The Great Win32 Computer Language Shootout

Back to the Win32 Shootout
Back to dada's perl lab

[The Original Shootout] [NEWS] [FAQ] [Methodology] [Platform Details] [Acknowledgements] [Scorecard]

[51 Language Implementations, 25 Benchmark Tests, 1275 Total Possible Programs, 983 Completed]

Here's a list of which solutions have/have not been implemented.

The Benchmark Tests
Ackermann's Function
Array Access
Count Lines/Words/Chars
Echo Client/Server
Exception Mechanisms
Fibonacci Numbers
Hash (Associative Array) Access
Hashes, Part II
Heapsort
Hello World
List Operations
Matrix Multiplication
Method Calls
Nested Loops
Object Instantiation
Producer/Consumer Threads
Random Number Generator
Regular Expression Matching
Reverse a File
Sieve of Erathostenes
Spell Checker
Statistical Moments
String Concatenation
Sum a Column of Integers
Word Frequency Count
(Not all languages are tested in every benchmark)

A benchmark comparison of a number of programming languages.

Note: **I'M NOT THE AUTHOR OF THE SHOOTOUT!** this is a port of the work done by Doug Bagley on the Win32 platform. I've added some Win32-specific languages (for which Doug is not to blame :-), but most of the material is copyright © Doug Bagley. The original shootout can be found here.

Intro

As stated in the note above, this is a port of the Great Computer Language Shootout by Doug Bagley on the Win32 platform. Doug made a hell of a good job with this test suite on its Linux box. Since most of the programming languages are implemented on Win32 without many differences, I started taking his source code and let it run on my Win32 box.

Please note that all the disclaimers, warnings, and notes by Doug apply to this shootout as well. This is ncluding, but not limited to:

- *...some solutions as currently presented are unoptimized...*
- *...please take the current results with a grain of salt...*
- *...you might even find that I have horrible bugs in my testing method...*
- *I'm doing it so that I can learn about new languages, compare them in various (possibly meaningless) ways, and most importantly, have some fun.*

And, above all:

- *"There are lies, damn lies, and benchmarks"*.

I should also add another very important disclaimer: **this is by all means not not be intended as a comparison between Linux and Windows!** The test platforms (both hardware and software configuration) are completely different, so the results from Doug's benchmarks can't be absolutely compared to mine. Please don't use this stuff to fuel your holy wars (eg. Windows suckz, Linux rulez, ...).

That said, the most important differences between the original shootout and this one are:

- I've added some Win32-specific languages:
  - Visual C++
  - VBScript
  - C#
- I've also added some languages not implemented by Doug:
  - Rexx
  - Pascal (Free Pascal, Virtual Pascal, Delphi)
  - Awka
  - Pliant
  - Mozart-Oz
  - Modula-3
  - elastiC
  - Parrot
- The benchmark results are somewhat more reliable (especially memory usage): I've used some Win32 API tricks to determine the process start time, end time and peak memory usage AFTER the process has finished, so we have data even for very quick processes.

The Languages			
Language		Imple-mentation (local summary)	Version (Official Homepage)
1.	Ada	***gnat***	GNAT 3.15p (20020523) Copyright 1996-2002 Free Software Foundation, Inc.

About the Languages

The languages that are in ***bold italics*** compile to machine code. The others are either byte-compiled or just interpreted.

A few Win32-specific notes:

- For many languages I've just used the binary distribution provided by the makers (or somebody else). The relevant information are available on the language page. I've done this mainly because I think that it is generally normal for Win32 users (as opposed to Linux/Unix users) to use an already compiled, ready to install program instead of compiling everything from

