Two WIPs at Duke

Things I keep swearing I'll finish, but who knows, maybe I should crowdsource some of it

[Matthew] Farrell
July 22(3?), 2015

Capture Lab Conference @
Stanford University Libraries

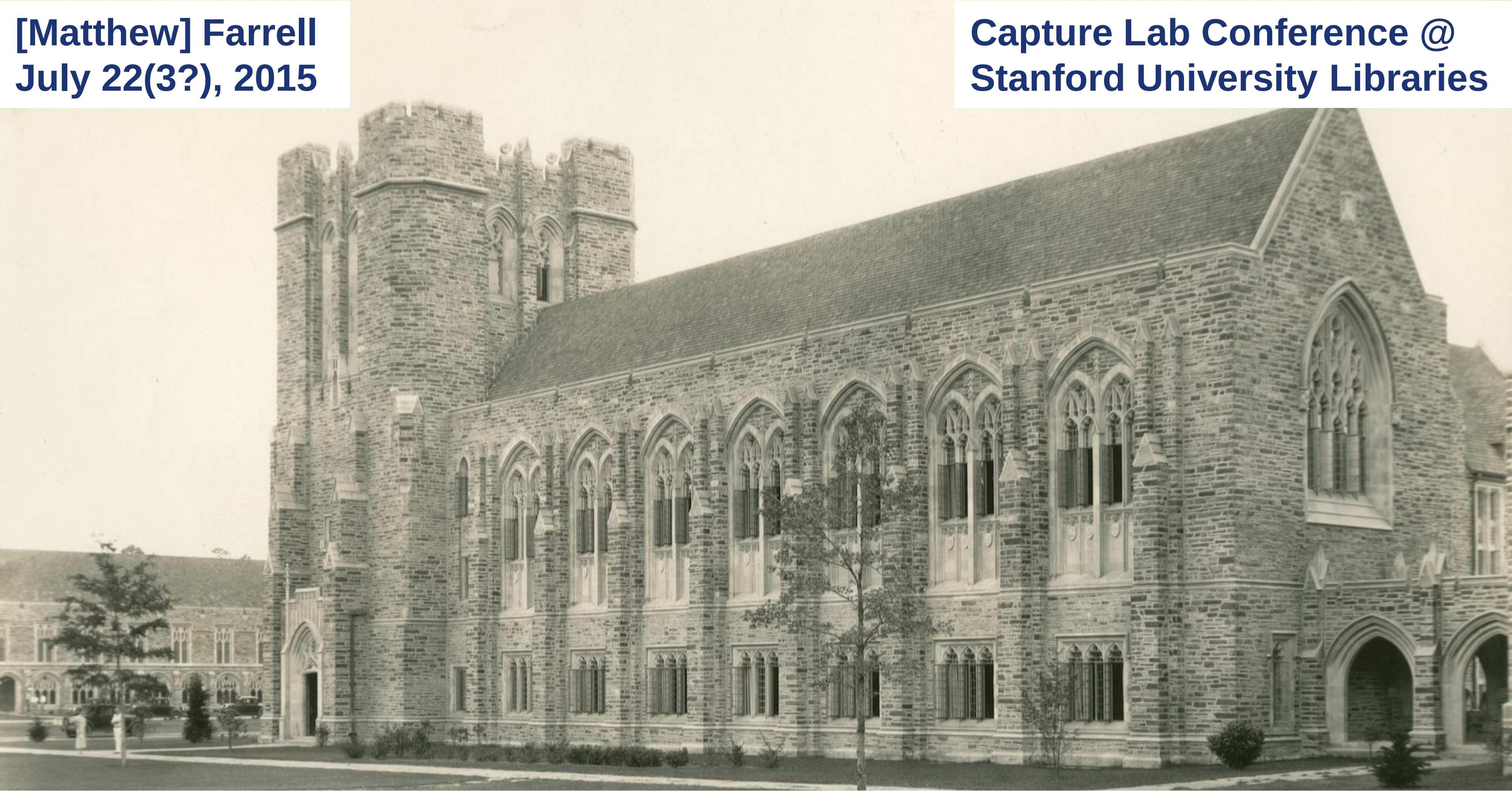
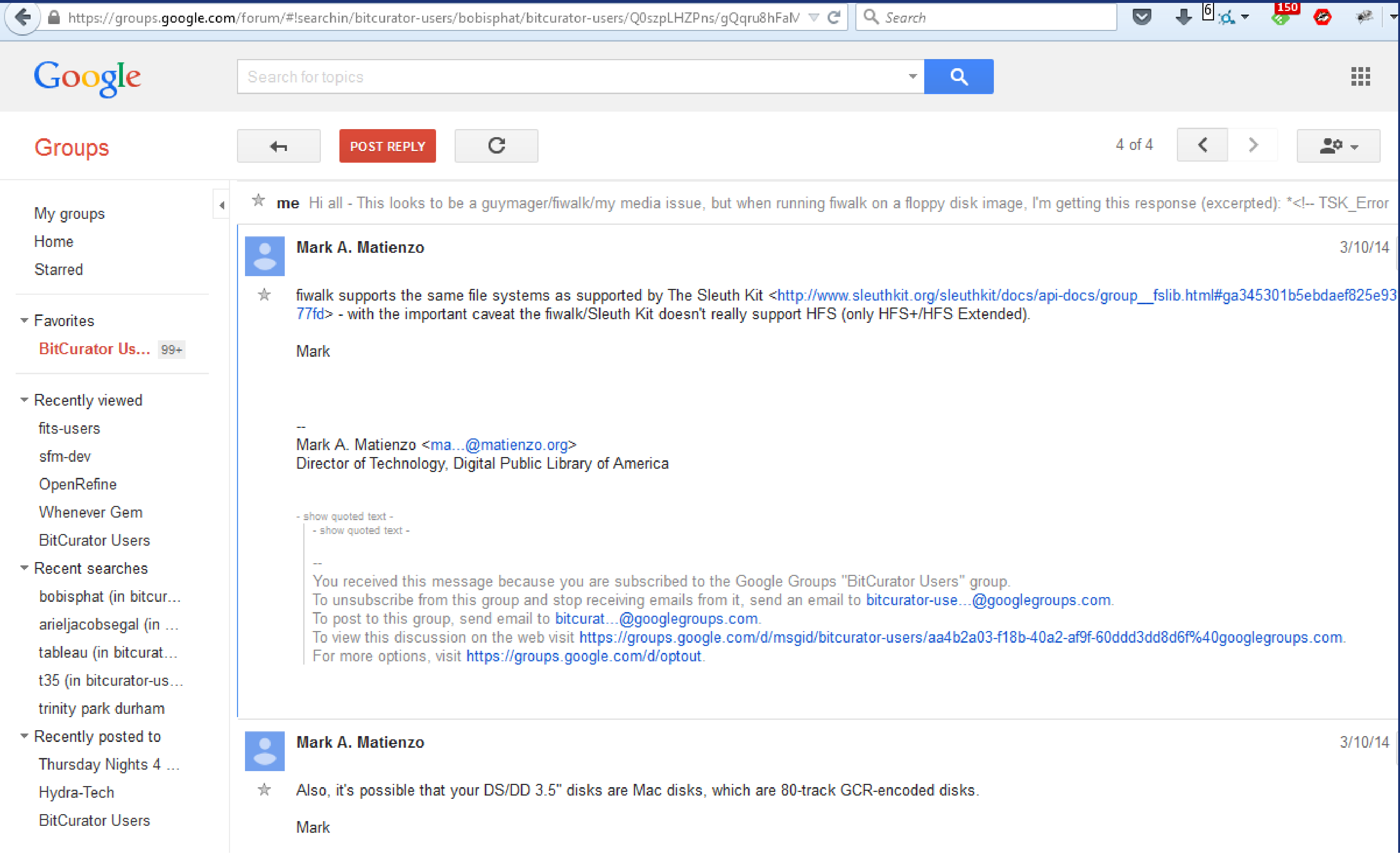


