Bob Bemer

Autor:
Data de publicació: 25-06-2004

Died 2004-06-22.. He worked for Lockheed Aircraft, Marquardt Aircraft, Lockheed Missiles and Space, IBM Corp. and the Univac division of Sperry Rand. He went to Paris in 1965 for a year to work for Bull General Electric, then back to the United States with GE and Honeywell Information Systems until his 1982 retirement. He promptly started his own software company, which he eventually sold to BigiSoft Inc., retaining the title of chief scientist.

Computer Pioneer Bob Bemer, 84

By Patricia Sullivan
Washington Post Staff Writer
Friday, June 25, 2004; Page B06

Robert W. "Bob" Bemer, 84, who helped invent the language used by most of the world's computers to translate text to numbers and who was the first scientist to warn of the Y2K problem, died of cancer June 22 at his home on Possum Kingdom Lake in Texas.

Without the invention of the computer code ASCII, there would be no e-mail, no World Wide Web, no laser printers and no video games. Mr. Bemer, known as "the father of ASCII," created the code in 1961 by assigning standard numeric values to letters, numbers, punctuation marks and other characters.

"We had over 60 different ways to represent characters in computers," Mr. Bemer told Computerworld magazine in 1999, describing the time before the American Standard Code for Information Interchange was created. "It was a real Tower of Babel."

He was well known in the computer industry (The Washington Post in 1999 said, "In the weenieworld of data processing, he is a minor deity"), but he broke into wider public consciousness when government and businesses began to panic about the "millennium bug" that threatened to shut down the computer systems on which society had grown so dependent.

Mr. Bemer had first published a warning in 1971 about the problems that would arise from using two digits instead of four to represent years in computer code. Unlike some of the doomsayers who came after him, he knew what he was talking about: He was involved in the original effort to create government standards for the computer industry.

Having learned from work he had done in the 1950s on genealogical records for the Church of Jesus Christ of Latter-day Saints, he realized that truncating a year's date was a penny-wise and pound-foolish solution to the cost of saving computer space. But Pentagon bureaucrats, among the largest computer users on Earth, refused to accept that 1999 was a better code than 99. The National Bureau of Standards went along, although it said programmers could voluntarily use four instead of two numbers.

What that decision led to was the fear, as 1999 turned to 2000, that data stored deep in computer code would misinterpret 00 not as the year 2000 but as 1900, or even 1000. Data might scramble, possibly causing nuclear reactors to go haywire, credit card transactions to vanish and automated processes that govern prison door locks, airline operations and giant dam gates to refuse to turn on or shut down.

"It was the fault of everybody, just everybody," Bemer told Time magazine and many others. "If [Adm.] Grace Hopper

[the founder of COBOL] and I were at fault, it was for making the language so easy that anybody could get in on the act."

Mr. Bemer kept up the alarm, even trying to get President Richard M. Nixon to declare 1970 "the year of the computer" in order to highlight the problem. He wrote about the problem for the technically literate in the Honeywell Computer Journal in 1971 and for the public in Interface Age in 1979. The response was derision -- when anyone bothered to respond. He continued the warnings until he retired in 1982. No one listened until it was almost too late.

Although the public may have thought the millennium warnings were overblown, an estimated $122 billion was spent in the United States alone to fix the Y2K problem, according to IDC, a technology and telecommunications research firm.

"I think he took a lot of pride in the fact that there wasn't a huge problem, and maybe people like him who were sounding the alarm were getting the companies to do what they needed to do," said his stepson, Glen Peeler.

Throughout his career, Mr. Bemer had a knack for being in the right place at the right time. He helped Hopper create the computer language that he named COBOL, or Common Business Oriented Language. He helped create the standard measurement of eight bits per byte. Computer users have Mr. Bemer to thank for the backslash character and for the escape sequence, which allows a computer to break from one language and enter another.

"I used to say that I never got a nickel for the escape sequence. A nice receptionist at the Dallas InfoMart did give me five pennies, but I spent them," he said on his Web site.

Mr. Bemer did not cash in on the financial bonanza of the computer revolution, his family said. His cars bore the vanity license tags ASCII and COBOL. He lived in a cliff-top house two hours west of Dallas on a reservoir, which he told visitors would have been handy in case he needed to "drain the lake" for drinking water. He collected Pogo Possum comic books and made lists of every airplane flight he'd ever taken, every country he'd visited and every trip to see his parents.

He was born in Sault Ste. Marie, Mich. He received a bachelor's degree in mathematics at Albion College in 1940 and a certificate of aeronautical engineering at the Curtiss-Wright Technical Institute in Santa Monica, Calif., the next year. He spent time as a machinist, furniture-maker and movie-set designer before he was hired as a programmer at RAND Corp. in 1951.

He worked for Lockheed Aircraft, Marquardt Aircraft, Lockheed Missiles and Space, IBM Corp. and the Univac division of Sperry Rand. He went to Paris in 1965 for a year to work for Bull General Electric, then back to the United States with GE and Honeywell Information Systems until his 1982 retirement. He promptly started his own software company, which he eventually sold to BigiSoft Inc., retaining the title of chief scientist.

"He was a hardworking man," Peeler said. "He was a relic, a throwback, old-school. He was always on a computer doing things."

In 2003, the Institute of Electrical and Electronic Engineers' Computer Society awarded him its Computer Pioneer medal.

He was married six times to five women and had five children by his first wife and a sixth by another. He also had two stepchildren, nine grandchildren and five great-grandchildren.

That Troublesome "Father of" (link)
Computer History Vignettes (link)

by Bob Bemer

I have a partial history of ASCII submitted to the Annals of the History of Computing. To a person, five reviewers have choked on my explanation for the sobriquet "Father of ASCII" as applied to me. Self-aggrandizement, they say. In other words, "pushy". I very much wanted the article published, so I took that part out before resubmission. You see it here, pretty much in the original verbiage.

The reviewers are wrong about my motives! Use of the term may confer credit. It may also confer opprobrium, misunderstanding, and nefarious motives, in the Internet world of slippery truth and outright lies. And it is graffiti on the ivory tower of the Unicode devotees.

So I did not want the current interpretation to stand, for my problem is that such a sobriquet implies too much, and very often the term is used inaccurately. Or is taken to imply my responsibility for things I don't want to be held or thought responsible for. So I'm tucking the story in with my other vignettes on my site, where I have some authoritarian control.

A Google search of the Web for "Father of ASCII" currently yields some 100 hits, all referring to this author. An editor of Computerworld once suggested that the term was self-assigned. And many places that the phrase is used are in error. So the problem really needs correction to the world.

I did not create the phrase "Father of ASCII". It came to me in the late 1960s as the parcel address for such a person at General Electric in Phoenix, where I was then working. Someone with knowledge forwarded it to me, and inside was a letter starting "Dear Bob". The sobriquet was then spread widely by Interface Age Magazine [1]. I accept it for being a major force in the creation and adoption of ASCII, as told in [2].

Please recognize first that the word "ASCII" is now generic, even though I tried (unsuccessfully) for a long time to have it called the "ISO Code". And we must recognize four large segments or aspects of the ASCII world:

ASCII Art
ASCII Art Offspring -- Emoticons
ASCII The Coded Character Set
ASCII The Base Platform for All Coded Character Sets, and the Web

So is "Father of ASCII" correct for all of these?

Well, yes and no.

ASCII Art

This one was surprising. There are thousands of websites devoted to making pictures of a sort with the graphic symbols of ASCII. Of course with a system of dot combinations, formerly called "rotogravure", they'd be even happier. That was sort of done once, in a registered ASCII variant called "Teletext".

But any parenthood could apply only to the set of graphic symbols. In [3] it is shown that I contributed, via accepted proposals, the characters { } [ ] ESC to ASCII, and that none of these had ever existed in the internal character set of any computer before STRETCH (with the possible exception of square brackets in the character set of the LGP-30).

US (Unit Separator), RS (Record Separator), GS (Group Separator), and FS (File Separator) are 4 other characters that I put in very early, but without those names. And they have no graphic shape, anyway. The backslash seems to be the character that would be most missed.

Parenthood of ASCII Art is tenuous, at least. A DNA test won't point at me.

ASCII Art Offspring -- Emoticons

The edition of the Oxford English Dictionary unveiled in the Fall of 2002 had "emoticon" among its list of new words. Apparently the addition was overdue, for 138,000 Web pages then contained that word!

The OED definition is not at hand, but apparently an "emotion-expressing-icon" is much used to compact one's e-mail messages, so much so that the users have evolved their own standard emoticons, as (again) constructed from the ASCII characters. E.g.:

~/ means "my glass is full".

(::()::) is a proffered bandaid, or offer of support.
:-{} is "blowing a kiss".

Any fatherhood by this author is totally denied!

ASCII The Coded Character Set

Most evidence, and the article submitted to the Annals, shows that I had substantial influence in putting the official standards efforts into operation, in the U.S. and worldwide. But I exerted little fatherly influence, leaving the selections and code placement to the international bodies who did that as their primary reason for existence. Dr. J.A.N. Lee said, in his bio for me in [4] that: "... He was not the inventor of ASCII ..."

