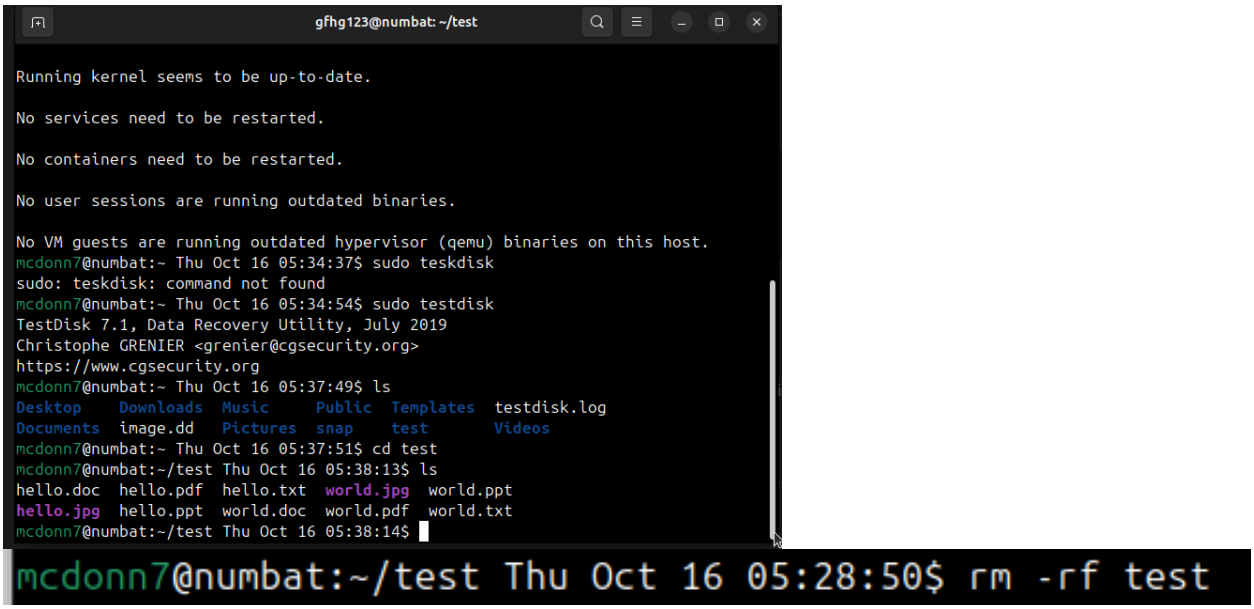
Discovery 2
Benjamin McDonnough

Question 1:
I had all files get recovered, all with their original names. I used testdisk to get my files back which allowed me to go through partitions and find the recovered one that I need, correct the partition table, and write it to the drive to restore access. Also, for clarification, the disk was dev/vda (screenshot 5), I just didn't have the correct one highlighted when I took the screenshot.



```
gfhg123@numbat: ~/test

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
mcdonn7@numbat:~ Thu Oct 16 05:34:37$ sudo teskdisk
sudo: teskdisk: command not found
mcdonn7@numbat:~ Thu Oct 16 05:34:54$ sudo testdisk
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org
mcdonn7@numbat:~ Thu Oct 16 05:37:49$ ls
Desktop    Downloads  Music     Public  Templates  testdisk.log
Documents  image.dd   Pictures  snap    test       Videos
mcdonn7@numbat:~ Thu Oct 16 05:37:51$ cd test
mcdonn7@numbat:~/test Thu Oct 16 05:38:13$ ls
hello.doc  hello.pdf  hello.txt  world.jpg  world.ppt
hello.jpg  hello.ppt  world.doc  world.pdf  world.txt
mcdonn7@numbat:~/test Thu Oct 16 05:38:14$
```

```
gfhg123@numbat: ~/test

TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org


TestDisk is free data recovery software designed to help recover lost
partitions and/or make non-booting disks bootable again when these symptoms
are caused by faulty software, certain types of viruses or human error.
It can also be used to repair some filesystem errors.

Information gathered during TestDisk use can be recorded for later
review. If you choose to create the text file, testdisk.log , it
will contain TestDisk options, technical information and various
outputs; including any folder/file names TestDisk was used to find and
list onscreen.

Use arrow keys to select, then press Enter key:
>[ Create ] Create a new log file
 [ Append ] Append information to log file
 [ No Log ] Don't record anything
```

```
gfhg123@numbat: ~/test

TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

  TestDisk is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):
>Disk /dev/mapper/ubuntu--vg-ubuntu--lv - 25 GB / 23 GiB
 Disk /dev/dm-0 - 25 GB / 23 GiB
 Disk /dev/vda - 53 GB / 50 GiB
 Disk /dev/loop0 - 4096 B (RO)
 Disk /dev/loop1 - 72 MB / 68 MiB (RO)
 Disk /dev/loop2 - 243 MB / 232 MiB (RO)
 Disk /dev/loop3 - 517 MB / 493 MiB (RO)
 Disk /dev/loop4 - 96 MB / 91 MiB (RO)
 Disk /dev/loop5 - 46 MB / 44 MiB (RO)
 Disk /dev/loop6 - 229 MB / 219 MiB (RO)
>[Previous]  [  Next  ]  [Proceed ]  [  Quit  ]

Note: Disk capacity must be correctly detected for a successful recovery.
If a disk listed above has an incorrect size, check HD jumper settings and BIOS
detection, and install the latest OS patches and disk drivers.
```

```
gfhg123@numbat: ~/test

TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org


Disk /dev/vda - 53 GB / 50 GiB

Please select the partition table type, press Enter when done.
>[Intel  ] Intel/PC partition
 [EFI GPT] EFI GPT partition map (Mac i386, some x86_64...)
 [Humax  ] Humax partition table
 [Mac    ] Apple partition map (legacy)
 [None   ] Non partitioned media