Image: General Library Exterior, before 1948. Source: Duke University Archives. License: cc-by-nc-sa.

Kryoflux Floppy Concordance



me Hi all - This looks to be a guymager/fiwalk/my media issue, but when running fiwalk on a floppy disk image, I'm getting this response (excerpted): *<!-- TSK_Error

Mark A. Matienzo 3/10/14

fiwalk supports the same file systems as supported by The Sleuth Kit <http://www.sleuthkit.org/sleuthkit/docs/api-docs/group__fslib.html#ga345301b5ebdaef825e9377fd> - with the important caveat the fiwalk/Sleuth Kit doesn't really support HFS (only HFS+/HFS Extended).

Mark

--
Mark A. Matienzo <ma...@matienzo.org>
Director of Technology, Digital Public Library of America

- show quoted text -
- show quoted text -

--
You received this message because you are subscribed to the Google Groups "BitCurator Users" group.
To unsubscribe from this group and stop receiving emails from it, send an email to bitcurator-use...@googlegroups.com.
To post to this group, send email to bitcurat...@googlegroups.com.
To view this discussion on the web visit https://groups.google.com/d/msgid/bitcurator-users/aa4b2a03-f18b-40a2-af9f-60ddd3dd8d6f%40googlegroups.com.
For more options, visit https://groups.google.com/d/optout.

