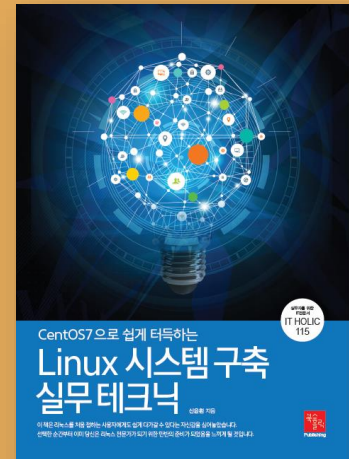


CHAPTER

02

리눅스 명령어 입문



Professor /

현재 시각과 날짜 출력

현재 시각과 날짜 출력 : date

리눅스 시스템에서 설정되어 있는 현재 시각과 날짜를 출력하기 위해서 사용되는 명령어는 date 입니다. date 명령어의 사용형식은 \$ 프롬프트 다음에 date 명령을 입력한 다음 **[Enter]**를 누르면 됩니다.

| 예제 2-1 |

현재의 날짜와 시각을 출력합니다.

\$ date

기능 시스템의 현재 시각과 날짜 출력

형식 date **[Enter]**



```
cskisa@localhost: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ date  
2016. 10. 03. (월) 13:10:11 KST  
[cskisa@localhost ~]$
```

[그림 2-2] date 명령어 실행 결과

현재 시각과 날짜 출력

시간동기화 : `timedatectl`

리눅스 시스템에서 구동되고 있는 날짜와 시각에 대해서 하드웨어 시계와 소프트웨어 시계, UTC, 타임존, 시간동기화 등을 출력할 때는 `timedatectl` 명령을 사용합니다.

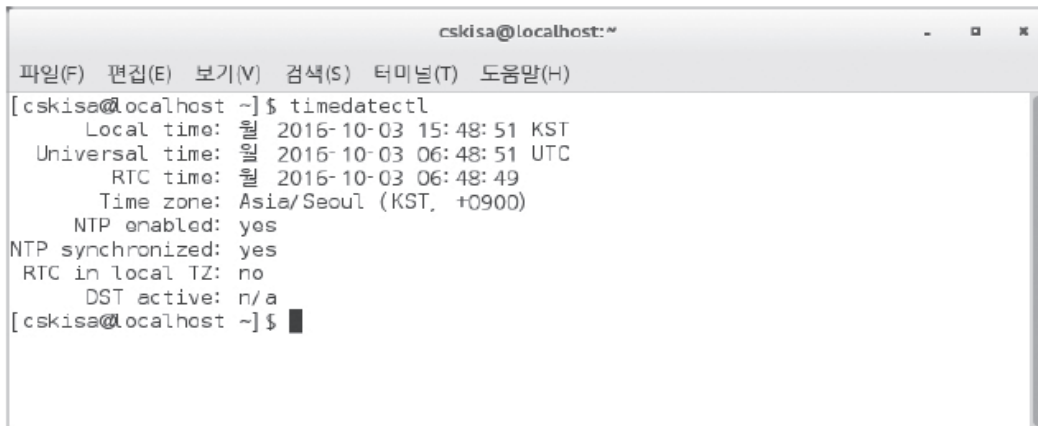
| 예제 2-2 |

시스템의 현재설정 상태를 확인합니다.

```
$ timedatectl
```

기능 하드웨어 시각, 소프트웨어 시각, 시스템 시각, 타임존, 시간동기화 등을 출력

형식 `timedatectl`



```
cskisa@localhost: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ timedatectl  
Local time: 월 2016-10-03 15:48:51 KST  
Universal time: 월 2016-10-03 06:48:51 UTC  
RTC time: 월 2016-10-03 06:48:49  
Time zone: Asia/Seoul (KST, +0900)  
NTP enabled: yes  
NTP synchronized: yes  
RTC in local TZ: no  
DST active: n/a  
[cskisa@localhost ~]$
```

[그림 2-3] `timedatectl` 명령어 실행 결과

연간 달력 출력

오늘의 날짜 출력 : `cal`

오늘 날짜를 기준으로 달력을 출력할 때는 `cal` 명령을 사용합니다. 실습을 통해 자세하게 살펴보도록 하겠습니다.


| 예제 2-3 |

오늘 날짜를 기준으로 해당하는 월의 달력을 출력합니다.

```
$ cal
```

기능 오늘 날짜에 해당하는 달력 출력

형식 `cal` [Enter]



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ cal  
    10월 2016  
일 월 화 수 목 금 토  
      1  
2  3  4  5  6  7  8  
9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30 31  
[cskisa@localhost ~]$
```

[그림 2-4] `cal` 명령어 실행 결과

연간 달력 출력

| 예제 2-4 |

특정 년도인 2020년의 달력을 출력합니다.

