In Search of the Original Fortran Compiler

Paul McJones
Software Preservation Group, Computer History Museum

In April 2002, Grady Booch sent out an email with the subject "Preserving classic software products." His appeal to provide top 10 lists and to suggest where to look for the source code received an enthusiastic response. Grady and the Computer History Museum organized a workshop in October 2003.[1] As a result, a month later, a group of Computer History Museum trustees, staff and volunteers (including me) established the Software Collection Committee[2] (SCC) to explore the software side of the museum's mission: "To preserve and present for posterity the artifacts and stories of the information age." I decided to take on the project of collecting Fortran materials.

John Backus's Fortran project at IBM developed one of the first higher-level programming languages as well as a set of techniques for code optimization that allowed Fortran to compete with hand-tuned assembly language.[3] The combination of ease-of-use and efficiency led to Fortran's rapid adoption and lasting impact on programming language design and implementation. Fortran's success on the IBM 704/709 led to Fortran compilers for many brands and models of computers, making it possible to port Fortran programs among different computers; Fortran program portability also made Fortran useful for nonnumerical projects. While Fortran's popularity waned as other languages grew up to fill various niches, some high-performance computing users continued to use Fortran, with periodic updates, to the present.

The history of the Fortran project is well documented in papers by Backus and others, covering the project, its impact, and technical analysis of the compiler itself.[4] However, the artifact itself—the source code for the original Fortran compiler—seemed to me an appropriate starting point to gain experience with collecting, preserving, and presenting historic software. By spring 2006, I had assembled a collection including listings, source and executable code, documentation, and more and had made the collection publicly accessible via a website. Take a look at that website now (http://www.softwarepreservation.org/projects/FORTRAN/)[5] to have it mind as you read this article, which describes my collection efforts. I hope my experience will be useful to other computing history researchers from the practitioner world considering similar efforts.

Editor's Note

Our anecdotes are typically on topics in computing history, but sometimes are about doing historical research; the current anecdote is the latter. At this time, it is more important than ever that historians and practitioners seek out original computer code and documentation before it is lost forever. The Computer History Museum (CHM) is one of the institutions encouraging such collection and archiving; for example, through its Center for Software History (http://www.computerhistory.org/softwarehistory), its Software Industry Special Interest Group (http://www.softwarehistory.org), and its Software Preservation Group (http://www.computerhistory.org/groups/spg/). Here we publish CHM volunteer Paul McJones' story of his collecting and archiving historical Fortran materials.

Getting Started

I started my search for Fortran by calling John Backus, with whom I had worked in the mid-1970s (on functional programming) but had not seen for about 25 years. He did not have a copy of the source code and directed me to Irv Ziller, who had been his first hire on the project. Irv responded to my request, saying that he didn't have the source code, but he recalled material being sent to the Smithsonian to become part of their collection. I visited the website of the Smithsonian National Museum of American History (NMAH) and located the group that had created the Computer History Collection,[6] but my initial attempts to establish communication with the staff were unsuccessful.[7]

One day I was chatting with a coworker, Jim King, who had worked at IBM Research for many years. Jim had used Fortran II and the Fortran Monitor System on an IBM 709 in college in the early 1960s and had some interesting anecdotes.[8] Jim suggested IBM's SHARE user group library as a potentially interesting source of material, and he also recalled the extensive IBM 7094-related library at Boeing when he worked there in the 1960s. Another member of the SCC, Dick Sites, recalled having seen "handwritten Fortran I documentation in a basement of the Sloan building at MIT in 1965 …. As I remember, it described the overall compiler design and optimizing algorithms. I remember a fairly neat but slightly small handwriting with hand-drawn boxes and arrows for some of the algorithms." Also, Len Shustek, Computer History Museum founder, pointed out this sentence in Backus's article[9] about Fortran at the first History of Programming Languages conference: "Shortly after the distribution of the system, we distributed—one copy per installation—what was fondly known as the 'Tome,' the complete symbolic listing of the entire compiler plus other system and diagnostic information, an 11 × 15 inch volume about four or five inches thick." Presumably this was different from what Sites had seen, since it was not handwritten.

Digging Deeper

At the monthly SCC meeting in February 2004, my early-Fortran quest became the committee's first official test of collection and preservation efforts. In addition to the source code, I decided to start asking about the "Tome." The next person I contacted was Bob Bemer, who entered the computer business in 1949 and led the development of FORTRANSIT,[10] the second Fortran compiler, for the IBM 650. Bob's website included an article "Who Was Who in IBM's Programming Research?—Early FORTRAN Days"[11] that reproduced the IBM Programming Research Newsletter from January 1957 with short descriptions of the Fortran team members, including the sentence "Hal [Harold Stern] is currently working on 'TOME' describing FORTRAN internally." Bob told me he hadn't saved a copy of the "Tome." Sadly, Bob died four months later.