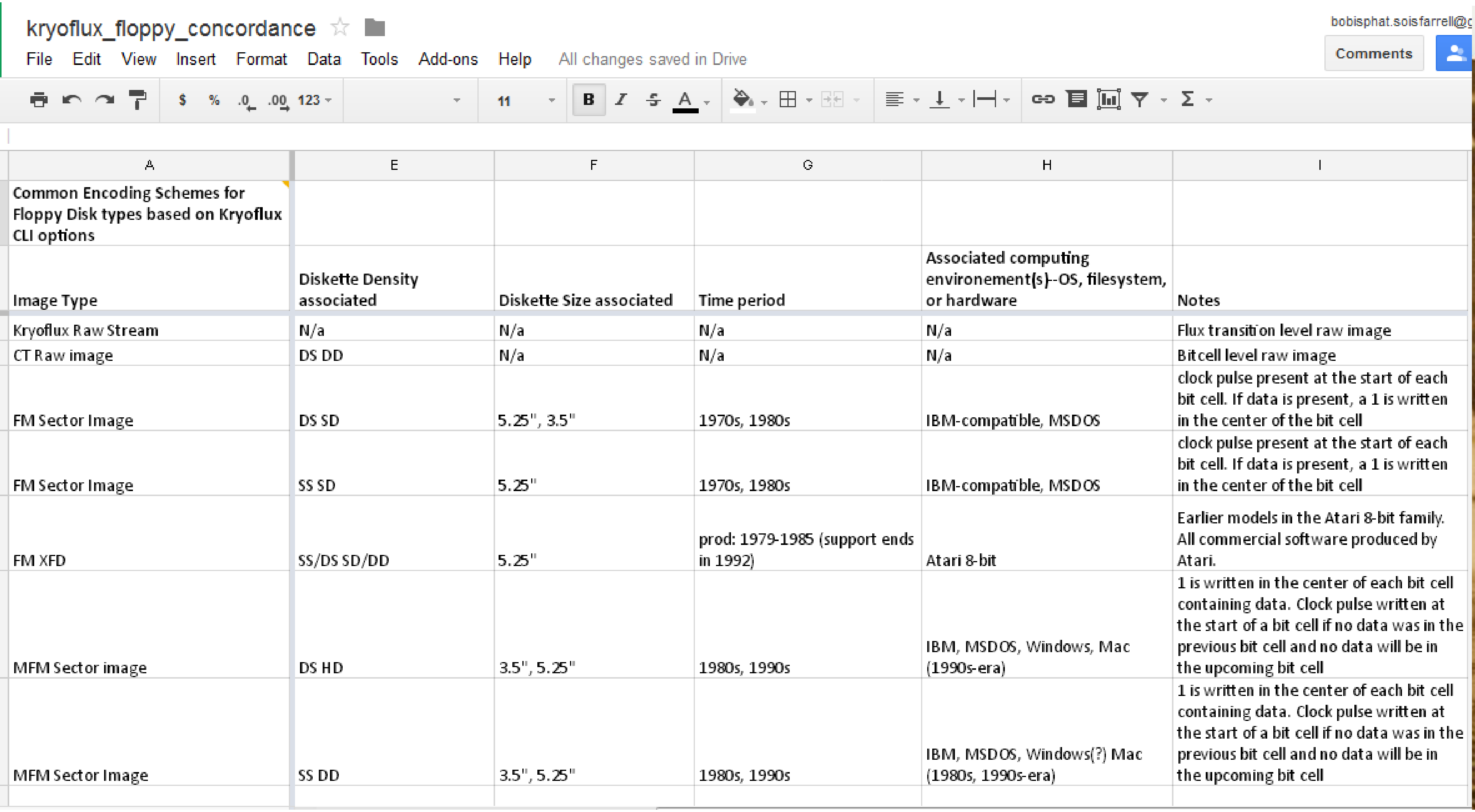
Mark A. Matienzo 3/10/14

Also, it's possible that your DS/DD 3.5" disks are Mac disks, which are 80-track GCR-encoded disks.

Mark

Mark Matienzo offers some helpful advice in 2014.

Concordance



A	E	F	G	H	I
Common Encoding Schemes for Floppy Disk types based on Kryoflux CLI options					
Image Type	Diskette Density associated	Diskette Size associated	Time period	Associated computing environement(s)--OS, filesystem, or hardware	Notes
Kryoflux Raw Stream	N/a	N/a	N/a	N/a	Flux transition level raw image
CT Raw image	DS DD	N/a	N/a	N/a	Bitcell level raw image
FM Sector Image	DS SD	5.25", 3.5"	1970s, 1980s	IBM-compatible, MSDOS	clock pulse present at the start of each bit cell. If data is present, a 1 is written in the center of the bit cell
FM Sector Image	SS SD	5.25"	1970s, 1980s	IBM-compatible, MSDOS	clock pulse present at the start of each bit cell. If data is present, a 1 is written in the center of the bit cell
FM XFD	SS/DS SD/DD	5.25"	prod: 1979-1985 (support ends in 1992)	Atari 8-bit	Earlier models in the Atari 8-bit family. All commercial software produced by Atari.
MFM Sector image	DS HD	3.5", 5.25"	1980s, 1990s	IBM, MSDOS, Windows, Mac (1990s-era)	1 is written in the center of each bit cell containing data. Clock pulse written at the start of a bit cell if no data was in the previous bit cell and no data will be in the upcoming bit cell
MFM Sector Image	SS DD	3.5", 5.25"	1980s, 1990s	IBM, MSDOS, Windows(?) Mac (1980s, 1990s-era)	1 is written in the center of each bit cell containing data. Clock pulse written at the start of a bit cell if no data was in the previous bit cell and no data will be in the upcoming bit cell

Current State

Some things have changed since March 2014

- Other sources of information
  - Anecdotal
  - No citations
  - Somewhat cited (Wikipedia)

