Digital forensics on a shoestring:
a case study from the University of Victoria

John Durno & Jerry Trofimchuk
2014.12.19

INTRODUCTION

Back in 2012, Marshall Breeding wrote:

"It's [..]interesting to consider the ways that technology has changed the challenges that libraries face as they accept the collections from retiring scholars or literary figures. In times past, such a collection sold or donated to a library or archive would consist of books and boxes of notes, manuscripts, correspondence, and other physical artifacts. Today, retiring scholars might also include their word processing files, research data sets, or even the computers themselves. The files might reside on media long obsolete. How many libraries, for example, still have equipment that can read 5.25" floppy diskettes that were standard in the early days of computers and word processors? Or magnetic tapes from a 1970s vintage mainframe? Processing such collections may involve increasing measures of digital archaeology."

Breeding's observation had particular resonance for us because shortly prior to its publication our University Archives had received a donation of physical materials with inventory lists on 5.25" floppies. This was not the first time donations had contained such materials, but so far all we had been able to do with them was to store them along with the other physical materials in the collection. Of course, this strategy was not optimal, both for the reason mentioned by Breeding and also because such media degrade over time. The longer we waited to attempt to extract their contents the less likely we would be able to do so. We decided the time had come to try.

As an obvious prerequisite, we would need to obtain equipment that could do the job. At one time, of course, our library had equipment capable of reading old media. But that was back when the media were not yet obsolete. Storage space being at a premium, over the years we conscientiously discarded older computers to make room for their replacements, and the last 5.25" floppy drive had long since left the building.

One possibility was of course to buy a complete older system from a vendor specializing in such things, but we were reluctant to go that route for two reasons:

1. Vintage equipment tends to be expensive, particularly if it has been refurbished.

2. Even with refurbished equipment there would be no guarantee how long it would continue to function. Mechanical parts wear out, capacitors burst, and rust never sleeps.

3. While there are off-the-shelf digital forensics solutions, such as the Forensic Recovery of Evidence Device (FRED) from Digital Intelligence, these solutions appeared to be overbuilt for our requirements, and with a correspondingly high price tag. Our hardware budget for this project was $2000, and we did not exceed it.

As a corollary to #2, it did occur to us that we will probably not require this kind of equipment to last forever. Eventually there will be no more donations of 5.25" and even

3.5" floppies, as the members of the generations that used such things join Eliot's dancers under the hill. And of course it is questionable whether such media will still be readable 20 years further down the line, even if we still have the equipment to do it.

Bearing that in mind, we decided to aim for a 20 to 25 year operational window. What could we do now, if anything, to ensure we maintained the capacity to read old floppy disks for the next quarter century? It was at this point we decided to look into how possible it was to build an older computer out of new parts. We did some research, concluded it was feasible, and obtained a grant from the Canadian Association of Research Libraries (CARL) to underwrite the cost of hardware purchase. The rest of this paper describes how we approached the problem, the design considerations as applied to hardware, software and the places where they intersect, and some of the preliminary outcomes.

