernel and then find that they need to debug the resulting system may need this option, but in such circumstances Codemist can obviously not guarantee anything!

`-y` is a short-hand for `-d*hankaku` and this flag is associated with Japanese language support in some versions of REDUCE. In versions distributed directly by Codemist such support is not available.

`-z` instructs the system that it should not load an initial heap image, but should run in "cold start" mode. This is only intended to be useful for system builders: it arises in a few of the rebuilding scripts that are included with REDUCE.

filename indicates a file that REDUCE should read input from in preference to accepting input from the terminal.

3 Image files

@@@@@@@@@@@@@@@@@@@

4 Initialisation files

When REDUCE starts up it checks for the presence of an initialisation file. If you need to avoid this you can specify `-dno_init_file` on the command line when starting REDUCE, and indeed the various REDUCE maintainance scripts do just this.

REDUCE tries to identify a "home directory". It does this by looking first for the value of a shell variable `home`, then `HOME` and then `HOMEDRIVE` and `HOMEPATH`. The first two such cases should work under almost all Unix environments and find your home directory. Under Windows you can arrange to set `HOME` or you can set `HOMEDRIVE` and `HOMEPATH` and the full home directory will be taken as the concatenation of these. If none of these environment variables are set then REDUCE will look in the current directory. In the directory so identified it looks for one of the following files

```
.r37.rc
r37.rc
r37.ini
```

and it will use the first of these files that it finds. It reads and executes REDUCE commands.

The various places that REDUCE searches are intended to mean that users of various flavours of Unix and of Windows can put an initialisation file in the place and with the name that looks most familiar to them.

Obviously an initialisation file can be used to set REDUCE flags or otherwise set up a custom configuration. When reporting any REDUCE problems either demonstrate them with `-dno_init_file` or remember to report the exact contents of your initialisation file since otherwise the behaviour that puzzles you may be hard to reproduce.

You may have several different initial behaviours that you want to have. If you copy the files `r37.exe` and `r37.img` and give them different names such as `myr37.exe` and `myr37.img` then this copy of REDUCE will look for its initialisation file as `myr37.rc` (or the other variants). Under Windows the command `r37c.exe` is a command-line version of REDUCE (as distinct from one that runs in a window). It uses the same image file and initialisation files as the normal version of REDUCE. If you copy a REDUCE executable to another location you should follow the convention that a windowed version has a name not ending in "c" and a command-line version has the same name but with "c" added to the end, as in `myr37c.exe`.