Where I’d like to see it go

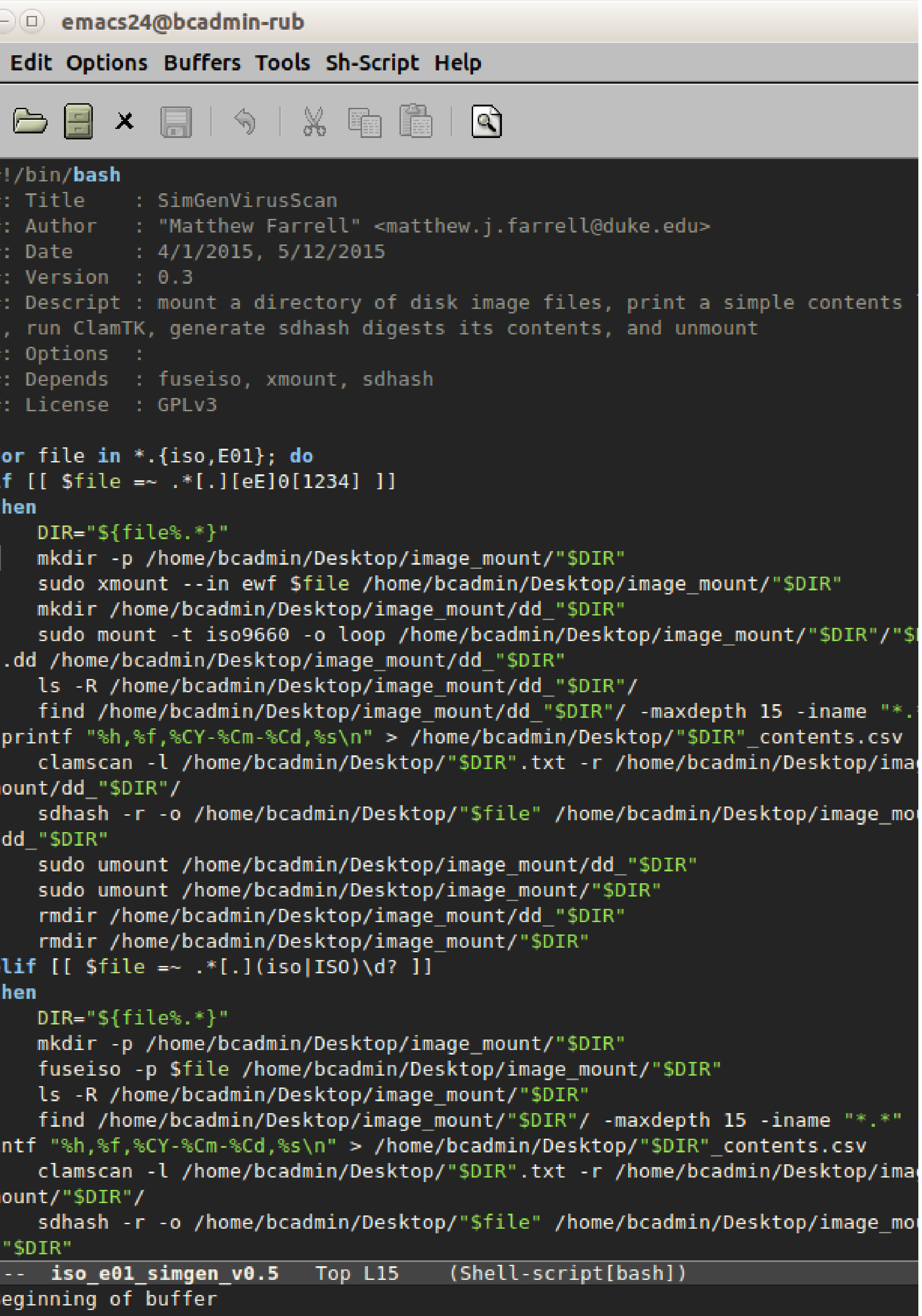
- Confirm the information found online
  - Lower hanging fruit (Wikipedia)
  - Other sources
- Photographs
  - Labeled, unlabeled
- Flesh out the notes
  - Tips, tricks
  - Contact info?
  - Particular hardware configurations