```
$ cal 2020
```

기능 옵션으로 지정한 년도에 해당하는 달력을 출력

형식 cal [옵션]

```

cskisa@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[cskisa@localhost ~]$ cal 2020
                2020

   1월           2월           3월
일 월 화 수 목 금 토   일 월 화 수 목 금 토   일 월 화 수 목 금 토
   1  2  3  4           1           1  2  3  4  5  6  7
   5  6  7  8  9 10 11   2  3  4  5  6  7  8   8  9 10 11 12 13 14
  12 13 14 15 16 17 18   9 10 11 12 13 14 15   15 16 17 18 19 20 21
  19 20 21 22 23 24 25   16 17 18 19 20 21 22   22 23 24 25 26 27 28
  26 27 28 29 30 31     23 24 25 26 27 28 29   29 30 31

   4월           5월           6월
일 월 화 수 목 금 토   일 월 화 수 목 금 토   일 월 화 수 목 금 토

```

[그림 2-5] cal 2020 명령어 실행 결과

실습 2-1 2025년도 달력을 출력하시오.

1. 2025년도 달력을 출력하는 명령어 사용하기
2. 금년도 오늘날짜가 포함된 달력 출력하기

\$ cal 2025

\$ cal

화면 지우기 : clear

| 예제 2-5 |

터미널 창에서 수행했던 모든 명령과 실행결과를 지웁니다.

```
$ clear
```

기능 터미널 창에 수행했던 모든 명령과 실행결과를 화면에서 지움

형식 clear

```
cskisa@localhost:~
```

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

```
[cskisa@localhost ~]$ clear
```

[그림 2-6] clear 명령어 실행

실습 2-2 지금까지 사용했던 모든 명령과 실행결과를 화면에서 모두 지우시오.

1. 지금까지 사용한 명령들의 목록 출력 : history 명령 사용
2. 터미널 창에 있는 모든 명령과 출력결과 삭제하기

```
$ history
```

```
$ clear
```


로그인 네임과 ID 확인

| 예제 2-6 |

리눅스 시스템에 접속하여 사용 중인 로그인 네임을 출력합니다.

\$ logname

기능 시스템에 접속하여 사용 중인 로그인 네임 확인

형식 logname



A terminal window titled 'cskisa@localhost:~' with standard window controls. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The command history shows the user typing '\$ logname' and pressing enter, resulting in the output 'cskisa'. The prompt returns to '\$'.

```
cskisa@localhost:~  
[cskisa@localhost ~]$ logname  
cskisa  
[cskisa@localhost ~]$
```

[그림 2-7] logname 명령어 실행 결과

로그인 네임과 ID 확인

| 예제 2-7 |

현재 리눅스 시스템에 접속한 사용자의 아이디를 출력합니다.

\$ users

기능 시스템에 접속하여 사용 중인 사용자들의 ID 확인

형식 users



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ users  
cskisa cskisa  
[cskisa@localhost ~]$
```

[그림 2-8] users 명령어 실행 결과

로그인 네임과 ID 확인

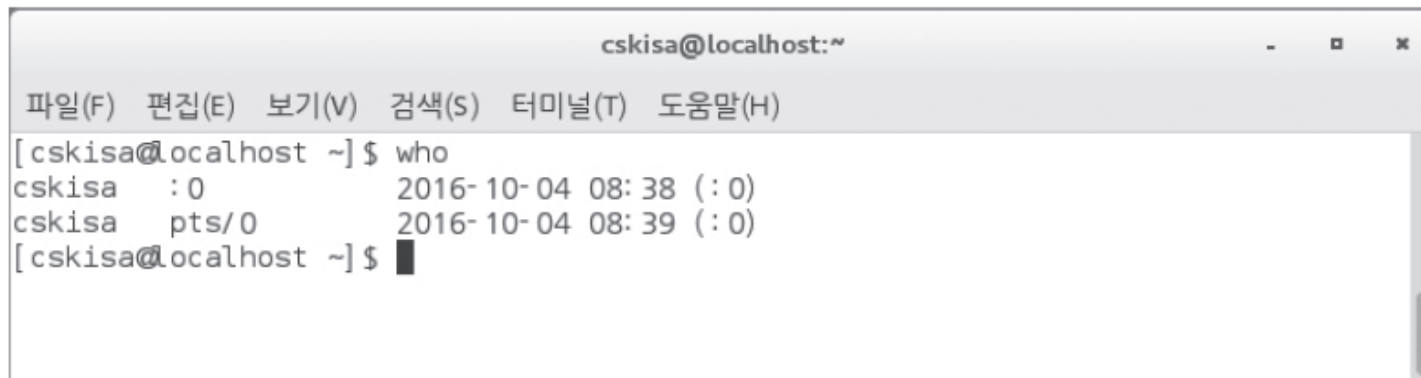
| 예제 2-8 |

리눅스 시스템에 로그인 한 모든 사용자의 계정과 터미널, 로그인 시각 등에 대한 정보를 출력합니다.

