Enhancing VICE

How to break things when fixing them

The problem of container formats for emulation and what communities will make out of them

by Christian Bartsch
The Software Preservation Society (Softpres/SPS)
http://www.softpres.org

Rev. 1.1 2014-07-25

For decades the field of emulation has been the realm of enthusiasts and hobbyists. Academia, museums, institutes and archives followed more slowly realising that in some cases emulation might be the only chance left to use or access old computer software. Consumers usually don't care about how things are implemented. To them it makes no difference if code is written well or wastes resources as long as the result is within limits. The average consumer does not need (usually also does not want) to know, how things work internally. To them it is important that things work to their satisfaction.

Digital containers for an analogue source media, like a record or a tape, only contain sampled representation of the information present on the source. Due to the differences of both worlds, it is not possible to store the analogue original in the digital domain. All is left is a digital approximation of the source, in many cases sampled at such high resolution that the loss of infinity present in the analogue world can be safely ignored. Nyquist–Shannon's sampling theorem[1] clearly defines that regarding audio, a sampling frequency of at least double the maximum frequency to be played back is sufficient. Today, engineers usually chose a multiple of such frequency, e.g. 192KHz at 24 bits resolution, allowing for a maximum frequency of 96KHz to be reproduced where the human ear already stops at about 16 KHz for an average adult. So even audio stored at resolutions of 96KHz at 24bits can for sure be regarded as fully preserved.

When it comes to digital media, many people seem to ignore the fact that digital recording[2] is based on analogue recording – at least for all recording media invented in the last century. Computer tapes, floppy disks and even hard disks can not store digital information (a continuous stream of zeroes and ones) on the lowest level, instead data is modulated with a recording scheme and then transferred to a by default analogue recording process. Because it is analogue, there are several restrictions that apply. This makes it impossible to store data directly. One reason for this is that an analogue medium can only record AC. Only AC generates an alternating magnetic field which in return will again generate AC when the recording material passes the read head and induces a current. If the user was to save a 100kB file filled with zeroes, with a binary pattern corresponding to only zeroes as well, this signal would result in DC being recorded, which won't generate any signal when read back.

1 http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem
2 http://en.wikipedia.org/wiki/Magnetic_storage

Depending on the tape (or disk) speed used there is a minimum and maximum frequency of the magnetic field changing to be obeyed. These restrictions enforce a recording scheme which will make sure that the requirements of the underlying analogue material are being met. With FM (frequency modulation) regular clock bits generate a defined change in polarity, which not only fulfills the demands of the analogue world, but also generates a reference clock to which the reading system can adjust to. Because of this, digital information is stored as polarity changes, so called flux changes[3], which happen for data, as well as for the clock bits.

It is possible to violate certain recording rules on purpose, although too much tampering also means trading reliability for other benefits, like enhancing data capacity.

When people started transferring old media, they often chose to store a complete medium in a digital container with the analogue foundation usually ignored. This is in fact not a problem as long as all of the data to be expected is well known, like with disks that use a defined formatting as provided by the operating system, e.g. Disk Operating System (DOS) disks. Images produced from such disks are usually stored as what is called sector dumps (ADF, IMG, D64 et al), referring to the logical addressing used by the operating system. This means that such a dump lacks the physical transport layer and only features the plain data as returned by reading such a disk, usually through an OS call. Such disks can be used in emulation like real disks, but lack a few features, like formatting the disk, which an emulator usually translates to emptying the corresponding disk image. The real formatting, actually waiting for e.g. an index pulse, as provided by the drive and then starting recording, does not work as the index, or the relation of the data to the index, can not be stored in such an image.

To avoid software theft, various forms of copy protection evolved since the 1970s, in many cases tied to the physical medium. While some protections only deal with varying parameters of the formatting to make it inaccessible to the OS (like changing sector sizes from 512 to 1024 bytes or the like), others completely take total control of the recording scheme and thus define their own coding.