Automating File Similarity Comparison

Problem: JHFNC Records

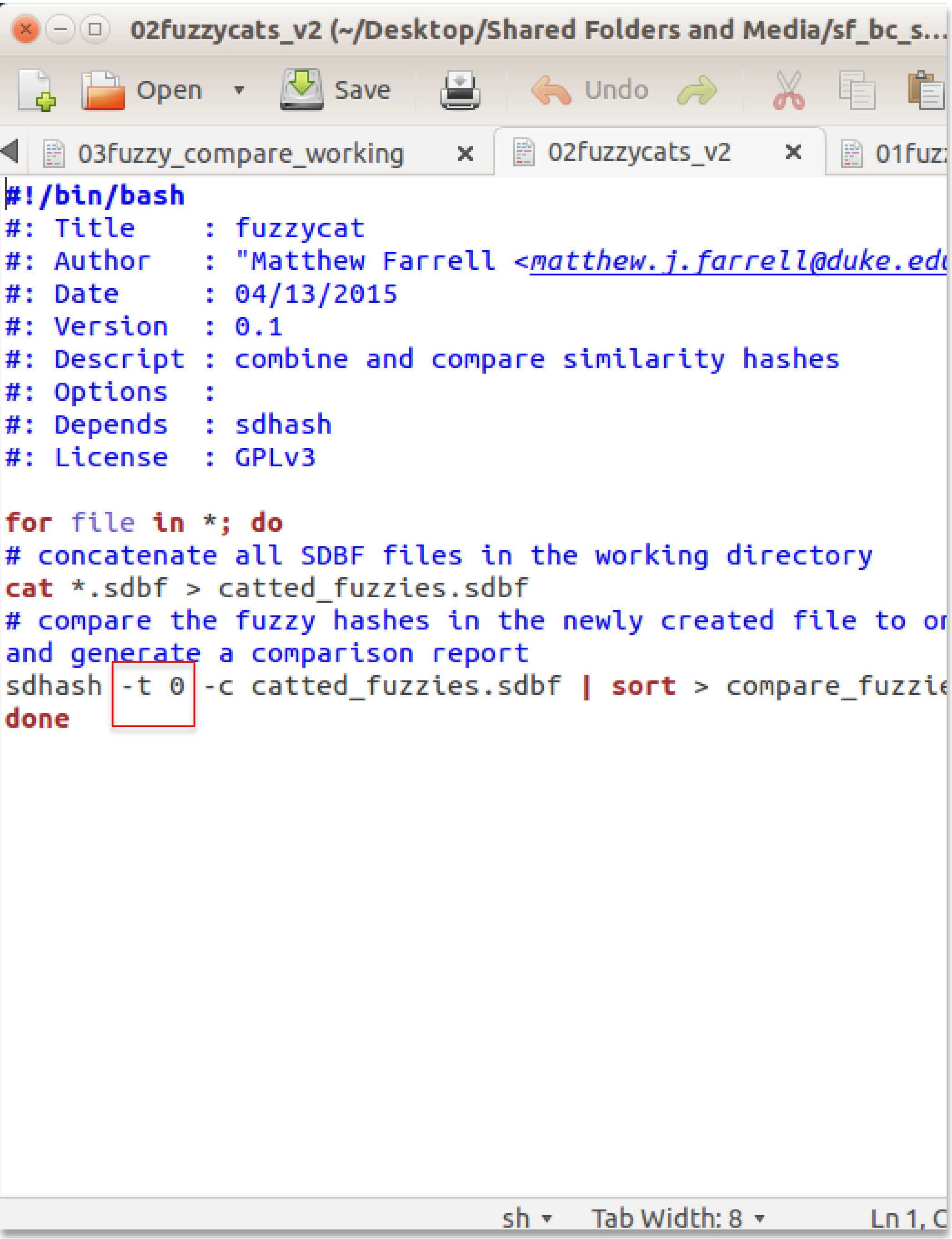
Idiosyncratic labeling/organization --> Untold duplication

First step



```
!/bin/bash
: Title    : SimGenVirusScan
: Author   : "Matthew Farrell" <matthew.j.farrell@duke.edu>
: Date     : 4/1/2015, 5/12/2015
: Version  : 0.3
: Descript : mount a directory of disk image files, print a simple contents
, run ClamTK, generate sdhash digests its contents, and unmount
: Options  :
: Depends  : fuseiso, xmount, sdhash
: License  : GPLv3

or file in *.{iso,E01}; do
f [[ $file =~ .*[.][eE]0[1234] ]]
hen
   DIR="${file%.*}"
   mkdir -p /home/bcadmin/Desktop/image_mount/"$DIR"
   sudo xmount --in ewf $file /home/bcadmin/Desktop/image_mount/"$DIR"
   mkdir /home/bcadmin/Desktop/image_mount/dd_"$DIR"
   sudo mount -t iso9660 -o loop /home/bcadmin/Desktop/image_mount/"$DIR"/"$
.dd /home/bcadmin/Desktop/image_mount/dd_"$DIR"
   ls -R /home/bcadmin/Desktop/image_mount/dd_"$DIR"/
   find /home/bcadmin/Desktop/image_mount/dd_"$DIR"/ -maxdepth 15 -iname "*.
printf "%h,%f,%CY-%Cm-%Cd,%s\n" > /home/bcadmin/Desktop/"$DIR"_contents.csv
   clamscan -l /home/bcadmin/Desktop/"$DIR".txt -r /home/bcadmin/Desktop/ima
ount/dd_"$DIR"/
   sdhash -r -o /home/bcadmin/Desktop/"$file" /home/bcadmin/Desktop/image_mo
dd_"$DIR"
   sudo umount /home/bcadmin/Desktop/image_mount/dd_"$DIR"
   sudo umount /home/bcadmin/Desktop/image_mount/"$DIR"
   rmdir /home/bcadmin/Desktop/image_mount/dd_"$DIR"
   rmdir /home/bcadmin/Desktop/image_mount/"$DIR"
lif [[ $file =~ .*[.](iso|ISO)\d? ]]
hen
   DIR="${file%.*}"
   mkdir -p /home/bcadmin/Desktop/image_mount/"$DIR"
   fuseiso -p $file /home/bcadmin/Desktop/image_mount/"$DIR"
   ls -R /home/bcadmin/Desktop/image_mount/"$DIR"
   find /home/bcadmin/Desktop/image_mount/"$DIR"/ -maxdepth 15 -iname "*.*"
ntf "%h,%f,%CY-%Cm-%Cd,%s\n" > /home/bcadmin/Desktop/"$DIR"_contents.csv
   clamscan -l /home/bcadmin/Desktop/"$DIR".txt -r /home/bcadmin/Desktop/ima
ount/"$DIR"/
   sdhash -r -o /home/bcadmin/Desktop/"$file" /home/bcadmin/Desktop/image_mo
"$DIR"
--  iso_e01_simgen_v0.5   Top L15    (Shell-script[bash])
eginning of buffer
```