HARDWARE

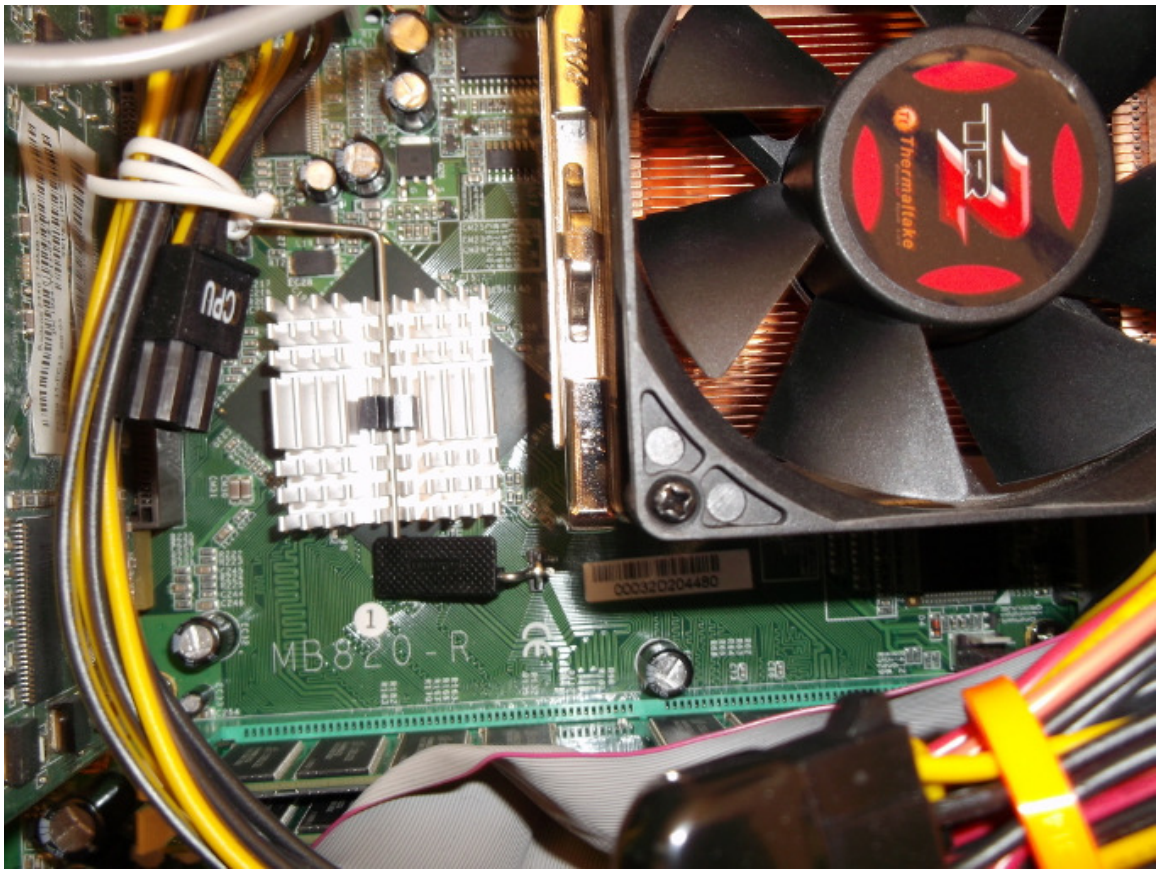
Of course, anyone at all conversant with the field of digital forensics could legitimately ask why we did not simply do what everyone else does: use a modern computer equipped with a USB floppy controller. (For 5.25" floppies, the "go-to" solution appears to be the FC5025 floppy controller from Device Side Data, or at the higher end the Kryoflux) (Reside, 2012). It's a good question, and before enumerating our reasons I will note that in fact we did build a second workstation with the FC5025, so we have a benchmark for comparison. For the rest of this paper, we will refer to these two computers as the "New/Old" computer (old computer built with new parts) and the "USB Floppy" computer (workstation with USB floppy controller).

Reasons why one might want to consider replicating an older hardware platform include the following:

1. Even if the FC5025 controller solves the problem of reading 5.25" disks very well (and early indications are that it does), it does not address the problem of reading 3.5" disks. External USB-attached 3.5" floppy drives are still commercially available, but there are reports these have more problems reading older media than do direct attached devices. Many can only read high density floppies, not the earlier double density variety (Reside, 2012).

2. Creating disk images is of course only the first step along the way toward preserving content, which is of course what we care about. You need to be able to mount the disk image or otherwise access the files it contains, and run a variety of older software and operating systems in order to be able to view documents or run older executables. While emulation and format conversion are both possible and necessary (as it would be unrealistic to attempt to duplicate every possible hardware configuration you might encounter), that approach does have its limits. Having the ability to hardware-boot older operating systems provides additional flexibility for extracting content from old media.

At least, that was the rationale we started with. We should stress at the outset that we are relative neophytes when it comes to recovering data from old computer media, and were even more so when this project got underway in 2012. Jerry is a Library Systems technician and hardware hacker with many years of experience; John is a Librarian and self-taught jack of all trades who manages the Library Systems department. Our goal here is to describe how we approached the problem in the hopes that it will have some value to others involved in the field, or anyone wanting to replicate the functionality of older hardware using new components, possibly for reasons other than simply reading data off of older media.



