CLOUDFLARE™

Turing's Curse

July 26, 2013

John Graham-Cumming

A LONG, LONG TIME AGO...

The Mother of All Demos - 1968

CLOUDFLARE

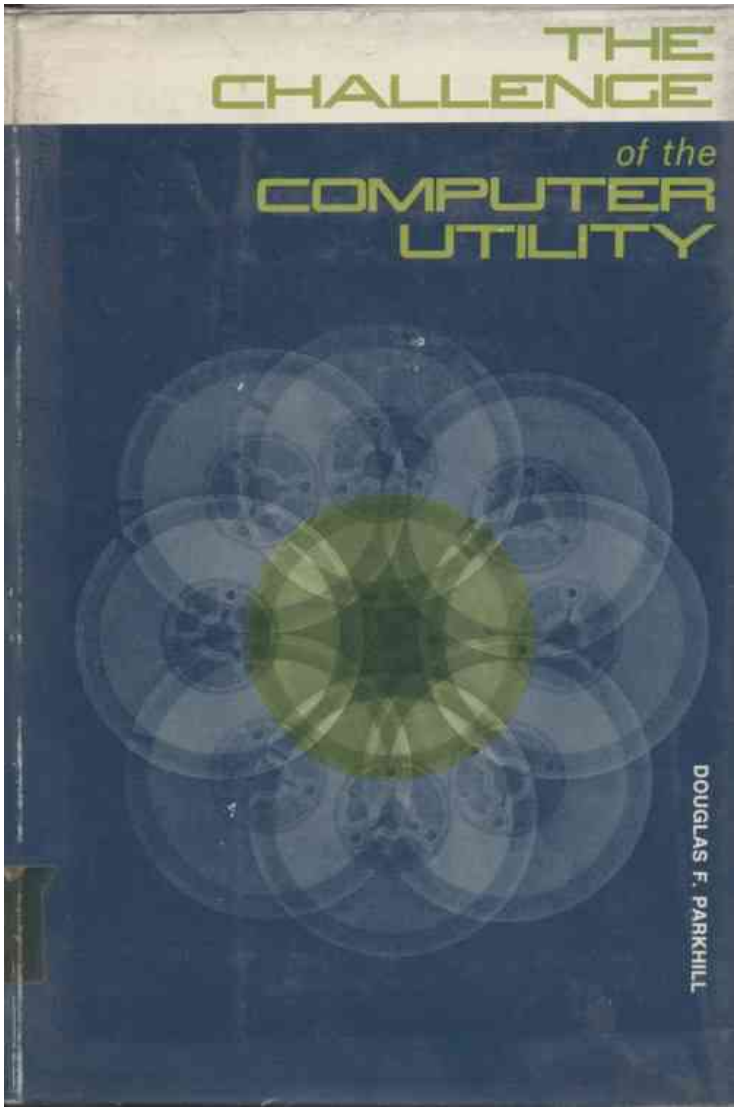
WHAT'S NEW?