- Bash script to mount disk image (ISO, EWF), create directory printout, run virus scan, create fuzzy hashes
- Result: some of the accessions-required reporting, fuzzy hash digest files



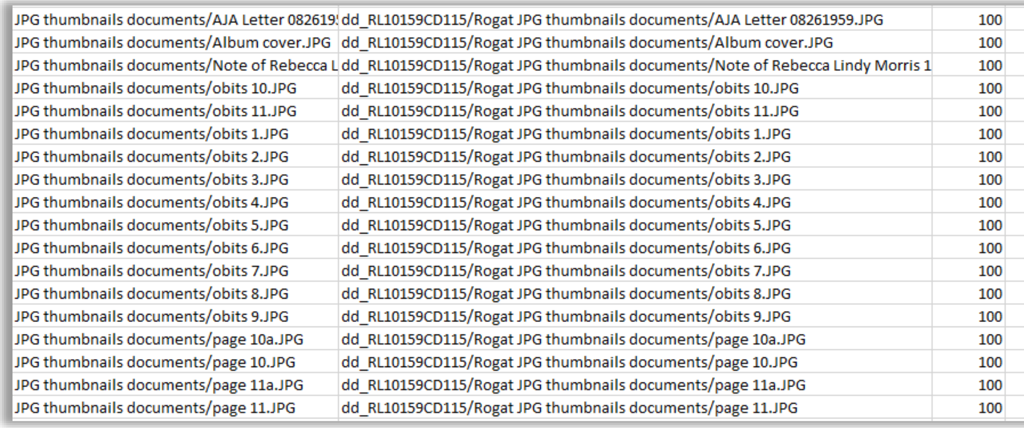
```
#!/bin/bash
#: Title     : fuzzycat
#: Author    : "Matthew Farrell <matthew.j.farrell@duke.ed
#: Date      : 04/13/2015
#: Version   : 0.1
#: Descript  : combine and compare similarity hashes
#: Options   :
#: Depends   : sdhash
#: License   : GPLv3

for file in *; do
# concatenate all SDBF files in the working directory
cat *.sdbf > catted_fuzzies.sdbf
# compare the fuzzy hashes in the newly created file to o
and generate a comparison report
sdhash -t 0 -c catted_fuzzies.sdbf | sort > compare_fuzzie
done
```

Second step

- Aborted simple script:
  - Combined all sdhash digests, mecha-digest to itself
  - Seemingly worked but output difficult to parse (the tolerance level highlighted resulted in 60K lines for comparison of 5 optical discs
  - Combining first results in disk image comparing to itself, which will is undesirable with mass storage volumes
- Some hope for this approach, when requiring strict confidence scores

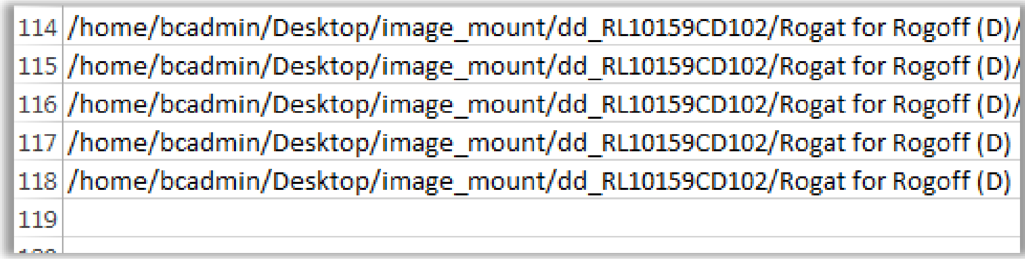
What I did instead (not recommended)

- Grouped discs with potential duplication
- Ran sdhash comparisons manually for each grouping
  - Looked at summaries for the comparison versus the contents for each disc image
- Laborious, but resulted in de-duplication
- Was it worth the time?