 [Sun    ] Sun Solaris partition
 [XBox   ] XBox partition
 [Return ] Return to disk selection



Hint: EFI GPT partition table type has been detected.
Note: Do NOT select 'None' for media with only a single partition. It's very
rare for a disk to be 'Non-partitioned'.
```

```
gfhg123@numbat: ~/test
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

Disk /dev/vda - 53 GB / 50 GiB
     CHS 104025 16 63 - sector size=512

 [ Analyse  ] Analyse current partition structure and search for lost partitions
>[ Advanced ] Filesystem Utils
 [ Geometry ] Change disk geometry
 [ Options  ] Modify options
 [ MBR Code ] Write TestDisk MBR code to first sector
 [ Delete   ] Delete all data in the partition table
 [ Quit     ] Return to disk selection

Note: Correct disk geometry is required for a successful recovery. 'Analyse'
process may give some warnings if it thinks the logical geometry is mismatched.
```

```
gfhg123@numbat: ~/test
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

Disk /dev/vda - 53 GB / 50 GiB - CHS 104025 16 63

     Partition                  Start        End    Size in sectors
> 1 P EFI GPT                 0   0  2 104025   6 22  104857599

 [  Type  ] >[Image Creation]  [  Quit  ]
                               Create an image
```

```
TestDisk 7.1, Data Recovery Utility, July 2019

Please select where to store the file image.dd (53687 MB), an image of the
partition
Keys: Arrow keys to select another directory
      C when the destination is correct
      Q to quit
Directory /home/gfhg123
 drwxr-x---  1000  1000      4096 16-Oct-2025 05:36 .
 drwxr-xr-x     0     0      4096 24-Sep-2025 03:55 ..
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Desktop
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Documents
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Downloads
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Music
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Pictures
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Public
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Templates
 drwxr-xr-x  1000  1000      4096 24-Sep-2025 04:07 Videos
 drwx------  1000  1000      4096 16-Oct-2025 05:21 snap
>drwxrwxr-x  1000  1000      4096 16-Oct-2025 15:05 test
 -rw-r--r--     0     0 13337288704 16-Oct-2025 05:37 image.dd
 -rw-r--r--     0     0      1830 16-Oct-2025 05:37 testdisk.log
```

```
gfhg123@numbat: ~/test
mcdonn7@numbat:~/test Thu Oct 16 05:28:48$ ls -l
total 40
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:24 hello.doc
-rw-rw-r-- 1 gfhg123 gfhg123 7 Oct 16 05:26 hello.jpg
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:27 hello.pdf
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:28 hello.ppt
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct  9 12:58 hello.txt
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:25 world.doc
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:27 world.jpg
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:27 world.pdf
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:28 world.ppt
-rw-rw-r-- 1 gfhg123 gfhg123 6 Oct 16 05:20 world.txt
mcdonn7@numbat:~/test Thu Oct 16 05:28:50$
```

