

리눅스시스템 Lab05

분반: 001

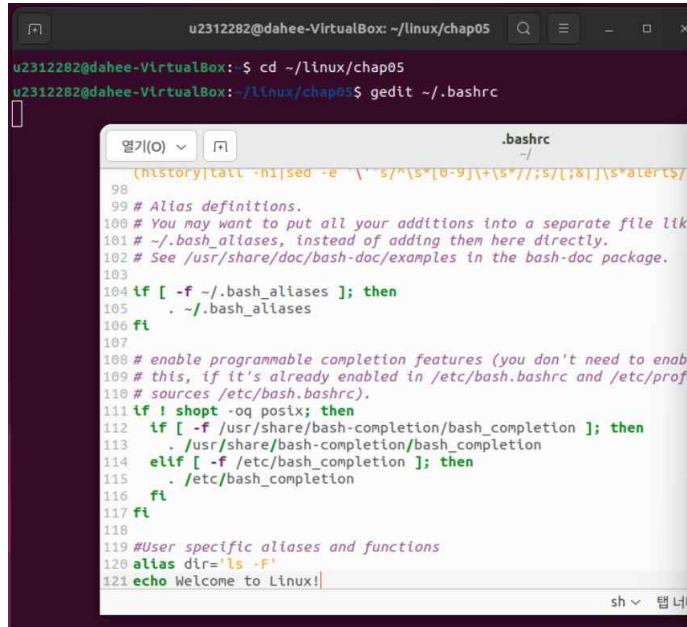
학과:컴퓨터과학전공

학번:2312282

이름:임다희

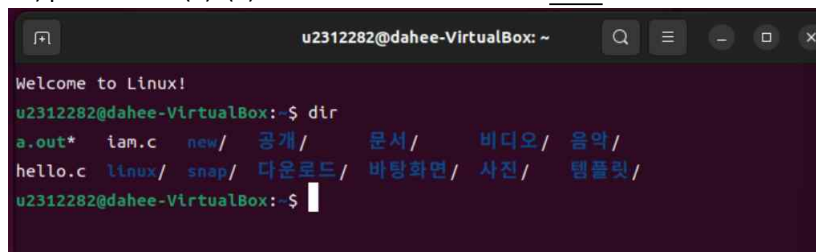
1. 시작 파일 실습

1) p3를 참고하여 gedit 명령어로 .bashrc 파일을 열어 지정된 명령을 추가한 뒤 화면을 캡처한다.



```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox: ~/linux/chap05$ gedit ~/.bashrc
[
. .bashrc
-~
(history|tail -n1sed -e '\`s/*[0-9]\{+\}/*;/s/[;|&]\{*\}alerts//
98
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 #User specific aliases and functions
120 alias dir='ls -F'
121 echo Welcome to Linux!
```

2) p4와 같이 (2)~(3)을 실행한 터미널 창을 캡처한다.



```
u2312282@dahee-VirtualBox: ~
Welcome to Linux!
u2312282@dahee-VirtualBox:~$ dir
a.out*  iam.c  new/   공개/   문서/   비디오/  음악/
hello.c linux/ snap/  다운로드/ 바탕화면/ 사진/   템플릿/
u2312282@dahee-VirtualBox:~$
```

3) [메뉴]에서 새로운 터미널 창을 열었을 때, 터미널 창에 출력되는 내용에 대해 설명한다.

~/.bashrc 파일은 새로운 셸이 만들어질 경우 적용되는 시작 파일로, .bashrc 파일에 echo Welcome to Linux! 명령어를 추가했기 때문에 새로운 터미널 창을 열면 자동으로 터미널 창에 Welcome to Linux! 라는 내용이 출력된다.

4) 새 창에서 dir 명령어를 실행하고 그 결과를 설명한다.

3)과 같은 원리로 .bashrc 파일에 alias dir='ls -F' 라는 명령어를 추가하였기 때문에 ls-F 명령어의 별칭을 dir로 설정하여, 새 창에서 dir를 입력하면 ls-F 명령어의 기능인 파일의 종류를 표시하여 출력하기 를 실행할 수 있다.