I’ve got a bad feeling about this

CLOUD COMPUTING

Cloud Computing - 1966

1966 - 1968



BIG DATA

Big Data in 1955

1955 - 1968

CLOUDFLARE

VIRTUAL MACHINE HYPERVISOR

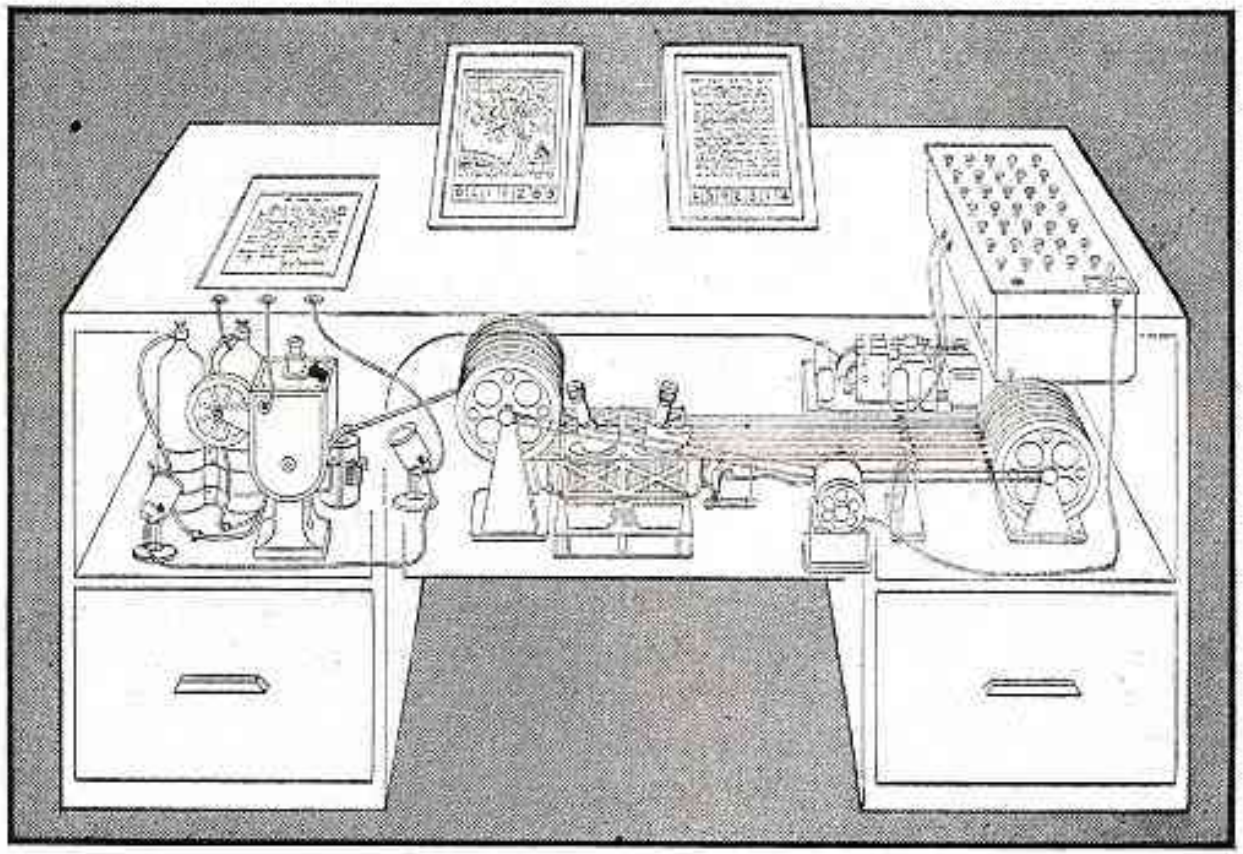
Virtual Machine Hypervisor - 1967

1955 - 1968

HYPERTEXT

Hypertext - 1945

1945 - 1968



MEMEX in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference.