True. ASCII was not "inventable". There was no way to achieve a standard code of the present universality without leaving it to a vast group of manufacturers and workers. Lee explained as much, adding that:

Bemer's forceful articulation and demonstration of the simplicity of the "escape sequence" solution achieved acceptance of a US standard that rendered as nonstandard all existing computer designs, software systems, and telecommunications hardware, including the whole repertoire of IBM equipment that used EBCDIC ..."
I once attempted to explain the "father" usage in a letter to BYTE Magazine [6], in response to a man who claimed [5] that ASCII was obsolete:

But it is not correct to say that I created ASCII. To say this insults many people who dedicated years of their lives working out compromises between dozens of existing codes for the resultant single code. I have never demurred at being called the "father of ASCII", as it is the "mother" who creates.
But a father is usually needed to get things started, like my maneuvering of the development work to be done on an international basis, ensuring the cooperation and support of all countries.

The following squib, from Honeywell's internal newspaper, circa 1979, tells of my efforts to spawn the ASCII project:

At Reata Pass Steakhouse

ASCII -- The Base for All Character Sets, and the Web

Now the real fathering, of the mechanisms that enable the Worldwide Web.

Upon initial interest in standard encoding methods, I saw the problem not as creating a single standard code, but as having a basic code and system from which the multitude of graphic symbols in the world could be encoded in groupings reached by temporary departure from the base system. I envisioned alternate coding tables, invoked dynamically as identified by

"an identifier self-identifiable as an identifier"
Thus the genesis of the "escape" character. Like "Now hear this" on a Navy ship, followed by a message to be obeyed, "escape" fits the "orderly provision for expansion and alternatives" clause of my original program-of-work statement.

References:

R.W. Bemer, "Inside ASCII - Part I",
Interface Age Magazine 3, No. 5, 96-102, 1978 May.

ASCII.HTM (current 2003 Dec 01)

R.W. Bemer, "The Great Curly Brace Trace Chase", BRACES.HTM (current 2003 Dec 01)

J.A.N. Lee, "Computer Pioneers", book, IEEE CS Press, 840 pp., 1995.

R. Collins, "Time to replace ASCII?", Letter to editor, BYTE Magazine, 420, 1990 Jan.

R.W. Bemer, Letter to editor, BYTE Magazine, 36, 1990 Jun.

BEMER MEETS EUROPE
(Computer Standards)
Computer History Vignettes

By Bob Bemer

My first trip to Europe occurred after I wrote a lengthy letter to Dr. Charlie De Carlo in justification of a trip to the ICIP Conference in Paris in 1959. John Backus was giving a paper on his BNF notation, and I argued that I had several technical fish to fry, in addition to giving a paper at the conference of the British Computer Society the following week.

My wife Marion also worked at IBM, as secretary for the Director of Customer Engineering. Somehow I arranged for her to come with me. Like CIA actions, the details are best left black. We left on 1959 June 12, to return on the 30th.

First we took a Lockheed 1049 Constellation, a propeller plane, to Madrid via Lisbon. Each pair of seats had sort of an opaque shower curtain one could draw about the area for privacy, especially while sleeping, for it was an 11-hour flight to land. A few seats behind us were Dr. Herb Grosch and his secretary. At Madrid, it was the summer made famous by Ernest Hemingway, about the memorable duel of fame between Dominguez and Ordoñez (brothers-in-law, or some such). We saw Ordoñez at Madrid's lesser bullring Vista Allegre.

Then to Paris (Orly Airport in those days) for the ICIP Conference. We felt important because all of us delegates had semi-official government status. We stayed at the Claridge Hotel. Dr. John Carr III, in his own memoirs, has recounted how I, astonished by the openness of the prostitutes outside the hotel, attempted to take pictures. Those handbags were heavy and well-wielded! I had to be rescued by the hotel staff.

The conference itself needs to be recounted in its own right. But when it was over we went to Le Toquet, where there was a channel-hopping plane that could carry automobiles. Imagine our surprise when a red Mercedes-Benz convertible came up for loading -- driven by Herb Grosch! Yes, he of "Grosch's Law" about the ratio of computer cost to performance.

From the tiny landing field in Britain, Herb drove us to London. The secretary was missing, so he had room. Then on to Cambridge University, where I was to give my talk. Our hotel was quite nice by English standards -- real tablecloths and all, with the waiter in formal clothes. Unfortunately Cambridge must have not had a cleaning establishment that the waiter could patronize.

We did get a chance to go punting on the river Cam. All the while I kept thinking of all of those limericks.

At the big formal dinner in Kings College, Prof. Eiichi Goto was on my left. Across our table were my wife and Prof. Sandy Douglas. With the dessert, both of them tried to shoot cherry pits at me with a spoon. They always missed, and always hit poor Prof. Goto, who bowed each time and said "So sorry"! Sandy got to like my wife, and many years later he and his son visited us in Phoenix. I gave young Malcolm a dead scorpion from our pool. He took it back to England in a jar, where it came back to vicious life!

After ICIP

Mr. Holland-Martin of the British firm ICT (International Computers and Tabulators) had apparently been in correspondence with John McPherson and Jim Birkenstock of IBM, and had noted the success of their plan to convert the Office Equipment Manufacturers Institute into the Business Equipment Manufacturers Association, with a charter in computer standards work, and sponsor of the ASA X3 committee. McPherson and Birkenstock had me draft a scope and program of work for the new committee, and it was adopted. Character codes and vocabulary were two of the six specific areas covered.

Oddly, Birkenstock's 47-page memoirs (in the 2000 January-March issue of the Annals of the History of Computing) made no mention of these actions. I think it was a lot more earthshaking than many of the things he recounted.

Holland-Martin wished to do much the same in England, and this expanded to the formation of ECMA, the European Computer Manufacturers Association. I will always believe that ECMA advanced computer standards much better than the USA's X3 did, probably because one Dara Hekimi, a Swiss, was named Secretary General. Plus Jean Besse, well known to world-class bridge players, as his righthand man. To win approval, Holland-Martin needed proof of the fast pace of similar work in the U.S. As I was familiar with the program we hoped would be adopted for X3, and had an active role in the COBOL work, McPherson sent me to give a status report to Holland-Martin's group.

This was only two weeks later, on Jul 14. I was so busy that all I had time to pack was a TWA flight bag. The UK immigration people of that time were suspicious of someone making a transatlantic trip with such a minimum of luggage, and questioned me for some time. But I only stayed two days. The trip served a dual purpose, as I also reported on COBOL progress to a group from the British Computer Society.

Fallout from the European Trips

That fall a John Gosden showed up at IBM. A side excursion was to see me, Frank Williams, and Howard Smith -- the subset of my group that was working on standard and extended character sets, prior to activation of ANSI X3. I recall that he got quite excited about what he saw, especially the 8-bit set, which was quite a novelty at that time.

He took this information back to the British Standards Institution, which had a working group on this topic. I was invited to show our work to them the next time I came to London, which was on 1960 Feb 23-27.

I don't now remember why I had to go through Rome and Paris to get to London! Again I juggled the travel accounts, as it was over a weekend. Instead of returning home I had my wife Marion come to London Thursday night. We packed the weekend with shows and such, and she returned home Sunday afternoon in time to be back at work at IBM Monday. A friend, telling her about their trip to Jamaica, chastised her for not really paying attention, asking scornfully "What did you do over the weekend?" The answer "Oh, I went to London" floored her, in those days.

One of the British members to whom I reported the character set work was Hugh McGregor Ross, by all means the main sparkplug of their studies, and very active even today, when he participates in the UNICODE work with his firm (Universe of Characters?). We hit it off very well (so much so that the code that was to become ASCII was first called the Bemer-Ross Code in Europe).

I must have told Gosden of some of my other interests, for after he was hired by Isaac Auerbach to work in the United States, Ike gave me a call and invited me to a luncheon with Gosden to discuss his new project for good practice, eventually the "Auerbach Standard EDP Reports". My air travel history gives no clue to when, for it was either in New York or Philadelphia, and those were not usually airplane trips for me.

IFIP Vocabulary

I do remember that the meeting ran on much longer than expected. Perhaps I gave some good advice, for Ike seemed

pleased. I think we must have agreed that a standard vocabulary would be a good thing, for both Gosden and I had been doing work on this, quite apart from what Grace Hopper was doing for the ACM. The problem with her committee was that they did not take any non-US usage into account. Working for the international company IBM, I found this unsatisfactory and provincial.

So did Ike Auerbach. When he founded the IFIP effort, with its first Council meeting in Rome in 1960 June, he saw it as an ideal place to rectify that. He thought correctly that to make a standard, one must first have an input as a basis. He thought that about most aspects of the work that ISO/TC97 was to do. I believe this was an important factor in his virtually singlehanded creation of IFIP, with rather corresponding committees. I was surprised to find that he had unilaterally named me the U.S. representative on the IFIP Vocabulary Committee. Perhaps Gosden's opinion counted. Perhaps he just asked around, found no objection to me, and asked if I would and could accept. Then it was "You are it!"

He may have made a good choice, because I put in a vast amount of effort on it. In correspondence with the other members, between physical meetings, I made meticulous and extensive annotations on the various definitions. I also sent all of this correspondence to John Gosden, to help him and to ensure as much feedback and support as possible.

The IFIP Vocabulary Committee was chaired by Geoffrey Tootill of the UK. We met in 1961 October (Copenhagen), 1962 March (Feldafing), 1963 January (London), 1963 April (Rome), and 1963 September (Oslo). Often these were in conjunction with IFIP Council meetings; we had to make reports.

I remember the London meeting because that was where I picked up my 1963 Sunbeam Alpine, which I still have in restored condition, bearing the Arizona license plate "ESQ SEQ". At that time the Customs rules were that one could take delivery in Europe, drive around a bit, and import the car as "used" for a lower duty. I had a terrible time with that scheme. I was able to drive it only 35 miles because (as a Dick Francis book later enlightened me) it was the worst UK weather in 300 years! Customs was kind, and okayed a "used" condition.

