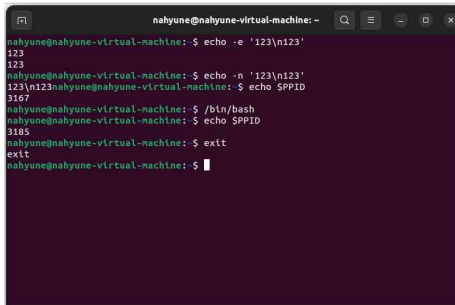


homework #8

인공지능학부 211852 조나현

homework #8

1.



```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ echo -e '123\n123'  
123  
123  
nahyune@nahyune-virtual-machine:~$ echo -n '123\n123'  
123\n123nahyune@nahyune-virtual-machine:~$ echo $PPID  
3167  
nahyune@nahyune-virtual-machine:~$ /bin/bash  
nahyune@nahyune-virtual-machine:~$ echo $PPID  
3185  
nahyune@nahyune-virtual-machine:~$ exit  
exit  
nahyune@nahyune-virtual-machine:~$
```

- `/bin/bash` 명령어를 실행하여 새로운 셸이 생성되었으므로, 새로운 셸의 부모 프로세스 ID를 출력합니다. 따라서 이전에 실행한 셸과는 다른 값이 됩니다.
- 여기서 `exit`는 새로운 셸 즉 `/bin/bash` 셸을 종료하는 것이기 때문이다.

homework #8

2.

```
nahyune@nahyune-virtual-machine: ~
nahyune@nahyune-virtual-machine:~$ (ls -l > file) >& errfile
total 128764
-rw-rw-r-- 1 wendy neverland 0 4월 14 00:15 a
drwxrwxr-x 5 nahyune nahyune 4096 4월 14 03:28 AntExample1
-rw-rw-r-- 1 nahyune nahyune 30720 4월 9 03:11 AntExample1.tar
-rw-rw-r-- 1 nahyune nahyune 0 4월 14 00:15 b
-rw-rw-r-- 1 nahyune nahyune 0 4월 14 00:15 c
drwxrwxr-x 11 nahyune nahyune 4096 4월 11 00:28 chnod_test
drwxrwxr-x 6 nahyune nahyune 4096 4월 9 02:43 copy_test
-rw-rw-r-- 1 nahyune nahyune 0 4월 14 00:15 d
drwxrwxr-x 2 nahyune nahyune 4096 4월 7 21:16 Desktop
-rw-rw-r-- 1 nahyune nahyune 0 4월 13 16:23 diary
-rw-rw-r-- 1 nahyune nahyune 360 4월 13 17:13 diary1.txt
-rw-rw-r-- 1 nahyune nahyune 722 4월 13 17:04 diary.txt
drwxrwxr-x 3 nahyune nahyune 4096 4월 9 02:00 directory_test
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Documents
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Downloads
-rw-rw-r-- 1 nahyune nahyune 0 4월 14 00:15 e
-rw-rw-r-- 1 nahyune nahyune 0 5월 24 21:11 errfile
-rw-rw-r-- 1 nahyune nahyune 0 5월 24 21:11 file
drwxrwxr-x 3 nahyune nahyune 4096 4월 11 00:37 homework3
drwxrwxr-x 25 nahyune nahyune 4096 5월 7 03:16 linux-5.19.12
-rw-rw-r-- 1 nahyune nahyune 131684622 9월 20 2022 linux-5.19.12.tar.xz
drwxrwxr-x 2 nahyune nahyune 4096 4월 7 21:16 Music
-rw-rw-r-- 1 nahyune nahyune 478 4월 9 03:16 myArchive.zip
-rw-rw-r-- 1 nahyune nahyune 0 4월 9 03:15 myfile_1.txt
-rw-rw-r-- 1 nahyune nahyune 0 4월 9 03:15 myfile_2.txt
-rw-rw-r-- 1 nahyune nahyune 0 4월 9 03:15 myfile_3.txt
drwxrwxr-x 3 nahyune nahyune 4096 4월 13 16:33 Pictures
-rw-rw-r-- 1 nahyune nahyune 0 5월 17 15:45 ps.out
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Public
drwxrwxr-x 2 nahyune nahyune 4096 4월 11 00:49 shared
drwxrwxr-x 2 nahyune nahyune 4096 4월 11 00:48 shared_t
drwxr-xr-x 4 nahyune nahyune 4096 4월 11 00:12 snap
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Templates
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Templates
drwxrwxr-x 3 nahyune nahyune 4096 5월 24 17:17 test
-rw-rw-r-- 1 nahyune nahyune 40960 4월 9 03:13 test2.tar
drwxr-xr-x 2 nahyune nahyune 4096 4월 7 21:16 Videos
drwxrwxr-x 2 nahyune nahyune 4096 4월 13 16:17 vi_homework
-rw-rw-r-- 1 nahyune nahyune 0 4월 14 00:15 x
nahyune@nahyune-virtual-machine:~$ cat errfile
nahyune@nahyune-virtual-machine:~$ (ls -z > file) >& errfile
nahyune@nahyune-virtual-machine:~$ cat file
nahyune@nahyune-virtual-machine:~$ cat errfile
ls: invalid option -- 'z'
Try 'ls --help' for more information.
nahyune@nahyune-virtual-machine:~$
```