★주의★ 3), 4) 문항은 .bashrc 수정 사항과 관련지어 설명하여야 한다.

2. 출력 재지정 실습

- 1) p5-6의 [예시 1]과 [예시 2]를 실행한 터미널 창을 캡처한다.

예시1

```
u2312282@dahee-VirtualBox: ~/linux/chap05

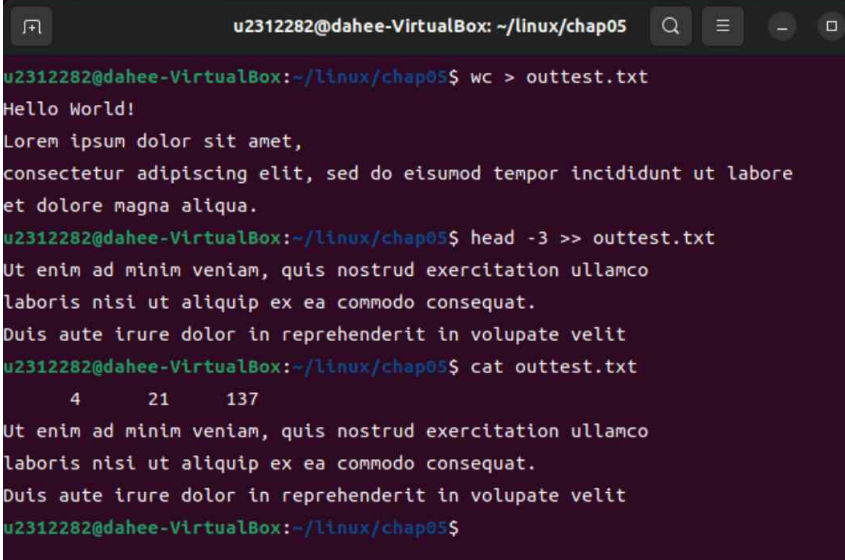
u2312282@dahee-VirtualBox:~/linux/chap05$ who > out1.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out1.txt
u2312282 tty2          2024-10-02 13:05 (tty2)
u2312282@dahee-VirtualBox:~/linux/chap05$ date >> out1.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out1.txt
u2312282 tty2          2024-10-02 13:05 (tty2)
2024. 10. 02. (수) 13:14:18 KST
u2312282@dahee-VirtualBox:~/linux/chap05$
```

예시2

```
u2312282@dahee-VirtualBox: ~/linux/chap05

u2312282@dahee-VirtualBox:~/linux/chap05$ cat > out1.txt
Hello World!
This is the first list.
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out1.txt
Hello World!
This is the first list.
u2312282@dahee-VirtualBox:~/linux/chap05$ cat >> out1.txt
Hi!
This is the second list.
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out1.txt
Hello World!
This is the first list.
Hi!
This is the second list.
u2312282@dahee-VirtualBox:~/linux/chap05$
```

2) p7을 참고하여 [예시 1], [예시 2]에 나온 명령어 외에 다른 명령어를 활용해 출력 재지정 명령어를 실습해보고 새로 생성한 파일 내용을 확인한다. 실행한 터미널 창을 캡처한다



```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ wc > outtest.txt
Hello World!
Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore
et dolore magna aliqua.
u2312282@dahee-VirtualBox:~/linux/chap05$ head -3 >> outtest.txt
Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit
u2312282@dahee-VirtualBox:~/linux/chap05$ cat outtest.txt
4      21      137
Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit
u2312282@dahee-VirtualBox:~/linux/chap05$
```

3) 본인이 실습한 2)번 결과를 바탕으로 출력 재지정 명령어 2개에 대해 각각 설명한다.

-설명

출력 재지정은 명령어의 표준출력을 모니터 대신에 파일에 저장하도록 한다.