SCC member Dick Gabriel mentioned Paul Pierce's extensive private computer collection.[12] I contacted Paul. He didn't have a copy of the "Tome," but he had a number of useful suggestions for further research and volunteered to scan more of his SHARE and SOS (SHARE Operating System) materials.

Tom Van Vleck was at MIT in the 1960s, and I thought he might have heard of the "Tome."[13] He hadn't, but he suggested several leads for me to follow: Jean Sammet, Lynn Wheeler, and other IBM folks I'd worked with in the 1970s, Doug McIlroy, and Frank da Cruz, who maintains the Columbia University Computing History website.[14] McIlroy told me that after he joined Bell Labs in 1958, he used Fortran II, and the source code was available then. He also told me, "Another document that came with Fortran, which everybody got to know, was the 'stop book.' The compiler did not issue diagnostics. Instead it halted. The machine operator would record the instruction counter from the console lights. The stop book told what the cause of each stop was, often very cryptically: 'trouble in the tag table, or some other cause.'"

In May 2004, I came across the special issue of the *Annals of the History of Computing* dedicated to the 1982 National Computer Conference Pioneer Day celebration of "FORTRAN's Twenty-Fifth Anniversary," which contains a number of edited session transcripts, articles, anecdotes, and an annotated bibliography.[15]

Around this time, John Backus told me that he'd donated his personal papers to the Library of Congress, and he gave me a spreadsheet listing the donated items.[16] John also gave me extra copies of papers and photographs, including a photocopy of a 29-page memo,[17] "Preliminary Report: Specifications for the IBM Mathematical FORmula TRANslating System, FORTRAN," dated 10 November 1954 (see Figure 1).

J.A.N. Lee's annotated bibliography in the *Annals* special issue says the authors were probably John W. Backus, Harlan Herrick, and Irving Ziller. He notes, "This is the first formal proposal for the language FORTRAN. It lists the elements of the language that are proposed to be included in the eventual implementation, together with some suggestions for future extensions. It is interesting to match this proposal with the Programmer's Reference Manual[18] and to note that many of the ideas of later FORTRANs as well as Algol appear to have been given birth in this document."

Daniel N. Leeson's article in the same issue[19] mentioned that materials they'd used for the Pioneer Day event were from private collections, "two of which are unusually noteworthy":

> Roy Nutt and Harlan Herrick have both made a special effort to retain material from their early days in computing. Nutt possessed a microfilm of (allegedly) every document in the FORTRAN development offices at the time the product was released. He generously donated a copy to the IBM historical archives. Herrick's collection of memorabilia was also extensive. For example, he owned the only known copy of IBM's first FORTRAN film, made in Poughkeepsie about 1958, that showed how FORTRAN could be used to program a solution to 'The Indian Problem' (a calculation demonstrating the effect of compound interest on the $24 said to have been paid for Manhattan Island).

I tracked down Leeson and called him. He thought it was very unlikely that any copies of the source code survived. He used to run the organization that handled binary program distribution for IBM, and he said that IBM didn't even want to archive the binaries once a particular machine went out of production. Leeson also speculated that Roy Nutt's personal collection may have been lost when he died in 1990. Leeson later donated his copies of a Fortran marketing film released in 1957 and the retrospective video created by IBM for the Pioneer Day event.[20] I later talked to the IBM corporate archivist, who had recently inherited a vast but poorly