\$ who

기능 시스템에 로그인 한 모든 사용자의 계정과 터미널, 로그인 시각 등 확인

형식 who



A terminal window titled 'cskisa@localhost:~' with standard window controls. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The command prompt is '[cskisa@localhost ~]\$'. The output of the 'who' command is displayed as follows:

```
[cskisa@localhost ~]$ who
cskisa   :0                2016-10-04 08:38 (:0)
cskisa   pts/0            2016-10-04 08:39 (:0)
[cskisa@localhost ~]$
```

[그림 2-9] who 명령어 실행 결과

로그인 네임과 ID 확인


| 예제 2-9 |

현재 리눅스 시스템에 동시에 접속된 사용자를 출력합니다.

```
$ whoami
```

기능 시스템에 동시에 로그인 한 사용자 확인

형식 whoami



A terminal window titled 'cskisa@localhost:~' with a menu bar containing '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The terminal shows the command '[cskisa@localhost ~]\$ whoami' being entered, followed by the output 'cskisa' on the next line. The prompt '[cskisa@localhost ~]\$' is shown again with a cursor.

[그림 2-10] whoami 명령어 실행 결과

로그인 네임과 ID 확인

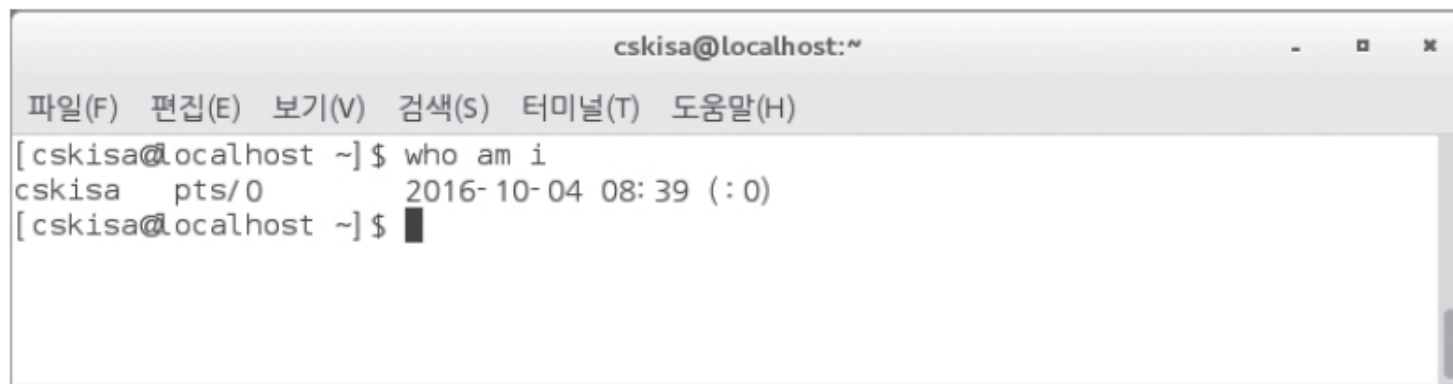
| 예제 2-10 |

현재 리눅스 시스템에 접속한 사용자 계정의 아이디와 pts, 접속한 시각 등 상세한 정보를 출력합니다.

```
$ who am i
```

기능 시스템에 접속한 사용자의 상세한 정보 확인

형식 who am I



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ who am i  