- "(ls -l > file) >& errfile" 은 파일 목록을 "file"에 저장하고, 표준 에러 메시지를 "errfile"에 저장하는 반면,
- "(ls -z > file) >& errfile"은 유효하지 않은 옵션으로 인한 에러 메시지를 "errfile"에 저장하는 차이가 있습니다.

homework #8

3.

```
nahyune@nahyune-virtual-machine: ~$ files="ls"
nahyune@nahyune-virtual-machine: ~$ wc $files
wc: ls: No such file or directory
nahyune@nahyune-virtual-machine: ~$ files="ls"
nahyune@nahyune-virtual-machine: ~$ wc files
wc: files: No such file or directory
nahyune@nahyune-virtual-machine: ~$ wc $files
 0 0 0
wc: AntExample1: Is a directory
 0 0 0
 0 49 38720 AntExample1.tar
 0 0 0
 0 0 0
wc: chnod_test: Is a directory
 0 0 0
wc: copy_test: Is a directory
 0 0 0
 0 0 0
wc: Desktop: Is a directory
 0 0 0
 0 66 360 diary1.txt
 0 141 722 diary.txt
wc: directory_test: Is a directory
 0 0 0
wc: Documents: Is a directory
 0 0 0
wc: Downloads: Is a directory
 0 0 0
 0 0 0
 2 11 64 errfile
 0 0 0
wc: homework3: Is a directory
 0 0 0

nahyune@nahyune-virtual-machine: ~$ wc $files
 0 0 0
wc: homework3: Is a directory
 0 0 0
wc: linux-5.19.12: Is a directory
 0 0 0
515343 2917742 131684652 linux-5.19.12.tar.xz
wc: Music: Is a directory
 0 0 0
 0 22 478 myarchive.zip
 0 0 0
 0 0 0
 0 0 0
wc: Pictures: Is a directory
 0 0 0
 0 0 0
wc: Public: Is a directory
 0 0 0
wc: shared: Is a directory
 0 0 0
wc: shared_t: Is a directory
 0 0 0
wc: snap: Is a directory
 0 0 0
wc: Templates: Is a directory
 0 0 0
wc: test: Is a directory
 0 0 0
 0 53 40960 test2.tar
wc: Videos: Is a directory
 0 0 0
wc: vl_homework: Is a directory
 0 0 0
515343 2918084 131757956 total
```