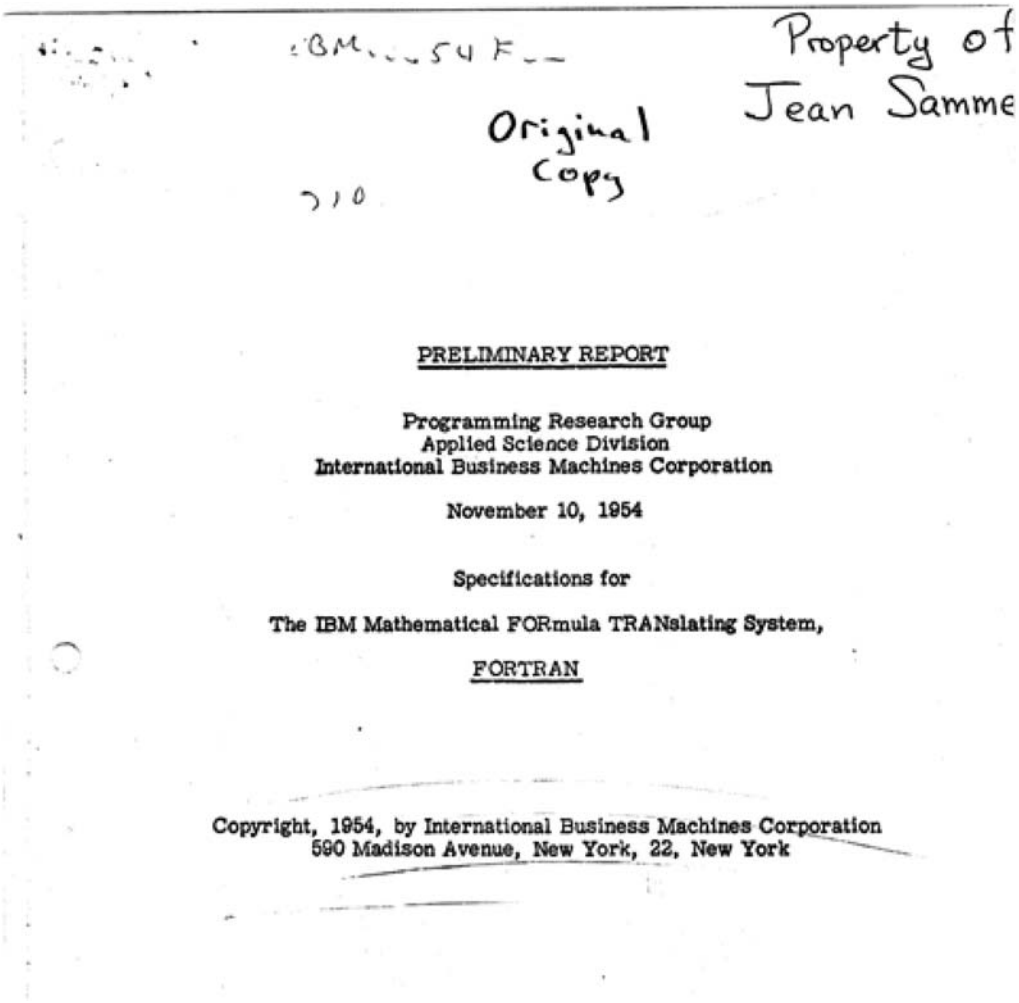

Figure 1. Preliminary Report, 1954.

cataloged archive. I never ended up getting access to any items in the IBM archive.

David Padua wrote an article[21] about the pioneering code optimization of the original Fortran compiler for a special issue of *Computing in Science and Engineering* on "the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century." I contacted David to see if, by any chance, he knew of someone with a copy of the source code. He didn't, but he and his colleague Sam Midkiff suggested contacting retired IBMers Fran Allen[22] or Marty Hopkins.

When I contacted Jean Sammet,[23] she also didn't know where to find the Fortran source, but suggested I contact Fran Allen and J.A.N. Lee.

J.A.N. Lee[24] responded to my email by saying, "I have asked about the original FTN compiler several times but without any success. Two sources would seem possible—John Backus himself … and the original recipient, Westinghouse-Bettis."[25] As it happened, I'd recently looked through several boxes of Fortran materials Lee had already donated to the Computer History Museum. They contained, among other things, photocopies of essentially all the papers included in his annotated Fortran bibliography.

First Results

During my search, I would periodically attempt to communicate with the Smithsonian. In June 2004, I was able to make contact with Alicia Cutler, a specialist in the Collection of Computers and Mathematics. She ran a search for me in their internal catalog, which produced a 12-page listing of documents related to Fortran. Three of them jumped out at me: three volumes totaling 1,321 pages, dated 1959, with

the comment, "This is a FORTRAN program listing for the 704. Well documented." After further study, Alicia told me that these volumes relate to Fortran II, not Fortran, and they are accompanied by a letter from A.L. Harmon in May of 1959 stating that these are the SAP listings for the final Fortran II. This was a wonderful discovery: the Fortran II compiler had additional features, but it was essentially a superset of the original Fortran compiler code base. The question of how to get a copy of this listing would occupy me for the next 20 months.

