姜海天 19307110022

```
# E1. Use the command uname to simultaneously show the name and version of the system
uname -sv

# E2. Use the command echo to show the full path of your home directory
echo $HOME

# E3. Use the command whoami to show the current login user.
whoami

# E4. Use the command chmod to make the file 1.sh in the current directory executable.
chmod +x 1.sh

# E5. Show the disk usage of the current directory, with depth 1, and human readable output.
du -hd 1

# E6. Use ssh to connect to 10.19.248.12, with user guest.
ssh guest@10.19.248.12

# E7. Use apt-get to install jq.
sudo apt install jq -y

# E8. Display the absolute path of the current directory.
pwd

# E9. Make a new file named 1.sh, with content: echo “hello world”.
echo 'echo "hello world"' > 1.sh
# but I will usually use `>1.sh`, and then type `echo "hello world"`, then <Enter> and <Ctrl-D> (this
only works in zsh, not bash)

# E10.Change the name of file 1.sh to 2.sh.
mv {1,2}.sh  # it is shorter form for mv 1.sh 2.sh

# E11.Make a new folder in current directory, named folder1.
mkdir folder1

# E12.Copy 3.sh in current directory to folder1, still named 3.sh.
cp 3.sh folder1/

# E13.Use the command cat to obtain the content of 2.sh and redirect the output to 3.sh.
cat 2.sh >> 3.sh  # in case 3.sh is nonempty

# E14.Create two new txt files, add line numbers to the contents of file 1, and then input file 1 to
file 2.
dd if=/dev/urandom of=1.txt bs=1k count=4  # create new file
dd if=/dev/urandom of=2.txt bs=1k count=4
cat -n 1.txt >> 2.txt
```

```
# E15.Show the name of all files and folders, including the hidden ones, in current directory.
ls -a

# E16.Randomly select a file directory to list the files modified in the current directory and its subdirectories in the last 10 days.
find . -mtime -10

# E17.Make the files and subdirectories of directory “dir” readable, unwritable, and executable to all users.
chmod -R 555 dir

# E18.Run a python program, query the pid number, and kill it with the kill command.
ps -elf | grep python  # to get the pid, suppose store it into $pid
kill -9 $pid

# N1. Show line 6-10 of file /etc/hosts.
head /etc/hosts | tail -n 5

# N2. Use grep to find ‘.txt’ files in the current directory and delete.
rm $(ls | grep .txt)

# N3. Copy the file 1.sh in the current directory to your home directory, the modification time should be preserved.
cp -p 1.sh ~/

# N4. Make the files and subdirectories of directory “dir” readable, writable, executable to all users, use mode.
chmod -R 777 dir

# N5. Use grep to find all matches of pattern “url” in file “commits.json”, print 1 line of leading and trailing context surrounding each match.
grep -C 1 url commits.json

# N6. Sort all the files and subdirectories in the current folder according to size in descending order and print the largest three with filename and size. Use pipeline.
du -s * | sort -nr | head -n 3

# H1. Use ps to find the process who start with “python ”and kill them.
ps -elf | grep "[[:space:]]pytho[n]" | awk '{print $4}' | xargs -n 1 kill -9
# [[:space::]] is used to make sure "python" is in starting position, [n] makes sure that the grep command itself is not grepped.
# print $4 in awk is because pid is in the 4-th field of `ps -elf`

# H2. Transform all uppercase letters to lowercase in the file “input.txt”, and output the result to “output.txt”.
tr '[:upper:]' '[:lower:]' < input.txt > output.txt
```

```
# H3. Use awk to show the first and the second column of command “ls -l”, the header should not be included(grep -v).
ls -l | sed '1d' | awk '{printf "%s\t%s\n", $1, $2}'
# I prefer to use sed other than grep because machine running in different language will show different first line.

# H4. Write a script to display “Morning”、“Noon”、“Evening”
#!/bin/bash
for i in "Morning" "Noon" "Evening"
do
    echo $i
done

# H5. Write a script to Calculate the total size of files which ends with ‘.conf’ in ‘/etc’
#!/bin/bash
du /etc/*.conf | awk '{sum += $1};END{print sum}'

# H6. Count the number of users currently logged in to the system, and judge whether it is more than three. If it is, display the actual number and give a warning message. Otherwise, list the account name and terminal of the logged-in user.
#!/bin/bash
users=$(who | awk '{print $1}' | sort | uniq | wc -l)
if ((users > 3)); then
    echo "[alert] Too many users:" $users
else
    echo "logged-in users:"
    who | awk '{printf "%s\t%s\n", $1, $2}' | sort
fi
```