

THE LIVES THEY LIVED 2011

The New York Times Magazine

December 25, 2011

THESE AMERICAN LIVES

Ordinary People, Extraordinary Stories

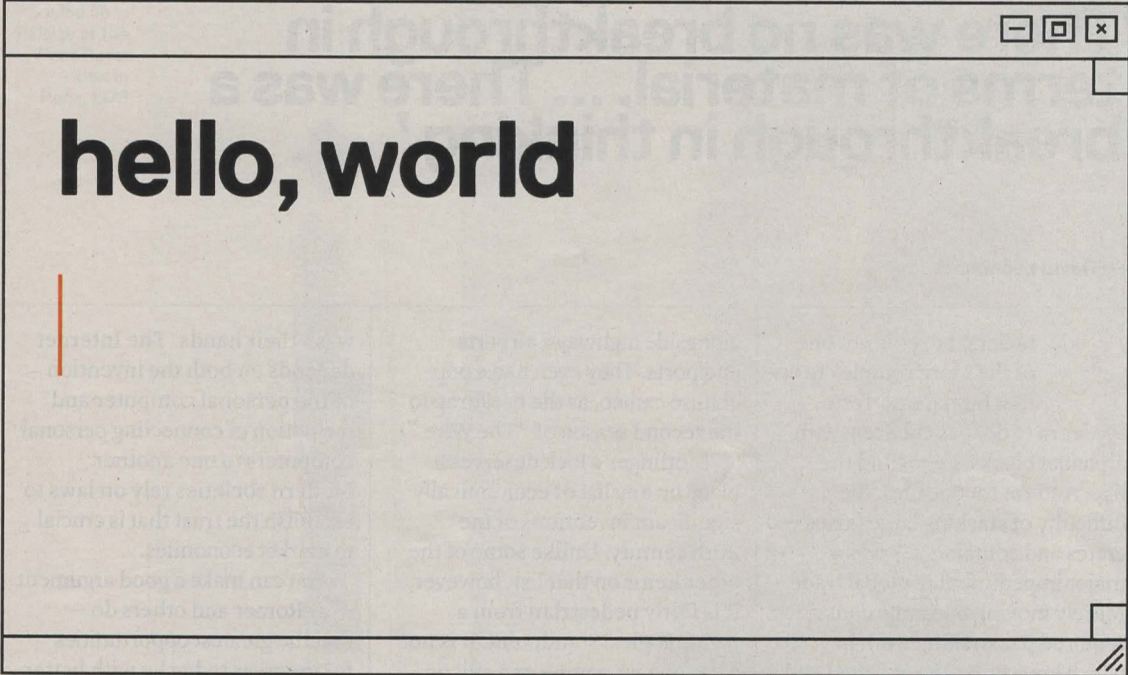
A special section edited by Ira Glass and the staff of "This American Life."

PLUS: Reality-TV Deaths, by **John Jeremiah Sullivan** Painting for Your Life, by **Michael Paterniti**
The Hollywood Know-'Em-All, by **Maureen Dowd** The Engineer of the Global Economy, by **David Leonhardt** The Late-Blooming Poet,
by **Philip Levine** The Tomboy Muse, by **Cathy Horyn** The Never-Ending Last Tango in Paris, by **Susan Dominus** and More.

'A programmer's need to explore, freely and openly, is powerful.'

Dennis Ritchie, b. 1941

By Ellen Ullman



hello, world

Hello, world: those were the words that appeared on the screen once you had programmed and run the iconic first example in the book “The C Programming Language,” which Dennis Ritchie, the creator of C, co-wrote with Brian Kernighan. I remember that slim volume’s revelatory power when I read it — its generous, collegial style, more a talk with presumed equals than a textbook. I still have on my shelf the copy I used, a first edition. The pencil scratches seem to indicate I was figuring out what the hell I was doing.

I was a self-taught programmer, and it was through Ritchie that I

Ellen Ullman is the author of “Close to the Machine” and “The Bug.” Her novel “By Blood” will be released in March 2012.

came to understand the layers of software that worked beneath the screens and printers and keyboards and mice. The newness of C’s conception — and the elegance of it — was that the language was both “high” and “low.” Higher-level languages — like Cobol and Fortran — kept you out of the innards of the machine. “Lower-level” languages — called “assembler” — worked on only specific hardware. Closed environments dominated the computing world of the 1970s and early ’80s. An operating system written for a Hewlett-Packard computer ran only on H.P. computers; I.B.M. controlled its software from chips up to the user interfaces.

But C and the operating system it was deeply intertwined with, Unix, designed by Ken Thompson, were made readily available. Programmers were free to poke

around to see and directly manipulate what was in the computer’s memory. The entire environment presumed you knew what you were doing, or trying to do. It let you fail spectacularly — bring down the system with one command — an annoying but essential part of any great experiment. The C/Unix system invited collaboration across time and space, what today we might call “crowd sourcing,” except that the members of this crowd — researchers in government, professors of computer science, students in universities — were deeply knowledgeable and often brilliant.

Sadly, we are returning to the “owned” past. Apple environments run only on Apple hardware; the Android system only on phones approved by Google. Wireless providers make deals with Apple,

Google and Microsoft, and these corporate contracts determine which specific equipment we can buy. Worse yet, Apple and Google are gatekeepers for the apps we can use. In the wireless world, what seems to be proliferating choice is really taking us into another digital archipelago.

But a programmer’s need to explore, freely and openly, is powerful. That is what I and others like me understood the first time we opened “The C Programming Language” and were magnetically drawn into the world Dennis Ritchie created. We were closer to the machines, yes, but also interconnected. We had the sense of being asked to join a heady conversation in which what could be said was limited by only talent, energy and imagination.

That conversation has now grown vast. We live in the midst of an ever-widening circle of people who understand technology and are incorporating it into their lives in an almost-interstitial way. From the “greats” who were Ritchie’s contemporaries to those contributing today: software engineers developing “open-source” systems; programmers writing apps for tiny start-ups; designers of personal Web sites; the millions “programming” their Facebook pages.

I believe they are experiencing the delight I felt when Dennis Ritchie let me peer into the deeper recesses of computing, the excitement of the great collaboration he and Ken Thompson began four decades ago.

In a sense, Ritchie has enabled us to all become programmers. And this alone should give us the power to create our own digital future.

Hello, new world. ♦