cskisa pts/0 2016-10-04 08:39 (:0)  
[cskisa@localhost ~]$
```

[그림 2-11] who am i 명령어 실행 결과

실습 2-3 현재 시스템에 접속된 사용자의 정보를 확인하시오.

1. 현재 접속하여 사용 중인 로그인 네임 확인하기
2. 리눅스 시스템에 접속된 사용자 아이디 확인하기
3. 시스템에 로그인한 모든 사용자의 계정과 터미널, 로그인 시각 등 확인하기

\$ logname

\$ users

\$ who

시스템 정보 출력 : uname

| 예제 2-11 |

현재 사용 중인 시스템에 대한 정보를 출력합니다.

\$ **uname** [옵션]

기능 시스템에 대한 정보 확인

형식 uname [옵션]

옵션 -a : 시스템의 모든 정보 확인

-m : 시스템이 사용 중인 하드웨어 정보 확인

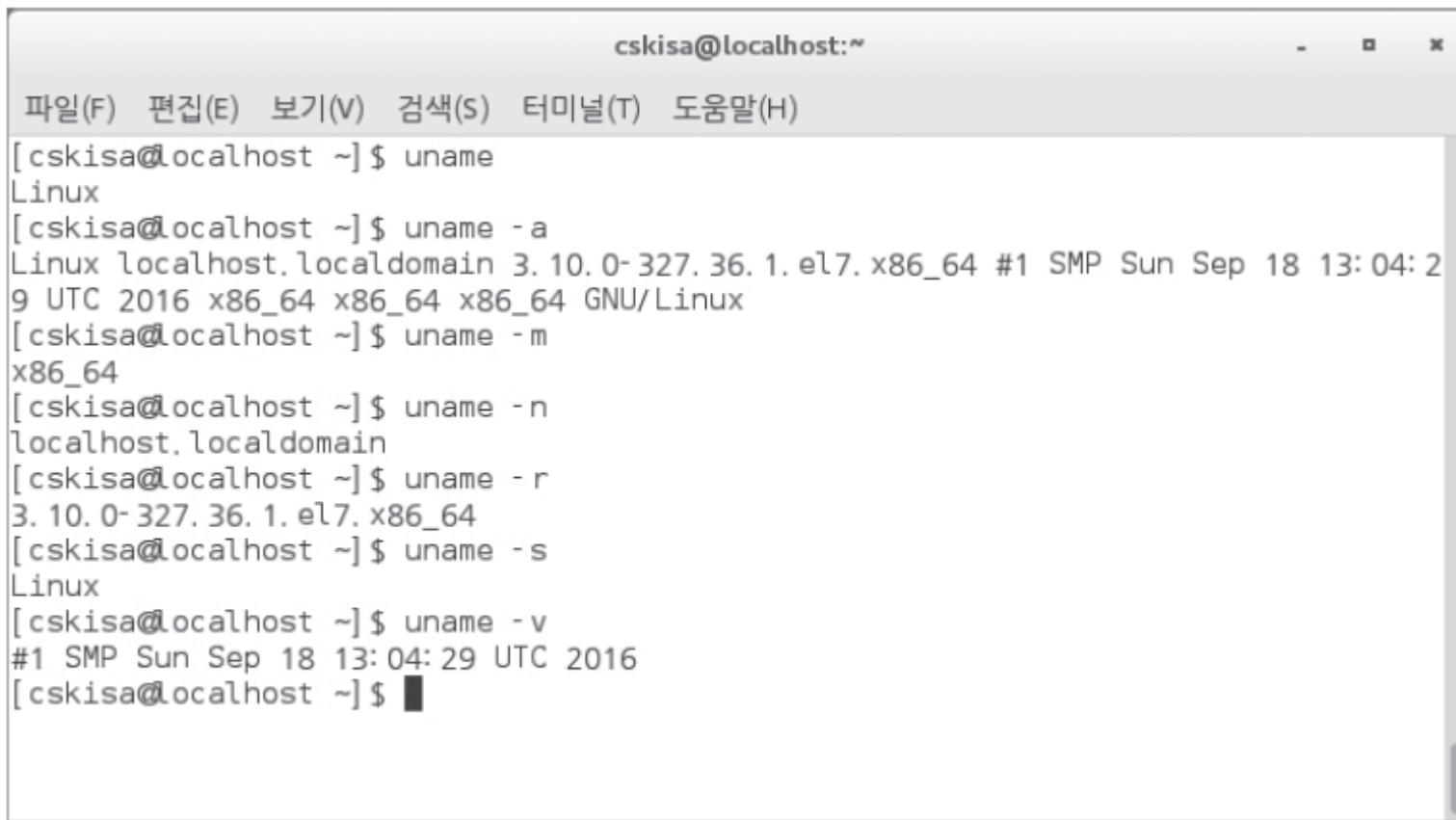
-n : 호스트네임 확인

-r : 운영체제의 릴리즈번호 확인

-s : 운영체제의 이름 확인

-v : 운영체제의 버전출시 일자 확인

시스템 정보 출력 : uname



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ uname  
Linux  
[cskisa@localhost ~]$ uname -a  
Linux localhost.localdomain 3.10.0-327.36.1.el7.x86_64 #1 SMP Sun Sep 18 13:04:29 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux  
[cskisa@localhost ~]$ uname -m  
x86_64  
[cskisa@localhost ~]$ uname -n  
localhost.localdomain  
[cskisa@localhost ~]$ uname -r  
3.10.0-327.36.1.el7.x86_64  
[cskisa@localhost ~]$ uname -s  
Linux  
[cskisa@localhost ~]$ uname -v  
#1 SMP Sun Sep 18 13:04:29 UTC 2016  
[cskisa@localhost ~]$
```

[그림 2-12] uname 명령어 실행 결과

실습 2-4 현재 사용 중인 시스템에 대한 정보를 출력하시오.

