



Homework 1: Linux Command Line Basics

Môn học: Hệ điều hành Linux và ứng dụng

CS111117 - 22MMT

Sinh viên: Nguyễn Hồ Đăng Duy - 22127085

Table of Contents

- [Task 1: System Investigation](#)
- [Task 2: Directory Setup](#)
- [Task 3: Log Review](#)
- [Task 4: Pattern Search](#)
- [Task 5: Stream Redirection](#)
- [Task 6: Text Processing](#)
- [Task 7: Cleanup Script](#)
- [Task 8: File Hunting](#)
- [Task 9: Joining Data](#)
- [Task 10: Line Numbering & Word Counts](#)

Task 1: System Investigation

1. Find out what directory you are currently in.

```
pwd
```

Explanation:

- `pwd` (print working directory) prints the full path of the current directory. It helps users know their current location in the file system. **Screenshot:**

```
(denver@kali)-[~]  
$ pwd  
/home/denver
```

2. Go to the Documents directory inside your home directory using a relative path.

```
cd ~/Documents
```

Explanation:

- `cd ~/Documents` changes to the `Documents` directory located in the current user's home directory (`~` stands for `/home/denver`).

Screenshot:

```
(denver@kali)-[~]  
$ cd ~/Documents  
  
(denver@kali)-[~/Documents]  
$
```

3. Confirm your new location.

```
pwd
```

Explanation:

- `pwd` is used again to confirm that the new current directory is `/home/denver/Documents`.

Screenshot:

```
(denver@kali)-[~/Documents]  
$ pwd  
/home/denver/Documents
```

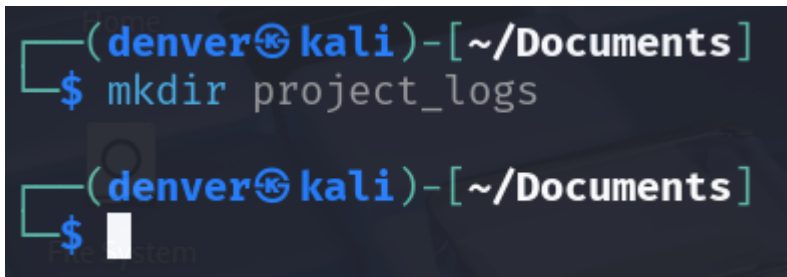
Task 2: Directory Setup

1. Create a folder named `project_logs` in your current directory.

```
mkdir project_logs
```

Explanation:

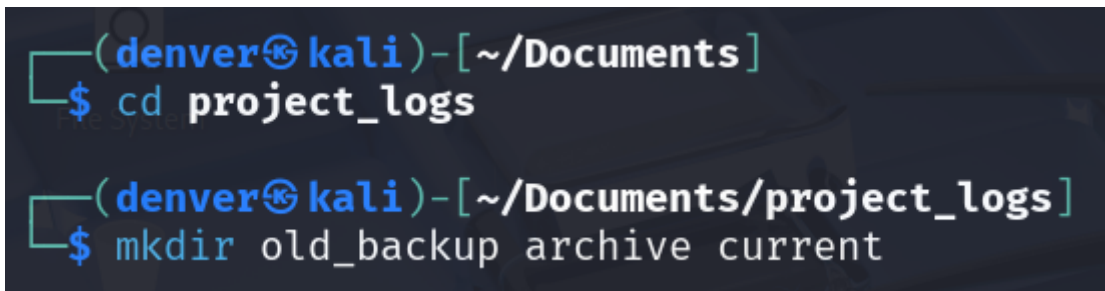
- `mkdir project_logs` creates a folder named `project_logs` in the current directory..

Screenshot:A terminal window with a dark background. The prompt is `(denver@kali)-[~/Documents]`. The user enters `$ mkdir project_logs`. The prompt changes to `(denver@kali)-[~/Documents]` and the user enters `$` followed by a cursor.**2. Inside it, create subfolders `old_backup`, `archive`, and `current`.**

```
cd project_logs
mkdir old_backup archive current
```

Explanation:

- `cd project_logs` navigates into the newly created folder.
- `mkdir old_backup archive current` creates three subfolders at once.