- "files='ls'"는 문자열을 변수에 할당하는 것이며, "files='ls'"는 명령어를 실행한 결과를 변수에 할당하는 것이다
- 따라서 "files='ls'"를 실행한 후에 "wc \$files"라는 명령어를 통해 변수 \$files에 저장된 파일의 문자, 단어, 줄 수를 세면 결과값이 나오고 "files='ls'"라는 명령어 후에는 파일이 없다고 뜨는 것이다.

homework #8

4.

```
nahyune@nahyune-virtual-machine: ~
nahyune@nahyune-virtual-machine:~$ mkdir test
mkdir: cannot create directory 'test': File exists
nahyune@nahyune-virtual-machine:~$ cd test
nahyune@nahyune-virtual-machine:~/test$ wget http://kyungbaekkin.jnu.ac.kr/data/temp/thread.c
--2023-05-24 21:57:18-- http://kyungbaekkin.jnu.ac.kr/data/temp/thread.c
Resolving kyungbaekkin.jnu.ac.kr (kyungbaekkin.jnu.ac.kr)... 211.248.97.119
Connecting to kyungbaekkin.jnu.ac.kr (kyungbaekkin.jnu.ac.kr)|211.248.97.119|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 459 [text/x-c]
Saving to: 'thread.c.3'

thread.c.3      100%[=====] 459 --.-KB/s  in 0s

2023-05-24 21:57:19 (20.6 MB/s) - 'thread.c.3' saved [459/459]

nahyune@nahyune-virtual-machine:~/test$ gcc -o thread -pthread thread.c
nahyune@nahyune-virtual-machine:~/test$ cd ..
nahyune@nahyune-virtual-machine:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/sn
nahyune@nahyune-virtual-machine:~$ thread
Command 'thread' not found, did you mean:
  command 'nthread' from deb nblaze (1.1-1)
Try: sudo apt install <deb name>
nahyune@nahyune-virtual-machine:~$ PATH=./test:$PATH
nahyune@nahyune-virtual-machine:~$ echo $PATH
./test:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap
nahyune@nahyune-virtual-machine:~$ thread
Thread 0 prints x value : 0
nahyune@nahyune-virtual-machine:~$
```

- 일반적으로 "thread"라는 명령어는 기본적으로 제공되지 않는다
- 그러나 "PATH=./test:\$PATH"를 실행한 후에는 "./test" 디렉토리에서 "thread" 명령어를 찾을 수 있게 되어, 이제 "thread"를 실행할 때 "command not found" 오류 대신에 명령어의 결과값이 나타나게 됩니다. 이는 PATH 변수에 새로운 디렉토리를 추가하여 해당 디렉토리에 위치한 프로그램을 찾을 수 있게 되었기 때문입니다.