I remember the Rome meeting because we actually worked in EUR, the city that Mussolini created to be the capital of Europe. Very run down, and the exposed electrical wiring was stapled or taped to the marble! I believe much of the movie "La Dolce Vita" was filmed there.

I remember the Oslo meeting in the first place from finding Vice President Johnson shopping at the Christiana Glasmagasin. My wife tried to strike up a conversation with the plainclothes guards, who didn't want that role to be visible. When they demurred she said "Oh, you guys are so obvious!"

Secondly because I was surprised to meet Eric Clamons of UNIVAC on the street; we have had long and fruitful collaboration in the coded character set business. It was Eric that, when I went to Univac, made the Univac 1050 the first true ASCII-based machine. Thirdly I remember it because during a lull I proposed playing "5-in-a-row", or Go-Mo-Ku (a simple derivative of the GO game), with Swedish member Olle Karlquist. For the first five games I was careless and lost all five. The next five I really paid attention, and still lost all five! I mentioned this anomaly to him, whereupon he told me he was the unofficial Go-Mo-Ku champion of Sweden!

The structure of the vocabulary we created was radical and new. Instead of taking a word(s) and writing a definition, we first took the concept, wrote the definition, and ONLY THEN assigned the term in each language considered. And we clustered by concepts, not name.

Among my definitions that I am proudest of are those for "data" as opposed to "information". And I made a great chart for logic operators. It's still in use.

Of the working members, the really active people were:

Geoffrey Tootill, Chmn - UK
Jim Wilde, secretary - UK
Robert Bemer - USA
Robert Mantz - Netherlands (Bull)
Paolo Ercoli - Italy
Olle Karlquist - Sweden
Rolf Basten - Germany

Mme. P. Fevrier - France
B. Vauquois - France
Pat Hume - Canada
Jean Besse - ICC and ECMA

I hired Jim Wilde for Univac because of this work. Flew him over for an interview, agreed, and flew him back again to start work.

Formation of ISO TC97, Computer Standards

This event occurred in Geneva in May of 1961. I chose to tell the story in a separate place. See it there.

The Great Curly Brace Trace Chase

Computer History Vignettes

By Bob Bemer

On my site is a pointer to an interesting page by Jukka Korpela of Finland [6], called "Character histories: Notes on some ASCII code positions". It tells how various interesting characters (then new to many computer character sets of the time) came into membership in the ASCII set.

Such research is difficult now, in an 8-bit byte, ASCII code, and soft-copy screen world, where typewriters, codes of different length, and computer word lengths have been pretty much forgotten. And their documentation is mostly in hardcopy libraries; not on the Web.

This vignette came about because I started to wonder:

If the curly braces exist in ASCII because of my efforts and examples, and/or
If I had been first to put curly braces, via IBM's Stretch, into the internal character set of any computer.

Here the only words subject to misinterpretation are "internal character set".
Webopedia helps us out by giving this definition:

"character set" -- "a defined list of characters recognized by the computer hardware and software".

That is not "hardware and/or software", but "hardware and software". Both. And the character set must have the same content, and size in bits, for both. For (1) above, remember that ASCII is (ASCI)Interchange.

These definitions controlled my search. The character set or repertoire of computer hardware is not at all necessarily the

same as that of input-output equipment that can be used. Conversations on the Web are bloated with instances of combinations of standard characters that are used to indicate a character, of the character set of the computer used, that is not enterable by keyboard. Typical are those of the C language, where "e" means the single escape character. Granted, that character itself is recognized by the hardware, but hordes of programmers have interrupted the scan code it generates on the keyboard to provide functions not those of the character itself.

Ah. Scan codes. Remember that keys on today's keyboard do not emit ASCII codes; they emit "scan codes", which are converted to ASCII according to the character layout of your keyboard. That is how the French maintain the "azerty", whereas the US has "qwerty" left to right, upper left, on its keyboards. Ditto for Cyrillic keyboards and such.

And seldom do adjacent tracks on magnetic tape match the adjacency of the bits of a character as stored internally as the code of the CPU itself. But there must be a universal 1-to-1 mapping.

Among other aberrations to consider is the possibility of encoding typewriter-like devices so that, although both upper and lower case alphabetic letters would print differently on paper, they would be encoded identically, and thus pass into the computer CPU as the same character.

Stretch

IBM's 7030 (Stretch) was thought to be the first production computer to have a byte size greater than 6 (in fact 8 bits), allowing lower case alphabet and other useful characters to be accommodated. The timing was:

1956 Sep 16-22 -- Large group of IBM people to Los Alamos (including Bemer from NYC Headquarters) (Note 1).
1956 Dec -- Dunwell paper "Design Objectives for the IBM Stretch Computer", EJCC paper
1959 Jan -- First two 7030s began assembly in Poughkeepsie (Note 2).
1960 Jan -- First publication of character set to outsiders. [2]
1961 Apr 16 -- First Stretch delivered (to Los Alamos)

Note 1. The travelog on my website, on the Lockheed-IBM page, shows that as my only visit to Los Alamos during that period, annotated "Stretch Planning". Be assured that the only bit of Stretch planning that I did was the character set. And perhaps advise a bit on software. The 8-bit byte decision had already been made when I got there, and I did say a strong Amen to that!

Note 2. For this to happen, especially for the most powerful machine built to that time, and certainly provided with FORTRAN, the character set would have to have been completely fixed at least 8 months prior to starting to build the CPU. I'm willing to estimate that increment to be cut to 4 months for any other computer, small, with an internal code containing any of [ ] { } . (Stretch had no escape character --
I didn't think of it until 1959 October).

Eric Fischer concluded from somewhere that the Stretch set dated from 1959 November, while my formula (above) gives 1958 April. We'll see if any competitors approach this date. The official character set publication [2], has its initial reference as [1], of 1959 September.

My investigation seemed like a good story, so follow along. First let's look at the ASCII characters that are known unequivocally to be due to me:

ESCAPE

See [3], published in 1960 Feb, but made known to workers in the coding field by 1959 Oct.

Four Information Separators

US (Unit Separator), RS (Record Separator), GS (Group Separator), and FS (File Separator) are the only 4 remaining of

the 8 information separators (ISi) (or data delimiters) that I put in in 1961 September. As I said in my Interface Age articles on ASCII, I got the idea from the Word Mark in the character-based IBM 1401.

Backslash

We may take it as absolute proof of genesis, from the early limited character sets, that IBM's STRETCH was the first computer to use the backslash character, for Reference [4] was actually published a couple of years after the design of that machine. And that uniqueness continued. [4] shows no 6-bit set with the backslash as a working character.

Here is an excerpt from both [15] and [5] (the latter is the source for Korpela's paper):

"I had called a joint meeting of IBM, SHARE, and GUIDE, to regularize the IBM 6-bit set to become the standard BCD Interchange Code ... Frequency studies of symbol occurrence had been prepared, particularly from ALGOL programs. The meeting of 1961 July 6 produced general agreement on a basic 60-64-character set, which included the two square brackets and the reverse slant, which was chosen in conjunction with "/" to yield 2-character representations for the AND and OR of early ALGOL. This is reflected in the set I proposed to ANSI X3.2 on 1961 September 18."
(Note: I had put the backslash in position 5/15. It enabled the ALGOL "and" to be "/\" and the "or" to be "/".)

SHARE and GUIDE representatives at the meeting were a little stubborn about accepting my proposed backslash, so I asked for a character more important to have. After much discussion they could not agree on a better candidate.

"At the 61 November 8-10 meeting, X3.2 constructed the first formal proposal, X3.21 ..." (which, much modified, was to become ASCII)
(Note: In this proposal the backslash was moved to position 5/12, and there it has remained ever since.)

Square Brackets

Having documented the introduction of square brackets to ASCII, careful research must be done to check for previous existence in any computer set. Here the backslash played a very discriminant role.

[7], [8] and [9] have very good information, but all have a basic flaw. The specific computers for which the coded sets are given are not identified to more than the manufacturer (in some cases not even that), nor are the years of their introduction.

Their author, Dik Winter, admits this and says he will try harder, but much documentation is lost, and it was characteristic of the times that nobody seemed to think character sets a very important feature of computers.

In [7], CD display, General Electric Internal, NCR, and Bull Scientific Internal all show both the brackets and the backslash, which would be a remarkable coincidence if they existed prior to ASCII. CDC Display Scientific shows brackets only, but Winter now says it could only be the 3600, introduced 1963 (too late). BCL Internal shows brackets only, but I am unable to find such a manufacturer in my list of nearly 3000 different computers to date.

In [8], [and] were shown for the RPC 4000 (too late at 1960 Nov), and a "MC" Flexowriter (I have to assume that Winter, of the Netherlands, speaking of ALGOL 60, means the computers built by the Mathematical Centre (1962 and later), or the Electrologica X-1 of 1960.

[9] shows two mag tape codes with [], but they're copycats. One is EBCDIC.

The LGP-30 and Square Brackets

Both [4] and [9] showed the LGP-30 with square brackets, as in a 7-bit set, so it had to be checked. The Web gave very few hits for this 1956 machine. Fortunately one was the museum of Dr. Tim Bergin's home base. He kindly faxed me a few pages from a training manual they had on exhibit. Ref. [14] specified a 32-bit word length, fit only for 8-bit bytes. If so, way ahead of its time.