Screenshot:A terminal window with a dark background. The prompt is `(denver@kali)-[~/Documents]`. The user enters `$ cd project_logs`. The prompt changes to `(denver@kali)-[~/Documents/project_logs]`. The user enters `$ mkdir old_backup archive current`.**3. Then, create a file `temp.txt`.**

```
touch temp.txt
```

Explanation:

- `touch temp.txt` creates an empty file named `temp.txt`.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ touch temp.txt

(denver@kali)-[~/Documents/project_logs]
$
```

4. Verify that they exist.

```
ls -l
```

Explanation:

- `ls -l` lists the files and directories in detail to confirm they were created successfully.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ ls -l
total 12
drwxrwxr-x 2 denver denver 4096 May 21 11:15 archive
drwxrwxr-x 2 denver denver 4096 May 21 11:15 current
drwxrwxr-x 2 denver denver 4096 May 21 11:15 old_backup
-rw-rw-r-- 1 denver denver  0 May 21 11:15 temp.txt
```

Task 3: Log Review

1. You received a system log file at `/var/log/syslog`.

```
sudo touch /var/log/syslog
sudo tee /var/log/syslog > /dev/null <<EOF
```

Explanation:

- `Kali Linux` doesn't have log file at `/var/log/syslog` so I created a temporary file to do this task

Screenshot:

```
(denver@kali)-[~]
$ sudo touch /var/log/syslog
[sudo] password for denver:

(denver@kali)-[~]
$ sudo tee /var/log/syslog > /dev/null <<EOF
Apr 01 10:00:00 denver systemd[1]: Started Session 1 of user root.
Apr 01 10:05:01 denver CRON[1234]: (root) CMD (apt update)
Apr 01 10:07:22 denver kernel: [123456.789] device eth0 entered promiscuous mode
Apr 01 10:15:43 denver systemd[1]: Stopping User Manager...
Apr 01 10:20:33 denver sshd[5678]: Failed password for invalid user admin from 192.168.1.100 port 5555
Apr 01 10:25:10 denver sudo[9999]: pam_unix(sudo:session): session opened for user root by user(uid=0)
Apr 01 10:30:42 denver app[7777]: ERROR: cannot connect to database
Apr 01 10:32:00 denver kernel: [123789.123] eth0: link down
Apr 01 10:33:14 denver app[8888]: ERROR: missing configuration file
Apr 01 10:35:00 denver systemd[1]: Reached target Shutdown.
EOF
```

2. Display the first 5 lines of the file.

```
head -n 5 /var/log/syslog
```

Explanation:

- `head -n 5` shows the first 5 lines of the file to give an overview of its beginning.

Screenshot:

```
(denver@kali)-[~]
$ head -n 5 /var/log/syslog
Apr 01 10:00:00 denver systemd[1]: Started Session 1 of user root.
Apr 01 10:05:01 denver CRON[1234]: (root) CMD (apt update)
Apr 01 10:07:22 denver kernel: [123456.789] device eth0 entered promiscuous mode
Apr 01 10:15:43 denver systemd[1]: Stopping User Manager...
Apr 01 10:20:33 denver sshd[5678]: Failed password for invalid user admin from 192.168.1.100 port 5555
```

3. Display the last 10 lines of the file.

```
tail -n 10 /var/log/syslog
```

Explanation:

- `tail -n 10` displays the last 10 lines, usually showing the most recent logs.

Screenshot:

```
(denver@kali)-[~]
$ tail -n 10 /var/log/syslog

Apr 01 10:00:00 denver systemd[1]: Started Session 1 of user root.
Apr 01 10:05:01 denver CRON[1234]: (root) CMD (apt update)
Apr 01 10:07:22 denver kernel: [123456.789] device eth0 entered promiscuous mode
Apr 01 10:15:43 denver systemd[1]: Stopping User Manager...
Apr 01 10:20:33 denver sshd[5678]: Failed password for invalid user admin from 192.168.1.100 port 5555
Apr 01 10:25:10 denver sudo[9999]: pam_unix(sudo:session): session opened for user root by user(uid=0)
Apr 01 10:30:42 denver app[7777]: ERROR: cannot connect to database
Apr 01 10:32:00 denver kernel: [123789.123] eth0: link down
Apr 01 10:33:14 denver app[8888]: ERROR: missing configuration file
Apr 01 10:35:00 denver systemd[1]: Reached target Shutdown.
```

4. Reverse the content of the file for a quick scan.

```
tac /var/log/syslog
```

Explanation:

- **tac** (reverse **cat**) displays the file from bottom to top, which can be helpful when viewing recent logs without scrolling.