Question 2:

1. One of the links is a self-link. For instance, if I had /home/gfhg123/testdir, then /home/gfhg123/testdir/. is the self-reference link. The second one is the entry from its parent directory that points to it.

2. These entries , . and .. are hardlinks for the directory itself and the parent directory respectively. This allows for easy transversal and allows for things like ls, find, and pwd to move around the filesystem to accomplish their purpose.

Question 3:

a. Loop devices are virtual block devices that let you mount a file as if it were a physical disk. The loop back a regular file to act like a block device. The kernel module “loop” manages them. Linux has several loop devices by default so that multiple files or disk images can be mounted at the same time.

```
gfhg123@numbat: ~
mcdonn7@numbat:~ Fri Oct 17 15:22:32$ dd if=/dev/zero of=looptest.img bs=1M count=10
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0212828 s, 493 MB/s
mcdonn7@numbat:~ Fri Oct 17 15:22:52$ mkfs.ext4 looptest.img
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 2560 4k blocks and 2560 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

mcdonn7@numbat:~ Fri Oct 17 15:23:05$ sudo mkdir /mnt/looptest
[sudo] password for gfhg123:
mcdonn7@numbat:~ Fri Oct 17 15:23:21$ sudo mount -o loop looptest.img /mnt/looptest
mcdonn7@numbat:~ Fri Oct 17 15:23:34$ df -h | grep loop\
> exit
mcdonn7@numbat:~ Fri Oct 17 15:23:55$ df -h | grep loop
/dev/loop8                        5.4M   24K  4.7M   1% /mnt/looptest
mcdonn7@numbat:~ Fri Oct 17 15:23:59$
```

b. TTY stands for teletypewriter. TTY devices represent text-based input/output interfaces (terminals). The reason for there being so many is because each physical/virtual terminal is a separate /dev/ttyN (N stands for a number). The kernel exposes many tty devices to support multiple user sessions, serial consoles, hardware ports, and pseudo-terminals.

c. TTY represents the physical/virtual console, is located in /dev/ttyN (N stands for a number) and is managed by the Kernel console subsystem. On the other hand, pts represents a virtual terminal(s) created by terminal emulators like GNOME terminal, SSH, xterm, etc. PTS’s are managed by ptmx (pseudo-terminal multiplexer) and the devpts filesystem.

d.

VCS (Virtual Console Screen) – contains plain-text representation of the current screen for a virtual console.

VCSU (Virtual Console Screen Unicode) – Same as vcs, but stores Unicode test (UTF-8)

VCSA (Virtual Console Screen + Attributes) – Stores the console contents and color/attribute data.

There are so many of all of these because each virtual console has its own set of these devices. Basically, if you have a ton of virtual consoles, you are going to have even more of these devices.

e. Virtual devices mimic hardware behavior while pseudo devices provide software interfaces for system-lever operations.

f.

/dev/null – discards anything written to it. Reading from it gives EOF immediately. This can be used to redirect unwanted output.

