How to Start Learning Whirlwind

g fedorkow, Mar 15, 2024

This note outlines resources for anyone wishing to learn about software restoration of code for the Whirlwind computer, built at MIT in 1949, active there until 1959. The material starts with background reading, then covers how to run the simulation, and from there, how to learn to read and even write Whirlwind programs.

Format this document with: > $ pandoc –number-sections -f markdown -t pdf How-to-Start-on-Whirlwind.md -o How-to-Start-on-Whirlwind.pdf

Reading Material

For a summary of the Whirlwind project, the book *Project Whirlwind: The History of a Pioneer Computer* by Redmond and Smith is a good starting point. This book primarily focuses on the travails of the Whirlwind team in obtaining and maintaining funding in a fluid post-war environment, as DoD priorities were making a rapid pivot from World World-War-II [1] demobilization to the new Cold War threat.

The work outlined in the document you're now reading is more focused on the technology of Whirlwind software and programming.
A recent summary of the machine's capabilities and the current restoration project can be found in guy's IEEE paper on WW.

> https://www.historia-mollimercium.com/whirlwind/Whirlwind-Software-Recovery-Project-v6d.pdf

(See IEEE Explore for the published version)

The starting point for Whirlwind programmers became a document with the identification number 2M-0277[2], *Whirlwind Programming Manual*, which effectively defines the architecture and instruction set.

> http://www.bitsavers.org/pdf/mit/whirlwind/M-series/2M-0277_Whirlwind_Programming_Manual_Oct58.pdf

You can see other background material, including blog postings, on Guy's web site

> https://www.historia-mollimercium.com/whirlwind/ww.html

[1] Most readers will think that WW-I happened during years 1914-1918, and WW-II was 1936-1942. In actual fact, WW-I (Whirlwind-I) was first operational in 1949, and WW-II, while planned, was never completed as such, but was re-oriented to the SAGE prototype XD-1 in (1955?).

[2] Whirlwind documents are all identified as with a couple letters to identify the Series, then sequential numbers within the series.
Commonly used series are 'M' for Memos, 'R' for reports, 'L' for administrative notes and meeting minutes, 'DCL' for Digital Computer Lab. The system fractured a bit as Lincoln Labs grew, i.e., '2M' probably indicates a memo from Lincoln Division Two.

For more of a look into how the Whirlwind programming system evolved, see the Comprehensive System document M-2539-2

> http://www.bitsavers.org/pdf/mit/whirlwind/M-series/M-2539-2_Comprehensive_System_Manual_Dec55.pdf

To get a sense for the scope of material available from the Whirlwind project, you can browse Bitsavers and the MIT Dome online archive

> http://www.bitsavers.org/pdf/mit/whirlwind/

> https://dome.mit.edu/handle/1721.3/37456

You can see all the raw bits that have been recovered[3] from Whirlwind paper and magnetic tapes at this link:

> http://www.bitsavers.org/bits/MIT/whirlwind/

(I have an ad-hoc tape catalog from the bit-level recovery done by CHM, but haven't figured out how to post it yet)

Running the Simulator

Want to play Blackjack against Whirlwind? You can read the blog article:

> https://computerhistory.org/blog/gambling-on-whirlwind-how-the-us-navy-spent-3-million-and-got-a-computer-game/

You can play the game, as implemented in WW code, using a stripped-down simulator that runs in your browser in javascript:

> https://www.historia-mollimercium.com/whirlwind/whirlwind-simulator

To run the simulator with other programs on your own machine, you can either download a pre-compiled version, or download the python source code to run it locally.

See user-guide notes on the WW simulator, assembler and other tools in Software Tools below.

Pre-Compiled Simulator

You can download the pre-packaged Windows or Mac binaries on GitHub

> https://github.com/gfedorkow/Whirlwind-Instruction-Simulator

Then download and run core files from Code Samples. There's no automated build of the pre-compiled versions, so they won't be "the newest".

[3]Thanks to Al Kossow and Len Shustek for tons of work actually reading the bits off the tapes!

Running the Native Python Simulator

You can also download the source for the simulator and run it on Windows, Mac or Linux.

- Install Python 3.10 or 3.11
- Install a few extra packages:
  - Numpy
  - screeninfo
- Set the Python search path environment var to pick up the Common directory.

Clone the source from:

> https://github.com/gfedorkow/Whirlwind-Instruction-Simulator

Examining a Simple WW program

A good way to learn a bit of how the instruction set actually works is to examine existing code. Here are some examples.

- examine the Whirlwind doc R-196 (link below), with the bouncing ball example on page 53. Note that this code (and document) was written very early in the WW project, prior to the introduction of the redesigned I/O system described in 2M-0277.

  http://www.bitsavers.org/pdf/mit/whirlwind/R-series/R-196_Programming_for_Whirlwind_I_Jun51.pdf

- examine Rainer's Vector Clock. This is a time-of-day clock which displays using the CRT and vector instructions. The program was written from scratch by Guy & Rainer in a slightly-more-modular style than the WW programmers would have used (i.e., it's not a historical artifact in any way)

  https://github.com/gfedorkow/Whirlwind-Instruction-Simulator/blob/master/Code-Samples/Vector-Clock/vector-clock.ww

- examine (what other samples would be good?)

Writing a WW program

- install the simulator (see above)
- read the Assembler notes (written, but not yet posted) (and there are enough options and knobs on the simulator that it probably needs a doc too)
- Start simple, for example, revise the R-196 Bounce program for R > 1
- Write Your Own Digital Clock (I'll provide underlying time-keeping in the sim)

Decoding a WW program

(Guy should provide a note on how to use the simulator flow graph extractor; that's proving to be increasingly valuable in understanding how codes work.)

Starter projects:

- comment the larger Bouncing Ball program
- comment blackjack

Advance Class in understanding WW code

- read M-1343 air defense
- why aren't the resulting tracks Optimal in the simulation?

Advanced Class in restoration - New work

- get a native assembler to work. (look for codes with "Basic Converter" in the name)
- Get Charles Adams' thesis code working ("Intact Ship Stability"). This is not likely to be easy – it was written as part of his thesis project, but completed before there was enough functional Whirlwind to actually run it, i.e., it's 1,000 lines of very carefully reviewed but untested code.

Software Tools

There's a 'user-guide' doc with the tools on Github:

> https://github.com/gfedorkow/Whirlwind-Instruction-Simulator/blob/guy-dev/Documentation/Whirlwind-Development-Environment.pdf

This doc describes the modern assembler-language format, as well as the numerous optional arguments for running the simulator.

Footnotes