IBASE MB820 Motherboard, CPU and Fan

The core component of our hardware strategy was an IBASE MB820 industrial motherboard, readily available from multiple suppliers. Industrial motherboards differ from their consumer counterparts in that they are designed to support operational continuity for specialized software frequently found in industrial applications. Typically, upgrading industrial software simply to keep it compatible with newer computer hardware is not worth the expense. It is more economical to keep the hardware environment constant for as long as possible. Thus the MB820 has some interesting and valuable characteristics from our perspective. [2]

The MB820 motherboard supports an Intel Pentium 4 processor, giving it the equivalent processing capacity of a computer from the early 2000s. In terms of supported components, it straddles the space that lies between older legacy devices such as 5.25" floppy drives, and PS/2 keyboard and mouse, as well as more recent devices such as SATA hard drives, an important consideration as IDE hard drives are becoming difficult to purchase. [3]

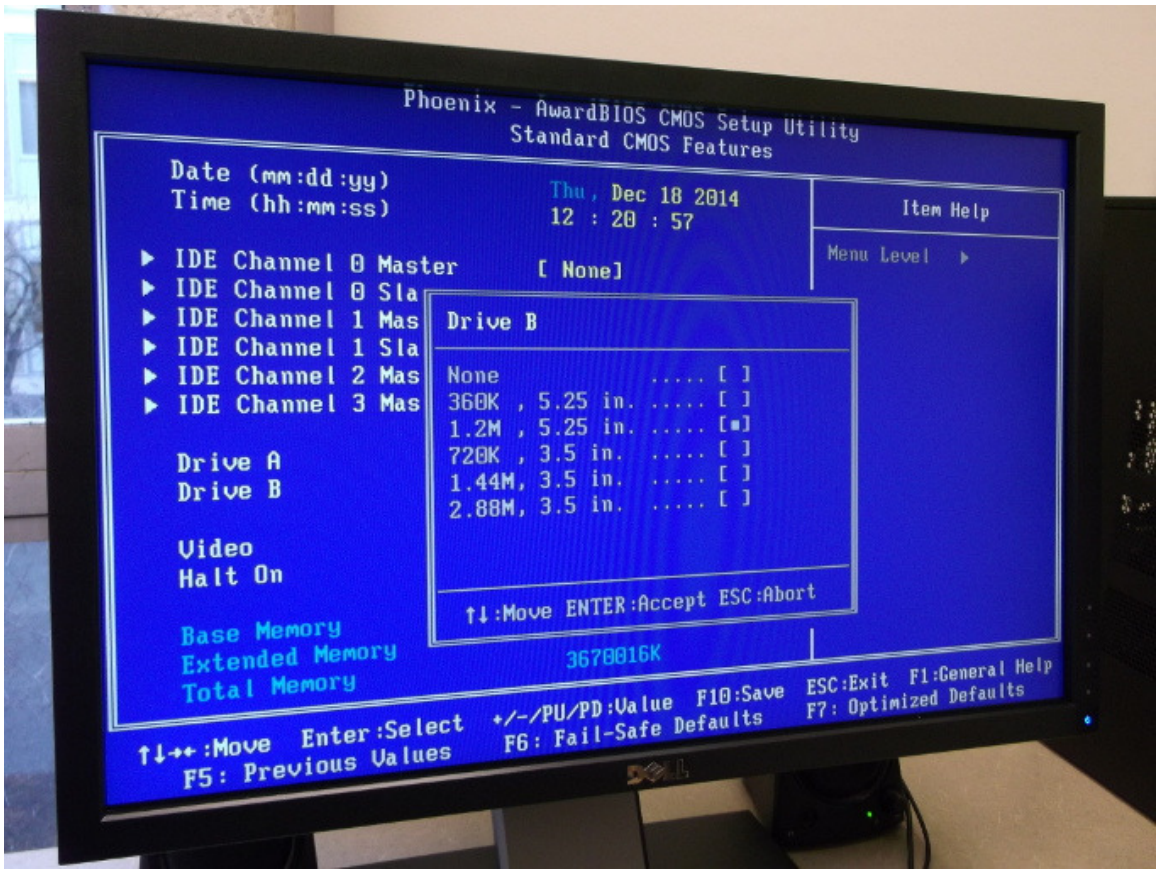
With the MB820 as our foundation, we were able to construct a system with the following components:

- Intel Pentium 4, 2.8 GHz processor with 512k L2 cache
- 4GB DDR memory
- 500GB SATA Seagate hard drive
- ATI Radeon 9200 AGP Graphics card
- Creative Labs SoundBlaster AWE 64 sound card (useful for audio in DOS)
- 5.25" floppy drive
- 3.5" floppy drive
- CDR/DVD IDE optical drive
- Logitech PS/2 keyboard
- Hewlett Packard PS/2 mouse

These components together allow us to run a rich cross-section of legacy software and operating systems, including MS-DOS and Windows 3.11, as well as more modern operating systems such as Windows XP Pro and CentOS Linux 5.10, to take advantage of the different capabilities of each. For example, the motherboard supports both USB and PS/2 input devices, which is useful given that DOS does not readily support USB. And of course, on board support for floppy drives obviates the need to utilize USB controllers, making it feasible to directly access the disk drives from within DOS, should we ever want to.