6-bit encodings, each assigned to two different characters, match what I had in [4], and include key or paper tape shifts

to upper case and lower case. Perhaps each shift condition added a 7th bit, selecting between the 6-bit combination assigned to the pairs of characters. The pairs themselves look like the typewriter key pairs. Perhaps the eighth bit was immaterial, or perhaps a parity bit.

But how would 7-bit encodings, formed from 6 input bits augmented by a shift bit (did the CAPS LOCK submit it anew for successive upper case?) drive the Flexowriter on output?

For now I am prepared to cede that the LGP-30 might have been the first and only other computer to have the square brackets in its character set. Until the Stretch computer came along, that is.

Others

Relative to ASCII, I had little to do with introducing the tilde, accent acute, accent grave, less than, greater than, or the standard typewriter special characters of that era, even though I used all except the accents for Stretch. Plus the vertical bar, which went into ASCII at the same time as the curly braces (but many computers had them previously).

So no claims are made.

Placement

I did get X3 to agree to move the alphabets down one position, reserving the three positions after z and Z for international usage in ASCII-alternate sets. I learned that by examining the Copenhagen telephone book while there in 1963 Sep, finding that the names starting with accented vowels followed the regular alphabet of 26.

Curly Braces

Now we get to the mystery. My search started with:

Date: Mon, 3 Dec 2001 14:22:49 -0600 (CST)
From: Eric Fischer
To: bob@bobbemer.com
> Bemer:

> I'm adding a question. Do you have source knowledge about the curly
> braces? I put them in the Stretch set, and now I cannot find any
> previous computer character repertoire that had them. I scarcely
> believe that I was the first to use them, being so common in literature,
> but incorporation in computer sets is a different thing.

Eric:

I remember that you also included them in the 256-character card code that you published in CACM, because I found it interesting that you proposed that they could be used in place of the Algol 'begin' and 'end' keywords, which is exactly how they were later used in the C language.

The next day he added:

Date: Tue, 4 Dec 2001
From: Eric Fischer
To: bob@bobbemer.com
It looks like the Lincoln Writer for the TX-0 at MIT Lincoln Lab probably had the curly braces before you did. Unfortunately the earliest listing of its characters I've seen is from January, 1960, but the letters from its designers that

are among the papers you donated to the Smithsonian give the impression that the machine had been designed and the first one was already being built in August, 1958. (I know there was a letter or article in CACM about it at some point too, but I don't have a copy and don't know the date, and I don't think it included the character list.) (Note: It did, but it was for the TX-2, which became operative in 1959.)

That's the only computer code I know of that seems to have included the curly braces before your Proposal.

The only curly braces found in Ref. [4] were indeed for Stretch and the Lincoln Writer, and they appear nowhere else in character sets prior to 1962, at least. But a recently-obtained copy of [16] indicates that linking this device with the TX-0 was erroneous. Vanderburgh states succinctly that "The Lincoln Keyboard was designed for use on TX-2 ... both for preparation of programs on punched paper tape and for direct console communication in program language".

But here the matter gets obscure. I know that Stretch (in [4]) had two lines of characters to accommodate encoding for more than 64 characters that actually were manipulated by the hardware. Thus the "Stretch character set.

And the Lincoln Writer had two lines of characters, without lower case alphabet shown. Except that shift characters for upper case and lower case were included. So the conclusion is that "it is doubtful that this is an internal character set for TX-2, even if it is an input device". It seems that I was too much in a hurry when I wrote [4], and did not make this distinction well enough, for other 2-line groups were for Flexowriters and other typewriters.

The TX-2 Set -- I/O Devices and the CPU

Vanderburgh's CACM paper [12] shows the set membership as:

26 Block English Letters
10 Standard Arabic Numerals
6 Greek Letters
12 Lower-case English letters ???
8 Punctuation Symbols -- , * ? ' ( ) { } (here they are)
11 Formula Symbols
7 Symbols for Symbolic Logic
8 Special Symbols

We can admit now that the incomplete character set, missing 14 lower case letters, was a little strange, but they claimed that the composition was studied carefully at the time, and that was their decision, strongly influenced by text processing needs.

Clues From the Web

"While TX-0 was still in possession of the big memory we wrote a program which allowed us to simulate a typewriter with 200 characters" (Jack Gilmore in [11]).
"A 36 bit operand word can be divided into one 36, one 27 and one 9, two 18, or four 9 bit subwords formed from the 9 bit quarters. The 9 bit quarters can be permutated among themselves. Any or all of the subwords can be used simultaneously" (TX-2 from [13]).
"Channels or tracks on the tape -- 10 Tracks/tape" [13]
"Lincoln Writer input -- 10 6 bit chars/sec" [13]
"Paper Tape Soroban punch -- 180 7 bit lines/sec" [13]
"Xerox printer -- 20 lines/sec, 1300 char/sec -- 88 characters can be printed in 2 sizes. 6 bit vert. & 9 bit horiz. axes resolution." [13]
"Lincoln Writer output -- 10 6 bit chars/sec" [13]
"One-of-a-kind research computer" [13]
"the keyboard's circuitry logic will sense what the case of the typewriter is and if it is not in the case of the selected

character then the necessary case code will be generated first and then the selected character's code will be generated." (Jack Gilmore in [11]).
"The overbar and underbar will not cause the platen to be shifted to the right (I think he means left) one space position. ... The same is true of the box and circle symbols. ... These four symbols allow us to modify characters and thereby change their meaning." (Jack Gilmore in [11]).

I admit that all the clues were there, but the mechanics were missing. I didn't know how to link them together until a phone call from Jack Gilmore on 2002 Jun 17.

How It Worked

Characters from the Lincoln Writer entered in 6-bit form. Those that were to be themselves were processed into 9-bit form by prefixing (or appending, which might be better for sorting) a specific 3-bit combination to form their internal 9-bit representation. All other characters entered with their own 6-bit form adjoined to a 3-bit representation derived from:

the shift keys (unlocked or locked)
a superscript key (causing half-size)
a subscript key (causing half-size)
the code of a nonspacing box, circle, underscore, or overscore

These pairs were processed into a 9-bit form made by adding the 3 bits to the regular 6-bit codes to indicate each of these modifiers.

Reasons Why TX-2 Had No Priority on Curly Braces

TX-2 was a "one-of". No interchange was considered. It was not a commercial product offering, and its users were captive.
MIT was never associated with the X3.2 work, and had no connections with the committees.
As Jack Gilmore told me on 2002 Jun 18, most of the work on Whirlwind, TX-0, and TX-2 was pretty secret. Thus there was little likelihood that their character repertoires would have been familiar to the ASCII committees.
It's curly braces could have been replaced with brackets, with new slugs, and nobody would have noticed the difference, least of all the CPU and the software.
In actuality, slugs for curly braces had existed for a long time for the basket of the IBM Model B typewriter.
In Figure 7 of [16] (Lincoln Writer Code), (and { are assigned as octal 52,) and } as octal 53. In Figure 8 (The Lincoln Type on the Flexowriter) these same characters are assigned as octal 31 and 21 respectively. Nowhere is it given what 9-bit codes these would have internally, even as their simple graphics, let alone as compound characters as overstruck with circle, square, underline, overline, or as the same characters in reduced size form.
Thus there was no physical internal character set per se for the TX-2. It was ephemeral, put in force by the software used.
And that software used was the equivalent of today's scan codes.

Amelioration

Having established my own priority for point 2 by the fact that there was no internal character set per se for the Lincoln Writer, I must admit that it was a very clever device, and I do not wish this paper to diminish the accomplishment. Early text editing was a major goal and usage, especially for printing reports. So it must have been exceptionally useful for that closed community.

Placing Curly Braces in ASCII

Although X3 had conceded on 1961 Jun 7-9 that ASCII would go from a 6-bit code to 7-bit, almost no characters were added except the lower case alphabet. The curly braces were not added until 1963 Dec 17-18. Here Eric Fischer was able to help again with the X3.2.4 minutes (he must have made copies of every document I deposited with the

Smithsonian Institution, for I recognized my own handprinting on the copies).

Document 12 was "Clamons Code Proposal dated 17 Dec 1963". It led to Item 7, which reads:

An ad hoc committee, comprised of Messrs. Clamons, Arne, Davis, Long, and Turner, was formed to consider the three positions following lower case z in the ISO draft proposal. X3.2.4 voted to recommend: ...
b. It is suggested that the three positions following
lower case z be: left brace, vertical line,
right brace. ...

To understand this, one needs to know that Eric Clamons had become a good friend when the ASCII development began, and continued when I went to UNIVAC as Director of Systems Programming where he was Manager of Product Planning for the UNIVAC 1050, which we tried to make the first ASCII-based computer ever. Together we handled all standardsmaking representation for UNIVAC. The proposal he carried to that meeting was mine.

We continued in the same warm mind-interchangeable relationship when I got him hired into GE when I went there and it continued until he died in 1999. So this is the proof for point 1, in addition to [1] and [2].

Conclusion

The above proves that I was the source responsible for placing 11 different characters into ASCII (point 1), and for at least 8 of these (with the possible exception of the vertical bar, and the square brackets of the LGP-30) it was the first placement in the internal character set of any computer (point 2).

Which is a partial explanation of why a parcel marked "Father of ASCII" (sometime about 1968), at General Electric in Phoenix, was forwarded to me, having inside a letter starting "Dear Bob".

