

3.5 bc: THE CALCULATOR

UNIX provides two types of calculators—a graphical object (the `xcalc` command) that looks like one, and the text-based `bc` command. The former is available in the X Window system and is quite easy to use. The other one is less friendly, extremely powerful and remains one of the system's neglected tools.

When you invoke `bc` without arguments, the cursor keeps on blinking and nothing seems to happen. `bc` belongs to a family of commands (called *filters*) that expect input from the keyboard when used without an argument. Key in the following arithmetic expression and then use `[Ctrl-d]` to quit `bc`:

```
$ bc
12 + 5
17
[Ctrl-d]
$
```

Value displayed after computation
The eof character

`bc` shows the output of the computation in the next line. Start `bc` again and then make multiple calculations in the same line, using the `;` as delimiter. The output of each computation is, however, shown in a separate line:

```
12*12 ; 2^32
144
4294967296
```

^ indicates "to the power of"

Maximum memory possible on a 32-bit machine

bc performs only integer computation and truncates the decimal portion that it sees. This shows up clearly when you divide two numbers:

```
9/5
1
```

Decimal portion truncated

To enable floating-point computation, you have to set `scale` to the number of digits of precision before you key in the expression:

```
scale=2
17/7
2.42
```

Truncates to 2 decimal places

Not rounded off, result is actually 2.42857.....

bc is quite useful in converting numbers from one base to another. For instance, when setting IP addresses (17.1.3) in a network, you may need to convert binary numbers to decimal. Set `ibase` (input base) to 2 before you provide the number:

```
ibase=2
11001010
202
```

Output in decimal—base 10

The reverse is also possible, this time with `obase`:

```
obase=2
14
1110
```

Binary of 14

In this way, you can convert from one base to the other (not exceeding 16). **bc** also comes with a library for performing scientific calculations. It can handle very, very large numbers. If a computation results in a 900-digit number, **bc** will show each and every digit!