Screenshot:

```
(denver@kali)-[~]
$ tac /var/log/syslog

Apr 01 10:35:00 denver systemd[1]: Reached target Shutdown.
Apr 01 10:33:14 denver app[8888]: ERROR: missing configuration file
Apr 01 10:32:00 denver kernel: [123789.123] eth0: link down
Apr 01 10:30:42 denver app[7777]: ERROR: cannot connect to database
Apr 01 10:25:10 denver sudo[9999]: pam_unix(sudo:session): session opened for user root by user(uid=0)
Apr 01 10:20:33 denver sshd[5678]: Failed password for invalid user admin from 192.168.1.100 port 5555
Apr 01 10:15:43 denver systemd[1]: Stopping User Manager...
Apr 01 10:07:22 denver kernel: [123456.789] device eth0 entered promiscuous mode
Apr 01 10:05:01 denver CRON[1234]: (root) CMD (apt update)
Apr 01 10:00:00 denver systemd[1]: Started Session 1 of user root.
```

Task 4: Pattern Search

1. Search for all lines that contain the word **error** (case-insensitive).

```
grep -i "error" /var/log/syslog
```

Explanation:

- **grep -i** searches for the word **error** without case sensitivity and redirects the output to **errors.log**.

Screenshot:

```
(denver@kali)-[~]  
$ grep -i "error" /var/log/syslog  
Apr 01 10:30:42 denver app[7777]: ERROR: cannot connect to database  
Apr 01 10:33:14 denver app[8888]: ERROR: missing configuration file
```

2. Count how many such lines exist.

```
grep -i "error" /var/log/syslog > errors.log  
wc -l errors.log
```

Explanation:

- Redirect the output of `grep -i "error" /var/log/syslog` to `errors.log`.
- `wc -l` counts the number of lines in `errors.log`, which is the number of matching lines.

Screenshot:

```
(denver@kali)-[~]  
$ grep -i "error" /var/log/syslog > errors.log  
  
(denver@kali)-[~]  
$ wc -l errors.log  
2 errors.log
```

Task 5: Stream Redirection

1. Redirect the list of all files in `/etc` into a file named `etc_list.txt` without showing it on screen.

```
ls /etc > etc_list.txt
```

Explanation:

- The `ls /etc` list all files in `/etc`.
- The `>` operator saves the output to `etc_list.txt` and does not display it in the terminal.

Screenshot:



```
(denver@kali)-[~]  
$ ls /etc > etc_list.txt  
  
(denver@kali)-[~]  
$
```

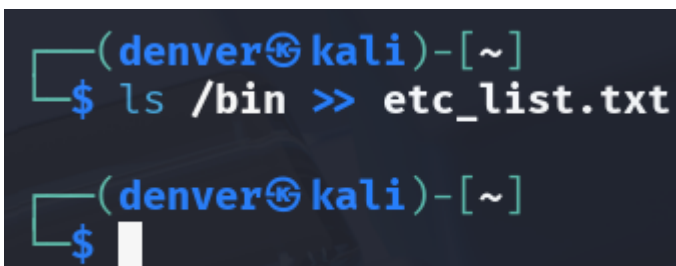
2. Append the list of files from `/bin` to the same file.

```
ls /bin >> etc_list.txt
```

Explanation:

- `>>` appends the output to `etc_list.txt` without overwriting the previous content.

Screenshot:



```
(denver@kali)-[~]  
$ ls /bin >> etc_list.txt  
  
(denver@kali)-[~]  
$
```