1. 현재 시스템이 사용 중인 하드웨어의 정보 확인하기
2. 현재 사용 중인 운영체제의 이름 확인하기
3. 현재 사용 중인 운영체제의 이름과 버전출시 일자 한꺼번에 확인하기

```
$ uname -m
```

```
$ uname -s
```

```
$ uname -sv
```

호스트 네임 출력 : hostname

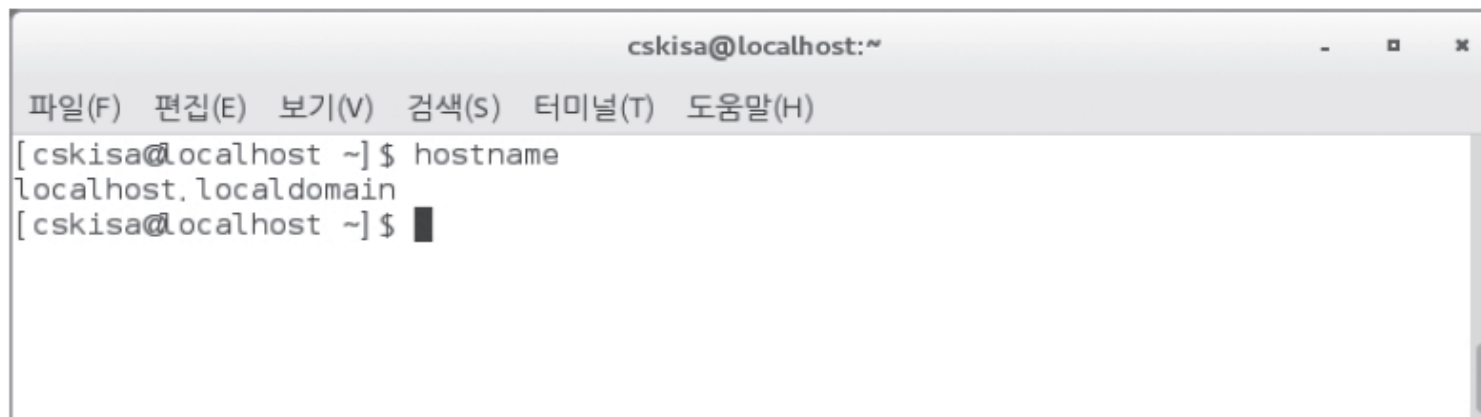
| 예제 2-12 |

현재 사용 중인 리눅스 시스템의 호스트 네임을 출력합니다.

```
$ hostname
```

기능 시스템의 호스트네임 정보 확인

형식 `hostname`



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ hostname  
localhost.localdomain  
[cskisa@localhost ~]$
```

[그림 2-13] hostname 명령어 실행 결과

하드웨어 정보 출력 : arch

| 예제 2-13 |

현재 시스템의 하드웨어 정보를 출력합니다.

\$ arch

기능 현재 사용 중인 컴퓨터의 CPU에 대한 정보 확인

형식 arch



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ arch  
x86_64  
[cskisa@localhost ~]$
```