Any form of aforementioned disk based protection can not be stored in a standard sector dump. Hence additional file formats needed to be conceived. Such evolution has happened on nearly every computer platform. The Amiga community got Extended ADF, for the C64 community G64 was developed in the late 90s. These formats have their shortcomings too. Looking at the medium the lowest level of data storage is flux changes. A format made for storing whatever information is present on the media would store such flux changes. This is why our floppy controller made for preservation, KryoFlux, exactly delivers this and our own stream file format saves the data as it comes off the disk's surface. However such technology was not available to the public over a decade ago, when the container formats in question were developed. All that was available was data read through the original devices or with specialised but less capable controller cards for the PC, which were not able to store multiple consecutive revolutions of raw data in the internal RAM. Standard PC controllers are also unable to ingest such data uninterpreted as the standard floppy controller built into many PCs utilises a Western Digital WD1771 chip which reads MFM[4] and, depending on assembly, also FM. Since it only supports two coding schemes and because it automatically discards clock bits (discarding 50% of data read before it even reaches the CPU), it does not serve the purpose of reading foreign formats, like C64 disks (which use GCR[5]) and is completely out of scope for doing preservation.

3 http://en.wikipedia.org/wiki/Floppy_disk#Operation
4 http://en.wikipedia.org/wiki/Modified_Frequency_Modulation
5 http://en.wikipedia.org/wiki/Group_Code_Recording

In the case of C64 disks, most of them were read through the floppy drive supplied by Commodore, the 1541[6]. The device is a small computer itself, with a processor similar to the one built into the C64. Because RAM was very expensive at the time, it only features 2kB of RAM, which makes it impossible to store a complete revolution of data in the device. And since data transmission to the host computer via the serial connection is slow, Commodore never invested into looking at streaming data to the host in real time. The engineers made the device a pre-processor instead which would perform the task of reading a certain sector from disk, decoding it, and only storing the decoded data in RAM. This is achieved with clever but primitive circuits, which will detect e.g. sync, a marker defining the start of a sector, and a GCR table with values stored in a ROM to compare against. The result is that the original flux changes are already discarded at the hardware level, at least most of it and especially if going through the drive's operating system. Modifications, like a RAM expansion or a parallel cable, will help getting more data or complete revolutions of data, but it's impossible to get data without some pre-processing applied in hardware, like the shift register. Another drawback is the missing index signal, which in replication is used for alignment of consecutive tracks.

The G64 format therefore stores data as it is seen by the 1541, not how it is present on the disk. Eight bits of data delivered by the drive are combined into one byte, which leads to some special restrictions, e.g. regarding the alignment of data, as raw data does not always start at the exact same position (or at a byte boundary). Since information in the image is only stored as bytes, the original density information (the exact time it takes from bit to bit) gets lost, so the only way to re-create average density based information is to know the rotational speed (usually 300rpm) and then divide it through the amount of bits per track.

The density definition built into the format back in 1998 is unusable due to the fact that it only defines densities that can be set by a 1541 when writing, but does not leave room to define densities as they are read, which can be different if written with a different device, e.g. a TRACE machine, a mastering device used to replicate floppy disks throughout the 1980s and '90s.

Another problem with G64 lies in the way the format is implemented in different emulators. Due to the fact that most emulators were conceived with D64 in mind (which was mostly based on second hand data – cracked software) and G64 added as an afterthought, many emulators were not prepared for cycle exact emulation of the 1541's inner workings. This lead to severely different interpretation of G64 files in emulators, making images emulator dependent and thus hard to interchange. To complicate things, many images needed to be hacked to work in certain emulators at all (there are even instructions available[7] for certain games and how to modify them), thus leading to even more versions of a simple dump being spread.

This means that as of today there is already a vast amount of images in circulation that are modified, not true to the source and only work with a certain emulator or a certain version of it. These images are not suitable for preservation as they have been tampered with and no one knows what has been patched, if these patches are good and they of course do not present what was present on the original disk.

In early 2012, István Fábián and Robert McIntyre of The Software Preservation Society decided to end this dilemma. The story preceding these events was intense C64 dumping and reverse engineering of classic games to create proper preservation containers for long term storage. It turned out that “true” G64 files (=identical to the data read from disk instead of being modified by the 1541) generated from disks for testing (G64 itself is only sufficient for an estimated 95% of original disks), this time read with the KryoFlux device, were not working in most of

6 http://en.wikipedia.org/wiki/Commodore_1541

7 http://c64preservation.com/dp.php?pg=bangui6

the emulators around. Differences in implementation as well as emulation faults led to unpredictable results.