One of the people Tom Van Vleck had mentioned was Frank da Cruz. From Frank's Columbia University Computing History website, I followed a link to the IBM 704 page of George A. Michael's website, "Stories of the Development of Large Scale Scientific Computing."[26,27] I sent him an email asking about the "Tome." He replied that although he was one of the early Fortran users, and Lawrence Livermore National Laboratory (LLNL) had four IBM 704s, he did not recall the "Tome." He noted that an LLNL programmer, Robert Hughes, had worked on the original Fortran project "on loan" from Livermore, and referred me to an oral history he had conducted with Hughes.[28] In that oral history, Hughes, who was one of the authors of the original 1956 IBM Fortran manual,[29] said: "I worked on what they called the 'first-level' documentation. And I made the biggest mistake of my life by not bringing a copy of that home. Now you understand why I missed making my first million dollars." So, apparently, he didn't bring a copy of the "Tome" back to LLNL. Hughes died in 2007 and Michael died in 2008.

I had learned from Alicia Cutler that Peter Zilahy Ingerman was the donor of the Fortran II listing at The Smithsonian.[30] I called Ingerman and had a very pleasant conversation. It turned out he'd donated the Fortran materials to The Smithsonian a number of years ago and did not remember the specific item I was interested in, but he volunteered to travel from his home in New Jersey to Washington if that would help.

In July 2004, I decided to try drawing attention to my search by creating an online journal called Dusty Decks. My initial post said, "I am using this weblog to discuss historic computer software and hardware, among other topics. For several months, I've been studying the early history of Fortran and trying to track down the source code for the original Fortran compiler. Although I just set up this weblog recently (June-July 2004), I've created backdated entries to document my quest in chronological order, starting here."[31]

As evidence for the power of the search engines, only three days after I published the blog, Micah Nutt, son of Fortran project participant Roy Nutt, commented on the post about Daniel N. Leeson. Micah refuted Leeson's speculation that Roy Nutt's personal collection may have been lost when he died: The family understood the significance of Roy Nutt's accomplishments and were carefully stewarding his papers. I followed up with Micah via email, and he told me, "The fiche I have is most likely the same as the set my father donated to the IBM library in 1982 for the 25th anniversary of FORTRAN. The contents appear to be the specifications, flowcharts, mathematical analysis and source code for FORTRAN along with (at least part of) the user's manual. The documents are a mix of handwritten and machine based (some with handwritten edits and notes)." Over the next months, Micah and I explored different approaches for having the microfiche digitized.

Another person who commented on the blog post was Leif Harcke, then a graduate student at Stanford as well as a knowledgeable practitioner of "retrocomputing."[32] Leif explained, "Fortran II was a strange beast; it ran under the Fortran Monitor System (FMS). FMS could either run the machine standalone, or it could run under IBSYS. Fortran II was link-compatible with the FAP assembler, the IBM product which superseded UA-SAP. … I'm not sure the compiler itself will be of any use without the FMS monitor and the FAP assembler."

Around this time, Alicia Cutler at the Smithsonian suggested that the Fortran II listing could be loaned to the Computer History Museum, where we could scan it, and that process was initiated.

In October 2004, two months after I wrote a blog post about Micah Nutt and Leif Harcke, Bob Abeles commented, "The IBSYS tapes on Paul Pierce's site contain the source for FAP, FORTRAN II, FMS (version that ran under IBSYS), plus lots of other goodies." I followed up with Bob by email, who supplied me with a utility program he'd written for working with the tape images from Pierce. This was an amazing discovery: a machine-readable copy of the source code for a "superset" of the original Fortran compiler. Fortran II extended Fortran with separate

compilation but preserved the original compiler's parsing, analysis, and code generation.

In January 2005, Peter Capek, who had worked at IBM Research for many years and was friends with several of the original Fortran project members, told me about a document he'd come across in his files: "a detailed description of the FORTRAN compiler, dated in 1960, and explicitly distinguishing between the 704 and 709 versions, but covering both. … It's a couple of hundred pages, and describes each section of the compiler, including table structure, in considerable detail." Peter kindly sent me a photocopy.[33]