HYPERTEXT WITH CLICKABLE LINKS

Hypertext with clickable links - 1967

1945 - 1968

MARKUP LANGUAGES

Markup Languages

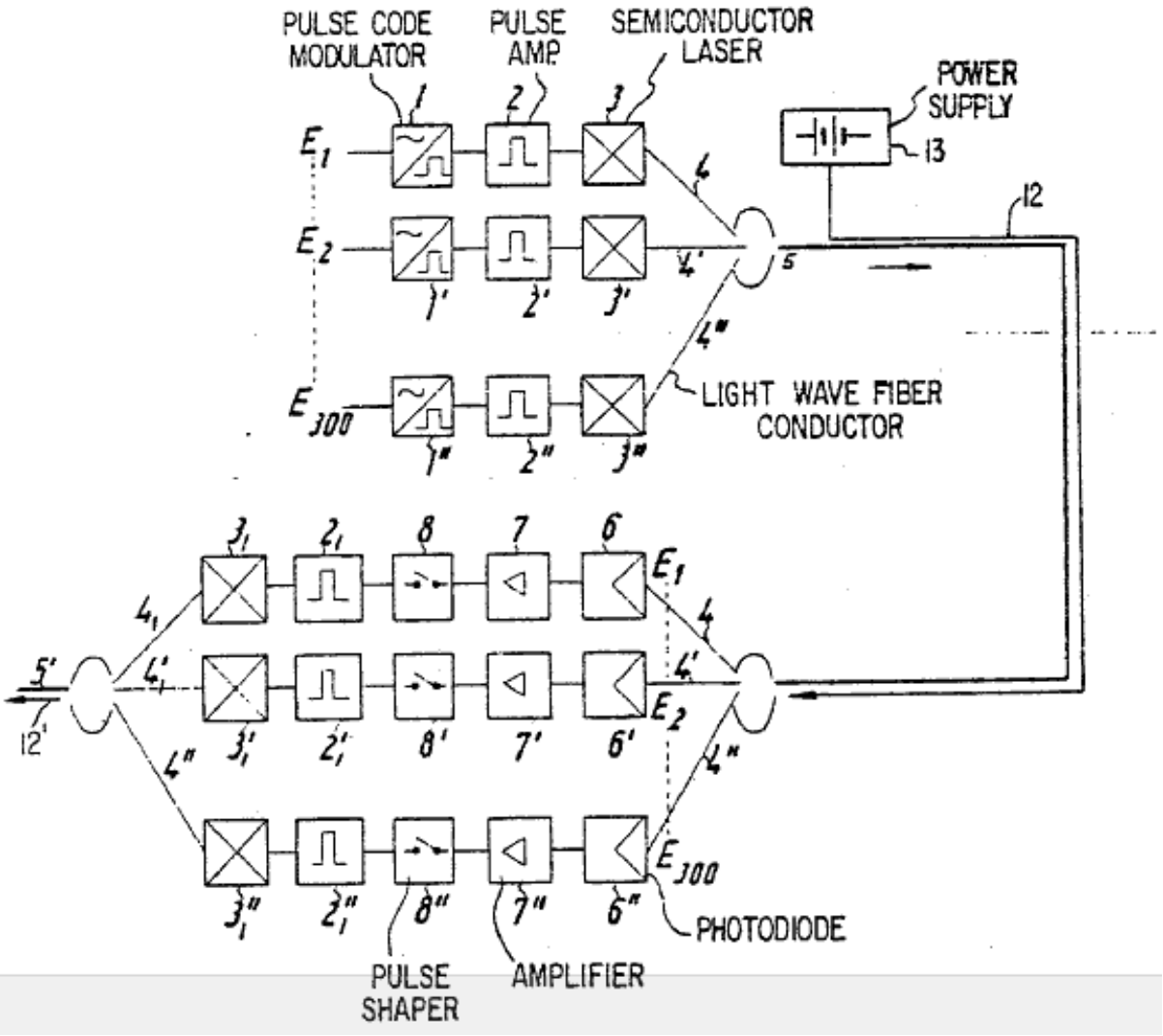
1945 - 1968

- IBM Generalized Markup Language – 1960s

FIBRE OPTIC NETWORKING

Fibre Optic Networking - 1966

1945 - 1968



CLOUDFLARE

WIRELESS NETWORKING

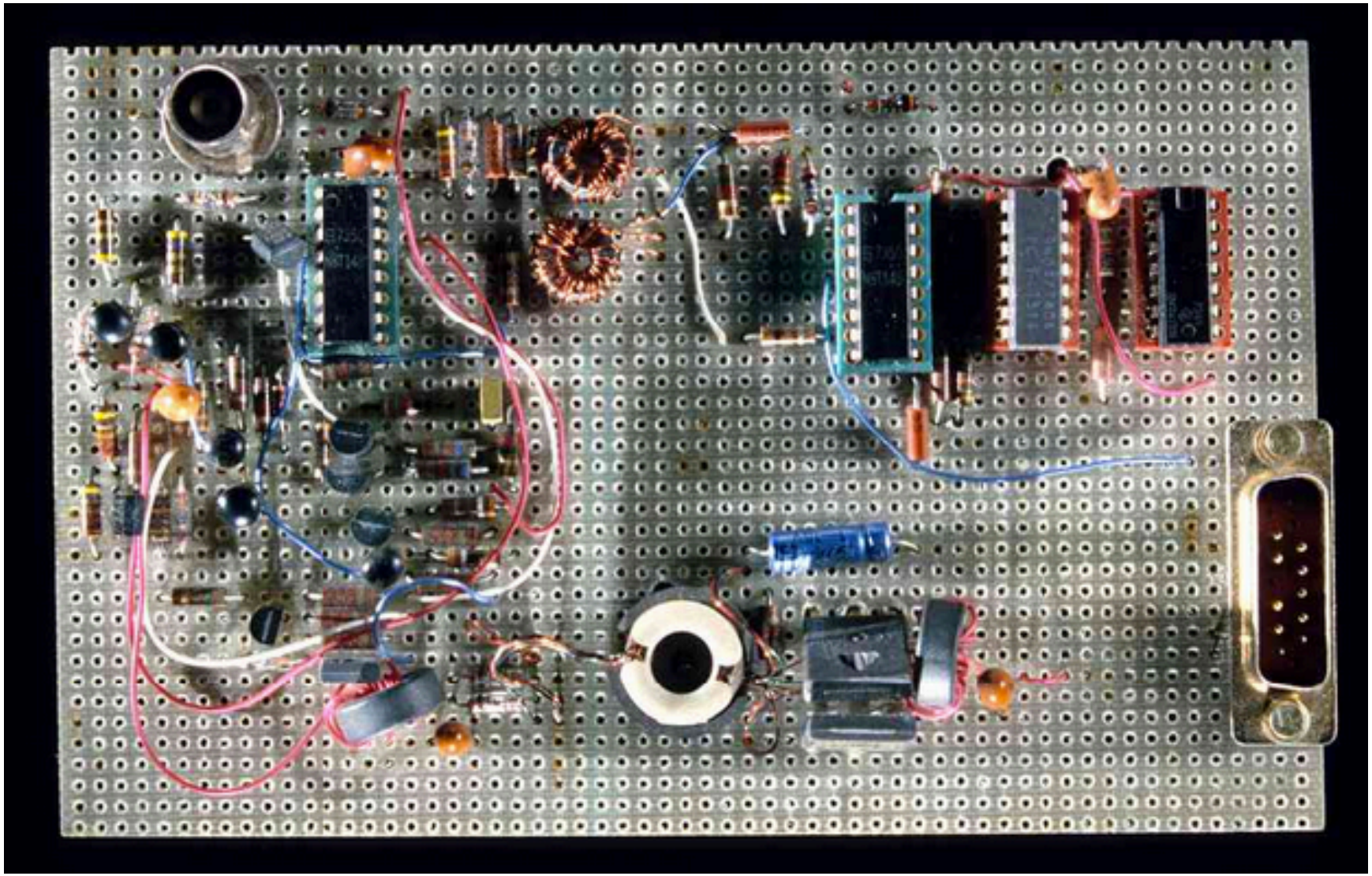
Wireless Networking – 1971

1945 - 1971

ETHERNET

Ethernet - 1973

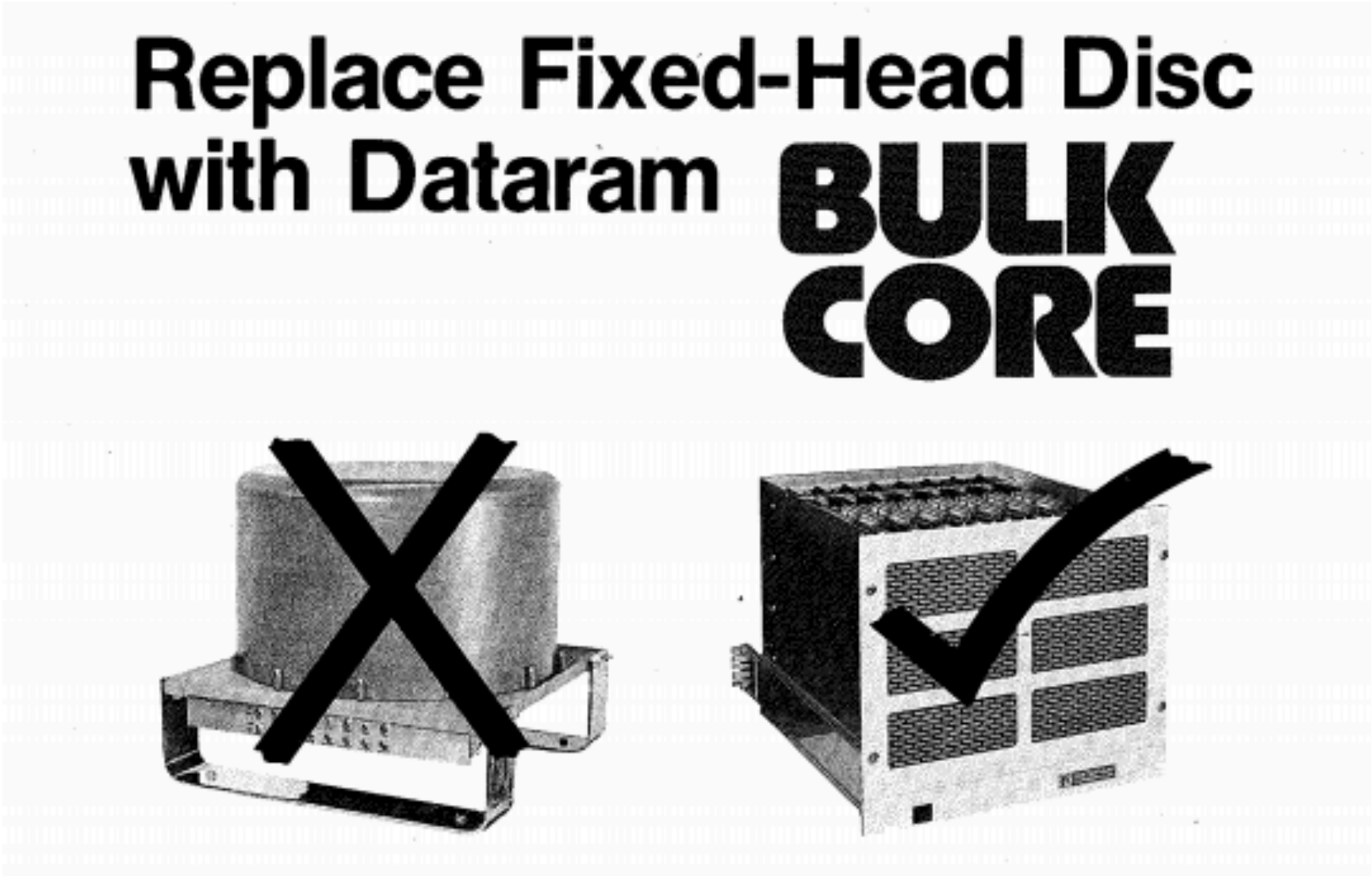
1945 - 1973



SOLID STATE DISKS

SSDs – 4MB in 1976

1945 - 1976



PROCESSOR ARCHITECTURE

Processor Architecture

1945 - 1976

- RISC (1960s Cray; 1980s)
- CISC (1960s IBM mainframes)
- Instruction and Data Caching (1960s; IBM System/360)
- Instruction Pipelining and Prefetch (1961 IBM Stretch; 1979 8086)
- Branch Prediction (1961 IBM Stretch)
- Vector processor (1974 STAR-100)

INTERNETWORKING

Internetworking

- Chat – 1967 (NLS later Unix Talk)
- File Transfer – 1971 (FTP)
- Email – 1971 (host to host)
- Remote Procedure Calls – 1975 (RFC 707)

GRAPHICAL USER INTERFACE

GUI – Xerox 8010 (1981)

1945 - 1981

GUI – Apple Lisa (1983)

1945 - 1983

INTERNET

TCP/IP – 1983

FUNCTIONAL PROGRAMMING