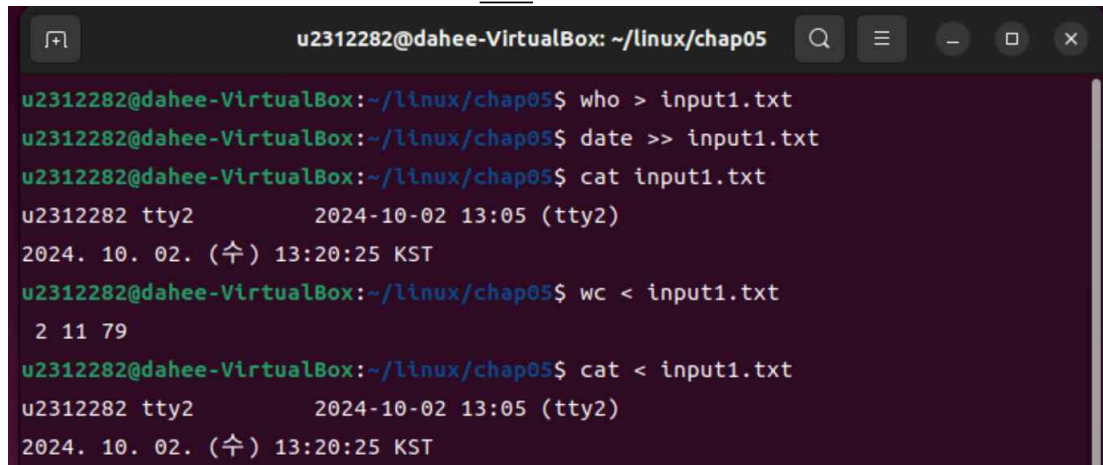
wc > outtest.txt 과 같은 출력 재지정을 통해 표준입력받은 내용(Hello World!~....magna aliqua)의 표준 출력(줄 수, 단어 개수, 문자 개수)을 outtest.txt 파일의 내용으로 저장한다.

head -3 >> outtest.txt 출력 재지정 명령어를 통해 표준입력받은 내용의 표준출력(head -3은 텍스트의 상위 세 줄을 의미)을 outtest.txt의 내용으로 덧붙인다. 출력 재지정 명령어의 >> 옵션을 기존 내용이 존재하는 파일에 사용하면 내용을 덮어쓰지 않고 파일에 새로운 출력 내용을 추가할 수 있다.

cat 명령어를 통해 outtest.txt의 내용을 출력하면 wc > outtest 명령어로 추가된 내용 (4 21 137)과 head -3 >> outtest.txt 명령어로 추가된 내용 (Ut enim...~..voluptate velit 의 3줄)을 모두 확인할 수 있다.

