

CoCoA-5 - Feature #719

Deadline: consider creating a textual interpreter with readline

21 May 2015 17:47 - John Abbott

Status:	Closed	Start date:	21 May 2015
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	5.00 hours
Target version:	CoCoA-5.1.2 summer 2015	Spent time:	5.10 hours
Description This was discussed in March 2009 via email. I will copy the essence of the messages here. Christof would greatly appreciate a readline textual version (since he's an emacs heathen).			
Related issues: Related to CoCoA-5 - Bug #745: Readline: avoid printing out READLINE twice Closed 09 Jul 2015			

History

#1 - 22 May 2015 16:46 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

I have downloaded and built readline-6.3.

Building was non-trivial: by default it compiled 32-bit (not compatible with CoCoALib).
CFLAGS="-m64 -O2" ./configure failed because it created incompatible shared libs (probably a bug with my old MacOSX-10.5.8?). Anyway I finally got a successful build by excluding the shared libs:
CFLAGS="-m64 -O2" ./configure --disable-shared
make

#2 - 22 May 2015 16:52 - John Abbott

- Estimated time set to 5.00 h

JAA does not want to make readline mandatory for compiling CoCoALib, nor even CoCoA-5.
So readline will become another external lib. This requires appropriate additions to configure, and also to autoconf.mk.

I plan first to make a working version of LineProviders.C assuming readline is present (in a system default place?). Then I'll think about putting in appropriate #ifdef macros.

#3 - 22 May 2015 17:45 - John Abbott

- Assignee set to John Abbott
- % Done changed from 10 to 20

I have an apparently working first prototype :-)

#4 - 22 May 2015 18:04 - John Abbott

I had to change just one line in Main.C (and add some #include directives).

What to do?

- **(A)** one possibility is to offer a command line arg to CoCoAInterpreter which tells it to use readline
- **(B)** or we could make use of readline the default, if it is present.

What should happen in **(A)** if the user requests readline and it has not been compiled in?

A quick test showed that the readline version works fine inside emacs, so hopefully there ought not to be any problem if we chose **(B)**.

#5 - 22 May 2015 18:17 - John Abbott

There is some problem with lines longer than 1023 chars. I think it is an Emacs/comint problem because cutting and pasting the same input (the value of factorial(1000)) in a plain terminal produced no problems. I recall that non-readline CoCoA-5 gives problems with not-so-very-long lines.

#6 - 22 May 2015 18:25 - John Abbott

Compilation of CoCoAInterpreter with require the option -lreadline if the readline library is to be used.

#7 - 22 May 2015 18:35 - John Abbott

The CoCoA-5 tests passed with readline active. So I have checked in (to reduce the risk of losing the code).

Still need to change configure, autoconf.mk and define a suitable flag to tell the linker to use readline library if necessary.

#8 - 22 May 2015 18:44 - Anna Maria Bigatti

John Abbott wrote:

What to do?

- **(A)** one possibility is to offer a command line arg to CoCoAInterpreter which tells it to use readline
- **(B)** or we could make use of readline the default, if it is present.

I'll check out and run some test within emacs.

IF there are indeed some problems with emacs then it would be great to have a command line arg so that we can disable readline when calling from emacs, but having readline by default when calling from a terminal.

#9 - 24 May 2015 14:26 - John Abbott

I have just updated configure. If a quick test passes, I'll check in.

Currently it is a bit of a hack: *e.g.* it assumes that libreadline.a is properly installed.

Still to do:

- treat readline like the other external libs.
- make configuration work even if readline is not properly installed.
- testing compatibility with emacs (should be OK unless you send weird chars, *e.g.* inside a string)

No error is given if readline is absent or is incompatible with compilation flags inherited from GMP; in these cases CoCoAInterpreter works as it has done until now. [well, that's the theory...]

#10 - 24 May 2015 14:29 - John Abbott

It's not perfect (e.g. prints out READLINE twice in the list of ext libs).
Checking in anyway, as I must go now.

