Homework #6

1. Do "clear", then do "su"(password is required), then do "dd if=/dev/sda of=mbr.bin bs=512 count=1", then do "od –xa mbr.bin"

- Take a screenshot 
- What is the content of mbr.bin?

2. Do "vi /etc/default/grub", then insert "#" character in front of "GRUB_HIDDEN_TIMEOUT=0" and "GRUB_HIDDEN_TIMEOUT_QUIET=true", then save the file and quit, (check 22 pages of the lecture slide 09), then do "update-grub", then do "shutdown –r now"

- Take a screenshot of grub selection page 

3. Start a new terminal, then do "su" (password required), then do "cd /boot", then do "cp vmlinuz-2.6.32-38-generic vmlinuz_test", then do "cp initrd.img-2.6.32-38-generic initrd.img_test", then do "cd /etc/grub.d/", then do "cp 40_custom 11_test", then insert the following sentences at the end of the file "11_test".

```
----------------------------
menuentry "homework_test"{
        insmod ext2
        set root='(hd0,1)'
        linux /boot/vmlinuz_test ro quiet splash
        initrd /boot/initrd.img_test
}
----------------------------
```

Then do "update-grub", then do "shutdown –r now"

- Take a screenshot of grub selection page 
- Why does the new selection locate after the original Ubuntu selection?

4. The following list indicates the files under a directory named as "/etc/rc5.d/". Describe the meaning of this directory in the aspect of booting the Linux.

```
-----------------------------------
S20kerneloops
S21quotarpc
S35networking
S50rsync
S70dns-clean
S80mysqld
-----------------------------------
```

5. Do "echo –e '123₩n123'", then do "echo –n '123₩n123', then do "echo $PPID", then do "/bin/bash", then do "echo $PPID", then do "exit"

- Take a screen shot
- Why the result of "echo $PPID" is different? Why does your terminal still work after doing "exit"?

6. Do "clear", then do "(ls –l > file) >& errfile", then do "cat file", then do "cat errfile", then do "(ls –z > file) >& errfile", then do "cat file", then do "cat errfile".

- Take a screen shot
- What is the difference? Why?

7. Do "clear", then do "files='ls'" (with quotes), then do "wc $files", then do "files=`ls`" (back quotes), then do "wc files", then do "wc $files".

- Take a screen shot
- Explain the result

8. Do "clear", then do "mkdir test", then do "cd test", then do "wget http://myweb.jnu.ac.kr/~kbkim/data/temp/thread.c", then do "gcc –o thread –pthread thread.c", then do "cd ..", then do "echo $PATH", then do "thread", then do "PATH="./test:$PATH", then do "echo $PATH", then do "thread"

- Take a screen shot
- Why the results of doing "thread" are different? Explain it.

9. Do "bash", then do "clear", then do "PS1="₩$"", then do "PS1="Hello This is ₩u₩$"", then do "PS1="Date ₩d User ₩u at ₩h ₩$"", then do "PS1="[₩d₩t]₩u@₩h[₩w]₩$"", then do "exit"

- Take a screen shot
- Explain why the prompt changes in different forms.

10. Do "clear", then do "alias smile="echo $USERNAME is smiling"", then do "alias hungry="echo $USERNAME is hungry", then do "alias getstory="wget http://myweb.jnu.ac.kr/~kbkim/data/3lpigs.txt"", then do "alias", then do "smile", then do "hungry", then do "getstory", then do "ls".

- Take a screen shot
- What are the result of doing "smile", "hungry", and "getstory"? why?

11. Create a file “test1.sh” containing following shell programming codes.

```
#!/bin/bash
#reading arguments
echo "Script file : $0"
echo "Number of Arguments : $#"
echo "Process ID is $$"
echo "Argument List ₩$* : $*"
echo "Argument 1 : $1"
echo "Argument 2 : $2"
```

Then do “clear”, then do “bash test1.sh 1 2 3 4”.

- Take a screen shot
- What happens if you add a line “echo $3”?

12. Create a file “test2.sh” containing following shell programming codes.

```
#!/bin/bash
k=/home/peterpan/test
echo "correct usage"
echo ${k%/*}
echo ${k%%/*}
echo ${k#*/}
echo ${k##*/}
a="xxy"
echo "$a"
echo "1:${a:="test1"}"
echo "1:$a"
echo "1n:${x:="test1"}"
echo "1n:$x"
echo "2:${a:-"test2"}"
echo "2:$a"
echo "2n:${b:-"test2"}"
echo "2n:$b"
echo "3:${a:+"test3"}"
echo "3:$a"
echo "3n:${c:+"test3"}"
echo "3n:$c"
echo "4:${a:?"test4"}"
echo "4:$a"
echo "4:${#a}"
echo "4n:${d:?"nonexist d"}"
```

Then do “clear”, then do “bash test2.sh”.

- Take a screen shot

13. Create a file “for.sh” containing following shell programming codes.

```
#!/bin/bash
for var in 1 2 3 4 5 6 7 8 9
do
        echo $var
done
read –p "number :" x
echo $x
```

Then, create a file “while.sh” containing following shell programming codes.

```
#!/bin/bash
var=1
tvar=0
while [ "$var" -le 9 ]
do
        echo $var
        var=$(($var+1))
        tvar=$(($tvar+$var))
done
echo “total value is $tvar”
```

Then do “clear”, then do “bash for.sh”, then do “bash while.sh”.

- Take a screen shot 
- What happens if you change the while condition to [“$var” –lt 9]?

14. Create a file “main.sh” containing following shell programming codes.

```
#!/bin/bash
name=peterpan
location=neverland
print_name()
{
   echo "name: $name"
}

print_all()
{
   echo "all name: $name"
   echo "all location: $location"
}

echo "start main"
print_name
print_all

export name
export -f print_all

bash sub.sh
```

Then, create a file “sub.sh” containing following shell programming codes.

```
#!/bin/bash
echo "start sub"
print_name
print_all
```

Then do “clear”, then do “bash main.sh”

- Take a screenshot 
- Explain the results.
- How to correctly use print_name on sub.sh?

Problems

1. Bash의 프롬프트를 다음과 같이 보이게 하는 명령어는?

(07:09:07)Hello kbkim@ubuntu[~]$cd vi_homework/

(07:09:10)Hello kbkim@ubuntu[~/vi_homework]$su

Password:

root@ubuntu:/home/kbkim/vi_homework#

---

2. 다음은 .profile이라는 configuration file의 일부분이다. 이 configuration 파일이 수행하는 내용을 설명하시오.

```
# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```