2.	Awk	gawk	GNU Awk 3.1.3
3.	Awk	***awka***	awka 0.7.5, 12 July 2001
4.	Awk	mawk	mawk 1.3.3 Nov 1996, Copyright (C) Michael D. Brennan
5.	C	***mingw32***	2.95.3-6
6.	C	***gcc***	gcc (GCC) 3.3.1 (cygming special) Copyright (C) 2003 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. (Cygwin Version 1.5.5(0.94/3/2))
7.	C	***lcc***	Logiciels/Informatique lcc-win32 version 3.8 compilation date: Jun 3 2002 11:11:13
8.	C	***vc***	Visual Studio 6.0 (Compiler Version 12.00.8168)
9.	C	***bcc***	Borland C++ 5.5.1 for Win32
10.	C#	***csharp***	Microsoft (R) Visual C# .NET Compiler version 7.00.9466
11.	C++	***vc++***	Visual Studio 6.0 (Compiler Version 12.00.8168)
12.	Common Lisp	poplisp	Version 15.53 06/04/2001 18:46:40
13.	D	***d***	Digital Mars D Compiler ALPHA v0.42
14.	Delphi	***delphi***	dcc (Borland Delphi for Windows) 14.01
15.	Eiffel	***se***	Release -0.74 (Tuesday May 7th 2002)
16.	ElastiC	elastic	Copyright (C) 1998-2001 Marco Pantaleoni. All rights reserved.
17.	Erlang	erlang	Erlang (THREADS) (BEAM) emulator version 5.2.3.3
18.	Forth	***bigforth***	bigforth 386-Win32 rev. 2.0.0
19.	Forth	gforth	gforth 0.5.0
20.	Haskell	***ghc***	The Glorious Glasgow Haskell Compilation System, version 5.04.2
21.	ICI	ici	ICI 4.0.5
22.	Icon	icon	Icon Version 9.3.1. May 4, 1998
23.	Java	***java***	Java(TM) 2 Runtime

scratch. I tried to resort to building the language only when the provided binary did not work out of the box.

Note that this also means that I'm accepting all the compilation flags and/or possible optimization as set by the original builder.

- Many languages (gcc, for example, but some others too) are depending on the Cygwin subsystem (more details about this).
- Other languages (Ocaml, for example, but some others too) make use of external C and/or Assembler compiler. Most of the times, I've accepted the provided default configuration. Additional information about the compiler(s) used are available on the language page.
-

			Environment, Standard Edition (build 1.4.1_03-b02)
24.	JavaScript	jscript	5.6.0.8513 (WSH Version 5.6)
25.	Lua	lua	Lua 4.0
26.	Lua	lua5	
27.	Mercury	***mercury***	Mercury 0.10.1
28.	Modula-2	***modula2***	O2/M2 development system v2.50 (c) 1991-2001 Excelsior, LLC. (build 20.02.2002)
29.	Modula-3	***modula3***	SRC Modula-3 version 3.5 (February 1, 1995)
30.	Nice	***nice***	ICI 4.0.5
31.	Ocaml	ocamlb	The Objective Caml compiler, version 3.06 (Byte Code)
32.	Ocaml	***ocaml***	The Objective Caml compiler, version 3.06
33.	Oz	***oz***	Mozart Compiler 1.2.5 (20030214) playing Oz 3
34.	PHP	php	PHP 4.3.1 (cli) (built: Feb 15 2003 23:07:00) Copyright (c) 1997-2002 The PHP Group Zend Engine v1.3.0, Copyright (c) 1998-2002 Zend Technologies
35.	Parrot	parrot	This is parrot version 0.0.9-devel built for i386-MSWin32
36.	Pascal	***vpascal***	Virtual Pascal Version 2.1.243 Copyright (C) 2000 vpascal.com
37.	Pascal	***fpascal***	Free Pascal Compiler version 1.0.4 [2000/12/30] for i386
38.	Perl	perl	This is perl, v5.8.0 built for MSWin32-x86-multi-thread
39.	Perl	cygperl	This is perl, v5.8.0 built for cygwin-multi-64int
40.	Pike	pike	Pike v7.4 release 10
41.	Pliant	pliant	Pliant dynamic reflexive compiler, release 81, debugging level 0, for i386 under Windows, compiled as a DLL using GCC (registers calling convention).
42.	Python	python	Python 2.3.2
43.	REBOL	rebol	
44.	Rexx	rexx	REXX-Regina_3.0.1 4.95 12 May 2002
45.	Ruby	ruby	ruby 1.6.7 (2002-03-01) [i586-mswin32]

46.	S-Lang	slang	S-Lang Library Version: 1.4.6
47.	SML	***smlnj***	Standard ML of New Jersey, Version 110.0.7, September 28, 2000 - -
48.	Scheme	guile	Guile 1.4
49.	Simula	***cim***	cim-3.36-i686-pc-cygwin
50.	Tcl	tcl	8.4.2
51.	VBScript	vbscript	5.6.0.7426 (WSH Version 5.6)
Languages that compile to native code are in ***Bold Italics*** .			

Back to the Win32 Shootout
Back to dada's perl lab

Send comments or suggestions to dada@perl.it