RPNcalc

–

A Reverse-Polish notation scientific calculator

James Tappin

July 9, 2012
Version 0.1

Abstract

rpncalc `https://github.com/jtappin/RPNcalc` is a fully functional scientific calculator emulator, written entirely in Fortran. It is both a usable tool and a demonstration of the gtk-fortran libraries `https://github.com/jerryd/gtk-fortran`.

1 SYNOPSIS

rpncalc [**options**] . . .

2 DESCRIPTION

A scientific decktop calculator using Reverse Polish notation.

3 OPTIONS

-c, - -closed start the calculator with the stack and memory display hidden.

-o, - -open start the calculator with the stack and memory display visible (this is the default).

-r, - -restore *file* Load the stack and memory contents from the given file.

-m, - -registers *number* Set the number of memory registers to create. The default is 10.

-R, - -radians Set the trigonometry unit to radians.

-D, - -degrees Set the trigonometry unit to degrees.

-G, - -grads Set the trigonometry unit to grads (1/100 of a right angle).

-h, - -help Print a help text and exit.

4 User's Guide

4.1 Introduction

The gtk-fortran RPN calculator is a fully functional scientific calculator emulator. It is not based on any particular model of calculator, but rather assembles those features that I find useful in a calculator. It uses the stack-based reverse Polish logic system for 2 reasons:

1. That's what I like to use, and most calculator emulators are algebraic notation.
2. I'm lazy and it's a lot easier to implement a reverse Polish system.

It is intended to be both a useful tool, and a demonstration of what can be done using gtk-fortran (and in particular the fortran-only high-level routines). The calculator is entirely written in Fortran 2003 (with a few 2008 features such as the `execute_command_line` subroutine.

4.2 Entering values:

All values with RPNcalc are stored as double-precision reals (Gtk+ has no support for long doubles).

Values can be entered either using the keypad, or by typing into the entry box. Values entered from the keypad make sanity checks for 2 decimal points or a decimal entered after the exponent has been started and the change-sign key works in a reasonably intelligent way. When values are typed or pasted into the entry box, characters that cannot be part of a number are rejected with a warning.

A value may be transferred from the entry to the stack by pressing the keyboard `Enter` key while focus is on the entry window, or by clicking the **Enter** key on the keypad. The **Dup** key copies the entry box to the stack without clearing the entry box. If the contents of the entry box are not a valid number (i.e. a Fortran `READ` statement cannot convert it to a floating point value) a message is displayed in the status bar and you may edit the entry box to correct the problem.

4.3 Operators:

The operators (**+**, **-**, *****, **/** and **y**x** and also the **atan2** function) operate on the entry box and the top element on the stack if there is anything in the entry box. If the entry box is empty, then they operate on the top 2 elements of the stack.

Operators may also be accessed by typing the operator into the entry window (N.B. The exponentiation operator is ^ rather than $**$ for convenience of implementation). The addition and subtraction operators will only work in this way if a sign would not be a valid part of a number where they are entered, notably + or − in an empty entry box is not treated as an operator. Note also that operators at the end of a multi-character paste are not accepted.

The result is placed on the top of the stack, and displayed in the result window.

4.4 Functions:

The functions operate on a single value, which is taken from the entry box if that has content or from the top of the stack otherwise. The result is placed on the top of the stack, and displayed in the result window.

If the **Inverse** checkbox is set, then functions are replaced by their inverses (e.g. **sin** becomes **asin**), in addition, the **y**x** operator becomes the corresponding root and the *roll down* button becomes *roll up*. Note that:

1. The less-used functions in the pulldown are not affected by this, as none have meaningful inverses.

2. The power operator ^ entered from the keyboard is not converted to a root.

The **Rad**, **Deg** and **Grad** radio buttons are used to select Radians, Degrees or Grads for the trigonometric functions.

The **HMS** and the various base-*n* keys are not proper functions, they don't remove or add anything to the stack. **HMS**displays the contents of the entry box or the top of the stack as if it were a number of hours converted to *HH***:***MM***:***SS.sss* format (or *D***d** *HH***h** *MM***m** *SS.sss***s** if the value is greater than 24). The **Hexadecimal**, `Octal` and **Binary** keys will display the integer representation of entry value or the top of the stack in the respective notations provided that the value is an integer.

