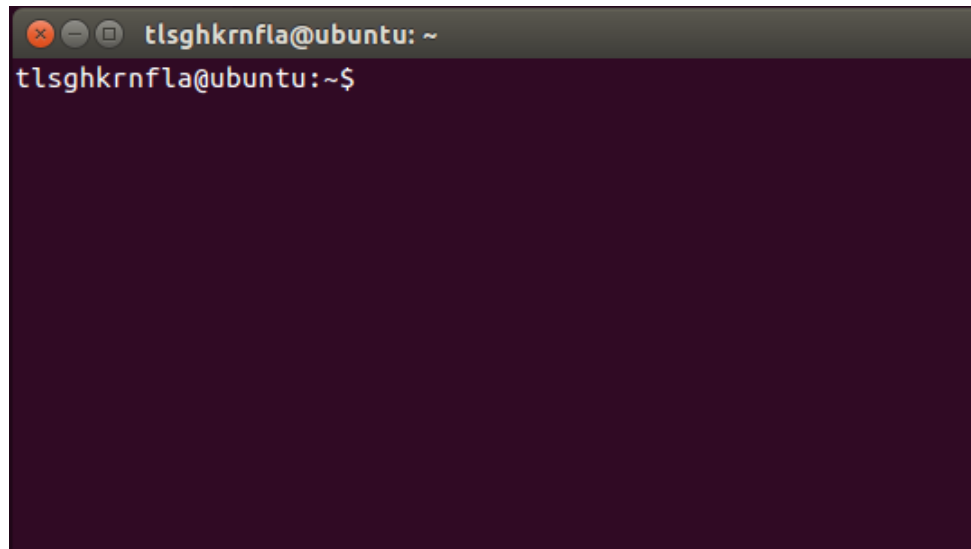


# Shell

- **Interface program between User & Linux**
  - Similar with Windows command prompt
  - Bash version : `/bin/bash --version`
  - Keyboard shortcut : `Ctrl + Alt + T`

A screenshot of a terminal window. The title bar shows the user 'tlsghkrnfla' on an 'ubuntu' machine, with a tilde '~' indicating the home directory. The terminal content shows the prompt 'tlsghkrnfla@ubuntu:~\$' followed by a blank line, indicating the shell is ready for input.

```
tlsghkrnfla@ubuntu: ~  
tlsghkrnfla@ubuntu:~$
```

# Redirection & Pipe

- **Redirection**

- Output redirection
- Input redirection
- `> / >>` (save or add stdout to file)
- `<` (transmit file to stdin)

- **Pipe**

- Process connecting

# Example

- **Redirection**

```
$ vi test.txt  
$ cat test.txt  
$ more test.txt  
$ more < test.txt  
$ ps  
$ ps >> test.txt  
$ cat test.txt
```

# Example

- **Find**

```
$ find . -iname "*.txt"
// find all files ".txt"
$ find . -iname "*.txt" | grep "aa"
// find all files that name has "aa"
$ find . -iname "*.txt" | xargs grep "aa"
// find all files that content has "aa"
```

- **Hard, soft link**

```
$ ln test.txt test2.txt
$ ln -s test.txt test3.txt
```

# Example

- **Echo**

```
$ echo "Today is festival"  
$ echo "class bb"
```

- **Pipe**

```
$ history  
$ history | grep bb
```

# Shell Programming

- **Line command**
- **Script**

```
$ vi test.sh

#!/bin/bash
echo 1

$ chmod 700 ./test.sh
$ ./test.sh
```

# Variable

- 모든 변수는 문자열로 인식
- 대소문자 구분
- 설정할 값에 띄어쓰기가 있는 경우 " " 사용
- 변수를 설정할 때 띄어 쓰면 error 발생
- 변수를 사용할 경우, 변수 앞에 '\$'

```
$ VAR=SW3  
$ echo VAR  
$ echo $VAR  
$ read VAR  
$ abcd  
$ echo $VAR
```

# Condition

- **[] / test**

```
#!/bin/bash

if test -f test.txt
then
...
fi
```

```
#!/bin/bash

if [-f test.txt]
then
...
fi
```



# Condition

- **if / elif / else**

```
#!/bin/bash
read VAR

if [ $VAR = "yes" ]; then
    echo "yes"
elif [ $VAR = "no" ]; then
    echo "no"
else
    echo "bb"
fi
```

# Condition

## ▪ for

- 문자열의 집합으로도 가능

```
#!/bin/bash

for foo in a b c
do
    echo $foo
done
```

```
#!/bin/bash

for file in $(ls a*.sh)
do
    echo $file
done
```

# Arithmetic Operation

- **Example**

```
#!/bin/bash

a=2
b=4
result=`expr $a + $b`
echo $result
result=`expr $a \* $b`
echo $result
```

# Stream EDitor

## ▪ SED

- `sed -n '/aa/p' test.txt`
  - 파일을 한 줄씩 읽으면서 aa문자를 찾으면 해당 줄을 출력
- `sed 's/aa/ab/' test.txt`
  - 파일의 aa을 ab로 치환하여 출력 (원본파일 변경 X)
- `sed -n '2,5p' test.txt`
  - 파일의 2~5줄만 출력

# PS

- 모든 프로세스 출력

- ps ax / ps -e / ps aux / ps -ef

- 사용자 이름으로 출력

- ps -fu username

- 프로세스 id 출력

- ps -p 1,4

- 시간으로 정렬

- ps aux -sort=-time | head -5