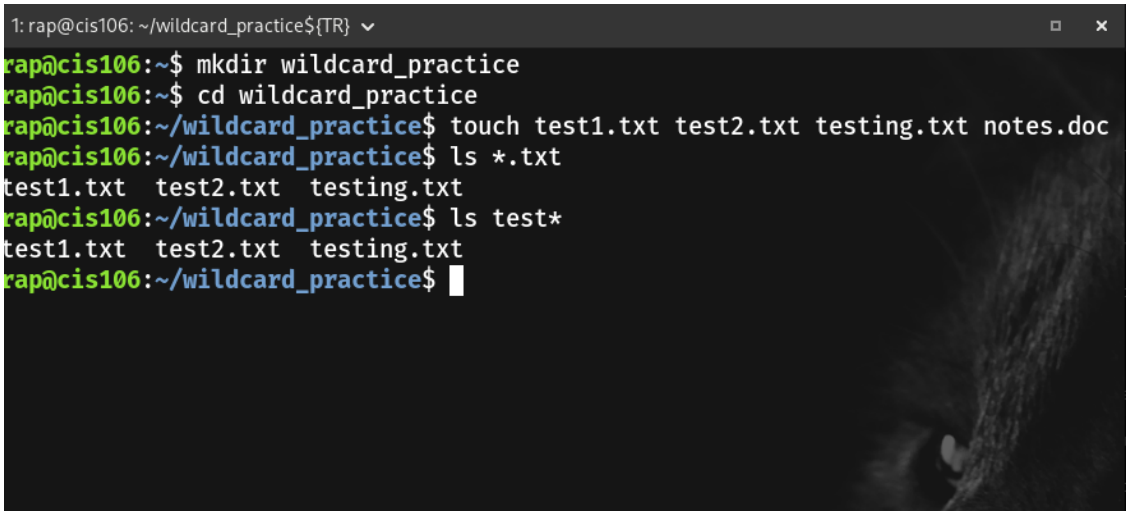
Week Report 7

1. Wildcards

- Wildcard: * (Asterisk)
  - Definition: Matches zero or more characters. It is the most flexible wildcard.
  - Examples: ls *.txt → lists all files ending in .txt
    - rm test* → deletes any file that starts with test (ex: test1, testfile, testing.txt)
    - cp * /backup/ → copies everything in the current directory

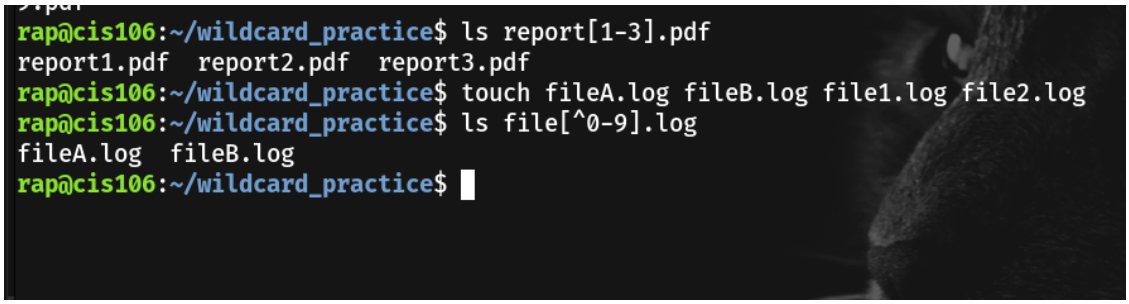


- Wildcard: ? (Question Mark)
  - Definition: Matches exactly one character.
  - Examples:
    - ls file?.txt → matches file1.txt, fileA.txt, but not file10.txt
    - mv photo?.jpg images/ → moves files like photo1.jpg, photo2.jpg
    - cat log?.txt → matches logs log1.txt, log2.txt, etc.