As mentioned above, the goal of the project was to build an older system out of new parts. We were mostly but not completely successful. The floppy drives themselves (both 3.5” and 5.25”) were purchased secondhand, as were the sound and graphics cards. We were not successful in locating suppliers for new iterations of these older components, and secondhand components were readily and cheaply available.

Buying spares of key components was already central to achieving our goal of a 25 year lifespan. Given that the used parts are less likely to survive for an extended period than were the newer components, we bought additional spares for these in case it proved difficult or expensive to procure replacements in future.



BIOS settings showing floppy drive capacities

SOFTWARE ENVIRONMENT

As noted above, our "New/Old" computer can boot multiple operating systems. However, for day to day use we find the CentOS 5.x (currently 5.11) partition to be the most flexible. We chose CentOS 5.11 because it supports both our 32-bit hardware and the CERT Forensics Tools package, which contains most of the utilities we need to acquire and examine disk images.[4] CentOS 6.x also comes in a 32-bit flavor but the CERT package does not appear to be available in a 32-bit version past CentOS 6.4, which meant we would not have been able to run updates going forward had we opted for the newer version of CentOS. Of course, CentOS 5.x updates will cease in 2017, at which point we will have to decide whether to simply freeze the machine (and block external network access for security reasons) or upgrade the OS.

It is worth noting that CentOS 5.x automatically saw and configured support for both floppy drives during the installation process. This was both unexpected and highly gratifying, as we had anticipated a certain amount of effort would be required to compile the appropriate drivers into the kernel.

Our "USB Floppy" computer runs Windows 7 Enterprise, with BitCurator running in Virtual Box.[5] However, as noted below, BitCurator is not currently operational in our environment.

DISK IMAGING

Imaging floppies is an important first step toward their preservation. Floppy disks were never the most durable of media, and older floppies in particular need to be handled extremely carefully. At the same time, it is not enough simply to extract their contents, since the goal is to preserve not only the individual files but also how they were organized, timestamps, and any additional metadata that might have been part of the original file system. Ideally as exact a copy as possible should be created with the least amount of handling of the original disk. Disk imaging addresses both those requirements: first a bit-level copy of the disk is created (hopefully in a single read) and any subsequent content extraction is done from a copy of the image, not the original disk (Woods & Lee, 2012)

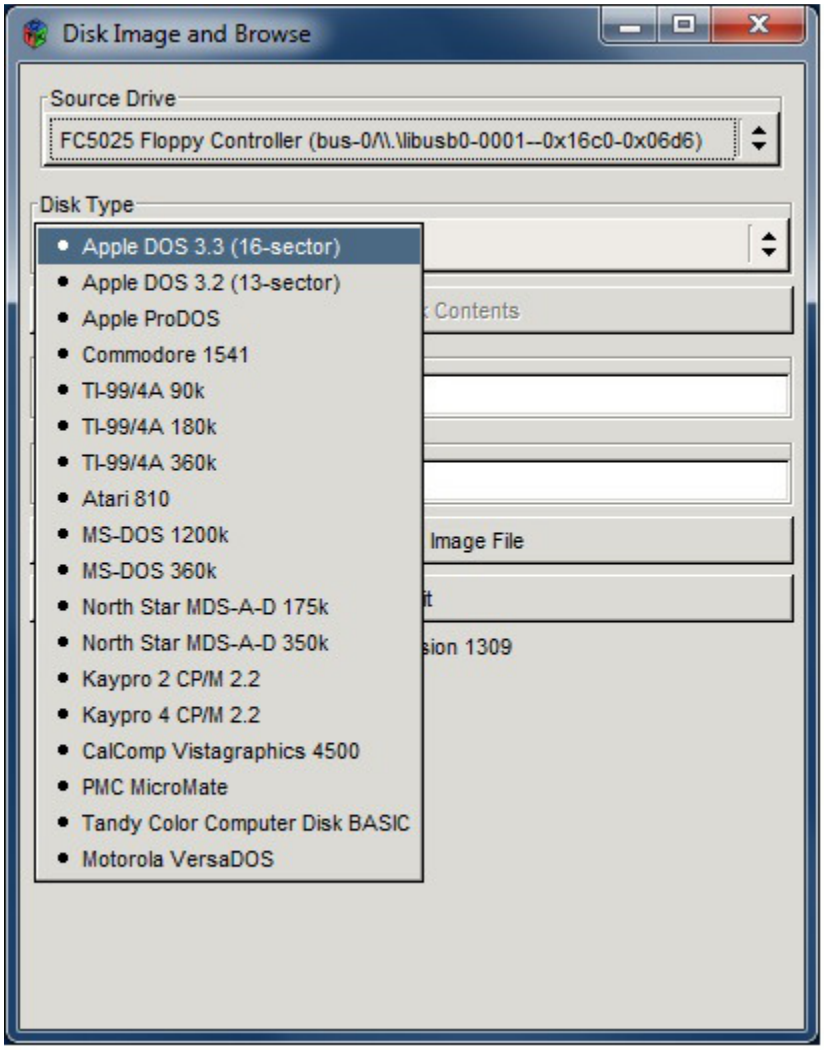
However, it turns out acquiring disk images is not altogether straightforward, even after you have the necessary hardware. A variety of imaging software is available, and images can be saved in multiple formats. What's the best approach?

