Hello World: An Instance Of Rhetoric in Computer Science

John Mount: jmount@mzlabs.com

February 19, 2008

Computer scientists have usually dodged questions of intent, purpose or meaning. While there are theories that assign deep mathematical meaning to computer programs[13] we computer scientists usually avoid discussion of meaning and talk more about utility and benefit. Discussions of the rhetorical meaning of programs is even less common. However, there is a famous computer program that has a clean an important rhetorical point. This program is called "hello world" and its entire action is to write out the phrase "hello world." The action is simple but the "hello world" program actually has a fairly significant purpose and meaning.

I would like to briefly trace the known history of "hello world" and show how the rhetorical message it presents differs from the rhetoric embodied in earlier programs. In this sense we can trace a change in the message computer scientists felt they needed to communicate (most likely due to changes in the outside world).

Since the late 1970's it has been a tradition in computer science that when facing a new system to start by writing the traditional "first program", which is called "hello world." The history of this tradition is not well documented, but WikiBooks claims the origin is in Brain Kerninghan's 1974 tutorial for the computer language "B."[14, 8]

The program itself is as follows:

```
main( ) {
  extrn a, b, c;
  putchar(a); putchar(b); putchar(c); putchar('!*n');
}
```

```
a 'hell';
b 'o, w';
c 'orld';
```

From the original context in [8] we can say this is not a pure “hello world” program. The purpose of this program is to illustrate a few functions of the language (“extrn” variables, use of multiple lines and so on) and not to test if the system is running.

The most famous (and a pure) example of “hello world” is found in Brian Kerninghan and Dennis Ritchie’s famous 1978 C book[9]. The code looks like the following:

```
main() {
        printf("hello, world\n");
}
```

Equally famous are the early versions in “BASIC”:

```
10 print "Hello, World!"
```

In both of these examples the purpose of “hello world” is clear: it is a trivial computer program that does nothing but print a single line of text. If this can not be made to work then nothing can be made to work. “hello world” is in fact a somewhat confrontational program. The author is saying “it isn’t obvious your computer system will work, so I am not going to invest a lot of time in it until I see it can at least print one line of text.”

Writing “hello world” as the first program on a new system was certainly a well known tradition by the late 1970’s. We can ask: did this tradition showed up late or early in the history of computer science?

The first modern computer is commonly credited to John von Neumann in 1945. This attribution of “first” was made before a lot of previously secret information on Bletchley Park’s and Konrad Zuse’s work in Germany were well known, but von Neumann’s work was definitely the known foundation that later work built on. Donald Knuth credits von Neumann with the first modern computer program[10]. In any case we can securely place the invention of the modern computer to the early to mid 1940’s. It seems like “hello world” was not written at that time because it made entirely the wrong rhetorical point.

In fact von Neumann's first program (in 1945) was a sorting algorithm (proving the computers could at least replace tabulators)- it was important to prove that not only could the computer turn on but that it could *do something.*

The 1957 description of FORTRAN starts with a program that solves for roots of an equation.[4] John Backus himself says the prior to 1954 "almost all programming was done in machine language or assembly language"[3] (which would make "hello world" an unlikely first program as numeric operations are typically much more succinct than string manipulations in these languages).

The 1958 UNIVAC Math-Matic Programming Manual first examples are (predictably) equations involving trigonometric functions.[5]

In 1959 the Algo 58 standard [2] had no concrete example programs- the purpose of this document was to show that program syntax could be specified and to discuss the techniques of specification (so a runnable example did not serve their argument).

John McCarthy's paper on recursive functions (LISP) starts with examples partial functions and conditionals (to get to terminating recursion as fast as possible).[11] The Lisp1 manual [6] concentrates on the translation of information into data-structures (an important point) and the Lisp1.5 manual [12] is written for the famous "bottom of page 13" moment where "eval" (essentially the semantic core of LISP) is defined in a few lines LISP code.

The computer language "BASIC" seems almost designed to support "hello world." Instead the 1964 BASIC manual[7] starts with a lively discussion of what a computer program is (as a process) and the first example program is a highly useful program that solves simultaneous equations. The point again being that useful work can be done.

Even as late as 1972 "hello world" does not seem to be the obvious message. For example the popular BASIC tutorial called "My Computer Likes Me, When I Speak in BASIC"[1] does indeed start with a simple program that only writes a line of text. But the line it writes is "MY HUMAN UNDERSTANDS ME." This is making a very different point than the point made by "hello world."

While I have skipped a number of languages of the period (Autocode, COBOL, APL, SNOBOL, PL/1, Logo, BCPL, Forth, Smalltalk ...) we can see the general trend: fist programs had to make a point. By the dates it seems that once invented the "hello world" tradition took off and spread very quickly. Writing such a program serves a useful purpose as a test so the practice probably spread quickly once the rhetoric of the earlier programs was no longer needed. That is “hello world” became popular once computer scientists no longer felt that society needed to be persuaded of the ultimate utility of computer systems.

With the increasing complexity of modern systems “hello world” is a more important test than ever. “hello world” is sometimes called the hardest application to deploy. The idea is that once you learn your lessons from deploying it then deploying a second, more sophisticated, application seems relatively easy. I often find that I can lean the true nature of a system by deploying “hello world.” If writing and deploying “hello world” is a sufficiently unpleasant task in a system then it is likely that every other task in that system will be similarly unpleasant.

In the end “hello world” serves the same purpose that it always has (testing if a system in fact works) and it stands as a rhetorical marker signifying that we are now living long after the skirmishes of the computer revolution.

References

[1] Albrecht, B. *My Computer Likes Me, When I Speak in BASIC.* Dymax, 1972.

[2] Backus, J. W. The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference.

[3] Backus, J. W. The history of fortran i, ii and iii. *ACM SIGPLAN Notices 13*, 8 (Aug 1978).

[4] Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., Nelson, R. A., Sayre, D., Sheridan, P. B., Stern, H., Ziller, I., Hughes, R. A., and Nutt, R. The fortran automatic coding system. *Proceedings of the Western Joint Computer Conference* (Feb 1957).

[5] Corp., R. R. U. *UNIVAC math-matic programming system.* Sperry Rand, 1958.

[6] Fox, P. Lisp 1 programmers manual. 165.

[7] Kemeny, J., and Kurtz, T. Basic.

[8] Kerninghan, B. W. A tutorial introduction to the language b.

[9] Kerninghan, B. W., and Ritchie, D. M. *The C Programming Language*. Prentice Hall, February 1978.

[10] Knuth, D. E. von neumann's first computer program. *Comp. Surveys 2*, 4 (1970), 247–260.

[11] McCarthy, J. Recursive functions of symbolic expressions and their computation by machine, part i.

[12] McCarthy, J., Abrahams, P. W., Edwards, D. J., Hart, T. P., and Levin, M. I. Lisp 1.5 programmers manual. 116.

[13] Tennent, R. D. The denotational semantics of programming languages. *Communications of the ACM 18*, 8 (Aug 1976), 437–453.

[14] Wikibooks. `http://en.wikibooks.org/wiki/Computer_programming/Hello_world#B`, 2008.