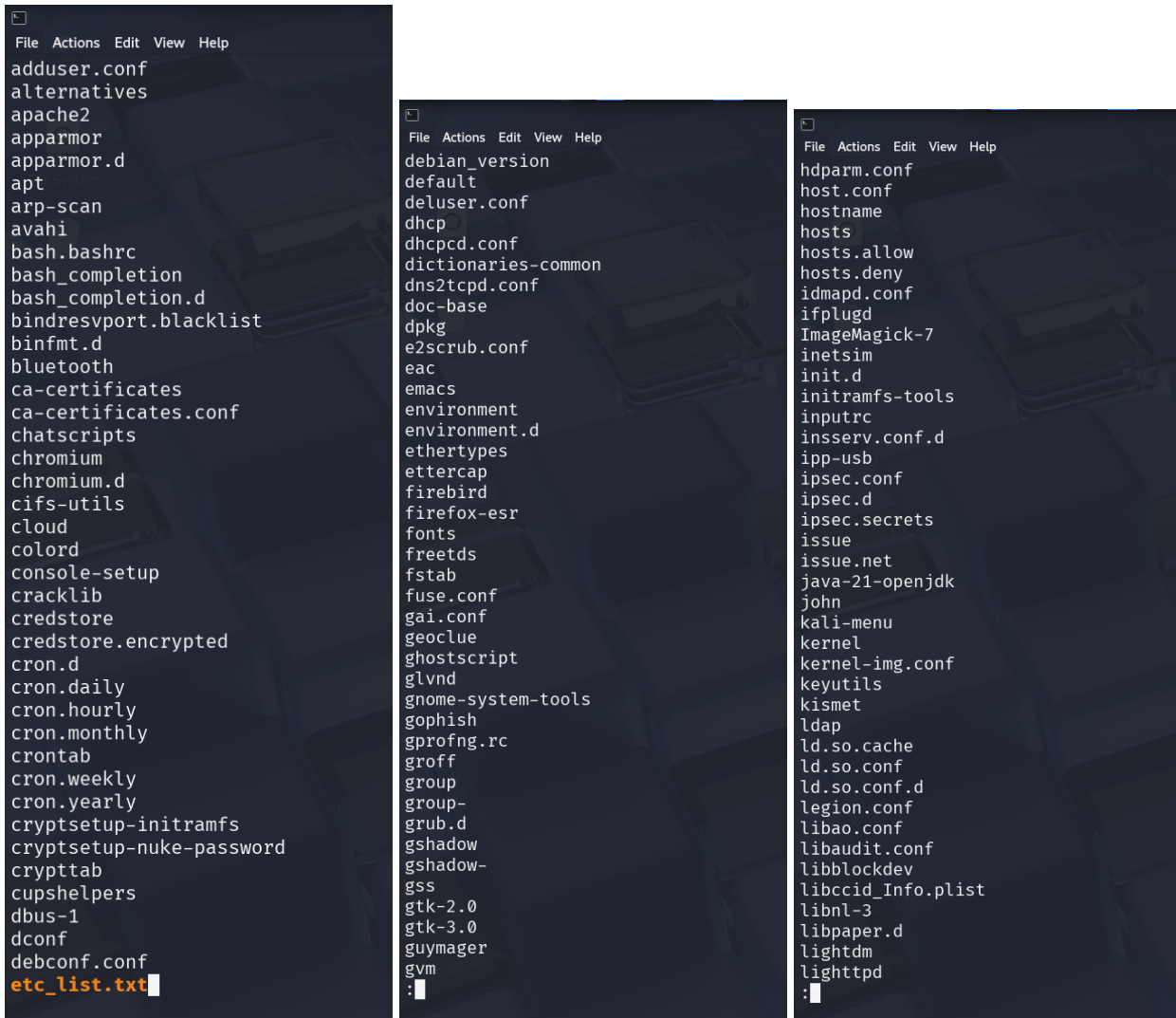
3. View the result one page at a time.

```
less etc_list.txt
```

Explanation:

- `less` allows viewing a file one screen at a time with scroll support.

Screenshot: The content of `etc_list.txt` is longer than 3 images below.



Task 6: Text Processing

1. Create a file `users.txt` with the content:

```
Alice:Admin
Bob:User
Carol:Guest
```

```
echo -e "Alice:Admin\nBob:User\nCarol:Guest" > users.txt
```

Explanation:

- `echo -e` prints multiple lines into `users.txt`, using `\n` to break lines.

Screenshot:

```
(denver@kali)-[~]
$ echo -e "Alice:Admin\nBob:User\nCarol:Guest" > users.txt

(denver@kali)-[~]
$ cat users.txt
Alice:Admin
Bob:User
Carol:Guest
```

2. Use `cut` or `awk` to extract usernames.

```
cut -d ":" -f 1 users.txt
```

Explanation:

- `cut -d ":" -f 1` extracts the part before `:` on each line:
 - `-d` stands for **delimiter**. In this case, the delimiter is a colon `:`, so each line will be split at every `:`.
 - `-f` stands for **field**. `-f 1` means take the first field - the part before the first `:` in each line
 - `users.txt` is the file that contains the data to be processed. **Screenshot:**

```
(denver@kali)-[~]
$ cut -d ":" -f 1 users.txt
Alice
Bob
Carol
```

3. Use `tr` to convert them to lowercase.

```
cut -d ":" -f 1 users.txt | tr 'A-Z' 'a-z'
```

Explanation:

- `|` is called the pipe operator. It is used to connect the output of one command to the input of another command.
- `tr 'A-Z' 'a-z'` converts all uppercase letters to lowercase.