homework #8

5.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ PS1="\$"  
$PS1="Hello This is \u\$"  
Hello This is nahyune$PS1="Date \d User \u at \h \$"  
Date 수 5월 24 User nahyune at nahyune-virtual-machine $PS1="[\d\t]\u@[\h\w]\$"  
[수 5월 2422:05:19]nahyune@nahyune-virtual-machine[~]$exit  
nahyune@nahyune-virtual-machine:~$
```

- PS1="\\$" => 프롬프트를 "\$" 기호로 설정하는 것
- PS1="Hello This is \u\\$" => 프롬프트를 "Hello This is [사용자 이름]\$ "로 설정하는 것
- PS1="Date \d User \u at \h \\$" => 프롬프트는 "Date [현재 날짜] User [사용자 이름] at [호스트 이름] [권한 기호]"로 설정하는 것

homework #8

6.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ alias smile="echo $USERNAME is smiling"  
nahyune@nahyune-virtual-machine:~$ alias hungry="echo $USERNAME is hungry"  
nahyune@nahyune-virtual-machine:~$ alias getstory="wget http://kyungbaekkin.jnu.ac.kr/data/3l  
nahyune@nahyune-virtual-machine:~$ alias  
alias alert='notify-send --urgency=low -i "${S? = 0}" && echo terminal || echo error)' "${histo  
+${s*//:/;[:&]}\\s*alert$/'\''}'  
alias egrep='egrep --color=auto'  
alias fgrep='fgrep --color=auto'  
alias getstory='wget http://kyungbaekkin.jnu.ac.kr/data/3lpigs.txt'  
alias grep='grep --color=auto'  
alias hungry='echo nahyune is hungry'  
alias l='ls -CF'  
alias la='ls -A'  
alias ll='ls -lF'  
alias ls='ls --color=auto'  
alias smile='echo nahyune is smiling'  
nahyune@nahyune-virtual-machine:~$ smile  
nahyune is smiling  
nahyune@nahyune-virtual-machine:~$ hungry  
nahyune is hungry  
nahyune@nahyune-virtual-machine:~$ getstory  
--2023-05-24 22:17:15-- http://kyungbaekkin.jnu.ac.kr/data/3lpigs.txt  
Resolving kyungbaekkin.jnu.ac.kr (kyungbaekkin.jnu.ac.kr)... 211.248.97.119  
Connecting to kyungbaekkin.jnu.ac.kr (kyungbaekkin.jnu.ac.kr)|211.248.97.119|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 5487 (5.4K) [text/plain]  
Saving to: '3lpigs.txt'  
  
3lpigs.txt 100%[=====] 5.36K --KB/s in 0s  
  
2023-05-24 22:17:15 (502 MB/s) - '3lpigs.txt' saved [5487/5487]  
  
nahyune@nahyune-virtual-machine:~$ ls  
3lpigs.txt diary linux-5.19.12 shared  
a diary1.txt linux-5.19.12.tar.xz shared t  
AntExample1 diary.txt Music snap  
AntExample1.tar directory_test myarchive.zip Templates  
b Documents myfile_1.txt test  
c Downloads myfile_2.txt test2.tar  
chmod_test e myfile_3.txt Videos  
copy_test errfile Pictures vt_homework  
d file ps.out x  
Desktop homework3 Public
```

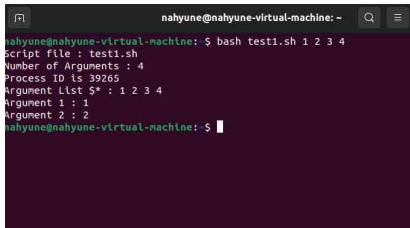
- nahyune is smiling
- nahyune is hungry
- 관련 링크를 얻은 것
=> 위 세가지가 각각 명령어의 결과이다.

이유는 alias가 별칭을 설정하는 명령어이기 때문이다.

homework #8

7.

- What happens if you add a line "echo \$3"? => 3이 출력된다.

A terminal window titled 'nahyune@nahyune-virtual-machine: ~' with search and menu icons. The prompt is 'nahyune@nahyune-virtual-machine: \$'. The user has entered 'bash test1.sh 1 2 3 4'. The script outputs: 'Script file : test1.sh', 'Number of Arguments : 4', 'Process ID is 39265', 'Argument List \$* : 1 2 3 4', 'Argument 1 : 1', and 'Argument 2 : 2'. The prompt is now 'nahyune@nahyune-virtual-machine: \$' with a cursor.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine: $ bash test1.sh 1 2 3 4  
Script file : test1.sh  
Number of Arguments : 4  
Process ID is 39265  
Argument List $* : 1 2 3 4  
Argument 1 : 1  
Argument 2 : 2  
nahyune@nahyune-virtual-machine: $
```


homework #8

8.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ bash test2.sh  
correct usage  
/home/peterpan  
  
home/peterpan/test  
test  
xxy  
1:xxy  
1:xxy  
1n:test1  
1n:test1  
2:xxy  
2:xxy  
2n:test2  
2n:  
3:test3  
3:xxy  
3n:  
3n:  
4:xxy  
4:xxy  
4:3  
test2.sh: line 25: d: nonexistent d
```

homework #8

9.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ bash for.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
number :4  
4  
nahyune@nahyune-virtual-machine:~$ bash while.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
"total value is 54"
```

- while ["\$var" -le 9] 는 var의 값이 9 이하인 동안 루프를 실행하는 것이고 , while [["\$var" -lt 9] 9미만인 동안 루프를 실행하는 것이므로 결과값이 1에서 8까지만 출력된다.

homework #8

10.