Sharing with Others

By April 2005, I decided the items I'd located, including the Fortran II source code, the Preliminary Report, the Systems Manual, and a number of manuals from Al Kossow's Bitsavers collection,[34] justified creating a project website. The Computer History Museum had set up a webserver with a web-based content-management system for the use of the SCC, so it was simple to create and maintain a project website.[35] A week or so after I announced the website on the Dusty Decks blog, John Van Gardner posted a comment. He had been one of the IBM Customer Engineers at Lockheed Aircraft in Marietta, Georgia. In May 1956, he helped install IBM 704 serial number 13, and in April 1957 he helped get Fortran running on that machine. There was a problem that caused the compiler to print "Machine error" and halt. John noted, "To solve this problem we needed a source listing of the compiler. It took the Branch Manager several days to get one for us, and it was on a roll of 35 mm microfilm. This had to be under IBM control at all times, and when we finished the bug, it was sent back to the Branch Office." John also wrote a two-page anecdote about the resourceful techniques he used in 1957 to debug a hardware problem that resulted in the Fortran compiler behaving in a nondeterministic manner.[36]

Supporting a program as complex as the Fortran compiler was clearly challenging. By 1959, IBM had written the FORTRAN I, II, and 709 Customer Engineering Manual of Instruction, which contained many details about the internal structure of the compiler, the system tape, etc.[37] A copy of this manual was provided by Mark Halpern as part of a large donation to the Computer History Museum. Mark's first assignment after joining the IBM Programming Research Department in 1957 was to study and document (via flow-charts) the Fortran compiler. I got in touch with him after encountering his online memoirs.[38]

The original Fortran system had a variety of built-in library functions and allowed the programmer to write single-statement function statements or to add additional library functions written in assembler, but there were no separately compiled Fortran subroutines or functions. By August 2005, I had run across three versions of a 1957 memo (unsigned, but probably written by Irv Ziller), "Proposed Specifications for FORTRAN II for the 704," dated August 28, September 25, and November 18, that show the evolution of the design of the subroutine feature of FORTRAN II. Around this time, I got in touch with Dennis Hamilton, who had started his career in software right around the time the Fortran II compiler shipped. He had interesting observations about the impact of subroutines:

> [T]he impact of small changes and improvements can be immense. The ability to build Fortran programs out of independently compilable modules and to have the ability to decompose into functions and subroutines using Fortran or any other tool that produced compatible code (usually the assembler, in those days) had an immense impact. In Fortran I, programs were one giant file, and there was no modularization structure. That small change in Fortran II was earthshaking in terms of software development and, I think, the endurance of Fortran as a technical-software programming tool.
>
> It also changed the way that computers had to operate to make software building and use work more smoothly. I think it is no coincidence that this paralleled increased interest in operating systems (called things like tape monitors, at the time) and the use of the computer for organizing the data processing workflows. (There was also a lot of resistance to operating systems in those days.)

In February 2006 I finally obtained scans of the 704 Fortran II listing from the Smithsonian. The proposed intermuseum loan had been blocked because of a misunderstanding about an earlier loan of ENIAC modules to the Computer History Museum's predecessor in Boston. Instead, the Computer History Museum hired a researcher near the Smithsonian to scan the listing.[39]

Running Fortran Under Emulation

As soon as I began my quest for Fortran, I encountered people who were writing emulators.[40] Paul Pierce, who owned a real IBM

709[41] and had digitized the IBSYS tapes, had written an emulator for the IBM 709. Rob Storey wrote utilities to manipulate Pierce's tape images, and then wrote an IBM 7094 emulator. James Fehlinger took the lead in getting Fortran IV to run on Storey's emulator. The Fortran IV compiler does not share code with the IBM 704/709 Fortran/Fortran II compiler, but this was an impressive achievement in its own right.

Next, Dave Pitts wrote a cross-assembler and succeeded in assembling the source code for Pierce's copy of IBSYS into a bootable image. Using a modified version of Pierce's IBM 709 emulator, he was able to run IBSYS and the Fortran II compiler, but the object code generated by that compiler would not run.

Finally, Rich Cornwell succeeded in executing both the Fortran II compiler and the code it generated. Rich used a modified version of Bob Supnik's SimH emulator.[42] It was Rich's enthusiasm that inspired me to finally obtain the scanned copy of The Smithsonian's IBM 709 Fortran II compiler listing. Later, Rich, Fausto Saporito, James Markevitch, Bob Abeles, and Robert Cicconetti transcribed the source code from this listing to an assembly code file,[43] and Rich reassembled it using the SAP assembler running on an emulated IBM 7090 computer.[44]

Reflections

I failed to locate the original Fortran compiler: neither source code images (e.g., cards or magnetic tape) nor an assembly listing. However, I found two different versions of the Fortran II compiler source code, whose code generation and optimization sections were essentially identical to the original compiler. While searching for the source code, I encountered a rich variety of other primary and secondary materials, allowing me to create a website of interest to a broader audience than the dedicated researchers and computer scientists I was originally targeting. I was also fortunate to meet and participate in the retrocomputing community, whose members write and run emulators, recreate source code from poor-quality listings, and pool their expertise in old hardware and software.