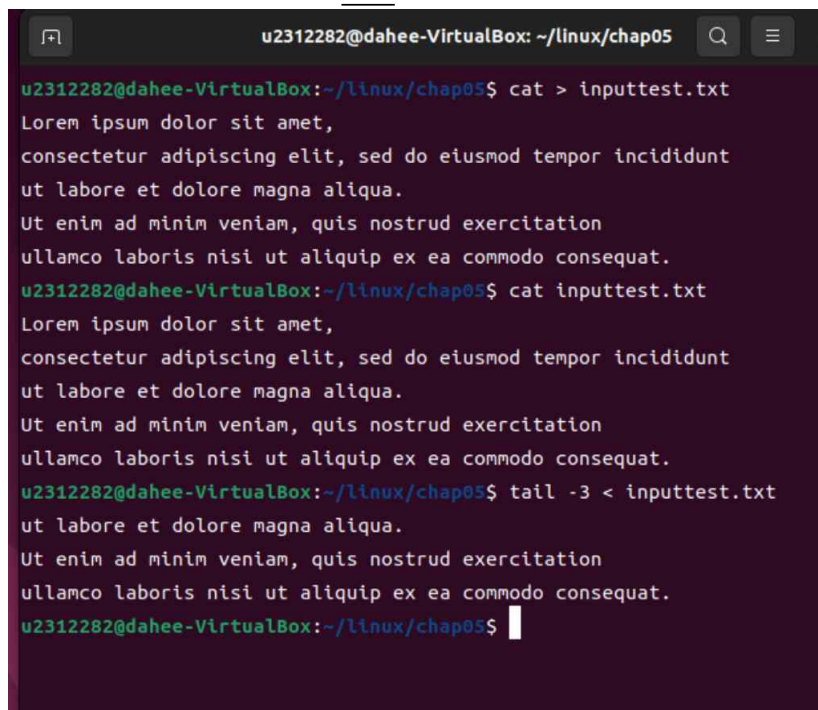
3. 입력 재지정 실습

1) p8의 [예시]를 실행한 터미널 창을 캡처한다.



```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ who > input1.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ date >> input1.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ cat input1.txt
u2312282 tty2          2024-10-02 13:05 (tty2)
2024. 10. 02. (수) 13:20:25 KST
u2312282@dahee-VirtualBox:~/linux/chap05$ wc < input1.txt
 2 11 79
u2312282@dahee-VirtualBox:~/linux/chap05$ cat < input1.txt
u2312282 tty2          2024-10-02 13:05 (tty2)
2024. 10. 02. (수) 13:20:25 KST
```

2) p9를 참고하여 [예시]에 나온 명령어 **외**에 다른 명령어를 활용해 입력 재지정 명령어를 실습해보고 실행한 터미널창을 캡처한다. 입력에 해당하는 파일 내용도 같이 출력하여 캡처한다.



```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ cat > inputtest.txt
Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
u2312282@dahee-VirtualBox:~/linux/chap05$ cat inputtest.txt
Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
u2312282@dahee-VirtualBox:~/linux/chap05$ tail -3 < inputtest.txt
ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
u2312282@dahee-VirtualBox:~/linux/chap05$
```

3) **본인이 실습한 2)번 결과를 바탕으로** 입력 재지정 명령어에 대해 설명한다.

입력 재지정 명령어는 명령어의 표준입력을 키보드 대신 파일에서 받도록 한다. cat > inputtest.txt 명령어를 통해 5줄의 텍스트 파일 inputtest.txt를 만들고, cat inputtest.txt 명령어를 통해 파일의 내용을 확인한다.

tail -3 < inputtest.txt 명령어를 통해 텍스트 파일 뒤에서부터 세 줄의 텍스트를 출력하는 명령어 tail -3의 표준입력을 키보드 대신 inputtest.txt 파일에서 받도록 한다. 그 결과로 처음 만들었던 파일 내용의 뒤에서부터 세 줄 텍스트(ut labore et....~...ea commodo consequat) 가 출력된다.