When we consulted the literature it appeared that the best practices for archival disk imaging had been derived from the field of digital forensics, the reason being that both archives and law enforcement have to credibly establish that their images are accurate and complete copies of the original (Duranti, 2009 & Fox, 2011). However, it is still possible to question how far best practices developed in one field should extend to the other, particularly as regards file formats for long term preservation.

The simplest image format is a raw, bit-level copy of the original source media. This format is output by utilities like the venerable dd command, which has existed in one form or another since the 1970s and is included on every POSIX-compliant unix system. However there seems to be some consensus in the literature that the Advanced Forensics Format (AFF) is preferred, largely due to its ability to store user-created metadata in the same file as the disk image, along with other features like digital signatures and more efficient storage capabilities. (Garfinkel et. al., 2006) But in the context of long-term preservation, and particularly in the context of the minimal storage requirements presented by floppy disk images, the use of a more complex and evolving format like AFF seems to us to present at least as many drawbacks as advantages. Our confidence was not increased when we determined that earlier versions of AFF have since been deprecated for future use (Forensics Wiki, n. d.). We believe raw images are more likely to be readable over the long term than are more specialized formats.

In addition to dd, we are exploring the use of other tools for imaging disks. One of these is a proprietary program that ships with Device Side Data's FC50525 USB floppy controller. For imaging 5.25" floppy disks this is turning out to be our best option, both

for its overall ease of use and for its ability to read disks that dd cannot. Its biggest weakness is that it does not log damaged sectors, even though it briefly display this information on screen while the reads are in progress. It outputs raw disk images.



Disk type options for the FC5025 USB Floppy Controller

BitCurator is another, more complex, suite of tools that is gaining significant traction in the digital preservation space, and we are only just beginning to explore its capabilities. A custom distribution of Ubuntu, BitCurator features the open source Guymager as its primary imaging tool. Guymager is flexible and relatively user-friendly imaging software that can output raw, AFF and EWF images along with checksums and metadata [6]. However, as of this writing we have had no success using BitCurator for imaging floppies of either the 3.5" or 5.25" variety. It has a 64-bit requirement and so does not run on our "New/Old" 32-bit machine. And while it will run under Virtual Box on our "USB floppy" workstation and does see the USB floppy drive, it freezes when attempting to read from it. We have not yet determined what is causing this problem.

However, it should be noted that while BitCurator is a very convenient way to obtain and use a wide range of forensics tools, most of its software components are available in other ways (such as via the CERT Forensics Tools package mentioned above) and they can compile and run quite readily in our 32-bit CentOS 5.x environment.