REFERENCES

R.W.Bemer, "A proposal for a generalized card code of 256 characters",
Commun. ACM 2, No. 9, 19-23, 1959 Sep
R.W.Bemer, W.Buchholz, "An extended character set standard", IBM Tech.
Pub. TR00.18000.705, 1960 Jan, rev. TR00.721, 1960 Jun
R.W.Bemer, "A proposal for character code compatibility", Commun. ACM 3,
No. 2, 71-7, (1960 Feb
R.W.Bemer, "Survey of coded character representation", Commun. ACM 3,
No. 12, 639-641, 1960 Dec
R.W.Bemer, "A view of the history of the ISO character code", Honeywell
Computer J. 6, No. 4, 274-286, 1972
Jukka Korpela, Finland.
See http://www.cs.tut.fi/~jkorpela/latin1/ascii-hist.html
www.cwi.nl/~dik/english/codes/7tape.html Dik Winter
www.cwi.nl/~dik/english/codes/internal.html Dik Winter
www.cwi.nl/~dik/english/codes/magtap.html Dik Winter
www.bobbemer.com/BACSLASH.HTM
Letter from Alexander Vanderburgh, Jr., to R. W. Bemer, 1958 August 15
(in National Museum of American History, 315 Box 4), as transmittal cover for
Letter from Jack Gilmore to Vanderburgh, re "The Lincoln Writer".
1958 August 14 (also in NMAH, 315 Box 4).
A. Vanderburgh, "The Lincoln Keyboard & a Typewriter Keyboard Designed for Computer Input Flexibility:, CACM 1, No. 7, 1958 Jul, p. 4.
ed-thelen.org/comp-hist/BRL61-1.html
http://www.computer-archiv.de
J.T.Gilmore, Jr., R.E.Savell, "The Lincoln Writer", M.I.T. Lincoln Lab.,

Group Report 51-8, 1959 October 06.

WHY IS A BYTE 8 BITS? OR IS IT?
Computer History Vignettes

By Bob Bemer

I recently received an e-mail from one Zeno Luiz Iensen Nadal, a worker for Siemens in Brazil. He asked "My Algorythms teacher asked me and my colleagues 'Why a byte has eight bits?' Is there a technical answer for that?" Of course I could not resist a reply to someone named Zeno, after that teacher of ancient times. Some people copied on the reply thought it a useful document, so (having done the hard work already) I add it to my site as further bite of history.

I am way behind in my work, but I just cannot resist trying to answer your question on why a "byte" has eight bits.

The answer is that some do, and some don't. But that takes explaining, as follows:

If computers worked entirely in binary (and some did a long time ago), and did nothing but calculations with binary numbers, there would be no bytes.

But to use and manipulate character information we must have encodings for those symbols. And much of this was already known from punch card days.

The punch card of IBM (others existed) had 12 rows and 80 columns. Each column was assigned to a symbol, a term I use here although they have fancier names nowadays because computers have been used in so many new ways.

The columns, going down, starting from the top, were 12-11-0-1-2-3-4-5-6-7-8-9. A punch in the 0 to 9 rows signified the digits 0-9. A group of columns could be called a "field", and a number in such a field could carry a plus sign for the number (an additional punch in top row 12 of the units position of the number), or a minus sign (an additional punch in row 11 just under that).

Then they started to need alphabets. This was accomplished by adding the 12 punch to the digits 1-9 to make letters A through I, the 11 punch to make letters J through R. For S through Z they added the 0 punch to the digits 2 through 9 (the 0-1 combination was skipped -- 3x9=27, but the English alphabet has only 26 letters). The 12, 11, and 0 punches were called "zones", and you'll notice them today lurking in the high-order 4 bits. Remember that this was much prior to binary representations of those same characters.

The first bonus was that the 12 and 11 punches without any 0-9 punch gave us the characters + and -. But no other punctuation was represented then, not even a period (dot, full stop) in IBM or telecommunication equipment. One can see this in early telegrams, where one said "I MISS YOU STOP COME HOME STOP". "STOP" stood for the period the machine did not have.

Then punctuation and other marks had combinations of punches assigned, but there had to be 3 punches in a column to do this. In most case the third punch was an extra "8".

In this way, with 10 digits, 26 alphabetic, and 11 others, IBM got to 47 characters. UNIVAC, with different punch cards (round holes, not rectangles, and 90 columns, not 80) got to about 54. But most of these were commercial characters. When FORTRAN came along, they needed, for example, a "divide" symbol, and an "=" symbol, and others not in the

commercial set. So they had to use an alternate set of rules for scientific and mathematical work. A set of FORTRAN cards would cause havoc in payroll!

With many early computers these punch cards were used as input and output, and inasmuch as the total number of characters representable did not exceed 64, why not use just 6 bits each to represent them? The same applied to 6-track punched tape for teletypes.

In this period I came to work for IBM, and saw all the confusion caused by the 64-character limitation. Especially when we started to think about word processing, which would require both upper and lower case. Add 26 lower case letters to 47 existing, and one got 73 -- 9 more than 6 bits could represent.

I even made a proposal (in view of STRETCH, the very first computer I know of with an 8-bit byte) that would extend the number of punch card character codes to 256 [1]. Some folks took it seriously. I thought of it as a spoof.

So some folks started thinking about 7-bit characters, but this was ridiculous. With IBM's STRETCH computer as background, handling 64-character words divisible into groups of 8 (I designed the character set for it, under the guidance of Dr. Werner Buchholz, the man who DID coin the term "byte" for an 8-bit grouping). [2] It seemed reasonable to make a universal 8-bit character set, handling up to 256. In those days my mantra was "powers of 2 are magic". And so the group I headed developed and justified such a proposal [3].

That was a little too much progress when presented to the standards group that was to formalize ASCII, so they stopped short for the moment with a 7-bit set, or else an 8-bit set with the upper half left for future work.

The IBM 360 used 8-bit characters, although not ASCII directly. Thus Buchholz's "byte" caught on everywhere. I myself did not like the name for many reasons. The design had 8 bits moving around in parallel. But then came a new IBM part, with 9 bits for self-checking, both inside the CPU and in the tape drives. I exposed this 9-bit byte to the press in 1973. But long before that, when I headed software operations for Cie. Bull in France in 1965-66, I insisted that "byte" be deprecated in favor of "octet".

You can notice that my preference then is now the preferred term. It is justified by new communications methods that can carry 16, 32, 64, and even 128 bits in parallel. But some foolish people now refer to a "16-bit byte" because of this parallel transfer, which is visible in the UNICODE set. I'm not sure, but maybe this should be called a "hextet".

But you will notice that I am still correct. Powers of 2 are still magic!

REFERENCES

R.W.Bemer, "A proposal for a generalized card code of 256 characters",
 Commun. ACM 2, No. 9, 19-23, 1959 Sep
 -- Computing Reviews 00025
 Early public hint of 8-bit bytes to come.
R.W.Bemer, W.Buchholz, "An extended character set standard",
 IBM Tech. Pub. TR00.18000.705, 1960 Jan, rev. TR00.721, 1960 Jun
 -- Computing Reviews 00813
R.W.Bemer, H.J.Smith, Jr., F.A.Williams,
 "Design of an improved transmission/data processing code",
 Commun. ACM 4, No. 5, 212-217, 225, 1961 May
 -- Computer Abstracts 61-1920
 ASCII in its original form.

HOW ASCII GOT ITS BACKSLASH
 Computer History Vignettes

By Bob Bemer

As we've shifted from thousands to hundreds of millions of computer users, much history is lost. Few realize that the backslash character did not exist in much text usage prior to 1961, and in no computer until 1958. A paper by Eric Fischer, submitted to the Annals of the History of Computing in early 2000 (and not yet published), unearthed a backslash on the keyboard of the Teletype Wheatstone Perforator, circa 1937-1945. But this was unknown to data processing people, who were stuck even up to the FORTRAN era (beginning 1955) with the Hollerith punch card code. John Auwaerter, V.P. of Engineering of that same Teletype Corp., and Chairman of X3, certainly never mentioned it.

Here is the story for the record.

GENESIS IN IBM's STRETCH COMPUTER

As an Algol enthusiast I envisioned a backslash for new computer character sets, where before and after pairing with the regular slash (or virgule or slant) could serve for the AND and OR operators of the theoretical Algol character set. I found the opportunity when invited by Dr. Werner Buchholz to do the main design of the 120-character set for the Stretch computer (the IBM 7030). This set, shown on page 12 of [1], shows both curly braces, both square brackets, and the reverse slash -- all of which eventually made their way into 7-bit ASCII.

[2] shows no ESCape character for STRETCH, for I did not think of that until after the character set was built into the hardware. There is one instance of ESCape, in a draft British proposal, for I had by that time alerted Hugh MacGregor Ross.

We may take it as absolute proof of genesis, from the early limited character sets, that IBM's STRETCH was the first computer to use the backslash character, for Reference [2] was actually published a couple of years after the design of that machine. And that uniqueness continued. Reference [2] shows no 6-bit set with the backslash as a working character.

From Reference [3]:

"I had called a joint meeting of IBM, SHARE, and GUIDE, to regularize the IBM 6-bit set to become the standard BCD Interchange Code [76]. Frequency studies of symbol occurrence had been prepared, particularly from ALGOL programs. The meeting of 1961 July 6 produced general agreement on a basic 60-64-character set, which included the two square brackets and the reverse slant, which was chosen in conjunction with "/" to yield 2-character representations for the AND and OR of early ALGOL. This is reflected in the set I proposed to ANSI X3.2 on 1961 September 18."
(Note: I had put the backslash in position 5/15. It enabled the ALGOL "and" to be "/\" and the "or" to be "/".)