This project taught me that patience and persistence are necessary. Old software and related materials are in constant danger of being lost, damaged, or destroyed; the people who created that software are aging and dying. When contacted, people are usually friendly, but they are almost always busy. Multitasking was crucial, along with maintaining notes suggesting who had been contacted, what the response had been and an indication of when or how to follow up. My previous career in software gave me a pool of contacts to help me get started. The traditional literature research combined with Internet search is a synergistic combination. For example, I learned about the special FORTRAN 25th issue of *Annals* from an old USENET post, which led me to Daniel Leeson's article and J.A.N. Lee's annotated bibliography. I was then able to locate and call Leeson by using a search engine. And publicizing my search via the blog and the project website resulted in strangers contacting me with additional information. I have used these techniques with some success on several later projects.[45]

Acknowledgments

John Backus gave me an early taste of computer history by sharing stories of his early days.[46] J.A.N. Lee and Henry S. Tropp's work on the 1992 Pioneer Day and the 1984 special issue of *Annals* celebrating Fortran's 25th anniversary provided a sound foundation for all subsequent work on the history of Fortran. Grady Booch helped awaken an interest in software preservation. Thanks also to all the individuals and institutions named in this paper and the website for answering questions, providing documents, and getting old code to run again. Burt Grad encouraged me to submit this "after action review" to the *Annals*.

References and Notes

1. "Preserve Classic Software," Computer History Museum, 2003, https://web.archive.org/web/20080420173659/http://www.computerhistory.org/PreserveClassicSoftware/.
2. Now the Software Preservation Group; see http://www.computerhistory.org/groups/spg/.
3. The compiler also served as the foundation for research on code optimization: F.E. Allen, "A Technological Review of the FORTRAN I Compiler," *Proc. Nat'l. Computer Conf.*, 1982, pp. 805–809.
4. See, for example, J.A.N. Lee, "An Annotated Bibliography of FORTRAN," *Annals of the History of Computing*, vol. 6, no. 1, 1984, pp. 49–58; doi:10.1109/MAHC.1984.10003.