[그림 2-14] arch 명령어 실행 결과

환경변수 확인 : env

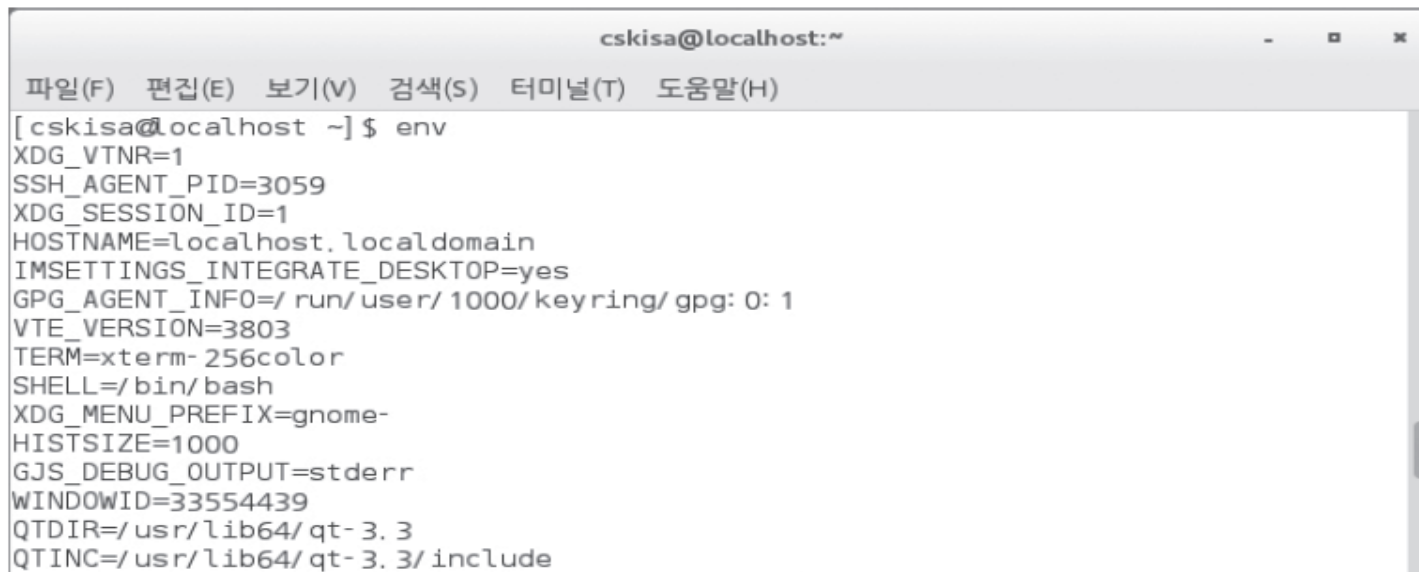
| 예제 2-14 |

현재 시스템에 설정되어 있는 환경변수에 대한 정보를 출력합니다.

```
$ env
```

기능 환경변수에 대한 정보 확인

형식 env



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ env  
XDG_VTNR=1  
SSH_AGENT_PID=3059  
XDG_SESSION_ID=1  
HOSTNAME=localhost.localdomain  
IMSETTINGS_INTEGRATE_DESKTOP=yes  
GPG_AGENT_INFO=/run/user/1000/keyring/gpg:0:1  
VTE_VERSION=3803  
TERM=xterm-256color  
SHELL=/bin/bash  
XDG_MENU_PREFIX=gnome-  
HISTSIZE=1000  
GJS_DEBUG_OUTPUT=stderr  
WINDOWID=33554439  
QTDIR=/usr/lib64/qt-3.3  
QTINC=/usr/lib64/qt-3.3/include
```

[그림 2-15] env 명령어 실행 결과

문자열 표준 출력 : echo

| 예제 2-15 |

지정한 문자열 'Have a good time.'을 표준형식으로 출력합니다.

```
$ echo Have a good time.
```

기능 지정한 문자열 및 환경변수 표준출력

형식 echo 출력할 문자열



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ echo Have a good time.  
Have a good time.  
[cskisa@localhost ~]$
```

[그림 2-16] echo 명령어로 간단한 문자열 출력

옵션 출력 : echo [옵션] 이스케이프

[표 2-1] 이스케이프 문자 종류 및 의미

이스케이프 문자	의미
"\a"	뻑~ 경고음
"\b"	백스페이스 적용
"\c"	마지막에 줄 바꿈 문자 출력 안 함
"\f"	폼 피드 형식으로 출력 (프린터에서 용지변경)
"\n"	줄 바꿈 문자 출력
"\r"	다음 행의 처음으로 커서 이동 (개리지 리턴)
"\t"	수평 탭 정렬
"\v"	수직 탭 정렬
"\w"	역슬래시 (또는 \) 출력
"\nnn"	아스키코드가 8진수(nnn)인 문자 출력

옵션 출력 : echo [옵션] 이스케이프

| 예제 2-16 |

줄 바꿈 옵션을 사용하여 하나의 문장을 두 줄로 출력합니다.

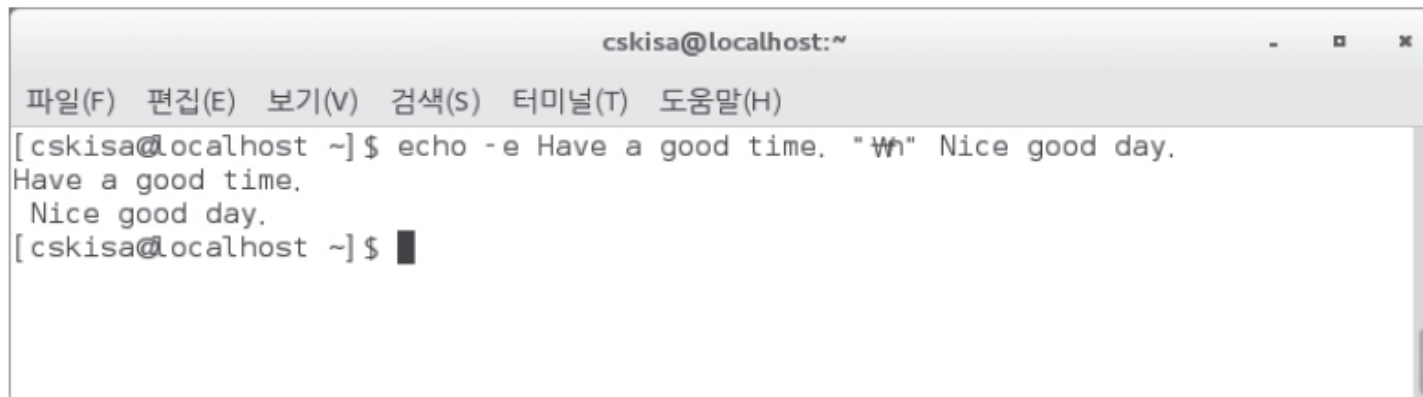
```
$ echo -e Have a good time. "\n" Nice good day.
```

