**Day 2 – Phase 2: File & Directory Management + Search**

**Tasks:**

**• Inside iot\_logger, create logs/temperature.log and scripts/sensor\_script.py.**

**A screenshot of a computer

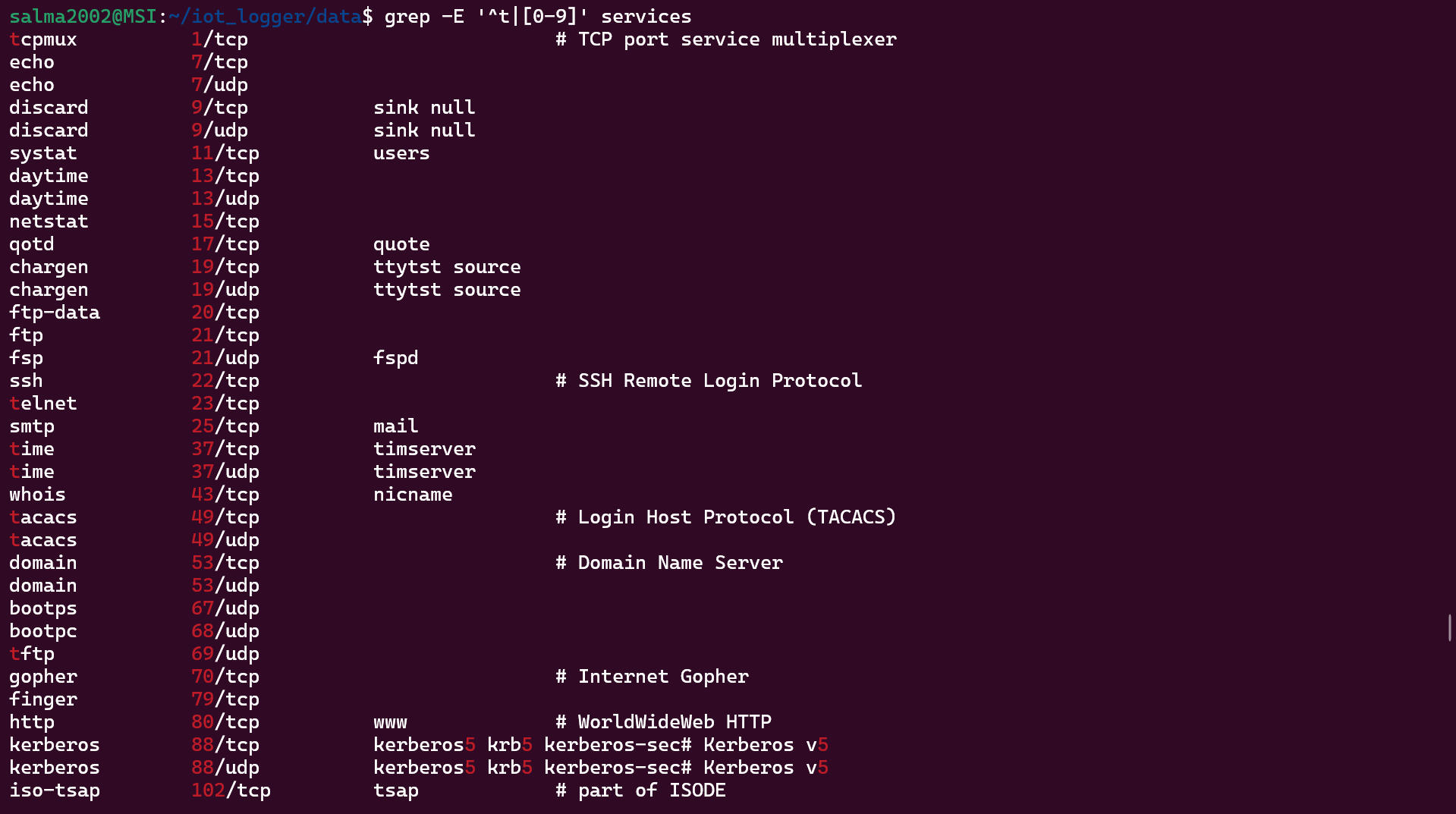
AI-generated content may be incorrect.**

**• Copy /etc/services into data and search for patterns like ssh or http.**

**A computer screen with white text

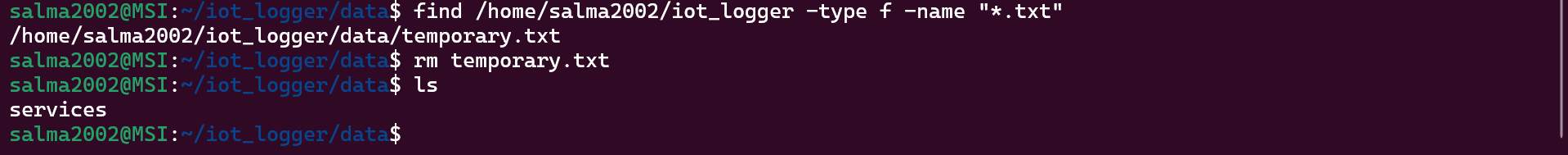
AI-generated content may be incorrect.**