5. The History of FORTRAN and FORTRAN II website contains design documents, source code, reference manuals, tutorials, papers, films, interviews, user stories, and other materials about this historic project. The Acknowledgements section of the website lists even more people who assisted my collection efforts than could be included in this paper.
6. See http://americanhistory.si.edu/comphist/.
7. I later learned they were in the final stage of mounting a major new exhibit: The Price of Freedom: Americans at War, http://americanhistory.si.edu/exhibitions/price-of-freedom.
8. For example, the compiler turned on a front panel indicator lamp once it determined there were no syntax errors.
9. J.W. Backus, "The History of FORTRAN I, II and III," *Proc. of the First ACM SIGPLAN Conf. on the History of Programming Language*, 1978. Also published in R. Wexelblat, ed., *History of Programming Languages*, ACM Monograph Series, Academic Press, 1981. Reprinted in *Annals of the History of Computing*, vol. 1, no. 1, 1979, pp. 21–37, and vol. 20, no. 4, 1998, pp. 68–78.
10. See http://www.bobbemer.com/FTRANSIT.HTM.
11. See http://www.bobbemer.com/PRORES.HTM.
12. See http://www.piercefuller.com/collect/index.html.
13. Van Vleck is the creator and maintainer of the Multicians website (http://www.multicians.org/), which "presents the story of the Multics operating system for people interested in the system's history."
14. See http://www.columbia.edu/acis/history/; for example, http://www.columbia.edu/acis/history/backus.html.
15. J.A.N. Lee and H.S. Tropp, eds., "Special issue: FORTRAN's Twenty-Fifth Anniversary," *Annals of the History of Computing*, vol. 6, no. 1, 1984.
16. See http://www.softwarepreservation.org/projects/FORTRAN/Backus%20-%20LOC%20-%20catalogue%20of%20papers.pdf (there is now a formal—and less gossipy!—finding aid: John W. Backus papers, 1951–2001, Library of Congress Manuscript Division, Washington, D.C.; http://hdl.loc.gov/loc.mss/eadmss.ms007001).
17. http://www.softwarepreservation.org/projects/FORTRAN/BackusEtAl-Preliminary%20Report-1954.pdf.
18. J.W. Backus, et al. *The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Reference Manual*. Applied Science Division and Programming Research Department, IBM Corp., 1956, http://archive.computerhistory.org/resources/text/Fortran/102649787.05.01.acc.pdf.
19. D.N. Leeson. "IBM FORTRAN Exhibit and Film," *Annals of the History of Computing*, vol. 6, no. 1, 1984, pp. 41–48.
20. Lot X2843.2005, Computer History Museum. See also http://www.softwarepreservation.org/projects/FORTRAN/index.html#Films/video.
21. D. Padua, "The Fortran I compiler," *Computing in Science and Eng.*, vol. 2, no. 1, 2000, pp. 70–75; doi: 10.1109/5992.814661.
22. See F.E. Allen, "The History of Language Processor Technology in IBM," *IBM J. of Research and Development*, vol. 25, no. 5, 1981, pp. 535–548; F.E. Allen, "A Technological Review of the FORTRAN I Compiler," *Proc. Nat'l Computer Conf.*, 1982, pp. 805–809.
23. Sammet has published extensively on programming languages and their history. Her book, *Programming Languages: History and Fundamentals*, Prentice-Hall, Inc., 1969, has been called the definitive work on early computer language development.
24. Lee had successive careers in civil engineering and computer science and has been active in the history of computing for many years, for example, as co-editor of the previously mentioned special issue on FORTRAN's Twenty-Fifth Anniversary, and a previous editor-in-chief of the *Annals*.
25. H.S. Bright, "FORTRAN Comes to Westinghouse-Bettis," *Computers and Automation*, vol. 20, no. 11, 1971, pp. 17–18.
26. See http://www.computer-history.info and http://www.computer-history.info/Page4.dir/pages/IBM.704.dir/index.html.
27. Michael was a computational physicist at Lawrence Livermore National Laboratory (LLNL) who worked in high-performance computing for most of his career.
28. G.A. Michael, "An Interview with Bob Hughes," May 9, 1997, http://www.computer-history.info/Page1.dir/pages/Hughes.html.
29. Backus, et al. *The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Reference Manual*.
30. Ingerman has published a number of books and papers in the area of programming languages and compilers.
31. The earliest post is http://www.mcjones.org/dustydecks/archives/2003/11/19/11/, and the introductory post is http://www.mcjones.org/dustydecks/archives/2004/07/04/2/.
32. See http://insar.stanford.edu/~lharcke/programming/.
33. Anonymous, *Systems Manual for 704 FORTRAN and 709 FORTRAN*, Applied Programming Department, IBM Corp., April 1960, http://archive.computerhistory.org/resources/text/Fortran/102653992.05.01.acc.pdf.
34. See http://www.bitsavers.org/.

35. The current location is http://www.softwarepreservation.org/projects/FORTRAN.
36. J. Van Gardner. "Fortran And The Genesis of Project Intercept," 2005, http://www.softwarepreservation.org/projects/FORTRAN/paper/John%20Van%20Gardner%20-%20Fortran%20And%20The%20Genesis%20Of%20Project%20Intercept.pdf.
37. Anonymous, *FORTRAN I, II, and 709: Customer Engineering Manual of Instruction*, IBM Corp., Form R23-9518-0, February 1959. Mark Halpern papers, Lot number X3762.2007, Computer History Museum, http://www.softwarepreservation.org/projects/FORTRAN/R23-9518-0.pdf.
38. M. Halpern, "On the Heels of the Pioneers—A Memoir of the Not-Quite-Earliest Days of Programming," *Annals of the History of Computing*, vol. 13, no. 1, 1991, pp. 101–111.
39. Catalog number 102660000, Lot number X3435.2006, Computer History Museum. See http://www.computerhistory.org/collections/catalog/102660000 and http://www.softwarepreservation.org/projects/FORTRAN/index.html#Source_code.
40. See http://www.softwarepreservation.org/projects/FORTRAN/index.html#Emulators.
41. Now catalog number 102728984 of the Paul Pierce Collection, Lot X7021.2014, Computer History Museum; see http://www.computerhistory.org/collections/catalog/102728984.
42. B. Supnick, "The Story of SimH," *Annals of the History of Computing*, vol. 37, no. 3, pp. 78–80. See also http://simh.trailing-edge.com/ and https://github.com/simh/simh.
43. See http://www.softwarepreservation.org/projects/FORTRAN/index.html#Source_code.
44. For a detailed description of techniques developed by Markevitch and others to convert a scan of an old listing into an accurate source file, see D. Walden and the IMP Software Guys, "The Arpanet IMP Program: Retrospective and Resurrection," *Annals of the History of Computing*, vol. 36, no. 2, 2014, pp. 28–39.
45. See for example http://www.softwarepreservation.org/projects/LISP, http://www.softwarepreservation.org/projects/ALGOL, and http://www.softwarepreservation.org/projects/c_plus_plus.
46. John Backus died in March 2007, the year of Fortran's 50th anniversary.