SHARE and GUIDE representatives at the meeting were a little stubborn about accepting my proposed backslash, so I asked for a character more important to have. After much discussion they could not agree on a better candidate.

"At the 61 November 8-10 meeting, X3.2 constructed the first formal proposal, X3.21 ..." (which, much modified, was to become ASCII)
(Note: In this proposal the backslash was moved to position 5/12, and there it has remained ever since.)

Of course the need to use pairs of existing symbols to represent a symbol not in the set is long past. So the backslash became that most useful of characters -- one nobody had used or preempted, just waiting there for a new use. The DOS system that Microsoft bought gave us that use in the directory mechanism!

Which is why I refer to the backslash as "my character", together with the other 10 characters I contributed to ASCII -- see this in story.

REFERENCES

R.W.Bemer, W.Buchholz, "An extended character set standard",
IBM Tech. Pub. TR00.18000.705, 1960 Jan, rev. TR00.721, 1960 Jun

R.W.Bemer, "Survey of coded character representation",
Commun. ACM 3, No. 12, 639-641, 1960 Dec

R.W.Bemer, "A view of the history of the ISO character code",
Honeywell Computer J. 6, No. 4, 274-286, 1972

That Powerful ESCAPE Character --
Key and Sequences
Computer History Vignettes

By Bob Bemer

The Birth of ASA Committee X3

On 1960 Jan 13 the American Standards Association convened an organizational meeting in New York City to discuss and approve the formation of Committee X3, Computers and Information Processing. The Scope and Program of Work, which I drafted at the behest of IBM VPs Birkenstock and McPherson, was accepted with a major component as:

"[to develop] a single standard for logical representation of characters and character format in the media used for interchange of instruction, data, and control information between data processing equipments, together with orderly provision for expansion and alternatives."
Ref. Annals of the History of Computing 9, No. 3/4, p.346 (1988)

Note that last clause - "together with orderly provision for expansion and alternatives". To attendees who did not understand at the time, I outlined the possible action of the Escape character, were it to be included as one of the characters in the basic set, and thus as manipulable as any graphic. They must have been somewhat convinced. We see the "expansion" in today's 8-bit and 16-bit sets, and the "alternatives" in the various national and other sets, especially in laser printers.

But also note that only "interchange" was covered. Had I gone for the internal code itself, all manufacturers would have rebelled. But I knew that reducing the internal-to-external conversion to zero would be tempting. Especially if there were a set content mismatch.

In any event, there was precedent.

First Appearances of "Escape"

In 1956 September it was decided that IBM's Stretch computer would have a 64-bit word, each character represented by 8 bits. I was there at Los Alamos when this was decided, and it made me very happy. I used to say at the time that "powers of 2 are magic". It was to prepare for the acceptance of 8-bit characters that I published [1].

The Stretch character set [2,6] was designed by Dr. Werner Buchholz (creator of the term "byte") and myself. I put in

the escape character as a set member, but a Stretch code for that did not appear in the 1960 survey [4] of internal computer character codes because the set was already frozen.

The only surveyed code to include ESC was a proposal by H. MacGregor Ross (of UNICODE fame) on Ferranti's behalf for the British Standards Institution. This followed on my presentation to the same body in February of 1960.

Page 3 of Reference [2] did contain this verbiage about ESC:

"4. Expansion of Set
Future expansion to a set larger than 120 may take place in two ways. One is to assign additional characters to presently unassigned 8-bit codes; allowance should be made for certain control codes which will be needed for communication and other devices and which are intended to occupy the high end of the code sequence. The second method is to define a shift character to "escape" to another character set. Thus, whenever the shift character is encountered, the next character (or group of characters) identifies a new character set, and subsequent codes are interpreted as belonging to that set. Another shift character in that set can be used to shift to a third set, which may again be the first set or a different set. Such additional sets would be defined only if and when there arise applications which require them.

ESC was certainly in the IBM proposal of 1961 May [5], for which Herb Bright said to forget the X3.2 committee and adopt this one!

But X3 did accept the escape concept from the beginning. ESC was in position 7/14 in the code that X3 showed to the Department of Defense in 1961 June. It never left the ASCII work, although it was moved to 1/11 in 1964 May, and stayed there ever since.

Moreover, my short article describing "escape" appeared in the next month's issue of the Communications of the ACM [3]. Publication always gives cachet. I learned that once in attending a decision meeting within IBM, where I seemed to be losing the argument. In disgust, I left and sent in Roy Goldfinger in my place. He, clever fellow, swung the decision to my viewpoint by quoting an article I had written and published! What I could not accomplish in person I could accomplish by having published it! Strange.

How Can One Recognize an Escape Sequence?

It will be useful, before going on, to admit a world of confusion.

Answer:

When it is started by the ESC character itself (customarily just for hardware), or
When, at a level above hardware, it is started by a character other than ESC that it is felt that a superimposed system will not need. And ended by convention or another similar character.

An example of the latter is the "\" as used by C and such. Ever see "t" in text cause a Horizontal Tab movement? I haven't. The reason is that it may be descriptive, as in general text, or announcing, as it is in source programs for the compiler for the programming language C and its offshoots. C could just as well have used "|announce|t", but the backslash fascinated with its brevity, preceding Microsoft's using it for their file hierarchy indicator.

HTML is a programming language, too, and it has four different forms of escape sequence. One is where "<" is the announcer. But what it announces is closed off by ">" -- a far better and more general method than that of C. Oddly, the HTML people refer to only one of those forms as an escape sequence -- not this one.

If you'd like to see escape sequences used properly, look at Internet design paper "rfc1554", which discusses

extensions of Japanese character sets via ISO 2022 [8].

How to Get to the Real Escape Character

Once in the early days of video terminals, e.g. DEC VT52, the ESC key actually emitted the ASCII ESC. And if you followed that by a capital A the cursor would be seen to move one position to the right. This was direct generation, unbuffered.

All that changed with the IBM PC. For excellent and sufficient reasons, its keys emitted not the character itself but a "scan code", which was then remapped to the hardware level, just as a compiler would. A pretty good technical explanation of all this has been made by Randall Hyde. See it for the nitty gritty understanding.

Now any compiler or application software just grabs the scan code from the ESC key and tailors the action to what it wants. Nothing wrong with that.

If you or your software kit have a way to enter the ESC character directly, as I have, some wonderful things can be done. I have a file of all of the original 64 combinations of 8 colors (x 2 for "bright" aspect) that I view in transparent mode (all the control characters show but don't function). From this I can cut the text for a specific foreground-background choice, paste it into my file, and return to regular mode. It's fun to show people how to cut and paste colors. Fortunately these 128 combinations still function even though we now have a much wider color spectrum evocable via the HTML codes.

Proof of Early Skepticism

It must be admitted that acceptance of the "escape" principle was hard-earned, which may surprise the millions of people who use billions of them every day.

From a letter from Herbert S. Bright to Charles W. Bachman, 1982 March 11, on the occasion of the second of my three nominations for the ACM's Turing Award (all of which failed!):

"Bob ... was a principal creator of the ASCII set. I well remember nailing Bob once for an explanation of the 'Escape Character' concept, which was new to me when Bob pushed it through his group. He convinced me that no set of any length would meet all requirements even then, certainly not in the future, and that the Escape concept would work. His Subcommittee, competitor's employees all, grudgingly agreed. After some months of study, they became the strongest boosters of the idea. Meanwhile, Bob's employer -- a large corporation -- seemed somewhat less than enthusiastic, if not downright reluctant, toward the whole ASCII set. It is a matter of public record that he changed employers about then.
(See the story of my departure from IBM).

The question has bugged me ever since; Is this an example of a solid company man placing his conviction about a public need above his well- developed instinct for bureaucratic self-preservation? There aren't too many in this business, who, like you, seemed to have displayed that bizarre character trait.

Having first raised, in Council, the idea of establishing a Turing Award, I have felt a fatherly concern about each selection.

I would feel comfortable if the award for this year goes to irritating, competent Bemer for the idea that will live after him."

I will admit to the "irritating". The energy I had then astounded even me, and I was indeed young enough to be impatient with people that could not see the benefit of many of my ideas.

Other Bows via the Turing Award

From Turing Award Recommendations in 1996 (my third nomination):

Eric Clamons (longtime chairman of the ASCII Committee):

"Bob's contributions ... one that affected the industry more profoundly than any other: the proposal to include the ESCAPE character in ASCII ... To a large extent the ESCAPE code made ASCII what it is ... The ESCAPE code also opened the way toward 'data streaming', an idea that made possible the widely-used softcopy control standard ..."

John Auwaerter (Chairman of X3):

"The impasse was broken by Mr. Bemer's "escape" code and registration of alternative meanings. ..."

Walter Carlson (ex-President of ACM):

"His idea ... was to use an "escape sequence", whereby each natural language had its own registered escape seqence code and associated set ..."

Auwaerter indicates that without the escape character and its functionality it may have been impossible to agree on and accept ASCII as a national and then worldwide standard. Surely the lack of a worldwide interchange alphabet would have inhibited development of the Internet and the Web!

IS THE ESCAPE SEQUENCE THAT IMPORTANT?

Care to speculate upon what the world would be like now if:

the escape sequence principle had not been created? or
if one had the power to now withdraw the escape sequence idea from usage? (IBM did not seek a patent; indeed, a $300 million dollar suit
against IBM was unsuccessful because the escape sequence had been put into the public domain)

The following list is illustrative but incomplete. Its purpose is to show how different the world would have been without ESCAPE! Think of your own input to the list.