기능 지정한 문자열 및 환경변수 출력

형식 echo [옵션] [문자열] [이스케이프 문자] [문자열] [Enter,↵]

옵션 -n : 문자열을 출력한 다음 줄을 바꾸지 않음

-e : 이스케이프 문자를 사용하기 위한 옵션



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ echo -e Have a good time. "\n" Nice good day.  
Have a good time.  
Nice good day.  
[cskisa@localhost ~]$
```

[그림 2-17] echo 명령어에 옵션과 이스케이프 문자를 사용한 실행 결과

실습 2-5 본인의 학과, 학번, 성명, 연락처를 다음과 같이 출력하시오.

1. 학과 : 빅데이터분석학과
2. 학번 : 201703125
3. 성명 : 홍길동
4. 전화 : 02-000-0000

```
$ echo -e 1.학과 : 빅데이터분석학과 "\n"2.학번 : 201703125 "\n"  
3.성명 : 홍길동"\n"4.전화 : 02-000-0000
```


명령어가 존재하는 디렉터리 경로 확인 : which

| 예제 2-17 |

알고 싶은 echo 명령어가 어느 디렉터리에 존재하는지에 대해서 알려줍니다.

```
$ which echo
```

기능 명령어가 어느 디렉터리에 존재하는지 확인

형식 which 명령어



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ which echo  
/usr/bin/echo  
[cskisa@localhost ~]$
```

[그림 2-18] which 명령어 실행 결과

실습 2-6 다음 항목에서 주어진 지시사항을 수행하시오.

1. cp 명령이 어느 디렉터리에 존재하는지를 출력하기
2. mv 명령이 어느 디렉터리에 존재하는지를 출력하기
3. cp와 mv 명령이 존재하는 디렉터리의 모든 파일목록 출력하기