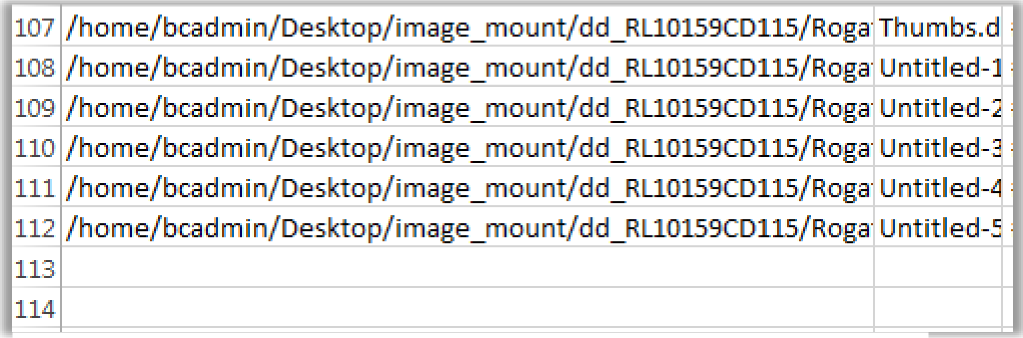
JPG thumbnails documents/AJA Letter 082619	dd_RL10159CD115/Rogat JPG thumbnails documents/AJA Letter 08261959.JPG	100
JPG thumbnails documents/Album cover.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/Album cover.JPG	100
JPG thumbnails documents/Note of Rebecca L	dd_RL10159CD115/Rogat JPG thumbnails documents/Note of Rebecca Lindy Morris 1	100
JPG thumbnails documents/obits 10.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 10.JPG	100
JPG thumbnails documents/obits 11.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 11.JPG	100
JPG thumbnails documents/obits 1.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 1.JPG	100
JPG thumbnails documents/obits 2.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 2.JPG	100
JPG thumbnails documents/obits 3.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 3.JPG	100
JPG thumbnails documents/obits 4.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 4.JPG	100
JPG thumbnails documents/obits 5.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 5.JPG	100
JPG thumbnails documents/obits 6.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 6.JPG	100
JPG thumbnails documents/obits 7.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 7.JPG	100
JPG thumbnails documents/obits 8.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 8.JPG	100
JPG thumbnails documents/obits 9.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/obits 9.JPG	100
JPG thumbnails documents/page 10a.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/page 10a.JPG	100
JPG thumbnails documents/page 10.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/page 10.JPG	100
JPG thumbnails documents/page 11a.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/page 11a.JPG	100
JPG thumbnails documents/page 11.JPG	dd_RL10159CD115/Rogat JPG thumbnails documents/page 11.JPG	100

Comparison of two disk images



114	/home/bcadmin/Desktop/image_mount/dd_RL10159CD102/Rogat for Rogoff (D)/
115	/home/bcadmin/Desktop/image_mount/dd_RL10159CD102/Rogat for Rogoff (D)/
116	/home/bcadmin/Desktop/image_mount/dd_RL10159CD102/Rogat for Rogoff (D)/
117	/home/bcadmin/Desktop/image_mount/dd_RL10159CD102/Rogat for Rogoff (D)
118	/home/bcadmin/Desktop/image_mount/dd_RL10159CD102/Rogat for Rogoff (D)
119	

End of contents for Disk 102

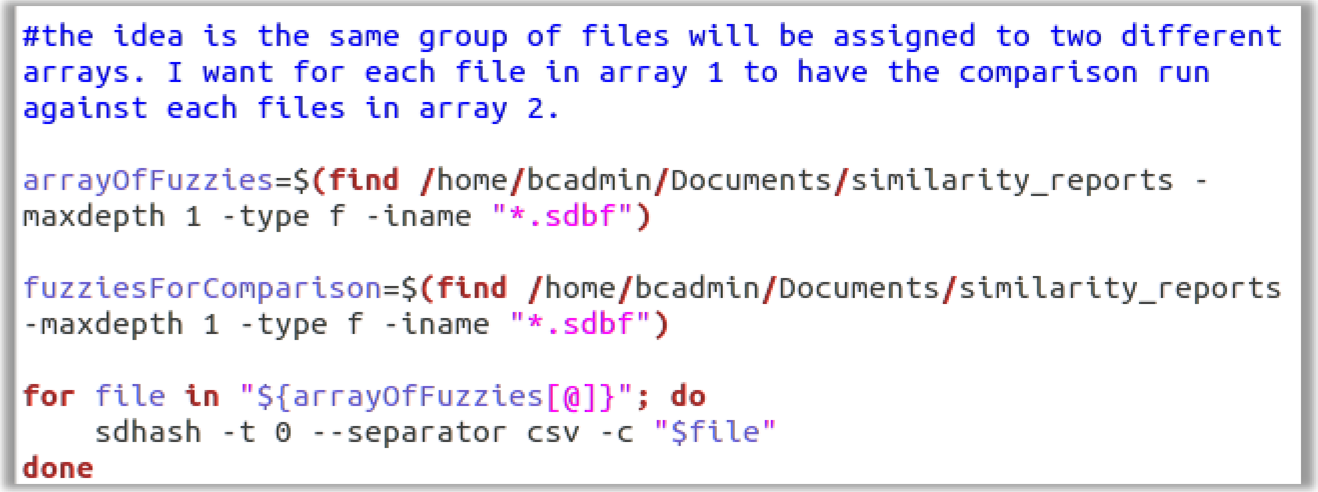


107	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Thumbs.d
108	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Untitled-1
109	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Untitled-2
110	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Untitled-3
111	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Untitled-4
112	/home/bcadmin/Desktop/image_mount/dd_RL10159CD115/Roga	Untitled-5
113		
114		

