A Brief System z Assembler History

SHARE 120, Session 12235

John R. Ehrman
ehrman@us.ibm.com

(Contents not guaranteed --- my memory is fallible)

IBM Silicon Valley Lab
555 Bailey Avenue
San Jose, CA 95141

Feb. 4, 2013
Bootstrapping an assembler

1. You have machine A (7094?), need an assembler for machine B (360)
2. Decide on a basic assembler language for B, BAL-B
3. Write a cross assembler ASMX on A for BAL-B in some language X
4. Verify that it generates correct object code OBJ-X for B (key step!)
5. Now write a basic assembler ASMB on A for BAL-B, in BAL-B
6. Assemble it using ASMX, compare its OBJ-B to OBJ-X
   • Fix BAL-B source on ASMX until OBJ's compare correctly
7. A test machine executing B instructions is now available...
8. Load OBJ-B on B, use it to assemble ASMB source
9. Compare OBJ-B from ASMB on machine B to OBJ-B from machine A
   • If there's a mismatch, fix whatever caused it
10. You now have an ASMB$_1$ executing on B:
    You have “bootstrapped” ASMX to ASMB!
11. Now, add features to ASMB$_1$ to create ASMB$_2$ for language BAL-B$_2$
    • Repeat previous step as needed, using ASMB$_n$ to create ASMB$_{n+1}$
Getting started on System/360

- Initial development on existing processors (7090/94, 7030 “Stretch”)
  - Emulated System/360 instructions
    - Very slow, not always consistent across emulators

- Assembler Language definition had to be fixed very early
  - Language was limited by what could (not should) be done

- Cross-assembled object code “bootstrapped” to early 360 processors

- First System/360-based assembler (“BOS”, Basic Operating System) had to run in 10K bytes
  - 4K for code, 3K for buffers, 3K for tables
• Assembler: the primary (only?) internal development language
  – Critical to OS’s and many key products (CICS, IMS, PL/I, Fortran, ...)

• System/360/370/390 assemblers for OS/360 and their descendants
  – E-Level Assembler (IET: 18K), F-Level Assembler (IEU: 44K): 1966
    — DOS/TOS assemblers (IJQ, 1968; IJY, 1966-7)
  – TESTRAN: debugging macros for Assembler Language programs: 1966
    — TESTRAN SVCs still used by many debuggers (TSO TEST, ASMIDF, ...)
  – Assembler XF (IFOX): 1972; DOS/VSE assembler (IPK): 1973
  – High Level Assembler (ASMA): May 1992

• Many university-based student-oriented assemblers:
  – SOS ((Brown), ZAP (Cornell), ESP (Iowa State), ASSIST (Penn State),
    SPASM (Stanford), STASS360 (British Columbia and Michigan),
    SWAP (Ohio State), FASTAM (Texas A&M), FGA (New Mexico Tech),
    TIGER1 (LSU), ASSIST/I (NIU)
  – ... and many assembler textbooks (some were pretty good ...)

© IBM 2013 SHARE 120, San Francisco
Assembler F

- 4 passes: 2 conditional assembly, 2 ordinary assembly

- Considerably faster than ASME
  - Larger memory simplified processing (44K vs. 18K)
  - Some new language features (e.g., OPSYN)

- Free ASMF source was used to build Univ. of Waterloo's Assembler G
  - Used more storage, combined Pass B and Pass 1, added load-to-memory option for student programs
• 4 logical passes; Pass B and Pass 1 merged where possible

• Better performance
• Some language enhancements
• Rewritten publications were very popular
Assembler H

- Started as an internal experiment; became first IBM Program Product

    ![Diagram of Assembler H process]

    - Conditional/Ordinary Assembly

    - Source Program
      - Edit, Generate, Assembly Pass 1
      - Macro/COPY Library
      - Lookahead
      - Symbol Table
      - Ordinary Assembly Pass 2
      - Object, Listing

- Much new function; far faster than all previous assemblers
  - Was used by IBM field staff to generate customer systems
    - SYSGENs took only an afternoon, rather than a weekend
  - Was almost “killed” but rapidly displaced all predecessors
  - The base for the popular “SLAC Mods”
• Supplanted all previous “mainframe” assemblers
• Many powerful new language features and functions
• I/O exits for all input and output files
• SYSADATA file contains all information about the assembly
• Supports *The world's most powerful macro language!*