Here are some things one would miss:

Video computer monitors (as on personal computers) are gone. With no escape sequences to move cursors, change foreground and background colors, set blinking or inverse, etc., we are back to Teletypewriters and other mechanical aids.
Writers are back to typewriters, for Word Processing is no more.
No more making one's own artwork or personal greeting cards.
Computer graphics hardly exist. No more pictures of a rotating auto chassis, no visual engineering design, no CAD (computer- aided design) for buildings, kitchens, highways, etc.
No laser printers, whose manuals show hundreds of escape sequences used for their control and operation.
Various phone services are gone, like checking a bank balance by typing in the account number followed by # and/or *, for these act as escape sequences to the bank computer.
Other screen applications are gone or reduced in functionality. Airline reservations, for example (we might have to

revert to inflexible and published schedules and fares -- which might not be an all-bad idea!).
Medicine reverts. No nurse's video stations, displays for treadmills, etc.
Forget air traffic control as we know it now, and all the intelligent displays in submarines and fighter planes.
Japan, Korea, and China are back to brushwork and linotype fonts for their character sets. The relationship to ASCII is gone.
Photocomposition cannot shift to subsets of the various symbols of the world, in various sizes, fonts, rotation, etc. Books will thus be more expensive, and rarer.
And -- No Internet or Web as they are now!

REFERENCES

R.W.Bemer, "A proposal for a generalized card code of 256 characters",
Commun. ACM 2, No. 9, 19-23, 1959 Sep
R.W.Bemer, W.Buchholz, "An extended character set standard",
IBM Tech. Pub. TR00.18000.705, 1960 Jan, rev. TR00.721, 1960 Jun
R.W.Bemer, "ESCape - a proposal for character code compatibility",
Commun. ACM 3, No. 2, 71-72 (1960 Feb)
R.W.Bemer, F.A.Williams, "Survey of coded character representation",
Commun. ACM 3, No. 12, 639-641, 1960 Dec
R.W.Bemer, H.J.Smith, Jr., F.A.Williams,
"Design of an improved transmission/data processing code",
Commun. ACM 4, No. 5, 212-217, 225, 1961 May
R.W.Bemer, W.Buchholz, "Character set",
Chapter 6 in Planning a Computer System, McGraw-Hill, 1962
R.W.Bemer, "The American standard code for information interchange",
Datamation 9, No. 8, 32-36, 1963 Aug, and ibid 9, No. 9, 39-44, 1963 Sep
[ISO2022], International Organization for Standardization (ISO),
"Information Processing -- ISO 7-bit and 8-bit coded characters sets --
Code extension techniques", ISO 2022-1986.

ORIGINS OF TIMESHARING
Computer History Vignettes

By Bob Bemer

Purpose of This Story

I know how Philo Taylor Farnsworth must have felt for much of his life. "Philo Who?" You know, the guy that invented television. "But I thought television was invented by RCA or Allen DuMont?" No, friends, Farnsworth invented electronic TV. But it had to be proved in court, and the truth was seldom acknowledged until after his death. RCA paid royalties. Seems they agreed.

I'm damned if I want to be a Farnsworth. (See) I lay at least a partial claim to the genesis of the timesharing concept, the basic reason you can read this on this Web site today. The myth that Christopher Strachey was first with the concept has been perpetuated in Datamation, Scientific American, and the Annals of the History of Computing. The remarkable aspect is that Strachey himself didn't agree! Nor does Prof. John McCarthy [6].

My claim is that I published the first paper to describe the aspects of computer timesharing for commercial usage, with users acting individually. It was in the public domain, published nationally. See what you think.

REFERENCES

Let's lay them up front, because they're more than references one may wish to make secondary referral to. They're the basis for the argument!

R.W.Bemer, "How to consider a computer", Data Control Section,
Automatic Control Magazine, 1957 Mar, 66-69
R.W.Bemer, "The status of automatic programming for scientific computation",
Proc. 4th Annual Computer Applications Symposium, Armour Research
Foundation, 1957 Oct 24-25, 107-117 (Panel discussion, pp. 118-126).
W.F.Bauer, "Computer design from the programmer's viewpoint",
Proc. Eastern Joint Computer Conf., Philadelphia, PA, 1958 Dec 3-5,
American Institute of Electrical Engineers, 1959 Jul, 46-51.
C. Strachey, "Time Sharing in Large Fast Computers", Proc. Intl. Conf. on
Information Processing, UNESCO, June 1959, B336-B341 (not on Web).
R.M.Fano, F.J.Corbató, Time-Sharing on Computers", Scientific American Magazine, 1966 September, 129-140.
See J.McCarthy, "Reminiscences on the History of Time Sharing",
1983 Winter or Spring.
M.Campbell-Kelly, "Christopher Strachey, 1916-1975, A Biographical Note", Annals History of Computing, 7, No. 1, 1985 Jan, 19-42.

My Unlikely Source for the Concept

I have told somewhere about attending Curtiss-Wright Technical Institute of Aeronautics prior to World War II. The school was in Glendale, CA. The date was the Fall of 1940. Another well-known institution there was the Bob's Big Boy hamburger. The original! At least it made the owner Mayor of Glendale. I went crazy for them, sometimes eating two in a row (I was young).

The drive-in was fascinating to this Midwestern boy. Parked cars around the restaurant, serviced by girls in attractive garb. But on going inside you saw what was to me an absolute first -- a revolving drum with elasticized or expanding bands. The waitresses put the orders in there, and the cooks referred to them often. The key word is "interspersion". Hardly any order ever got produced in its entirety at one time. The production sequence was "shared" by all customers and by all cooks. That idea stuck with me for 17 years!

Contribution of Robert W. Bemer (Ref. [1] - 1957 March)

The final section of this paper, commissioned by Editor Evan Herbert, described aspects of future commercial timesharing:

"Future computer operation, which strongly influences the design of the programming languages, has some vitally interesting possibilities. In this glimpse, the picture presented here is dependent upon three features:

Faster computers always lower the dollar cost per problem solved, but not all companies will be able to afford the high prices of the next generation of super-computers. They simply may not have enough problems to [fully] load one.
Producing a spectrum of machines is a tremendous waste of effort and money on the part of both the manufacturers and the users.
Availability of a huge central computer can eliminate the discrete acquisition of multiple smaller computers, homogenize the entire structure of usage, and allow a smaller, and more numerous class of, user into the act, thus tapping a market many times the size presently projected with current practice in computer access."

"Assuming the availability of practical micro-wave communications systems, it is conceivable that one or several computers, much larger than anything presently contemplated, could service a multitude of users. They would no longer rent a computer as such; instead they would rent input-output equipment, although as far as the operation will be concerned they would not be able to tell the difference. This peripheral equipment would perhaps be rented at a base price plus a variable usage charge on a non-linear basis. The topmost level of supervisory routine would compute these charges on an actual usage basis and bill the customer in an integrated operation.
These program features are, of course, recognizable to Operations Research people as the Scheduling and Queuing problems. Using commutative methods, just as motion pictures produce an image every so often for apparent continuity, entire plant operations might be controlled by such super-speed computers.

These future hardware capabilities (and few competent computer manufacturers will deny the feasibility, even today, of super-speed and interleaved programs) demonstrate a pressing need for an advanced common language system so all users can integrate their particular operations into the complex of control demanded by an automated future ..."

(Note: This call for a "common language" was issued two years before the start of COBOL work.)

This excerpt shows two biases. 1) Equipment rental as a way of life, for I was then an IBM employee, and rental was then their policy, and 2) Manufacturing control as a primary example, for this was the audience for the publication.

Contribution of Robert W. Bemer (Ref. [2] - 1957 October)

On p. 126, Bemer's reply was:

Q: (Omitted from the publication, but I think the word "time-sharing" was used in the question. I used the word myself in a talk to the Franklin Institute the previous February, but you can find several previous papers, particularly from the SAGE project, that used the same phrase. But strictly in a hardware sense.)
A: "I am in favor of the short-order-cook policy that I think will come into effect perhaps five or ten years from now. It might resolve at least a certain class of problems as between the small and large computers. If one had an extremely large, extremely fast centralized computer with various lines radiating out, and with terminal facilities such as a person now only gets in in the form of input-output devices at the computer, and if one could have high-speed transmission to and from this centralized computer, it would be like a short-order cook. It takes the orders off the lines and, so to speak, heats up the griddle and sees that the toast is ready while it is pouring the coffee. It will be self-scheduling, self-regulating, and self-billing to the customer on the basis of use of the input-output device. I think, since the larger and faster computers, as far as production problems are concerned, always produce more problems solved per dollar once the problems are in the machine, that this is the obvious direction to go. I agree that at the present time there are many small computers that seem to take less trouble than a large one; but I think that, in the long run, we will use the largest computers and will start thinking in terms of compatibility of languages and ultimately in terms of a single language."

Contribution of Dr. Walter F. Bauer (Ref. [3] - 1958 December)

Here Walter Bauer describes his "Ultradatic, A Conjectural Computer". The part of the paper pertinent to timesharing says:

"It is fostered by the idea that the large computer is more economical to use today than the small computer, or a number of small computers, as long as the total workload is sufficient to keep the large computer busy, or as long as unnecessary expense is not incurred by idle time. The idea further stems from the fact that with increasing frequency one sees computers being operated from remote locations by means of communications over voice channels, or at least extensive input or output of data is being fed to computers from remote locations.