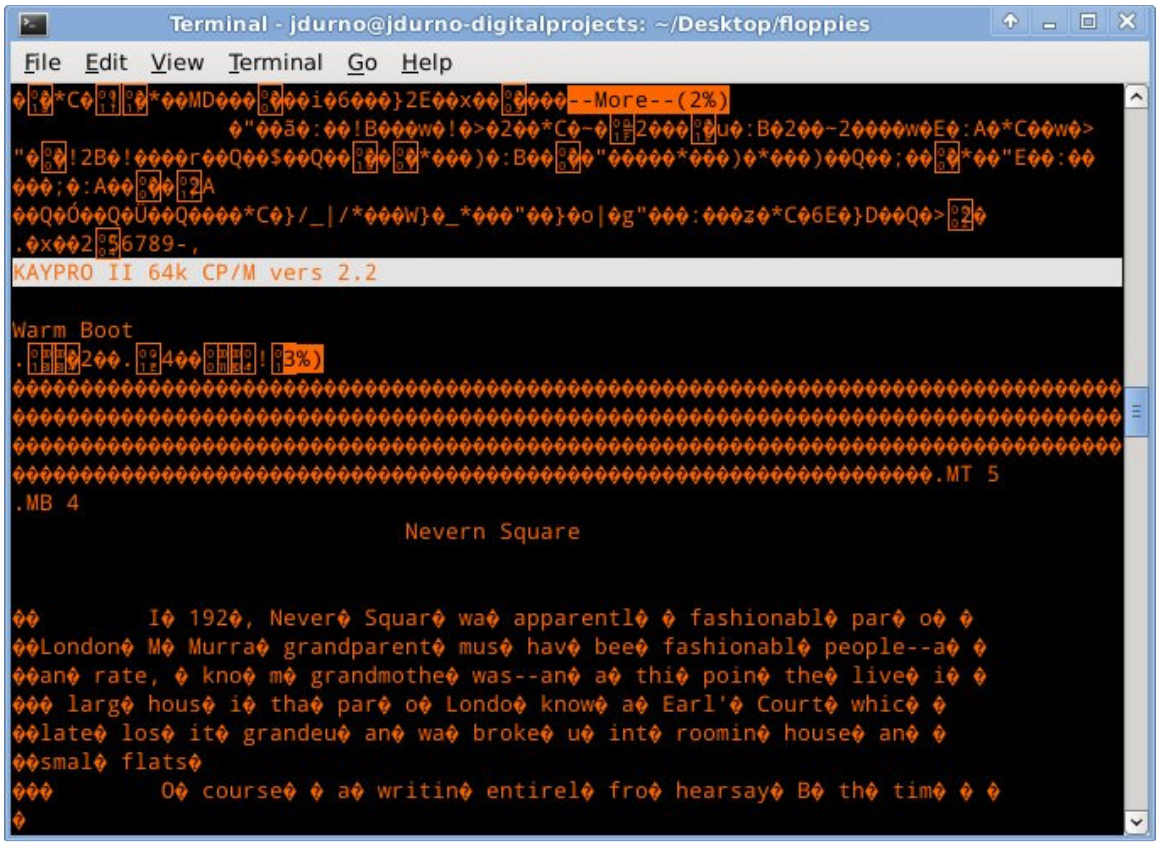
Finally, one of the bigger imaging challenges we have encountered to date has been simply determining what combination of technologies wrote data to the disks in the first place. Even taking a bit-level copy of a disk, it can be critical to know, for example, whether the data it contains was saved by an Apple II or a DEC Rainbow 100, and which version of operating system it was running at the time. It is perhaps surprising how infrequently this kind of metadata accompanies the older floppy disks in our archives.

CONTENT RETRIEVAL

To illustrate the kinds of challenges posed by older media we will describe one of our earliest projects, imaging and extracting content from a box of twelve 5.25" disks in the fonds of a local (Vancouver Island) poet, Rona Murray. At the outset we knew that the disks were approximately 30 years old, from a variety of manufacturers, all single-sided double density. A label pasted on the side of the box indicated they contained files created in Wordstar 2.2 and in addition, each disk was labelled with a list of the files it contained. However, we did not know what kind of computer the author had used.

Our first attempts to image the disks failed. We used 'dd' with some standard configuration options but it was unable to pull an image from any of the disks we tried. Thinking the disks may have been corrupted we then employed a program called 'ddrescue' to see if we could recover data that way, as ddrescue reads at a lower level than dd. Using ddrescue we did manage to extract a considerable amount of data but the logs indicated significant error rates.

We were not able to mount the ddrescue images so we were not able to view their contents using standard file system tools. Grasping at straws, we then dumped the binary contents of the images to screen using the unix 'more' command. And this is what displayed:



It is interesting to note that, even if we had not been able to further refine our process much of the content would still have been readable, thanks to the fact that WordStar saved files primarily in plain text (albeit 7-bit ASCII). But even better, WordStar also indicated which version of itself created the file, in this case the "Kaypro II 64k CP/M" version. Armed with this information, we were able to go back and pull some very good images off ten of the twelve disks using the USB floppy drive, this time using Device Side Data's proprietary imaging tool (which has a built-in setting for Kaypro II CP/M disks). Unfortunately two of the twelve disks were visibly corroded and many sectors were unreadable, but overall this was a much higher success rate than we anticipated.