- ☐ (Character Class)
  - Definition: Matches one character from a list or range.
  - Examples:
    - ls report[123].pdf → matches report1.pdf, report2.pdf, report3.pdf

    - rm file[a-c].txt → deletes filea.txt, fileb.txt, filec.txt
    - cp img[0-5].png pics/ → matches images numbered 0–5

- Wildcard: [^] (Negated Character Class)
    - Definition: Matches one character NOT inside the brackets.
    - Examples:
        - ls file[^0-9].txt → matches fileA.txt, fileB.txt, but not file1.txt
        - rm test[^ab].log → deletes logs not ending in a or b
        - mv doc[^s].md backup/ → moves files not ending in s



- Brace Expansion
    - Definition:

      Brace expansion generates multiple strings before a command runs. It is used to quickly create many files or directories, including full directory trees.

        - Example 1: Create multiple directories
        - mkdir project/{docs,images,scripts}

- Creates:

  project/docs

  project/images

project/scripts

```
rap@cis106:~/wildcard_practice$ mkdir project/{docs,images,scripts}
mkdir: cannot create directory ‘project/docs’: No such file or directory
mkdir: cannot create directory ‘project/images’: No such file or directory
mkdir: cannot create directory ‘project/scripts’: No such file or directory
rap@cis106:~/wildcard_practice$ ls project
ls: cannot access 'project': No such file or directory
rap@cis106:~/wildcard_practice$ ls project
ls: cannot access 'project': No such file or directory
rap@cis106:~/wildcard_practice$
```

- Example 2: Create entire folder structures mkdir -p music/{jazz,rock,pop}/{mp3,ogg,flac}

Creates:

music/ ├─ jazz/ │ ├─ mp3 │ ├─ ogg │ └─ flac ├─ rock/ │ ├─ mp3 │ ├─ ogg │ └─ flac └─ pop/ ├─ mp3 ├─ ogg └─ flac

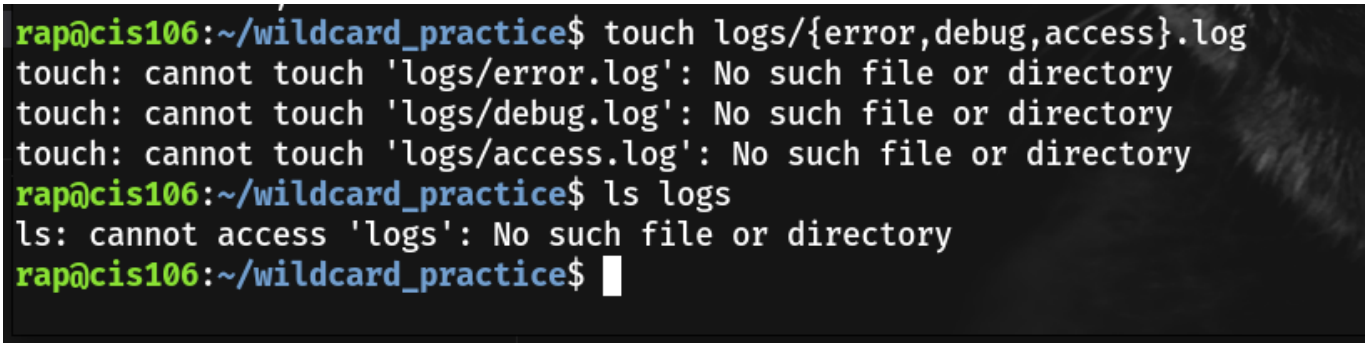
```
rap@cis106:~/wildcard_practice$ mkdir -p music/{jazz,rock,pop}/{mp3,ogg,flac}
rap@cis106:~/wildcard_practice$ tree music
music
├── jazz
│   ├── flac
│   ├── mp3
│   └── ogg
├── pop
│   ├── flac
│   ├── mp3
│   └── ogg
└── rock
    ├── flac
    ├── mp3
    └── ogg

13 directories, 0 files
rap@cis106:~/wildcard_practice$
```

- Example 3: Create multiple files at once touch logs/{error,debug,access}.log
- Creates:

error.log

debug.log

access.log


rap@cis106:~/wildcard_practice$ touch logs/{error,debug,access}.log
touch: cannot touch 'logs/error.log': No such file or directory
touch: cannot touch 'logs/debug.log': No such file or directory
touch: cannot touch 'logs/access.log': No such file or directory
rap@cis106:~/wildcard_practice$ ls logs
ls: cannot access 'logs': No such file or directory
rap@cis106:~/wildcard_practice$