Hacking the System/3

System/3 was from a user perspective a very well designed system with great stability.

During my eight and a half year experience I only recall a single bug!

That has changed a lot since the PC and especially Windows have done its part!

Of cause the functionality was limited compared to today's systems – but we didn't know that we were missing subdirectories and wildcard filenames etc.

So "hacking" was not the name of the game at that time. But as always you could do some "personalization".

In the beginning the RPG source code was kept on the 96-column punched cards but after moving to the 3740 diskette it was customary to use the source library on the disk to store the programs and only copy it to a diskette for modification before copying it back again before compilation.

When using CCP (the online monitor equivalent to CICS) on the model 12 running in DPF-mode (Dual Program Feature) the problem was that ÅMAINT, which was used for the copying could only be run in "dedicated"-mode. The same was true for the model 8 were your could not interrupt a running program ("roll-out") and copy the source to diskette before continuing the original program ("roll-in").

I found out that it was only a matter of a single bit in the attribute of the ÅMAINT-program in the object-library. So the solution was to copy the ÅMAINT-program to diskette, change the attribute and then copy the program back with a new name. As far as I remember you got at checksum error, but it was possible to ignore that.

My boss at the last System/3 installation where I was engaged had still contact with his former company and I did the modification there for the price of a bottle of wine and I think my boss also got one as "commission". So the price of one bit was one bottle of wine!

= = = o O o = = =

Later when running a model 15D with CCP, we bought a FDP (Field Developed Program) which could copy the source from the source library to a direct-file (relative-file in Cobol-terminology) where it could be modified by a (primitive) full-screen editor on the 3278 display before being copied back to the library. Another FDP-program could submit jobs from a display by the way of $QCOPY which copied the OCL to the reader-queue from where the system picked it up and executed the job.

The problem was that you had to have a reader device defined at sysgen-time when you enabled input-spooling. This device was allocated at IPL-time and could not be used by the system as a "normal" device, which was what we want. Of course if you used the 3741 as a reader it could still be used as offline data-entry station.

My idea was to define a "dummy" MFCU on the system and use that as a reader device for input-spooling. I found out that you could patch the configuration record on the system-disk to cheat the system to think it had a MFCU. But that was not enough you also had to have the device-code for the MFCU otherwise you got an error at IPL-time. So when we were at the IBM Data Center to perform some other job, I copied the $$MFxxx routines to a diskette and when I came home and copied the routines to our system it worked as hoped.

= = = o O o = = =

Even at that time programming errors was invented! But it was often harder to correct the wrong data, as your had no SQL or general purpose data file manipulation utilities, you often had to make a special (one-time) program to correct the data.

We had another FDP-program called LOOK, which could find data in an indexed file (probably by key – I am not sure) and display it on the 3278 screen. As an extra bonus it also displayed the physical disk-address (Cylinder, Head and Sector – Offset?). With that information it was possible to use the $PATCH-program normally used by the IBM CE (Customer Engineer) to apply PFTs (Program Temporary Fix'es). As far as I remember you specified the disk-address, the old value, the new value and maybe a checksum (where you where able to ignore the checksum-error). Of course it was a bit dangerous (like playing with fire) as I remember one time we entered the disk-address wrong and thereby changing a byte somewhere on the disk without knowing what was there before so we couldn't change it back and we had no idea of what the consequences could be. But luckily we never experienced any problem!