```
nahyune@nahyune-virtual-machine: ~  
nahyune@nahyune-virtual-machine:~$ bash main.sh  
start main  
name: peterpan  
all name: peterpan  
all location: neverland  
start sub  
sub.sh: line 3: print_name: command not found  
all name: peterpan  
all location:  
nahyune@nahyune-virtual-machine:~$
```

=>echo "start main"을 실행 한 후, 차례로
print_name 함수와 print_all을 실행한다.
export name은 name 변수를 현재 셸의 환경
변수로 내보내고,

export -f print_all은 print_all 함수를 현재 셸
의 환경 변수로 내보낸다.

이후, sub.sh 스크립트를 실행하는데, 이 스
크립트는 현재 셸의 환경 변수로 내보낸
print_all 함수를 사용하여 결과를 출력합니
다.

따라서 print_name 함수와 print_all을 실행
해도 print_all의 결과만 출력된다.

=>print_name 함수를 제대로 동작하게 하
기 위해서는 'export -f print_name' 명령어를
추가하면 된다.

homework #8 - problems

1.

- PS1="(\t)Hello \u@\h[\w]\\$" 라는 명령어가 정답이다.

```
nahyune@nahyune-virtual-machine:~$ PS1="(\t)Hello \u@\h[\w]\$"
(22:29:17)Hello nahyune@nahyune-virtual-machine[~]$cd vi_homework/
(22:30:47)Hello nahyune@nahyune-virtual-machine[~/vi_homework]$su
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@nahyune-virtual-machine:/home/nahyune/vi_homework# su
```

homework #8 - problems

2.

-먼저, 현재 실행 중인 셸이 Bash인지 확인합니다. 이를 위해 `$BASH_VERSION` 환경 변수의 값이 비어있지 않은지 확인합니다.

-Bash 셸이라면, `.bashrc` 파일이 존재하는 경우 해당 파일을 포함시킵니다. `.bashrc` 파일은 사용자의 Bash 셸에 대한 구성과 사용자 지정 명령을 포함할 수 있는 파일입니다. 따라서, `.bashrc` 파일은 사용자가 Bash 셸을 로딩할 때마다 실행되어 특정 환경 변수, 별칭, 함수 등을 설정할 수 있습니다.

-그 후, 사용자의 개인 bin 디렉토리 (`$HOME/bin`)가 존재하는지 확인합니다. 만약 해당 디렉토리가 존재한다면, `PATH` 환경 변수에 해당 디렉토리를 추가하여 사용자의 개인 바이너리 파일이 포함된 디렉토리를 사용할 수 있도록 합니다. `PATH`는 실행 가능한 프로그램을 검색하는 데 사용되는 환경 변수로, 여기서는 `$HOME/bin` 디렉토리를 우선적으로 검색하도록 설정됩니다.

homework #8 - problems

3.

```
== Simple Calculator ==
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 1
Add X and Y
X?
2
Y?
3
[1]X + Y = 5
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 2
Subtract Y from X
X?
3
Y?
2
[2]X - Y = 1
```

```
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 3
Multiply X and Y
X?
2
Y?
3
[3]X * Y = 6
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 4
Devide X by Y
X?
4
Y?
2
[4]X / Y = 2
```

```
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 5
History
==HISTORY==
[1]X + Y = 5
[2]X - Y = 1
[3]X * Y = 6
[4]X / Y = 2
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 7
!! Please select correct operation
Which operation?:
1) Add X and Y
2) Subtract Y from X
3) Multiply X and Y
4) Devide X by Y
5) History
6) Quit
#? 6
Quit
## Bye Bye, Have a nice day
```