Selected topics in software development

Today:

Test automation

Speaker:

Jyrki Katajainen

1. the development of software to automatically test a system can be as big a job as the development of the system software itself
2. there is no **The System**; there are only **System Versions**

Testing a killer application

A script for regression testing runs the old version (`old_ka`) and the new version (`new_ka`) for a large number of different test data files, and complains about each one for which the outputs are not identical. Such a script should usually run silently, producing output only if something unexpected occurs.

```
for i in ka_data.*
do
  old_ka $i >out1
  new_ka $i >out2
  if ! cmp -s out1 out2
  then
    echo $i: BAD
  fi
done
```

[Brian W. Kernighan & Rob Pike, The practice of programming, §6.3]

Testing Awk

Many language constructions were tested by running specified inputs through tiny programs and checking that the right output was produced. ...test run the new version of Awk on a short Awk program to produce output in one file, write the correct output to another file with `echo`, compared the files, and reported an error if they differed.

For simple expressions, we created a small specialized language for describing tests, input data, and expected outputs. ... An Awk program converted each test into a complete Awk program, then ran each input through it, and compared actual output to expected output; it reported only those cases where the answer was wrong.

Whenever the program was changed, even in a trivial way, the whole test suite was run; it took only a few minutes.

[Brian W. Kernighan & Rob Pike, The practice of programming, §6.3]

Testing `memset`

The function `memset(s, c, n)` set `n` bytes to the byte `c`, starting at `s`, and returns `s`.

```
  big = maximum left margin + maximum n + maximum right margin
  s0 = malloc(big)
  s1 = malloc(big)
  for each combination of test parameters n, c, and offset:
    set all of s0 and s1 to known pattern
    run slow memset(s0 + offset, c, n)
    run fast memset(s1 + offset, c, n)
    check return values
    compare all of s0 and s1 byte by byte

offset = 10, 11, ..., 20
c = 0, 1, 0x7F, 0x80, 0xFF, 0x11223344
n = 0, 1, 2, 3, 4, 5, 7, 8, 9, 15, 16, 17, 31, 32, 33, ..., 65535, 65536, 65537
```

[Brian W. Kernighan & Rob Pike, The practice of programming, §6.4]

Testing a raster graphics library

Consider a raster graphics library involving an operator that copies blocks of pixels from one image to another.

1. Write by hand simple code (slow) that correctly handles a single pixel and use this to test the library's version (fast).
2. Do the same for a single horizontal row of pixels.
3. Continue this way using rows to build rectangles, rectangles to build tiles, and so on.
4. If a test failed , the tester printed out a detailed analysis to aid understanding what went wrong, and also to verify that the tester was working properly itself.

[Brian W. Kernighan & Rob Pike, The practice of programming, §6.4]

Testing the Markov program

A shell script generated necessary input data, ran and timed the tests, and printed any anomalous output. The script was configurable so the same tests could be applied to any version of Markov [C, Java, C++ Awk, Perl], and every time we made a set of changes to one of the programs, we ran all the tests again to make sure that nothing was broken.

[Brian W. Kernighan & Rob Pike, The practice of programming, §6.8]

Summary

- You should know at least one scripting language!
- A little language for writing tests makes it easy to create a lot of them; using a program to write a program to test a program has high leverage.