Exercises on symbolic links and hard links.

Explore Inodes

1. Create a few sample files in your home directory:

```
cd ~
touch file1 file2 file3
```

2. Run `ls -li` and note the inode numbers.
3. What do identical inode numbers indicate?

   (Hint: Two names with the same inode point to the same data on disk.)

Understand what an inode represents and how hard links use inode numbers.

Create Hard Links

1. Create a hard link from `file1` to `hard1`:

```
ln file1 hard1
```

2. Run `ls -li` again. Compare the inode numbers of both files.
3. Modify `file1` with `echo "test" >> file1`. Then read both files.

```
cat file1
cat hard1
```

4. Delete the original file:

```
rm file1
```

Is `hard1` still accessible and containing the same data?

Verify that hard links share the same inode and data, and persist even after the original filename is removed.

Create Symbolic Links

1. Create a symbolic (soft) link:

   ```
   ln -s hard1 soft1
   ```

2. Use `ls -l` to see how symbolic links are displayed (look for the `l` file type and the arrow `->`).
3. Use `readlink soft1` to display its target.
4. Now remove the target:

   ```
   rm hard1
   ```

What happens if you try `cat soft1`?

Observe that symbolic links refer to file paths, not inodes, and that removing the target creates a *dangling link*.

Mixed Practice Tasks

Try solving the following small challenges:

1. **Cross-filesystem test:** Create a file on `/boot` and try to make a hard link to it from your home directory. What happens, and why?

   ```
   sudo touch /boot/afile
   ```

   ```
   ln /boot/afile ~/link
   ```

2. Create a symlink named `log` pointing to `/var/log` and verify its target using `ls -ld log`.
3. Use `find` to locate all symbolic links under `/etc`:

   ```
   find /etc -type l | head
   ```

4. Use `stat` on both a hard-linked and a symlinked file. Identify which fields differ.

Quick Review

- Use `ln` for hard links and `ln -s` for symbolic links.
- Hard links share inodes and cannot cross filesystems.
- Symbolic links point by pathname and can cross filesystem boundaries.
- Removing a file affects its hard links differently than its symbolic links.