Screenshot:

```
(denver@kali)-[~]  
$ cut -d ":" -f 1 users.txt | tr 'A-Z' 'a-z'  
alice  
bob  
carol
```

4. Use `sort` to display in reverse alphabetical order.

```
cut -d ":" -f 1 users.txt | tr 'A-Z' 'a-z' | sort -r
```

Explanation:

- `sort -r` sorts the list in reverse alphabetical order (Z to A).

Screenshot:

```
(denver@kali)-[~]  
$ cut -d ":" -f 1 users.txt | tr 'A-Z' 'a-z' | sort -r  
carol  
bob  
alice
```

Task 7: Cleanup Script

1. Remove an empty directory named `old_backup`.

```
rmdir old_backup
```

Explanation:

- `rmdir` removes empty directories. It will fail if the directory is not empty.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current  old_backup  temp.txt

(denver@kali)-[~/Documents/project_logs]
$ rmdir old_backup

(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current  temp.txt
```

2. Delete a file named `temp.txt`.

```
rm temp.txt
```

Explanation:

- `rm` deletes a file in the Linux system.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current  temp.txt

(denver@kali)-[~/Documents/project_logs]
$ rm temp.txt

(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current
```

3. Move a file `report.txt` to the `archive` folder.

```
mv report.txt archive/
```

Explanation:

- `mv` moves `report.txt` into the `archive/` folder.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ touch report.txt

(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current  report.txt

(denver@kali)-[~/Documents/project_logs]
$ mv report.txt archive/

(denver@kali)-[~/Documents/project_logs]
$ ls
archive  current

(denver@kali)-[~/Documents/project_logs]
$ ls archive/
report.txt
```

4. Copy `archive/report.txt` to `current/report_backup.txt`

```
cp archive/report.txt current/report_backup.txt
```

Explanation:

- `cp` copies the file from `archive` to `current` with a new name `report_backup.txt`.

Screenshot:

```
(denver@kali)-[~/Documents/project_logs]
$ cp archive/report.txt current/report_backup.txt

(denver@kali)-[~/Documents/project_logs]
$ ls current/
report_backup.txt
```

Task 8: File Hunting

1. Find all `.conf` files under `/etc`.

```
find /etc -type f -name "*.conf"
```

Explanation:

- `find` with `-type f` and `-name` finds all files ending in `.conf` under `/etc`:
 - `find` used to search files and directories in a directory hierarchy based on various criteria (name, type, permissions, size, etc.).
 - `/etc` is the starting point of the search.
 - `-type f` limits the search to **files** only.
 - `-name "*.conf"` filter the search by **filename pattern**.

Screenshot:

```
(denver@kali)-[~]
$ find /etc -type f -name "*.conf"
/etc/inetsim/inetsim.conf
/etc/debconf.conf
/etc/modprobe.d/ath9k_htc.conf
/etc/modprobe.d/kali-defaults.conf
/etc/modprobe.d/amd64-microcode-blacklist.conf
/etc/modprobe.d/intel-microcode-blacklist.conf
/etc/smi.conf
/etc/dhcpd.conf
/etc/gvm/gvmd_log.conf
/etc/gvm/pwpolicy.conf
/etc/gvm/ospd-openvas.conf
find: '/etc/ipsec.d/private': Permission denied
/etc/request-key.conf
find: '/etc/vpnc': Permission denied
/etc/netsniff-ng/oui.conf
/etc/netsniff-ng/tcp.conf
/etc/netsniff-ng/geoip.conf
/etc/netsniff-ng/udp.conf
/etc/netsniff-ng/ether.conf
/etc/NetworkManager/NetworkManager.conf
/etc/responder/Responder.conf
find: '/etc/redis': Permission denied
/etc/libao.conf
/etc/nfs.conf
/etc/gtk-3.0/im-multipress.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-lgc-sans-mono.conf
/etc/fonts/conf.avail/30-droid-oto-mono.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-lgc-serif.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-sans.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-serif.conf
/etc/fonts/conf.avail/58-dejavu-lgc-sans-mono.conf
/etc/fonts/conf.avail/65-fonts-lmodern.conf
/etc/fonts/conf.avail/65-droid-sans-fallback.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-sans-mono.conf
/etc/fonts/conf.avail/20-unhint-small-dejavu-lgc-sans.conf
/etc/fonts/conf.avail/57-dejavu-sans.conf
/etc/fonts/conf.avail/58-dejavu-lgc-serif.conf
/etc/fonts/conf.avail/57-dejavu-sans-mono.conf
```

2. Find all files with 755 permission in your home directory.