To create a working emulation environment, VICE[8] was chosen for its licence (GPL), which would mean full access to source as well as the ability to contribute changes back into the codebase[9]. The VICE emulation model for the 1541 works with a 1MHz master clock (CPU clock), whereas the original hardware operates at a resolution of 16MHz (inner workings of discrete, analogue, logic on the board, not the operating frequency of the CPU built into the drive). Since several protections make use of it, physical layer emulation needed to be extended by generating fake sub-cycles, up to 16 per one CPU cycle. Another major change was adding full “half-track” support. The 1541 uses an 80 track mechanism, but a head made for disks with 40 tracks only. This is why the 1541 double steps and only addresses 40 (OS only 35) of 80 possible tracks. Programmers dealing with the floppy directly and bypassing the OS can chose between odd and even tracks, or mix, as appropriate. Until addressed by the changes, VICE would simply source every second track (if accessed by only stepping once instead of twice) from the preceding one.

Because of the above, fixing and enhancing VICE also meant breaking support for many old, bad or modified, images. As it turns out, compromises made and shortcuts taken earlier lead to a vast amount of dumps that are known to be modified or inferior to the original. Such dumps create a dangerous reality: Such titles might be considered preserved while they are not and search for or acquisition of titles might be suspended or stopped.

As a conclusion, here are three rules that will help keeping access to whatever assets you might have:

a) Always store the original dump data, regardless of how good your preservation container for representation might be. At Softpres, every single dump processed to produce an IPF file since 2000 has been archived. Even though we haven't found a single bad image until today, we can always go back to the magnetic information stored on the disk in the first place and check if something went wrong.

b) Make sure to have a preservation container which can store all data, also meta data, as needed. If the data does not fit in there, do not modify data. Instead, extend the format to properly hold whatever information needs to be stored. This is why SPS developed he Interchangeable Preservation Format (IPF) which uses scripted data, which is proven to be authentic and free of errors. Versioning is a must.

c) Always keep various emulator versions used, especially if images were intentionally (or unintentionally) modified to make them work with that very version of an emulator. The latest release, even though it might be better, might break support to images tampered with to work around shortcomings in earlier versions. For emulators that are not open source, always check OS requirements and make sure to keep a copy to be able to also emulate the host OS, just in case. Even slow and stuttering double emulation will be better than having nothing.

8 http://sourceforge.net/projects/vice-emu/

9 http://vice-emu.svn.sourceforge.net/viewvc/vice-emu/trunk/vice/src/drive/rotation.c?revision=25988&view=markup
SPS changes documented starting at line 1120 (22.07.2012)

About The Software Preservation Society: SPS, or Softpres, formerly the Classic Amiga Preservation Society (CAPS), is a privately funded association of art collectors and computer enthusiasts striving for the preservation of computer art, namely computer games. Unlike games from the 1970s (delivered on solid state ROM-modules) and games from and after the mid-1990s (delivered on optical media like CD-ROMs and DVDs which are supposed to last for decades), computer games from the 1980s and early 1990s were delivered on magnetic media like tapes or floppy disks and are now on the brink of extinction. Founded by computer expert and preservation pioneer István Fábián in 2001, SPS already has digitally archived about 8,000 games and applications produced for the Commodore Amiga, Atari ST, CPC and Spectrum to name just a few. SPS is working closely with other associations like museums and libraries around the globe to preserve the early years of gaming history. The technology and tools developed by SPS, like the Interchangeable Preservation Format (IPF) or the custom made floppy controller KryoFlux, are unparalleled when it comes to forensic inspection and long term storage of magnetic media.

For more information:

www.softpres.org
www.facebook.com/softpres
www.softpres.com/glossary:kryoflux
www.kryoflux.com

Contact:

Softpres UK
Kieron Wilkinson
80 Allington Way
Maidstone, Kent
ME16 0HN
United Kingdom

email: kieron@softpres.org

KryoFlux Germany
Christian Bartsch
Auf dem Blick 26
48329 Havixbeck
Germany

email: cb@kryoflux.com