LISP – 1958

APL - 1964

ML – 1973

OBJECT ORIENTED PROGRAMMING

Simula 1 – 1967

C++ - 1979

Smalltalk - 1980

CONCURRENT PROGRAMMING

CSP – 1978

Ada - 1979

Occam – 1983

EVENT DRIVEN PROGRAMMING

PL/1 - 1966

DECLARATIVE LANGUAGES

SQL – 1979

Regular Expressions – 1970s

HIGH WATER MARK IS (ROUGHLY) **1983**

IMPLICATIONS

THE THING YOU ARE DOING HAS BEEN DONE BEFORE

There's great value in computer science education

WE ARE IN THE AGE OF GREAT PRODUCTIVITY

Be thankful all that stuff's been invented already

WE HAVE YET TO CONQUER UNRELIABILITY

Ada Lovelace (1840s)

My Dear Babbage. I am working very hard for you; like the Devil in fact; (which perhaps I am). I think you will be pleased. I have made what appears to be some very important extensions & improvements. . . .

I am delighted with Note D. It is in your usual clear style and requires only one trifling alteration which I will make. This arises out of the circumstance of our not having yet had time to examine the outline of the mechanical part. . . .

Think of my horror . . . at just discovering that the Table & Diagram (over which I have been spending infinite patience & pains) are seriously wrong, in one or two points. I have done them however in a beautiful manner, much improved on our first edition of a Table & Diagram. But unluckily, I have made some errors.

Grace Hopper (1947)

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 795 cor
13"UC (032) MP - MC ~~1.5821442000~~ ~~2.13047641 5 (-3)~~ 4.615925059(-2)
(033) PRO 2 2.13047641 5
correct 2.13067641 5
Relays 6-2 in 033 failed special speed test
In relay " 11.00 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.
1545 
Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1700~~ 1630 Arctangent started.
1700 closed down.

Relay 2145
Relay 3370

Maurice Wilkes (1950s)

- As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. **I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.**

Donald Knuth (1990s)

- Question from audience: “Which programming language do you prefer Java or C++?”
- Knuth: “Which has the better debugger?”

TURING'S CURSE

1936: There is no program that, given a description of an arbitrary computer program, can decide whether the program finishes running or continues to run forever.

WORK ON RELIABILITY

Help programmers make fewer mistakes

Help programmers find their mistakes