End of contents for Disk 115

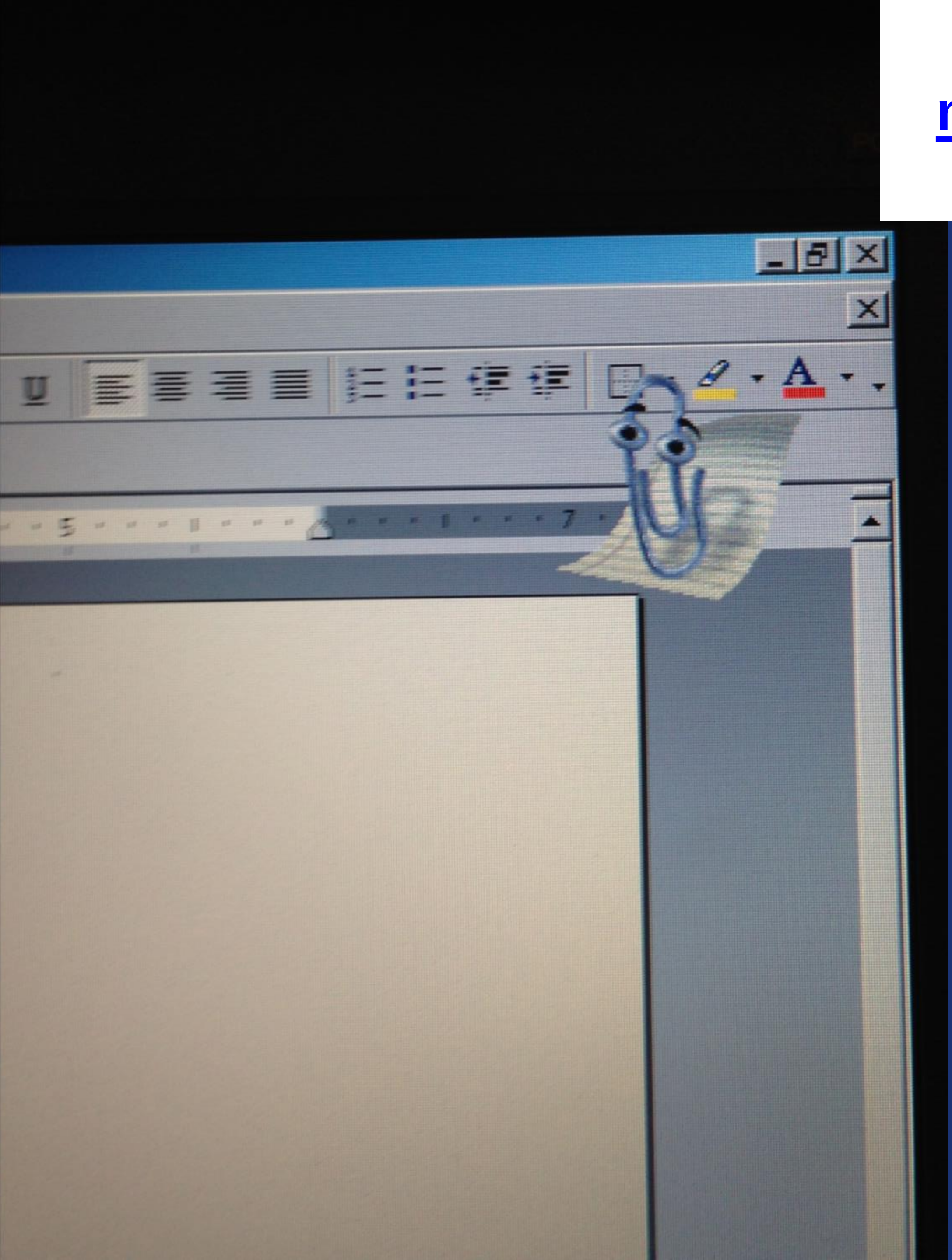
What I would rather do (next steps)

- Internal decisions about sdhash’s use
- Script to, for each digest file detected, *sdhash –c* against each other digest file
- Bash?
  - Worked on but haven’t troubleshot
- Python
  - More flexible, haven’t had time yet to address



```
#the idea is the same group of files will be assigned to two different
arrays. I want for each file in array 1 to have the comparison run
against each files in array 2.

arrayOfFuzzies=$(find /home/bcadmin/Documents/similarity_reports -
maxdepth 1 -type f -iname "*.sdbf")

fuzziesForComparison=$(find /home/bcadmin/Documents/similarity_reports
-maxdepth 1 -type f -iname "*.sdbf")

for file in "${arrayOfFuzzies[@]}"; do
    sdhash -t 0 --separator csv -c "$file"
done
```

[Matthew] Farrell
matthew.j.farrell@duke.edu
@laissezfarrell



Thanks!