/dev/zero – provides and endless stream of zero bytes (\0). This is helpful for creating blank files or initializing memory

/dev/urandom – provides pseudo-random data generated by the kernel's random number generator.

```
mcdonn7@numbat:~ Fri Oct 17 15:25:57$ echo "This disappears" > /dev/null
mcdonn7@numbat:~ Fri Oct 17 16:15:10$ dd if=/dev/zero of=sample.img bs iM count=5
dd: unrecognized operand 'bs'
Try 'dd --help' for more information.
mcdonn7@numbat:~ Fri Oct 17 16:15:26$ dd if=/dev/zero of=sample.img bs=1M count=5
5+0 records in
5+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.0117474 s, 446 MB/s
mcdonn7@numbat:~ Fri Oct 17 16:15:38$ head -c 32 /dev/urandom | hexdump
0000000 cef2 f2a8 7487 e17e dc1e d0a5 66eb 4a71
0000010 7f6a d31d 64a7 cba7 0bbc 7ebc 0b49 a22c
0000020
mcdonn7@numbat:~ Fri Oct 17 16:16:16$
```

Question 4:

Cron is used to automate repetitive tasks by running them at fixed times, dates, or intervals. Anacron is similar, but is designed for systems that aren't always running.

```
mcdonn7@numbat:~ Thu Oct 16 15:26:17$ mkdir -p ~/scripts
mcdonn7@numbat:~ Thu Oct 16 15:26:21$ nano ~/scripts/backup.sh
mcdonn7@numbat:~ Thu Oct 16 15:31:10$ chmod +x ~/scripts/backup.sh
mcdonn7@numbat:~ Thu Oct 16 15:32:59$ mkdir -p ~/backups
mcdonn7@numbat:~ Thu Oct 16 15:33:54$ nano ~/scripts/disk_usage.sh
mcdonn7@numbat:~ Thu Oct 16 15:35:16$ nano ~/scripts/disk_usage.sh
mcdonn7@numbat:~ Thu Oct 16 15:35:37$ chmod +x ~/scripts/disk_usage.sh
mcdonn7@numbat:~ Thu Oct 16 15:36:10$
```

```
  GNU nano 7.2                /home/gfhg123/scripts/backup.sh *
#!/bin/bash
tar -czf /home/ben/backups/backup_$(date +%F_%H-%M).tar.gz /home/ben/testdata

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
gfhg123@numbat: ~
  GNU nano 7.2          /home/gfhg123/scripts/disk_usage.sh *
#!/bin/bash
df -h >> /home/ben/system_report.log

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
  GNU nano 7.2                                                    /tmp/crontab.uXAk
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

# Daily backup ay 2am
37 12 * * * /home/ben/scripts/backup.sh

# Disk Usage Log
23 12 * * * /home/ben/scripts/disk_usage.sh
```

```
crontab: installing new crontab
mcdonn7@numbat:~/backups Thu Oct 16 12:49:25$ ls /home/gfhg123/backups
backup_2025-10-16_12-50.tar.gz
```

Question 5:

```
gfhg123@numbat: ~/backups
mcdonn7@numbat:~/backups Thu Oct 16 12:52:35$ mkdir -p /home/gfhg123/logdemo
mcdonn7@numbat:~/backups Thu Oct 16 12:52:47$ echo "Initial log entry at $(date)" > /home/gfhg123/logdemo/test.log
mcdonn7@numbat:~/backups Thu Oct 16 12:53:14$ ls -l /home/gfhg123/logdemo/
total 4
-rw-rw-r-- 1 gfhg123 gfhg123 53 Oct 16 12:53 test.log
mcdonn7@numbat:~/backups Thu Oct 16 12:53:23$ cat /home/gfhg123/logdemo/test.log
Initial log entry at Thu Oct 16 12:53:14 PM EDT 2025
mcdonn7@numbat:~/backups Thu Oct 16 12:53:37$

mcdonn7@numbat:~/logdemo Thu Oct 16 12:58:17$ for i in {1..200}; do echo "Log line $i at $(date)" >> ~/logdemo/test.log; done
mcdonn7@numbat:~/logdemo Thu Oct 16 12:58:38$ ls -lh /home/gfhg123/logdemo/test.log
-rw-rw-r-- 1 gfhg123 gfhg123 9.4K Oct 16 12:58 /home/gfhg123/logdemo/test.log
```

```
GNU nano 7.2
/home/gfhg123/logdemo/test.log {
        size 1k
        rotate 4
        compress
        delapcompress
        missingok
        notifempty
        create 644 gfhg123 gfhg123
}
```

```
mcdonn7@numbat:~/logdemo Thu Oct 16 13:04:56$ sudo logrotate -f /etc/logrotate.d/testlog
mcdonn7@numbat:~/logdemo Thu Oct 16 13:04:59$ ls -lh /home/gfhg123/logdemo
total 12K
-rw-r--r-- 1 gfhg123 gfhg123    0 Oct 16 13:04 test.log
-rw-rw-r-- 1 gfhg123 gfhg123 9.4K Oct 16 12:58 test.log.1
mcdonn7@numbat:~/logdemo Thu Oct 16 13:05:14$
```