```
find ~ -type f -perm 755
```

Explanation:

- Searches for files (**-type f**) in the home directory (**~**) with permission **755**:
 - **~** home directory
 - **-perm 755** tells the **find** to match files with specific permission mode:
 - **755** means:
 - Owner: read, write, execute (**7 = rwx**)
 - Group: read, execute (**5 = r-x**)
 - Others: read, execute (**5 = r-x**)

Screenshot:

```
(denver@kali)-[~]  
$ find ~ -type f -perm 755  
/home/denver/.local/share/nautilus/scripts/Terminal  
  
(denver@kali)-[~]  
$
```

3. Find all directories under **/usr that contain the word **lib****

```
find /usr -type d -name "*lib*"
```

Explanation:

- Finds all directories (**-type d**) with names containing **lib** under **/usr**:
 - **-type d** find directories only
 - ***lib*** is a **glob pattern** (wildcard-based string pattern). It means:
 - ***** matches zero or more characters.
 - **lib** is the literal string being searched for
 - ***lib*** matches any name that contains the word **lib** anywhere.

Screenshot:


```
(denver@kali)-[~]
$ find /usr -type d -name "*lib*"
/usr/lib
/usr/lib/firmware/libertas
/usr/lib/firmware/intel/sof-ipc4-lib
/usr/lib/postgresql/17/lib
/usr/lib/postgresql/17/lib/bitcode/postgres/lib
/usr/lib/postgresql/17/lib/bitcode/postgres/libpq
/usr/lib/llvm-19/lib
/usr/lib/llvm-19/lib/clang/19/lib
/usr/lib/llvm-19/lib/clang/19/include/llvm_libc_wrappers
/usr/lib/gcc/i686-w64-mingw32/13-win32/adalib
/usr/lib/gcc/x86_64-w64-mingw32/13-win32/adalib
/usr/lib/python3/dist-packages/dploit/lib
/usr/lib/python3/dist-packages/reportlab-stubs/lib
/usr/lib/python3/dist-packages/win32-stubs/lib
/usr/lib/python3/dist-packages/sgmllibk-1.0.0.egg-info
/usr/lib/python3/dist-packages/types_html5lib-1.1.dist-info
/usr/lib/python3/dist-packages/zlib_wrapper
/usr/lib/python3/dist-packages/mpmath/libmp
/usr/lib/python3/dist-packages/types_httplib-2.022.dist-info
/usr/lib/python3/dist-packages/matplotlib-3.8.3.egg-info
/usr/lib/python3/dist-packages/importlib_metadata-8.0.0.dist-info
/usr/lib/python3/dist-packages/setuputils/_vendor/importlib_metadata
/usr/lib/python3/dist-packages/playwright/driver/package/lib
/usr/lib/python3/dist-packages/types_polib-1.2.dist-info
/usr/lib/python3/dist-packages/passlib-stubs
/usr/lib/python3/dist-packages/msldap/external/bloodhoundpy/lib
/usr/lib/python3/dist-packages/types_oauthlib-3.2.dist-info
/usr/lib/python3/dist-packages/httplib2-stubs
/usr/lib/python3/dist-packages/build/lib
/usr/lib/python3/dist-packages/fontTools/misc/plistlib
/usr/lib/python3/dist-packages/macholib
/usr/lib/python3/dist-packages/zlib_wrapper-0.1.3.egg-info
/usr/lib/python3/dist-packages/types_passlib-1.7.dist-info
/usr/lib/python3/dist-packages/rumel.yaml.lib-0.2.12.dist-info
/usr/lib/python3/dist-packages/scipy/lib
/usr/lib/python3/dist-packages/pyqtgraph/flowchart/library
/usr/lib/python3/dist-packages/importlib_metadata-8.6.1.dist-info
/usr/lib/python3/dist-packages/matplotlib_inline-0.1.6.egg-info
/usr/lib/python3/dist-packages/theHarvester/lib
/usr/lib/python3/dist-packages/Xlib-stubs
/usr/lib/python3/dist-packages/importlib_metadata
/usr/lib/python3/dist-packages/lasgger/ui5/static/lib
/usr/lib/python3/dist-packages/passlib-1.7.4.egg-info
/usr/lib/python3/dist-packages/types_python_xlib-0.33.dist-info
/usr/lib/python3/dist-packages/sympy/plotting/backends/matplotlibbackend
/usr/lib/python3/dist-packages/passlib
/usr/lib/python3/dist-packages/certipy/lib
/usr/lib/python3/dist-packages/scapy/libs
/usr/lib/python3/dist-packages/hamcrest/library
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/third_party/2/six/moves/urllib
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/third_party/3/six/moves/urllib
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/third_party/2and3/requests/packages/urllib3
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/stdlib/3/urllib
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/stdlib/3/importlib
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/stdlib/2and3/msilib
/usr/lib/python3/dist-packages/jedi/third_party/typeshed/stdlib/2and3/lib2to3
/usr/lib/python3/dist-packages/python_libmmap-0.7.3.egg-info
/usr/lib/python3/dist-packages/wafw00f/lib
/usr/lib/python3/dist-packages/html5lib
/usr/lib/python3/dist-packages/Python/lib
/usr/lib/python3/dist-packages/pandas/tests/tslibs
/usr/lib/python3/dist-packages/pandas/tests/libs
/usr/lib/python3/dist-packages/pandas/_libs
/usr/lib/python3/dist-packages/pandas/_libs/tslibs
/usr/lib/python3/dist-packages/pandas/plotting/matplotlib
/usr/lib/python3/dist-packages/matplotlib_inline
/usr/lib/python3/dist-packages/bloodhound/lib
/usr/lib/python3/dist-packages/pip/_internal/metadata/importlib
/usr/lib/python3/dist-packages/pip/_internal/resolution/resolvelib
/usr/lib/python3/dist-packages/pip/_vendor/resolvelib
/usr/lib/python3/dist-packages/pip/_vendor/urllib3
/usr/lib/python3/dist-packages/pip/_vendor/distlib
/usr/lib/python3/dist-packages/oauthlib-stubs
/usr/lib/python3/dist-packages/dnslib-0.9.25.egg-info
/usr/lib/python3/dist-packages/types_requests_oauthlib-2.0.dist-info
```

Task 9: Joining Data

1. Create file `names.txt`

