

Adam's Technology

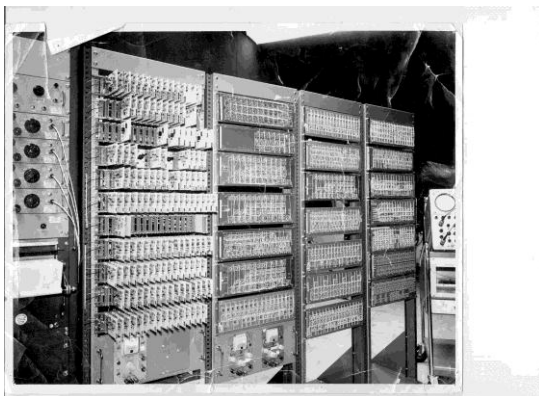
I am a computer geek. Well, I was once. Now I just use computers to research arguments between me and my wife, write stuff like this article and store the pictures I take sporadically with my adorable little digital camera.

Speaking of the word “digital,” my archeological friend, Fred, who has discovered several ancient scrolls similar to the Dead Sea Scroll, has shared with me some interesting and amazing facts that were left out or expunged by the censors of the Bible over time. For example, I was amazed to find out that the very first things Adam saw immediately after he was created by God, rolled up in fetal position, were his *cojones*, not his ten fingers. So he began counting in binary and according to Fred's scroll, all mankind counted in binary for thousands of years until the Sumerians noticed their fingers and complicated things. I suppose Adam could be called the legitimate father of the modern computer, but it took around six thousand years for man to invent electronic devices that mimicked his initial discovery. These devices count by twos using “1”s and “0”s corresponding to a positive electronic pulse or the absence of it.

Incidentally, the Mayans counted in binary, so Fred thinks it is entirely possible that the Garden of Eden was in Guatemala or here in Mexico, not Israel or Turkey as was supposed by tradition.

I am inclined to believe Fred's findings. After all, the entire universe that God created is analog. Only Man invented digital things like computers and cameras.

My first computing device was a slide rule. It worked well, computed trig functions and logarithms to about two decimal places. I name my calculating devices, so I called my slide rule “Elmer.”



My first computer, one I designed and built myself, used transistors instead of big, hot, clumsy vacuum tubes the “mainframes” used to do its “1”s and “0”s. “Earl,” as I called it, shown here, was a marvel of efficiency for its time but it was just about as capable as my hand calculator

today. My current desktop computer, “Melvin,” is more powerful than the combined computing capabilities of the entire planet fifty years ago.

Originally, programmers made up the vocabulary for this new thing not created by God. A mistake in a program was called a “bug,” needing the programming equivalent of DDT to fix it. A binary “1” or “0” was named a “bit”. A “byte” was eight of them, deliberately spelled differently to distinguish them from sandwich nibbles. Bytes are still used to identify the amount of information a computer can store, around 20 billion in Melvin, the equivalent of around half a million copies of Tolstoy’s “War and Peace.”

But then businessmen took control of naming things, ignoring the clever culinary allegories. How unimaginative is a “word” full of bytes? Words are blah-blah by salesmen. Then teachers probably named the next largest unit, the “instruction.” Many instructions make up a “program”, interesting only to convention organizers. If there are enough instructions, they comprise an “application.” Proctologists probably came up with that name. Who else would care about an “application” other than the Preparation H Corporation?

If I had been one of the original old timers, I would have called a “word” a “filling,” an instruction a “sandwich,” a program a “meal,” and an application a “feast.” So a bit of a byte of a sandwich for a birthday picnic would be how we would describe what a computer does. Makes much more sense, but nobody asked me.

Old time programmers always seemed to have eccentric senses of humor. One day I took some punched cards representing one of Earl’s programs, to the giant IBM computer room to be tabulated. The computer operator took one look at me, a newcomer, saw a live one and put my program into the card reader. I waited next to the printer as the IBM monster began cranking out the computations I had asked it to do. Fully expecting to wait about 20 minutes for the calculations to finish and then a short printout of some results, the huge, mechanical printer suddenly began cranking and clunking away. “Must be a bug in my program,” I thought, mentally calculating what level of DDT I would need. Alarmed, I followed the large piece of printout paper slowly coming out of the printer. A picture of a giant hand with an extended finger gradually appeared. The damn thing was giving me the bird! A rigid digit program had been surreptitiously slipped in my deck of programming cards. I looked around. The operator and three programmers were weeping with laughter.

So having read this article, you are now a certified computer geek. Sing after me: “The bit bones connected to the byte bones, the byte bones connected to

the word bones, the word bone connected to the instruction bones, the instruction bone connected to the program bone, the program bones connected to the application bones....now hear the word of the Lord.” And so on, everything in perfect harmony with the binary findings of Adam’s *cojone* technology.