4. 파이프 실습

1) p10의 모든 명령어를 실행한 터미널 창을 캡처한 후, 2개의 명령어의 결과를 각각 설명한다.

```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ ps -ef
UID          PID    PPID  C  STIME TTY          TIME CMD
root           1        0  0  13:04 ?        00:00:02 /sbin/init splash
root           2        0  0  13:04 ?        00:00:00 [kthreadd]
root           3        2  0  13:04 ?        00:00:00 [pool_workqueue_release]
root           4        2  0  13:04 ?        00:00:00 [kworker/R-rcu_g]
root           5        2  0  13:04 ?        00:00:00 [kworker/R-rcu_p]
root           6        2  0  13:04 ?        00:00:00 [kworker/R-slub_]
root           7        2  0  13:04 ?        00:00:00 [kworker/R-netns]
root          12        2  0  13:04 ?        00:00:00 [kworker/R-mm_pe]
root          13        2  0  13:04 ?        00:00:00 [rcu_tasks_kthread]
root          14        2  0  13:04 ?        00:00:00 [rcu_tasks_rude_kthread]
root          15        2  0  13:04 ?        00:00:00 [rcu_tasks_trace_kthread]
root          16        2  0  13:04 ?        00:00:04 [ksoftirqd/0]
root          17        2  0  13:04 ?        00:00:00 [rcu_preempt]
root          18        2  0  13:04 ?        00:00:00 [migration/0]
root          19        2  0  13:04 ?        00:00:00 [idle_inject/0]
root          20        2  0  13:04 ?        00:00:00 [cpuhp/0]
root          21        2  0  13:04 ?        00:00:00 [kdevtmpfs]
root          22        2  0  13:04 ?        00:00:00 [kworker/R-inet_]
root          24        2  0  13:04 ?        00:00:00 [kauditd]
root          25        2  0  13:04 ?        00:00:00 [khungtaskd]
```

```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ ps -ef | grep u2312282
u2312282      1153        1  0  13:05 ?        00:00:00 /lib/systemd/systemd --user
u2312282      1154      1153  0  13:05 ?        00:00:00 (sd-pam)
u2312282      1160      1153  0  13:05 ?        00:00:00 /usr/bin/pipewire
u2312282      1161      1153  0  13:05 ?        00:00:00 /usr/bin/pipewire-media-session
u2312282      1162      1153  0  13:05 ?        00:00:00 /usr/bin/pulseaudio --daemon
size=no --log-target=journal
u2312282      1170        1  0  13:05 ?        00:00:00 /usr/bin/gnome-keyring-daemo
n --daemonize --login
u2312282      1178      1153  0  13:05 ?        00:00:00 /usr/bin/dbus-daemon --sessi
on --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
u2312282      1181      1153  0  13:05 ?        00:00:00 /usr/libexec/xdg-document-po
rtal
u2312282      1182      1131  0  13:05 tty2      00:00:00 /usr/libexec/gdm-wayland-ses
sion env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
u2312282      1186      1153  0  13:05 ?        00:00:00 /usr/libexec/xdg-permission-
store
u2312282      1188      1182  0  13:05 tty2      00:00:00 /usr/libexec/gnome-session-b
inary --session=ubuntu
u2312282      1255      1153  0  13:05 ?        00:00:00 /usr/libexec/gnome-session-c
tl --monitor
u2312282      1268      1153  0  13:05 ?        00:00:00 /usr/libexec/gvfsd
u2312282      1277      1153  0  13:05 ?        00:00:00 /usr/libexec/gvfsd-fuse /run
```

- 설명

ps -ef 명령어는 현재 실행되고 있는 모든 프로세스를 표준출력으로 보여준다. grep 명령어는 특정 단어를 찾는 명령어이다. 따라서 ps -ef|grep 사용자명 과 같은 명령은 ps -ef의 표준출력인 실행중인 모든 프로세스를 grep 명령어의 표준입력으로 입력하며, 실행중인 모든 프로세스 중 사용자명이 들어가 있는 프로세스(특정 사용자의 프로세스)를 찾아 결과를 출력한다.

2) p11의 [예시]를 실행한 터미널 창을 캡처한 후, 명령어의 결과를 설명한다.

```
u2312282@dahee-VirtualBox: ~/linux
u2312282@dahee-VirtualBox:~/linux/chap05$ touch out1.c
u2312282@dahee-VirtualBox:~/linux/chap05$ touch out2.c
u2312282@dahee-VirtualBox:~/linux/chap05$ touch out3.c
u2312282@dahee-VirtualBox:~/linux/chap05$ ls ./*.c
./out1.c ./out2.c ./out3.c
u2312282@dahee-VirtualBox:~/linux/chap05$ ls ./*.c | wc -w
3
u2312282@dahee-VirtualBox:~/linux/chap05$ cd ..
u2312282@dahee-VirtualBox:~/linux$ ls chap05/*.c | wc -w
3
u2312282@dahee-VirtualBox:~/linux$
```

-설명

touch 명령어를 통해 .c 타입의 파일을 3개 생성한다. ls ./*.c 명령어는 현재 디렉터리 내에서 .c 타입의 파일만을 찾아 출력하는 명령어이다. ls ./*.c | wc -w 와 같이 두 명령어를 파이프로 연결하면 ls ./*.c의 출력을 표준입력의 단어 수를 세는 명령어 wc-w의 입력으로 줄 수 있다. chap05 내의 .c 파일은 touch를 통해 생성한 3개이므로 ls ./*.c | wc-w의 결과는 3이다. 현재 작업 디렉터리를 chap05의 상위 디렉터리 linux로 이동하여 ls chap05/*.c | wc-w와 같이 실행해도 결과는 3으로 같다.