Some less-used functions are in the **More** pulldown. The **atan2** function computes `arctan(y/x)` removing the quadrant ambiguities.

Functions whose arguments are out of range will produce an error message and the stack is left unchanged.

4.5 Stack operations

- **CE** clears the entry box, or if that is empty deletes the top entry on the stack.
- **CA** clears the entry box and all entries on the stack.
- The *up* button moves the selected item in the stack up one place. If the top item (or nothing) is selected then it is exchanged with the entry box.
- The *down* button moves the selected entry on the stack down one place.
- The *roll down* button, moves the last element of the stack to the top and all others down one place.

4.6 Constants:

There are a number of built in fundamental physics constants that are build in to the calculator, these can be entered from the **Phys** pull-down menu.

4.7 Memory Registers:

The calculator also has memory registers (numbered from 0). The contents of these registers can be viewed by selecting the "registers" tab. The default number of registers is 10, but this may be set with the **-m** option, or by setting a new number in the spin box at the bottom of the registers tab.

These can be accessed in one of two ways:

1. Select a register in the registers tab, and then click a memory operation. In this case the value used will be the entry box or the top of the stack if the entry is empty.
2. Enter a register number in the entry box and click the memory operation. The value used is the top of the stack.

The operations are:

- **STO:** Store the value in the selected register.
- **RCL:** Copy the selected register to the top of the stack
- **M+:** Add the value to the selected register
- **M-:** Subtract the value from the selected register
- **MCL:** Clear the selected register.
- **MCA:** Clear all registers

4.8 Statistics:

If the **Live stats** toggle is enabled, then a summary of the statistical properties of the contents of the stack is maintained in the "Statistics" tab of the display area.

4.9 Save & Restore:

The stack, registers and entry box can be saved to and restored from a text file with the **File->Save** and **File->Restore** menu items.

The file format is a plain text file with the floating point values written in hexadecimal – this allows the retention of full-precision but is endian-independent. Obviously any machines that do not use IEEE floating point values will not be able to read files from other machines. Also any machine with a `c_double` that is not 8-bytes will not work.

4.10 Settings:

In the current version, there are 2 user-definable settings accessed through the **Edit** menu:

4.10.1 Result Format:

Specify the format to use in the result box. You have the options to select one of the standard formats:

Fixed: A fixed number of decimal places (set in the precision spin box). The actual format used is "(F0.*n*)". WARNING: this may be a GNU extension.

Sci: Scientific format. Specify the number of decimals, and the width of the exponent in the spin boxes. The total width is calculated automatically.

Eng: Engineering format, similar to scientific, except that the exponent is always a multiple of 3.

Free: Use a list-directed write (the default).

Alternatively you can type an explicit Fortran format statement into the combo box (with or without the enclosing parentheses). This may be any Fortran formatting code valid for a REAL type. Setting it to "*" or an empty string will use the default list-directed output (as will an invalid format).

This menu also includes an option to show or suppress the leading zeroes in base-*n* displays.

4.10.2 Show degrees:

If this is enabled, then use angular rather than time notation for the HMS display (the button will be relabelled **DMS**).

4.11 Cut & Paste:

The **Edit** menu has options to cut or copy the selected text in the entry window (or the result window in the case of copy) to the clipboard. The current clipboard item my also be pasted into the entry box, or the selected text may be deleted. The usual keybindings for these operations are also available.

4.12 Help system:

This manual can be accessed in a number of ways:

1. A manpage is generated and can be accessed using `man rpncalc`.
2. The "Help" item in the menu will display either the text version or (if the environment variable `RPNCALC_VIEWER` is set to a PDF viewer) the PDF version.
3. The text and PDF versions are installed in the `share/docs/rpncalc` subdirectory of the installation directory.

4.13 Accelerators:

The menu items have accerators to save mouse clicking:

- Save – ctrl-s
- Restore – ctrl-o
- Quit – ctrl-q
- Cut – ctrl-x
- Delete – ctrl-shift-x
- Copy – ctrl-c
- Paste – ctrl-v
- Set Format – ctrl-f
- Help – ctrl-h
- About – ctrl-a
- About gtk-fortran – ctrl-shift-a

5 AUTHORS

James Tappin (jtappin at gmail dot com).

6 LICENCE