Paul McJones *retired in 2009 from a career alternating between software research and product development. In retirement, he has been a Computer History Museum volunteer in the area of software preservation; see http://www.mcjones.org/paul/, which also includes contact information for him.*

CHM Happenings

Dag Spicer
Computer History Museum

Editor: Chigusa Kita

The year 2016 was another banner year for the Computer History Museum (CHM), with record attendance, multiple new exhibits, hundreds of new donations accepted into the Museum's permanent collection, our usual unique and engaging lecture series, and the establishment of two major new research centers.

Nearly 200,000 people, representing over 140 countries, came to CHM in Mountain View in 2016 to learn and explore our main exhibit on the history of computing, *Revolution: The First 2,000 Years of Computer History,* as well our other exhibits on Ada Lovelace, self-driving cars, and the world's smallest computer.

Just as impressive were the Museum's online statistics: more than 2 million unique visitors to the Museum's website, computerhistory.org, with the majority of visitors exploring our online Timeline of Computing History. CHM's YouTube channel, now with over 500 videos on the history of computing, had over 900,000 views and over 48,000 subscribers.

Our major new 10,000 square foot exhibit on the history, impact, and technology of software opens January 28, 2017. Entitled *Make Software: Change the World,* the exhibit features seven focus stories that reflect the range, power, and protean nature of software: Photoshop, World of Warcraft, Texting, MRI, Car Crash Simulation, Wikipedia, and MP3 (see Figures 1 and 2). Each of the seven galleries features an interactive station to engage visitors in hands-on learning and reinforce key concepts, and the exhibit also features a software learning lab for visitors and school groups.

Figure 1. Overview of Make Software: Change the World's seven galleries. (credit: Doug Fairbairn)

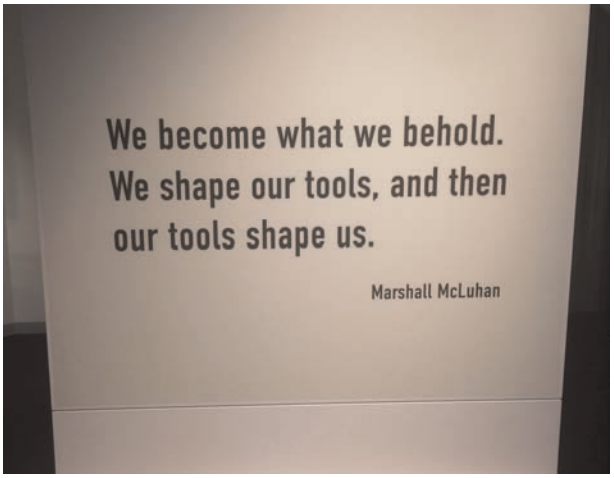

Figure 2. Quote from Marshall McLuhan reminds exhibit visitors of the interplay between software and user.

Research Centers

CHM has dramatically expanded the depth of its research capabilities with the creation of two new research centers. The CHM Exponential Center focuses on entrepreneurialism, the Silicon Valley ecosystem, and the processes through which companies start, grow, and succeed (or fail). In November, for example, the Center held a "Day of the Dead" event to conduct "postmortems" of failed companies. The Center has attracted widespread interest and support, particularly from the Silicon Valley VC community, and is led by Marguerite Gong Hancock, formerly director of the Project on Emerging Companies at Stanford University.

The Center for Software History, led by historian David Brock, is tasked with growing the collection and interpretation of the Museum's software collection (already one of the largest in the world). David and colleague Hansen Hsu have already made a number of highly significant additions of personal papers and software to the collection and are actively engaged in the larger historical community. Both Centers extend a welcome hand to anyone wishing to contribute knowledge, documents, or software, or to conduct research using the Museum's permanent collection. For assistance with any research requests, simply email research@computerhistory.org.

Off the Museum's main lobby, two temporary exhibits also provided visitors with a small glimpse of earlier technologies. *Let ERMA Do It* was about the groundbreaking SRI/GE ERMA system, which fostered and