3) p12의 명령어를 실행한 터미널 창을 캡처한 후, 명령어의 결과를 설명한다.

```
u2312282@dahee-VirtualBox: ~/linux
u2312282@dahee-VirtualBox:~/linux$ ls /bin | sort | more
GET
HEAD
POST
VGAuthService
X
X11
Xephyr
Xorg
Xwayland
[
aa-enabled
aa-exec
aa-features-abi
aconnect
acpi_listen
acpidbg
add-apt-repository
addpart
addr2line
airscan-discover
alsabat
alsaloop
alsamixer
```

- 설명

ls /bin | sort | more와 같이 파이프를 이중으로 사용한 경우 ls /bin의 표준출력이 sort의 표준입력으로 들어가고, 또한 그 결과로 나온 sort의 표준출력이 more의 표준입력으로 들어가는 경우이다. ls /bin 명령어는 bin 디렉터리 내의 모든 파일을 리스팅하며, sort는 해당 결과를 입력으로 받아 오름차순으로 정렬한다. 최종적으로 more는 오름차순으로 정렬된 bin 디렉터리 내의 파일 리스트를 입력으로 받아 페이지 단위로 내용을 출력한다.

5. 대표문자 실습

1) p14의 [예시]를 실행한 터미널 창을 캡처한다.

```
u2312282@dahee-VirtualBox: ~/linux
u2312282@dahee-VirtualBox: ~/linux/chap05$ touch abc.c
u2312282@dahee-VirtualBox: ~/linux/chap05$ touch h.c
u2312282@dahee-VirtualBox: ~/linux/chap05$ cd ..
u2312282@dahee-VirtualBox: ~/linux$ cp chap05/?.* /tmp
u2312282@dahee-VirtualBox: ~/linux$ ls /tmp
VMwareDnD
h.c
snap-private-tmp
systemd-private-e440e85c2cd249b38b7141b2cca8c035-ModemManager.service-UQ00Xz
systemd-private-e440e85c2cd249b38b7141b2cca8c035-colord.service-2lzuC0
systemd-private-e440e85c2cd249b38b7141b2cca8c035-fwupd.service-ZyE8je
systemd-private-e440e85c2cd249b38b7141b2cca8c035-power-profiles-daemon.service-k
IDaCT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-switcheroo-control.service-NYaB
QT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-logind.service-JfAvnC
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-oomd.service-enKtjV
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-resolved.service-npLwMb
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-timesyncd.service-fdVJT
A
systemd-private-e440e85c2cd249b38b7141b2cca8c035-upower.service-TsJnoE
```

```
u2312282@dahee-VirtualBox:~/linux$ cp chap05/*.c /tmp
u2312282@dahee-VirtualBox:~/linux$ ls /tmp
VMwareDnD
abc.c
h.c
out1.c
out2.c
out3.c
snap-private-tmp
systemd-private-e440e85c2cd249b38b7141b2cca8c035-ModemManager.service-UQ00Xz
systemd-private-e440e85c2cd249b38b7141b2cca8c035-colord.service-2lzuC0
systemd-private-e440e85c2cd249b38b7141b2cca8c035-fwupd.service-ZyE8je
systemd-private-e440e85c2cd249b38b7141b2cca8c035-power-profiles-daemon.service-k
IDaCT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-switcheroo-control.service-NYaB
QT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-logind.service-JfAvnC
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-oomd.service-enKtjV
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-resolved.service-npLwMb
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-timesyncd.service-fdVJT
A
systemd-private-e440e85c2cd249b38b7141b2cca8c035-upower.service-TsJnoE
```

```
u2312282@dahee-VirtualBox:~/linux$ cp chap05/*.* /tmp
u2312282@dahee-VirtualBox:~/linux$ ls /tmp
VMwareDnD
abc.c
h.c
input1.txt
out1.c
out1.txt
out2.c
out3.c
snap-private-tmp
systemd-private-e440e85c2cd249b38b7141b2cca8c035-ModemManager.service-UQ00Xz
systemd-private-e440e85c2cd249b38b7141b2cca8c035-colord.service-2lzuC0
systemd-private-e440e85c2cd249b38b7141b2cca8c035-fwupd.service-ZyE8je
systemd-private-e440e85c2cd249b38b7141b2cca8c035-power-profiles-daemon.service-k
IDaCT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-switcheroo-control.service-NYaB
QT
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-logind.service-JfAvnC
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-oomd.service-enKtjV
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-resolved.service-npLwMb
systemd-private-e440e85c2cd249b38b7141b2cca8c035-systemd-timesyncd.service-fdVJT
A
```