In this case, of course, our ability to run and emulate a range of historic computing environments was not helpful for accessing the contents of the disk images, given that CP/M is not yet one of the environments we can emulate. Fortunately we were able to use Michael Haardt's cpmtools package [7] on a modern Ubuntu linux workstation to list the contents and extract individual files. Although we currently lack a copy of WordStar, we were able to open the files easily in a variety of backwards-compatible applications, such as WordPerfect 12.

Our experience so far has been software like dd and Guymager have a much higher success rate for imaging floppies dating from the late 1980s through the 1990s, when disk geometries had become more standardized and the biggest compatibility issues were

between Microsoft and Apple systems. Imaging and retrieving content from disks dating from the early 1980s requires higher levels of effort and expertise.

Not all of our recovery projects have been successful. Our attempt to recover files from a set of 5.25" floppies that accompanied a 1984 Masters' thesis did not result in anything useful. In this case, the disks had been stored in the back of a hardbound thesis on the public shelves in one of our branch libraries, not in our climate-controlled archives. They contained executable programs written on an Apple II in an early educational software called PILOT, and data had been written on both sides, (so-called 'flippy disks,' a fairly common practice back in the day, but not encouraged by the disk manufacturers, who controlled for quality only on one side). Given our minimal success in reading any data at all from these disks, we suspect they may have passed through a library desensitizer at least once in the previous 30 years. An object lesson, if one is needed, that floppy disks are fragile media and need to be treated as such. We are pleased to report that the hardbound thesis is still quite legible, however.

LONG TERM STORAGE

Of course, simply having the technological capability to image old media is only a first step [8]. We are only beginning to work out how we want to operationalize the storage of images going forward. We will need to work through a number of issues, such as:

1. Metadata: What metadata do we want to capture for each image? In addition to the usual subject classification, it would be very useful to note any additional information we can gather about the original software environment, as well as what tools and settings we used to capture the images. It would also be useful to capture any sector read errors reported during the disk creation process. Then there is the question of metadata formats. Guymager provides some useful functionality in this area, but may not be the best choice for imaging all of the disks in our collection, particularly the early ones.

2. Format: What image formats do we want to preserve? As noted above, the raw file format seems to us like a reasonable choice. Others may disagree. Given the small size of these images it would be possible to store them in multiple formats if necessary.

3. Content: How much effort do we want to put into extracting and forward-migrating the files contained on the disk images? Creating images is fairly trivial if you know what was used to write to the disk in the first place. Extracting content from the images is a variable process: it can be relatively time consuming and technical, as illustrated in our example above; or it can be easy if the disk images can be mounted and explored using standard file system utilities.

Given that there is a pressing need to image many of these disks before they become unreadable, it may be better to focus our efforts on imaging and only do minimal content sampling to ensure the imaging process was successful.

4. Organizing files: What is the best way to organize the files so that their conceptual and physical relationships are preserved in a digital storage system?

5. Storage system. Our library recently acquired an instance of Archivematica, an open source, OAIS compliant digital preservation system from Artefactual Systems. This would appear to be our best long-term storage solution, particularly given that it recently (version 1.2, September 2014) acquired the ability to ingest forensic disk images [9]. We have not yet tested this capability, however.

CONCLUSIONS

Although we are in the early stages, we can provisionally conclude that having a variety of hardware and software options is a good thing when it comes to working with early media and their contents. In general we have so far found that our "New/Old" workstation provides more flexibility when it comes to working with older media than does our "USB Floppy" workstation, and consequently it has been the one we use first when confronting a set of archival floppies for the first time. Yet as noted in the example above, the "USB Floppy" workstation is better for certain specialized applications, notably imaging 5.25" disks from the early 1980s.

So far the ability to boot into multiple operating systems has not been all that useful for retrieving older content. Most of the early documents we have so far encountered have been in some version of ASCII with minimal proprietary formatting, and conversion tools for many early document types are still readily available. Not to say that we won't ever encounter cases where having these capabilities will be of help to us, only that we haven't yet.