The central idea here is that each large metropolitan area would have one or more of these super computers. The computers would handle a number of problems concurrently. Organizations would have input-output equipment installed on their own premises and would buy time on the computer much the same way that the average household buys power and water from utility companies. In fact, the charges for the services would depend on the type of services rendered. If the problem required extensive use of the higher priced circuits, for example, high-speed multiplying, the rental rate for the time used would be higher than in other cases. The user would be charged for that time used and only that time used, and the accounting procedure would reflect the customer's detailed use."

Bauer further says that "the Ultradatic idea was first described in talks given by the author to the San Diego and Rio Grande chapters of the Association for Computing Machinery in late 1957 and early 1958".

He refers to "parallel programming" and Stan Gill's 1957 paper on that subject.

At the Munich IFIP meeting in August of 1962, someone mentioned my paper to Bauer, who until that time thought he had published the first paper on commercial timesharing. We met in a restaurant to discuss the timings and content. Dr. Bauer, with his Ramo Wooldridge experience, and as founder of Informatics (later bought by Sterling Software), was usually quite up-to-date in the computer field.

Contributions of Fano and Corbató (Ref. [5] - 1966 September)

This paper goes into much detail about working TSS systems. Only the beginning is pertinent to the purpose of this history, where the authors credit Christopher Strachey of the United Kingdom as the one "who first proposed (in 1959) a time-sharing system".

Contributions of McCarthy (Ref. [6])

This paper has problems of inaccuracies, suppositions, and bad dates. McCarthy's first thinking about time-sharing "might have been around 1955". But he does say that his "first attempts to do something about time-sharing was in the Fall of 1957 ..." And that to get the requisite hardware from IBM "took a long time ... perhaps a year, perhaps two." One suspects he did not keep his records.

Nor did he publish, for this paper [6] has no references. He admits that "In all this, there wasn't much publication. I wrote a memo to Morse dated January 1, 1959 ..." But in the next sentence he admits that "it has been suggested that the date ... should have been 1960."

I also complained to Eric A. Weiss, of the Annals of the History of Computing, re the genesis of timesharing. I cannot find an answer. What I appreciate is that McCarthy complained about Strachey's credits in [5] to the Scientific American. So did I. They replied to me that "The point you raise is most interesting, and I have put it before Professors Fano and Corbató".

The authors did reply to Scientific American, but oddly enough backed off to the 1959 Jan 01 internal paper by McCarthy, 22 months after my external paper. But McCarthy says that "Corbató was surprised to find my 1959 memo in the files".

Also in their reply they credited Strachey with "substantial detail". I am sure that both Bauer and I could have engendered detail to go with the concept. I, in particular, was limited by space available in a popular journal.

And so we come full circle. The entire chorus of credits for Strachey hits a bad chord.

The really interesting part of McCarthy's paper is its Appendix, containing Strachey's reply to a query from Don Knuth about origins. Here are the significant parts of Strachey's reply:

"It was mainly about multi--programming ... I did not envisage the sort of console system which is now so confusingly called time sharing ... Halsbury ... was certainly right to say that in 1960 'time sharing' as a phrase was much in the air. It was, however, generally used in my sense rather than in John McCarthy's sense ..."

Strachey's reply also said that the IFIP Conference was in 1960. It was in 1959. As he died just a year after writing the letter, the memory failure is forgivable.

Contribution of Christopher Strachey (Ref. [7])

This remarkably complete story of Strachey's life confirms my tale in many ways:

Strachey's Stretch visit to IBM, and to the ACM Conference in Houston in June of 1957, where we talked, three months after I had published my timesharing paper.
"his name is often associated with the origination of multiaccess timesharing computers. For example, Fano and Corbató ... credit him with the original conception." But that is second hand reporting.
"The paper [5] was influential and has come to be regarded by some as a significant step in the development of time-sharing ...". (no doubt this is true, if misguided). "Strachey was himself surprised to receive so much credit. ... Strachey's view, somewhat naively, was that time-sharing was in the air and that he was merely articulating those ideas".
Nevertheless, "Strachey filed a patent application for time-sharing in February 1959. ... The patent (British patent 924672) was granted in 1963." Now that's a strong articulation! The patent seems to have had no more influence than the one recently granted in the U.S. for the windowing method for Year 2000 repair, now roundly laughed at.
"A definite statement on the extent of Strachey's part in the invention of time-sharing will have to await a more thorough analysis."

(TIME)Sharing the Credit

It has been said that great ideas often occur simultaneously to several people. In the pertinent period there was a ferment, and one probably cannot ascribe the entire idea to a particular person as we are able to do with Da Vinci.

Much could have been gotten from me. As my first publication elicited a suggestion from a Dr. Franz (???) that I should be fired because it was not IBM's policy, that may be why IBM instigated no patent applications. Maybe I had put it in the public domain, as I did with publication of my escape sequence concept. But it is no secret that I was very active professionally in the 1957-1959 period.

Even when the New Yorker Magazine profiled me in 1957 January I mentioned the community computer idea, which we see again in Bauer's paper.
I was speaking pretty much for IBM, both professionally in ACM and to the customers personified by SHARE and GUIDE.
My Armour Research Foundation paper became well-known,
I knew McCarthy well, antedating the internal memorandum that Fano and Corbató finally found, and attended his 1959 Apr 16 Symbol Manipulation conference.
Strachey attended the ACM meeting in Houston in 1957 June, as I did. I know, because I talked to him, as mentioned above.
Strachey also visited IBM's STRETCH project then.

So I stand by my original claim. I did not create a timesharing system. I did not create a practical design for such. But I was the very first person to publish in some medium (and it was national) the concept of sharing a computer remotely amongst many users via input/output equipment for commercial usage!

Final Note on Historical Accuracy

As I prepare historical material for this site, I am astounded at how many dates the historians (and those they write about) have gotten wrong. In the fast-moving computer industry, I would think it important to keep accurate records of idea generation, patents, public talks and publications.

Accordingly I submit for general use the nonpersonal subset of my travel log from 1954 (when I first kept records) to

date. I hold no copyright, and anyone wishing to flesh out the time line may use it as a starting point. After all, busy though I was in computer activities, I could not go to every meeting!

Each trip is identified as the nth in that year. These could be interspersed in the text as identifiers, if anyone thinks that a useful idea.

You can find the path to these logs in the "Interesting Computer History" page, but I must admit that their interest is limited to this one purpose. Otherwise, I agree. They're pretty dull!

Back to History Index Back to Home Page

Bob Bemer
Scripps Howard
News Service

If the Web, the Internet,
ASCII, and laser printers
printers are very useful
to you --
click here

License plate of Bob Bemer's Ford Expedition
Steve Hebert / Abilene Reporter-News

Computer Software Consultant, Futurist and Raconteur:
Specialties: Standards, history, alternative methods and inventive solutions
Key responsibilities: Architectural & visionary lead consultant for major software
and marketing development projects.
Motto (original) for a Half-Century: ((((DO SOMETHING!) SMALL) USEFUL) NOW!)

Contact Information - Bob & Bettie Bemer:
bob@bobbemer.com is our preferred e-mail address
Home Phone: 940-779-4016 Fax: 940-779-2296

Computer Pioneer Bob Bemer, 84 (washingtonpost.com)

www.washingtonpost.com/.../A4138-20...?Tradueix aquesta pàgina

per Patricia Sullivan - en 33 cercles de Google+

25/06/2004 - Robert W. "Bob" Bemer, 84, who helped invent the language used by ... Without the invention of the computer code ASCII, there would be no e-mail, ... characters in computers," Mr. Bemer told Computerworld magazine in ...

Bob Bemer

www.thocp.net/biographies/bemer_bob.htm?Tradueix aquesta pàgina

08/03/2013 - biography of Bob Bemer. ... Without the invention of the computer code ASCII, there would be no e-mail, no World Wide Web, no laser printers ...

Robert W. Bemer - IEEE Computer Society

www.computer.org/portal/web/awards/bemer?Tradueix aquesta pàgina

Rob Bemer, becoming a programmer in early 1949, has worked at RAND ... He was the major force in developing ASCII, contributing 10 characters ... He invented the escape sequence and registry concept, and is called the “Father of ASCII”. ... Wall Street Journal, New Yorker, New York Times, Time Magazine, Vanity Fair, ...

Bob Bemer, father of ASCII, dies.(American Standard Code ...

business.highbeam.com › ... › Computer Weekly?Tradueix aquesta pàgina

13/07/2004 - Related newspaper, magazine, and journal articles ... Robert W. "Bob" Bemer, 84, who helped invent...invention of the computer code ASCII, ...

Bob Bemer, father of ASCII/COBOL/8-bits-per-byte, dies at age 84 ...

arstechnica.com/civis/viewtopic.php?t=518687?Tradueix aquesta pàgina

26/06/2004 - Robert W. "Bob" Bemer, 84, who helped invent the language used by ... characters in computers," Mr. Bemer told Computerworld magazine in ...

ASCII - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/ASCII?Tradueix aquesta pàgina

1 History; 2 ASCII control characters. 2.1 ASCII control code chart .... Bob Bemer introduced features such as the escape sequence. His British colleague Hugh ...

Mr. Robert Bob Bemer - ZoomInfo.com

www.zoominfo.com/p/Robert-Bemer/540296710?Tradueix aquesta pàgina

WASHINGTON - Robert W. "Bob" Bemer, who helped invent the language used by ... Without the invention of the computer code ascii, there would be no e-mail, ... to represent characters in computers," Bemer told Computerworld magazine in