2) [예시]의 결과를 참고하여 p13의 명령 3개에 대해 차이점을 중심으로 각각 설명한다.

각 명령들은 어떤 파일을 선택하여 복사하는지에서 차이를 보인다.

\$ cp 디렉터리/*c/tmp: 대표문자 *는 빈 스트링을 포함하여 임의의 스트링을 나타내는 문자이다. .c로 끝나는 모든 파일들을 복사하는 명령이다.

\$ cp 디렉터리/?c/tmp: 대표문자 ?는 길이가 1인 임의의 한 문자를 표현할 때 사용한다. .c로 끝나면서 파일명이 한 글자인 파일들만을 복사하는 명령이다.

\$ cp 디렉터리/* /tmp: 대표문자 *는 홀로 사용되면 모든 파일을 나타낸다. 조건에 따라 다른 파일을 선택하여 복사하는 앞의 두 명령과 달리 디렉터리 내의 모든 파일들을 복사하는 명령이다.

6. 명령어 열/그룹 실습

1) p15의 [예시]를 실행한 터미널 창을 캡처한다.

```
u2312282@dahee-VirtualBox: ~/linux/chap05
u2312282@dahee-VirtualBox:~/linux/chap05$ date; pwd; ls
2024. 10. 02. (수) 13:37:13 KST
/home/u2312282/linux/chap05
abc.c  h.c  input1.txt  out1.c  out1.txt  out2.c  out3.c
u2312282@dahee-VirtualBox:~/linux/chap05$ (ls -al; who; date)
합계 16
drwxrwxr-x 2 u2312282 u2312282 4096 10월 2 13:33 .
drwxrwxr-x 5 u2312282 u2312282 4096 10월 2 13:07 ..
-rw-rw-r-- 1 u2312282 u2312282 0 10월 2 13:33 abc.c
-rw-rw-r-- 1 u2312282 u2312282 0 10월 2 13:33 h.c
-rw-rw-r-- 1 u2312282 u2312282 79 10월 2 13:20 input1.txt
-rw-rw-r-- 1 u2312282 u2312282 0 10월 2 13:30 out1.c
-rw-rw-r-- 1 u2312282 u2312282 66 10월 2 13:17 out1.txt
-rw-rw-r-- 1 u2312282 u2312282 0 10월 2 13:30 out2.c
-rw-rw-r-- 1 u2312282 u2312282 0 10월 2 13:30 out3.c
u2312282 tty2          2024-10-02 13:05 (tty2)
2024. 10. 02. (수) 13:37:24 KST
u2312282@dahee-VirtualBox:~/linux/chap05$ date; pwd; ls > out1.txt
2024. 10. 02. (수) 13:37:40 KST
/home/u2312282/linux/chap05
```

```
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out1.txt
abc.c
h.c
input1.txt
out1.c
out1.txt
out2.c
out3.c
u2312282@dahee-VirtualBox:~/linux/chap05$ (date; pwd; ls) > out2.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ cat out2.txt
2024. 10. 02. (수) 13:38:41 KST
/home/u2312282/linux/chap05
abc.c
h.c
input1.txt
out1.c
out1.txt
out2.c
out2.txt
out3.c
u2312282@dahee-VirtualBox:~/linux/chap05$
```

