Would you trust your nuclear missiles to a floppy disk?

May 27 2016, by Rob Miles

Credit: George Chernilevsky

What can you do with a floppy disk these days? Apple retired floppy drives with its original 1998 iMac, PC manufacturer Dell followed suit in 2003 and now they are but a distant memory. But, according to a recent US government public accounts committee report, legacy computer systems using floppy disks are still used today for controlling part of the US Nuclear Command.

In the old days a "legacy" was something valuable that you might be left when a great aunt passed away. In the world of computers, they are a lot less fun: a legacy computer system is one that is old – obsolete even – but one which you are stuck with, often simply because it does the job and replacing it is too complex, too difficult, too expensive, or all three.

You might think such systems are rare in this modern day, but in fact whenever you buy something on a credit or debit card, conduct banking transactions, or take a flight, among other things, computer systems dating back many decades are an integral part of the process. Banks, air traffic control systems and many core civil or military functions rely on technology that is well past its sell-by date.

To put this into context, the system in question, the Pentagon's [Strategic Automated Command and Control System](), is part of a 53-year old computer network built around the [IBM Series/1 minicomputer](). Priced at between US$10,000-100,0000 when launched in 1976, it has around the same power as today's [Arduino microcontroller]() boards for home-brew projects that can be bought for a few pounds. The huge 8" floppy disks hold a 500 kilobytes (0.5 megabytes) of data. You'd need around four of these disks to store just [one average MP3 music file]().

But amazing though it something as antiquated as this Series/1 minicomputer sounds, the [report]() indicates that it alive and well in the bowels of the Pentagon to "handle functions related to intercontinental ballistic missiles, nuclear bombers and tanker support aircraft". One hopes it is still working well and that there is a cupboard full of spare System 1 boards, chips and power supplies to keep it running.

But legacy systems are not just about hardware. There is a load of legacy software still in use today too. For example, the programming language COBOL was invented in the 1950s and has been used to create many of the systems that underpin modern business. That's also true of

FORTRAN, a language which has been a mainstay of scientific programming for over 60 years. In fact, if I was giving careers advice to a young developer hoping to make their mark, I'd be strongly inclined to advise them to spend some time with these dinosaurs of computing – because as specialists of yesteryear grow old and retire while the systems they created and maintained show no sign of doing so, the expertise to keep them running is drying up and skills are in high demand.

In for the long haul

Having said that, I'm quite relaxed about the age of these systems. Software that has been behaving itself for 40 years is probably going to keep working. So long as we can keep the hardware going as well, there's no reason to suppose it couldn't continue for another 40. I'm a big fan of computer systems designed to be small and simple, with less to go wrong: modern hardware and software is capable of astonishing things, but the complexity and interdependence found in today's systems can bring horrendous problems, as we saw recently when one developer managed to [break thousands of programs in a fit of pique](link).

Why rush to replace something that works with another that may bring a whole new world of problems? Software doesn't wear out, and so long as the context in which it is being used and the requirements we have of it don't change, there is no pressing need to change it. As Pentagon spokeswoman Lt Col Valerie Henderson told AFP news agency: "This system remains in use because, in short, it still works."

I think the ultimate solution to our legacy computing problems lies in the way that computers are extremely good at behaving like other computers. It turns out to be comparatively easy to write a program that can behave like, or emulate, another computer processor. Fans of more commonplace obsolete computers such as the Sinclair Spectrum, Commodore 64 or BBC Micro can get their retro gaming fix via the

huge number of emulators that can run software for old computers on current PCs or even smartphones.

All we need to do is create a software emulation of the IBM Series/1 that runs on off-the-shelf PC hardware and all that complex, messy and probably un-recreatable programming designed for an obsolete system can run on into the future, code that becomes almost immortal.

This story is published courtesy of [The Conversation](#) *(under Creative Commons-Attribution/No derivatives).*

Source: The Conversation