#11 - 25 May 2015 12:17 - John Abbott

Strictly the headers for readline are needed only for compiling LineProviders.C.
Similarly libreadline.a is needed only when linking CoCoAInterpreter.

The flag CoCoA_WITH_READLINE is needed only in LineProviders.C, and **should not** be passed to the qmake files for compiling the GUI.

JAA thinks that the readline defines, flags, etc. need to be handled separately from those of other external libs.

#12 - 26 May 2015 15:51 - Anna Maria Bigatti

- % Done changed from 20 to 30

(I had to do a bit of updating and adaptations) it compiles and works well.

Under emacs, both in shell and in comint, the input line is echoed (I wonder why!!), so the command line option "--no-readline" would be nice.
I think I can do it.... but not today.

I'll also take care of the flags to the GUI compilation.

#13 - 02 Jun 2015 11:27 - Christof Soeger

My first try was not successfull. Durign the configure I got

```
/tmp/ccdhDbN0.o: In function `main':
test-readline.C(.text+0xf): undefined reference to `readline'
collect2: error: ld returned 1 exit status
```

I now investigated it and the problem is the compile command. The library is put in front of the .C file and therefore not used to resolve dependencies from the .C, at least on my compiler (g++ 4.8.2). Here is a patch for readline-check-cxxflags.sh:

```
-$CXX" $CXXFLAGS $* test-readline.C -o test-readline
+$CXX" $CXXFLAGS test-readline.C $* -o test-readline
```

#14 - 02 Jun 2015 12:05 - Christof Soeger

And it also works fine for me. Is it also possible to include a tab-completion? :)

Only problem is that the tests of CoCoAInterpreter fail because the .found now also includes the input commands.

And one minor point, on start it reports readline twice:

```
With CoCoALib and external libraries NORMALIZ READLINE READLINE
```

#15 - 02 Jun 2015 12:45 - John Abbott

Clearly, I was too hasty with my hack to add readline :-(

I know READLINE appears twice at the start -- don't know why, not so important.

The echoing is a nuisance; Anna had already noticed it when running CoCoA inside emacs. I suppose readline turns echoing on when the input is not from a tty. There's probably an easy flag to turn it off... just have to read the documentation (and add the right line somewhere).

#16 - 02 Jun 2015 13:11 - John Abbott

I have just read some documentation for readline and also looked online. We're not the first people to have problems with undesired echoing, and it seems there is no very clean solution (*i.e.* guaranteed portable standard C++).

There is a function called isatty but that is designed to work with C file descriptors rather than C++ streams.

Probably the safest solution is to have a command line option which disables readline, or maybe an internal flag which is set when the interpreter knows that it is reading from a file...

#17 - 02 Jun 2015 13:19 - Christof Soeger

A commandline option to disable readline is probably a good idea. It is the most flexible. You can then also add some automatic behaviour (then include also a "force enable readline" option).

#18 - 28 Jun 2015 18:49 - John Abbott

- % Done changed from 30 to 50

I have added a new command flag **--no-readline**; it seems to work as desired.

I have also modified cocoa5.el so that emacs disables readline; the new cocoa5.el will cause trouble with an old binary (but that's asking for trouble anyway).

I have also modified src/CoCoA-5/tests/RunTests.sh so that it disables readline. Oddly, I did not have the echoing problems (in the tests) which Christof had.

Updated percentage to 50%; someone still has to deal cleanly (see comment 9) with the external readline library, so I'm not increasing the percentage any higher for now.

#19 - 30 Jun 2015 11:27 - Anna Maria Bigatti

- *Status changed from In Progress to Feedback*

Tried: it works fine for me. Close?

#20 - 28 Jul 2015 15:56 - Anna Maria Bigatti

- *Status changed from Feedback to Closed*

- *% Done changed from 50 to 100*