**• Use regex to find lines starting with t or containing numbers.**

****

**• Locate .txt files in /home/ and remove temporary ones if needed.**

****

****

**• Create hard and symbolic links for temperature.log.**

**A computer screen shot of white text

AI-generated content may be incorrect.**

**• Display directory structure to confirm organization.**

**A computer screen with white text

AI-generated content may be incorrect.**

**Open-Ended Questions:**

**• Explain the different types of files in Linux (regular, directory, symbolic link, device, etc.) and how to check them with commands.**

**1. Regular Files**

Regular files are the most common type, used to store data. They can be:

* Text files → Contain human-readable characters (e.g., .txt, source code files).
* Binary files → Contain compiled programs or other non-readable data (e.g., executables).
* Media files → Store multimedia content like images, videos, and music (e.g., .jpg, .mp4).

**2. Directory Files**

A directory is a special type of file that holds references to other files or subdirectories.

* Example: /home/user/ contains a user’s files and folders.

**3. Symbolic Links (Symlinks)**

A symbolic link is a pointer (shortcut) to another file or directory.

* It doesn’t contain data itself, only the path to the target.
* Example: /usr/bin/python -> /usr/bin/python3.11
* If the target is deleted, the symlink becomes broken (dangling).

**4. Character Device Files**

Character device files represent devices that transfer data one character at a time.

* Common examples: keyboards, mice, and serial ports.
* Typically found in /dev/.
* Example: /dev/input/mouse2
* Can be created using the mknod command.

**5. Block Device Files**

Block device files represent devices that handle data in fixed-size blocks.

* Commonly used for disks and storage devices.
* Found in /dev/.
* Example: /dev/sda1 (a partition on a hard disk).

**6. FIFO (Named Pipes)**

FIFO files are used for inter-process communication (IPC).

* They allow processes to pass data so that it is read in the same first-in, first-out order it was written.
* Created with the mkfifo command.
* Example:
* mkfifo mypipe

**How to Check File Types**

Linux provides several ways to check file types:

* ls -l → Displays file type in the first character (- = regular, d = directory, l = symlink, c = character device, b = block device, p = pipe).

**Reference:** [**Types of files in Linux**](https://medium.com/@naskararijit212/types-of-files-in-linux-7092aef006d5?utm_source=chatgpt.com)

**• What’s the difference between a hard link and a symbolic link? Give real examples of when to use each.**

In Linux, files are represented by inodes, which store metadata and point to the actual file data on disk. A file in the filesystem is essentially a name (link) that refers to an inode.

**Hard Links**

* A hard link is an additional filename that points directly to the same inode as the original file.
* Deleting the original file does not remove the data, because the inode remains as long as at least one link exists.
* Renaming or moving the original file also does not affect the hard link, since both names point to the same inode.
* Any modification to the data in the inode is reflected in all hard-linked files.
* Limitation: Hard links can only exist within the same filesystem.

**Example use case:**When you need multiple filenames pointing to the same physical data, such as maintaining different references to a configuration file without duplicating storage.

**Symbolic Links (Symlinks)**

* A symbolic link is a special file that stores the path to another file.
* Unlike hard links, a symlink does not point directly to the inode.
* If the target file is deleted, renamed, or moved, the symlink becomes a broken link (dangling).
* Symlinks can span across different filesystems because they reference names, not inodes.

**Example use case:**When you want a shortcut or alias to a file or directory, such as linking /etc/nginx/sites-enabled/ to files in /etc/nginx/sites-available/.

**A diagram of a computer network

AI-generated content may be incorrect.**

**Reference:**[**What is the difference between a symbolic link and a hard link? – Stack Overflow**](https://stackoverflow.com/questions/185899/what-is-the-difference-between-a-symbolic-link-and-a-hard-link?utm_source=chatgpt.com)

**• Is rmdir the same as rm -r when deleting directories? Explain**

No, they are not the same.

* rmdir: This command can only delete a directory if it is completely empty. If the directory contains any files or subdirectories, the command will fail.
* rm -r: This command recursively deletes a directory along with all its contents (files and subdirectories). It is more powerful and also more dangerous because it removes everything without requiring the directory to be empty.

**Reference:** [**Difference between rmdir and rm -r**](https://unix.stackexchange.com/questions/374804/difference-between-rmdir-and-rm-r?utm_source=chatgpt.com)