However, we are beginning to give some thought to contexts in which the multiple boot and emulation options could be very useful indeed. It seems probable to us that systems constructed around industrial motherboards could have a wider application than disk imaging. As legacy systems become increasingly difficult to maintain and expensive to purchase, systems like our "New/Old" machine could bridge the gap for researchers wishing to study old software running in its natural habitat.

Douglas Reside (2010) has observed that "… some of the most useful tools for migrating data from an obsolete to a modern (or at least slightly less obsolete) format are those computers that were manufactured at a moment when a popular new media format or transfer protocol had just emerged. Such computers often have ports or drives, along with associated drivers, capable of using older, and in their time more common, technologies as well as new ones." He calls these kinds of computers "Rosetta machines". In Reside's view the Rosetta machine "par excellence" is the Macintosh Wallstreet Powerbook, because having been manufactured in mid-1998 it came with swappable CD, DVD, floppy drives capable of reading 400K and 800K disks, and could accommodate a swappable Zip drive. It had an Ethernet port and USB capability (via onboard PCMCIA

slots), and it is capable of running older versions of Linux. Its' one main drawback is the lack of a 5.25" floppy drive.

Although we did not set out to build a Rosetta machine according to Reside's specifications, we were pleased to note that our "New/Old" workstation meets or exceeds every one of them. This, coupled with the fact that the supply of Wallstreet Powerbooks is finite (and ever more so as time takes its toll), suggests to us that rather than focusing on keeping old computers alive, building them out of a combination of new and secondhand parts may represent a more viable strategy for reading older media over the medium to long term.

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the Canadian Association of Research Libraries (CARL) for providing a research grant to support this project.

Notes

[1] See http://www.digitalintelligence.com/products/fred/

[2] For complete specs, see http://www.ibase-i.com.tw/mb820.htm

[3] Whereas in the fall of 2013 it was still possible to purchase IDE drives through NCIX, as of a year later IDE drives were not available from the same supplier.

[4] For more information on the CERT Linux Forensics Tools repository, see https://forensics.cert.org/

[5] BitCurator is available at http://www.bitcurator.net/

[6] See http://guymager.sourceforge.net/

[7] Michael Haardt's cpmtools package: http://www.moria.de/~michael/cpmtools/

[8] In fact, developing the technology before developing a preservation plan is probably not the optimal order of precedence. See Ricky Erway (2012), You've got to walk before you can run: First steps for managing born-digital content received on physical media. Dublin, Ohio: OCLC Research. Retrieved from http://www.oclc.org/research/publications/library/2012/2012-06.pdf

[9] See https://www.archivematica.org/wiki/Archivematica_Release_Notes

References

Breeding, M. (2012). From disaster recovery to digital preservation. *Computers in Libraries, 32(4).* Retrieved from http://librarytechnology.org/ltg-displaytext.pl?RC=16821

Duranti, L. (2009). From digital diplomatics to digital records forensics. *Archivaria 68*. Retrieved from http://journals.sfu.ca/archivar/index.php/archivaria/article/view/13229/14548

Forensics Wiki.(n. d.). AFF. Retrieved from http://www.forensicswiki.org/wiki/AFF

Fox, R. (2011). Forensics of digital librarianship. *OCLC Systems & Services: International digital library perspectives, 27(4).* Retrieved from http://dx.doi.org/10.1108/10650751111182560

Garfinkel, S. L., Malan, D. J., Dubec, K., Stevens, & C., Pham, C. (2006). Advanced forensic format: An open, extensible format for disk imaging. Retrieved from http://nrs.harvard.edu/urn-3:HUL.InstRepos:2829932

Kirshenbaum, M., Ovenden, R., & Redwine, G. (2010) *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*. Washington, D.C.: Council on Library and Information Resources. Retrieved from http://www.clir.org/pubs/reports/pub149

Reside, D. (2010). Rosetta computers. In M. Kirshenbaum, R. Ovenden, & G. Redwine, *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*. Washington, D.C.: Council on Library and Information Resources. Retrieved from http://www.clir.org/pubs/reports/pub149

Reside, D. (2012). Digital archaeology: recovering your digital history. Retrieved from http://www.nypl.org/blog/2012/07/23/digital-archaeology-recovering-your-digital-history

Woods, K. & Lee, C. (2012). Acquisition and processing of disk images to further archival goals. *Archiving 2012,* Copenhagen, Denmark, June 12-15, 2012.Retrieved from http://ils.unc.edu/callee/archiving-2012-woods-lee.pdf