

We hereby announce the availability of a SIMH based 3705 emulator capable of running an unmodified IBM NCP.

Project origin and brief summary of the activities:

We (Edwin and Henk) both IBM MVS professionals and fans had a talk in September 2020 about writing an 3705 emulator that could run in a machine separate from Hercules as well as running NCP.

So, why not write such an IBM 3705 emulator?

We got the time (Covid-19 lock-down), documentation and knowledge, let's do it!

That became our goal for the next 11 months.

After some discussion SIMH was selected as running framework. We only had to add multiple thread support to SIMH. The Raspberry Pi was chosen as the target hardware (but any Intel/AMD will work too).

We were lucky in finding via the Internet a copy of the V1.2 NCP macro's and object modules, which are free(!) from IBM Copyright. We installed it in an MVS3.8 environment and were able to successfully generate a small NCP (1 chan adap, 1 scanner, 1 line, 1 PU and some LU's).

We found in the IBM hardware manuals a listing of IBM miniROS for IPL via a T1 channel adaptor. It was OCR-ed and assembled with RC=0.

We started implementing all IBM 3705 instructions as described in the PoO. This manual has a lot of small print which caused some serious headaches.

The comm3705.c in Hercules was almost fully stripped and turned into a message exchange between Hercules (CCW's) and the 3705emu (via TCP/IP). Some of the original code from comm3705.c has been redeployed at the PU/LU side of the 3705 emu.

Next, the 3705 channel Type 1 adapter hardware capabilities were coded. To load the NCP, we restored the original IFLOADRN module in MVS3.8.

After some (understatement) debugging, we were able to load the NCP. However, we learned that TK4 environments had it's 3705's sysgen-ed with Type 2 channel adaptors. Bummer...

So, we started writing a T2 chan adap which is much more complicated than the Type 1. The miniROS was replaced with a maxiROS version which support the T2 chan adap.

After a "V NET,ACT,ID=NCP3705" IBM loads first IFL3705A, which loads the full 3705 diagnostics program IFL3705E into the 3705.

Note: IFL3705E contains 600+ hard stop instructions.

Now we discovered this diag program is intended for an IBM 3705 with 18 bits Extended Addressing (> 64k storage). We had to add EA support to the CPU much earlier than anticipated.

After some (this is really an understatement) debugging the full diagnostics program ran without any hard stop in program level 1 of our 3705 emu.

Next, the interrupt handler and program run levels 2, 3, 4 & 5 were added to the CPU.

We were now able to load the small genned NCP into the 3705 emu.

The 3705 can be equipped with 3 different types of communication scanners:

- Type 1 - Needs a CPU interrupt after every transmitted or received bit.
- Type 2 - CPU interrupt after every Tx or Rx byte.
- Type 3 - Interrupt after a whole frame.

Type 1 is for slow speed lines and generates a lot of 3705 CPU overhead. The type 3 is much too complex, so we started coding a type 2 scanner. Type 2 has 11 modes of operations. Currently only the SDLC half duplex mode is implemented.

An SDLC line set simulator was developed. Automatically stuffing a 0 after five consecutive 1's in the data, is not implemented. Too complex and creates too much overhead. Our solution is very simple, it is 100% transparent as long as the data stream doesn't contain X'470F7E'. An FCS (aka CRC) is not calculated.

A SNA PU2 simulator was coded which supports currently the minimal required SNA commands. (ACTPU, ACTLU, BIND, CLEAR, SDT, UNBIND, INITSELF)

TN3270 server support added. Copied and adapted from the original comm3705.c.

Multiple channel support was added to the channel adapter code; however, we are unable to test this as we need NCP v1.3 or higher. (Anybody?)

End result:

We are able to run a SNA network with a real NCP and able to logon with tn3270 to TSO through a running NCP.
We consider this stage as our beta version.

To do:

- Multiple LU's per PU support.
- Running an NCP with Extended Addressing enabled. (NCP V1 supports max 256k)
- Multiple online hosts (we need NCP V1.3 or higher for this. Anybody?)
- Solving some last bugs, including some performance issues.
- Docs!!!
- General code cleanup before we make it general available to the H390 community.

Future:

- Coding a 3705 emu with a 'real' remote NCP is our next goal.
For this we are still searching for a ROS listing from a remote IBM 3705. Anybody?

We want to thank Mark Waterbury for his great support and advice in this project.

Edwin Freekenhorst and Henk Stegeman