2) [예시]에 나온 명령어 중 **2개 이상을 바꿔** 명령어 열/그룹을 실습해보고 실행한 터미널창을 캡처한다.

```
u2312282@dahee-VirtualBox:~/linux/chap05$ cat commandtest.txt
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore magna aliqua.
Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris
u2312282@dahee-VirtualBox:~/linux/chap05$ wc commandtest.txt; which passwd; uname
  5  27 182 commandtest.txt
/usr/bin/passwd
Linux
u2312282@dahee-VirtualBox:~/linux/chap05$ (tail -3 commandtest.txt; which passwd; uname)
sed do eiusmod tempor incididunt ut labore magna aliqua.
Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris
/usr/bin/passwd
Linux
```

```
u2312282@dahee-VirtualBox:~/linux/chap05$ head -2 commandtest.txt; which passwd; uname > commandtest2.txt
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
/usr/bin/passwd
u2312282@dahee-VirtualBox:~/linux/chap05$ cat commandtest2.txt
Linux
```

```
u2312282@dahee-VirtualBox:~/linux/chap05$ (head -2 commandtest.txt; which passwd; uname) > commandtest2.txt
u2312282@dahee-VirtualBox:~/linux/chap05$ cat commandtest2.txt
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
/usr/bin/passwd
Linux
```

3) p16을 참고하여 **본인이 실습한 2)번 결과를 바탕으로** 명령어 4개에 대해 **차이점을 중심으로** 각각 설명한다.

5줄의 텍스트 파일 commandtest.txt를 생성한다.

\$ wc commandtest.txt; which passwd; uname 명령어를 실행하면 wc commandtest.txt(commandtest.txt의 글자수를 세어 출력한다), which passwd(passwd 파일의 위치 출력), uname(시스템 정보 출력)의 실행 결과가 순차적으로 출력된다.

\$ (tail -3 commandtest.txt; which passwd; uname) 명령어를 실행하면 괄호 안의 명령어 tail -3 commandtest.txt(commandtest.txt의 끝에서부터 세 줄 텍스트를 출력한다), which passwd, uname의 실행 결과가 순차적으로 실행된다.

\$ head -2 commandtest.txt; which passwd; uname > commandtest2.txt를 실행하면 head -2 commandtest.txt(commandtest.txt의 처음 두 줄을 출력한다), which passwd, uname > commandtest2.txt(uname의 출력 결과를 commandtest2.txt 파일에 입력한다)가 순차적으로 실행된다. commandtest2.txt 파일의 내용을 확인해 보면 uname의 출력만이 들어가 있다.

\$ (head -2 commandtest.txt; which passwd; uname) > commandtest2.txt 를 실행하면 괄호 안의 모든 명령어 head -2 commandtest.txt, which passwd, uname이 순차적으로 실행되고 해당 출력 결과가 모두 commandtest2.txt의 입력으로 들어간다. commandtest2.txt 파일의 내용을 확인해 보면 세 가지 모두의 출력이 순차적으로 들어가 있다.

이를 바탕으로 정리하면

\$ 명령어1; ...; 명령어n: 명령어 1부터 명령어 n까지의 명령어가 단순히 순차적으로 실행된다.

\$(명령어1; ...; 명령어n): 명령어 1부터 명령어 n까지의 괄호 안 명령어가 한 그룹으로 묶여서 순차적으로 실행된다.

\$ 명령어1; ...; 명령어n > out1.txt: 명령어 1부터 명령어 n > out1.txt까지의 명령어가 단순히 순차적으로 실행된다. out1.txt 의 내용으로는 명령어n의 출력만이 들어간다.

\$(명령어1; ...; 명령어n) > out2.txt: 명령어 1부터 명령어 n까지의 모든 괄호 안 명령어가 한 그룹으로 묶여서 순차적으로 실행되며, 그 다음 순서로 괄호 안 명령어들의 모든 출력이 out2.txt의 내용으로 들어간다.

각 경우에는 기본적으로 명령어를 순차적으로 실행하되 특정 명령어들이 괄호 안에서 그룹처럼 실행되는지 아닌지의 차이가 존재한다.