Question 6:
Pipe: A named pipe, also called a FIFO (First-In-First-Out) special file, is a method of communication between processes. It allows one process to write data into the pipe and another process to read that data in the same order it was written. Unlike anonymous pipes (created automatically by the shell with |), named pipes are persistent objects that exist as files in the filesystem until deleted. They are created using the mkfifo command or the mknod command. These act as temporary data channels between unrelated processes.

Socket: A socket is a special file that provides a communication endpoint between processes. While most sockets are used for network communication (TCP/IP), Unix domain sockets exist entirely within the filesystem and allow efficient, bidirectional communication between local processes. Sockets are usually created by services or daemons using the socket() system call, but you can also create one manually with tools like nc -lU (netcat using a Unix socket) or using programming languages that support sockets. Sockets enable inter-process communication (IPC) within the same machine.

```
mcdonn7@numbat:~/logdemo Thu Oct 16 13:07:25$ sudo find / -type p 2>/dev/null | head -n 10
/run/user/1000/gnome-session-leader-fifo
/run/user/1000/systemd/inaccessible/fifo
/run/initctl
/run/dmeventd-client
/run/dmeventd-server
/run/systemd/inhibit/23.ref
/run/systemd/inhibit/13.ref
/run/systemd/inhibit/12.ref
/run/systemd/inhibit/11.ref
/run/systemd/inhibit/4.ref
mcdonn7@numbat:~/logdemo Thu Oct 16 13:07:49$ ls -l /run/user/1000/gnome-session-leader-fifo
prw-rw-r-- 1 gfhg123 gfhg123 0 Oct  9 08:57 /run/user/1000/gnome-session-leader-fifo
mcdonn7@numbat:~/logdemo Thu Oct 16 13:08:29$ ls -l /run/user/1000/systemd/inaccessible/fifo
p--------- 1 gfhg123 gfhg123 0 Oct  9 08:57 /run/user/1000/systemd/inaccessible/fifo
mcdonn7@numbat:~/logdemo Thu Oct 16 13:08:50$
```

```
mcdonn7@numbat:~/logdemo Thu Oct 16 13:09:07$ sudo find / -type s 2>/dev/null | head -n 10
/var/lib/fwupd/gnupg/S.gpg-agent.browser
/var/lib/fwupd/gnupg/S.gpg-agent
/var/lib/fwupd/gnupg/S.gpg-agent.extra
/var/lib/fwupd/gnupg/S.gpg-agent.ssh
/run/gdm3/dbus/dbus-TfMmbYWo
/run/gdm3/dbus/dbus-nds3u9j6
/run/uuidd/request
/run/snapd-snap.socket
/run/snapd.socket
/run/lxd-installer.socket
mcdonn7@numbat:~/logdemo Thu Oct 16 13:09:36$ ls -l /run/snapd.socket
srw-rw-rw- 1 root root 0 Oct  9 08:54 /run/snapd.socket
mcdonn7@numbat:~/logdemo Thu Oct 16 13:10:11$ ls -l /run/lxd-installer.socket
srw-rw---- 1 root lxd 0 Oct  9 08:54 /run/lxd-installer.socket
mcdonn7@numbat:~/logdemo Thu Oct 16 13:10:24$
```