```
echo -e "1 Alice\n2 Bob\n3 Carol" > names.txt
```

Explanation:

- Uses `echo` to simulate a list of IDs and names.

Screenshot:

```
(denver@kali)-[~]
$ echo -e "1 Alice\n2 Bob\n3 Carol" > names.txt

(denver@kali)-[~]
$ cat names.txt
1 Alice
2 Bob
3 Carol
```

2. Create file `roles.txt`

```
echo -e "1 Admin\n2 User\n3 Guest" > roles.txt
```

Explanation:

- Similar to `names.txt`, but for corresponding roles.

Screenshot:

```
(denver@kali)-[~]
$ echo -e "1 Admin\n2 User\n3 Guest" > roles.txt

(denver@kali)-[~]
$ cat roles.txt
1 Admin
2 User
3 Guest
```

3. Join them into one file with both name and role.

```
join names.txt roles.txt > names_roles.txt
```

Explanation:

- `join names.txt roles.txt`: merges the two files based on the first field (ID).
- `>` redirects the result into a new file named `names_roles.txt`.

Screenshot:

```
(denver@kali)-[~]
$ join names.txt roles.txt > names_roles.txt

(denver@kali)-[~]
$ cat names_roles.txt
1 Alice Admin
2 Bob User
3 Carol Guest
```

Task 10: Line Numbering & Word Counts

1. Use `nl` to number the lines of `users.txt`

```
nl users.txt
```

Explanation:

- `nl` stands for number lines. It displays the contents of the file while automatically adding line numbers at the beginning of each line.

Screenshot:

```
(denver@kali)-[~]  
$ nl users.txt  
1 Alice:Admin  
2 Bob:User  
3 Carol:Guest  
  
(denver@kali)-[~]  
$
```

2. Use `wc` to count number of lines, words, and bytes

```
wc users.txt
```

Explanation:

- `wc` stands for word count. It provides three statistics for the specified file:
 - Number of lines.
 - Number of words.
 - Number of bytes (or characters).

Screenshot:

```
(denver@kali)-[~]  
$ wc users.txt  
3  3 33 users.txt
```