Linux Calculator Overview

By Steve Litt
http://www.troubleshooters.com

Partial Linux Calculator List

- bc
- dc
- calc
- orpie
- GNU Octave
- x48
- Scheme as calculator

Calculators by Category

RPN
- dc
- orpie
- x48

Infix
- bc
- calc
- Octave

Prefix
- Scheme

Scriptable
- dc
- bc
- calc +
- Octave +,++
- Scheme

CLI
- dc
- bc
- calc
- Scheme

Curses
- Orpie

GUI
- Octave*
- x48

+ With suitable grep

++ CLI mode only

*** Besides GUI, Octave has a CLI mode**

Prefix, Infix and Postfix

- Think “position of the operator”
- Prefix
  - “(+ 3 2)”
- Infix
  - “3 + 2”
- Postfix
  - “3 2 +”

Infix

- “3 + 2”
- Sometimes called “Normal”
- Used in C, Java, Python, most calculators, etc.
- Parentheses if going against operator precedence

Postfix

- “3 2 +”
- Often called “Reverse Polish Notation”, or RPN
- Usually stack based
- Used in old HP Calculators, Forth, Postscript
- No operator precedence or parentheses
- Number of operands defined by operator
- Most tolerant of lack of planning
  - 1r/ replaces stack top with its reciprocal

Prefix

- “(+ 3 2)”
- Used in Lisp, Scheme, Guile
- Parentheses always
- No operator precedence
- Number of operands open

Stack Basics

- Resembles a stack of plates
- Can only access the top one (pop or look)
- Can only add to the top (push)
- Last In First Out (LIFO)
- Operands go on stack
- Operators pop operands from stack
- Operators push result back on stack

Reverse Polish Notation

- RPN for short
- Uses a stack
- Used in dc, Orpie, x48
- Most operators do the following process:
  1) Pop value(s) from stack
  2) Operate on the value(s)
  3) Push the result to the stack
- No parentheses, no operator precedence
- Tends to need less keystrokes
- More tolerant of lack of planning
- But RPN is unfamiliar to most folks

dc basics

- f: Look at whole stack, stack unchanged
- p: Look at top of stack, stack unchanged
- k: Pop number and use as precision
- +-*/%^: Pop 2 from stack, push result
- sd: Pop 1 from stack, store in register d
- ld: Push value of register d. Register d unchanged.
- d: Push another copy of stack top
- r: Reverse top 2 stack entries
- Warning: No sin(), ln(), roots, etc.

Canned dc Programs

- Iteration
- Fibonacci with shell vars
- Fibonacci entirely in dc
  - Pop value(s) from stack
  - Operate on the value(s)
  - Push the result to the stack
- No parentheses
- Tends to need less keystrokes
- More tolerant of lack of planning
- But RPN is unfamiliar to most folks

Calculator Experiments

- Any calculator
- Any task