**Understand the Commands**

**File Listing & Management**

ls -ltr

📌 Description: Lists files in the current directory with:

* -l: long listing format (permissions, owner, size, date)
* -t: sorted by modification time (newest first)
* -r: reverse order (oldest first)

ls -ltr

mv old\_name new\_name

📌 Description: Renames or moves files.

mv NSBPay.jar NSBPay\_stable.jar

mv /home/asela/NSBPay-1.0.0-SNAPSHOT.jar NSPay.jar

jar -uf filename.jar path/to/file

📌 Description: Updates a .jar file by replacing or adding a file inside it.

jar uf NSBPay.jar BOOT-INF/classes/application.properties

**Service Management**

systemctl restart servicename

📌 Description: Restarts a service

systemctl restart nsb-app.service

**Log Viewing**

tail -f logs/spring.log

📌 Description: Continuously watches the end of a file

tail -f mylog.txt

**Practice Commands**

**Create a Practice Directory**

mkdir ~/jar-practice

cd ~/jar-practice

**Create Dummy Files**

touch NSPay.jar NSPay\_stable.jar

mkdir -p BOOT-INF/classes

echo "dummy=config" > BOOT-INF/classes/application.properties

**Try the Commands jar**

jar uf NSPay.jar BOOT-INF/classes/application.properties

**Rename JAR File**

mv NSPay.jar NSPay\_stable.jar

mv NSPay\_stable.jar NSPay.jar

**What tail Does**

* tail shows the **last lines of a file**.
* tail -f follows a file **live** — updates as the file are written to

**Practice tail Command**

**Create a Dummy Log File**

mkdir ~/log-practice

cd ~/log-practice

echo "Log line 1" > mylog.txt

**Add More Lines**

echo "Log line 2" >> mylog.txt

echo "Log line 3" >> mylog.txt

**Show N Number of Lines**

tail -n 2 mylog.txt

show two lines

**Live View**

tail -f mylog.txt

**50 lines to a file and then view the last 20 lines using tail.**

**Create or Clear the File**

> mylog.txt

**Add 50 Lines Using a Loop**

for i in {1..50}; do echo "This is log line $i" >> mylog.txt; done

**View the Last 20 Lines**

tail -n 20 mylog.txt

**Monitor File While Adding Lines**

**A live viewer in one terminal**

tail -f mylog.txt

**Another terminal, run**

for i in {51..100}; do echo "This is log line $i" >> mylog.txt; sleep 0.5; done

This simulates live logs being written.