\$ which cp

\$ which mv

\$ ls /usr/bin

지금까지 사용한 명령어 리스트 확인 : history

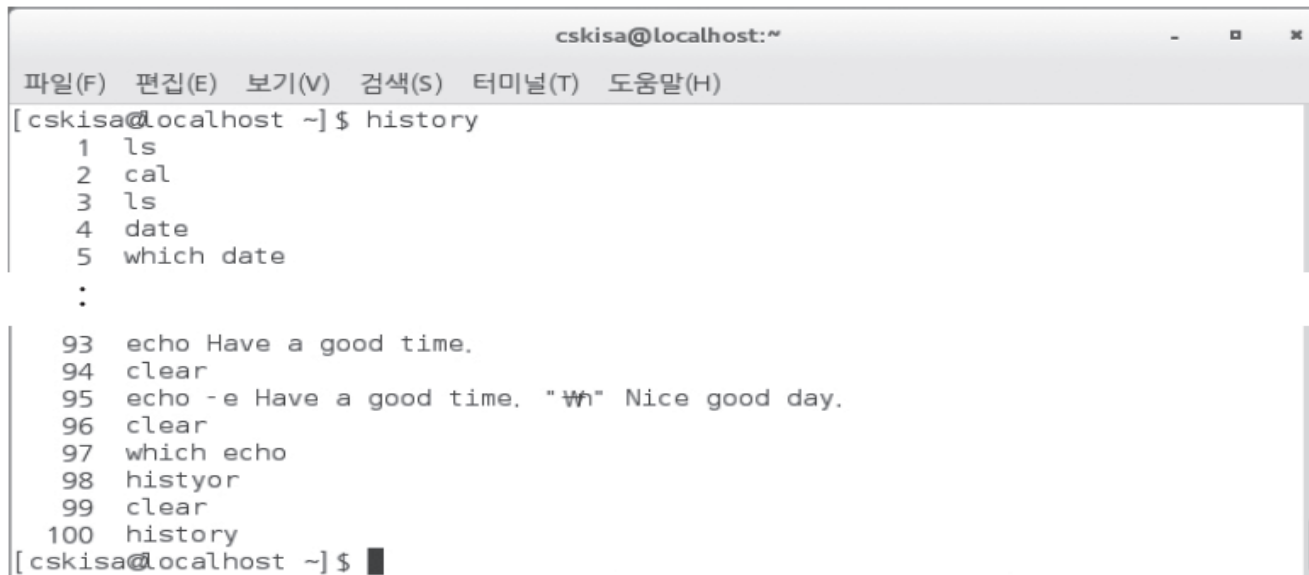
| 예제 2-18 |

리눅스 시스템의 터미널 창에서 지금까지 사용한 모든 명령어 리스트를 출력합니다.

```
$ history
```

기능 : 지금까지 사용했던 모든 명령어들을 보여줌

형식 : history



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[cskisa@localhost ~]$ history  
 1  ls  
 2  cal  
 3  ls  
 4  date  
 5  which date  
:  
93  echo Have a good time.  
94  clear  
95  echo -e Have a good time. "w" Nice good day.  
96  clear  
97  which echo  
98  histyor  
99  clear  
100 history  
[cskisa@localhost ~]$
```

[그림 2-19] history 명령어 실행 결과

사용했던 명령을 라인번호로 다시 실행 : ![라인번호]

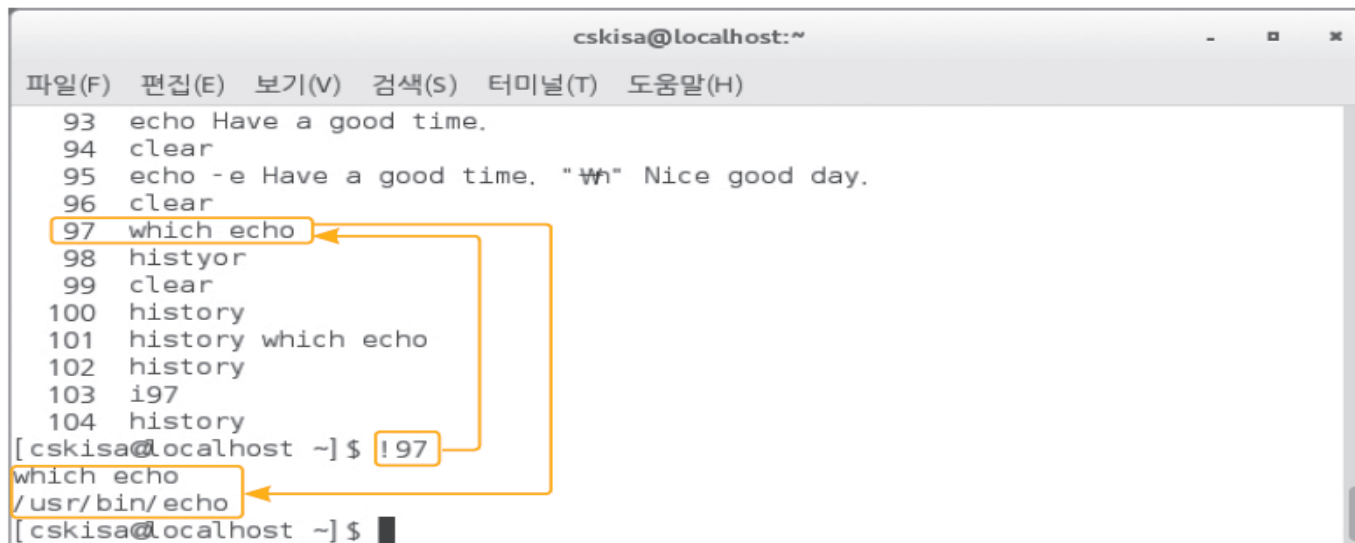
| 예제 2-19 |

리눅스 시스템에서 사용했던 명령들 중에서 특정 명령을 수행하기 위해 라인번호를 느낌표와 함께 지정하여 수행했던 명령을 다시 실행하도록 합니다.

\$!97

기능 : 사용했던 모든 명령어들 중에서 97번째 수행한 명령어를 다시 사용

형식 : [옵션][라인번호] [Enter]



The screenshot shows a terminal window titled 'cskisa@localhost:~'. The command history is displayed as follows:

```
93 echo Have a good time.
94 clear
95 echo -e Have a good time. "\n" Nice good day.
96 clear
97 which echo
98 histyor
99 clear
100 history
101 history which echo
102 history
103 i97
104 history
```

Below the history, the user enters the command `[cskisa@localhost ~]$!97`. This command is highlighted with an orange box. An arrow points from this box to the command `which echo` at line 97 of the history. Below the execution of `!97`, the output of the command is shown: `which echo` and `/usr/bin/echo`. Another arrow points from the `!97` command to the output line.

[그림 2-20] history 명령어 후 옵션명령 !97 실행 결과

라인번호를 지정하여 명령 삭제 : history -[옵션] [라인번호]

| 예제 2-20 |

리눅스 시스템의 터미널 창에서 수행했던 명령들 중에서 특정 라인번호를 지정하여 선택적으로 수행했던 명령을 삭제합니다.

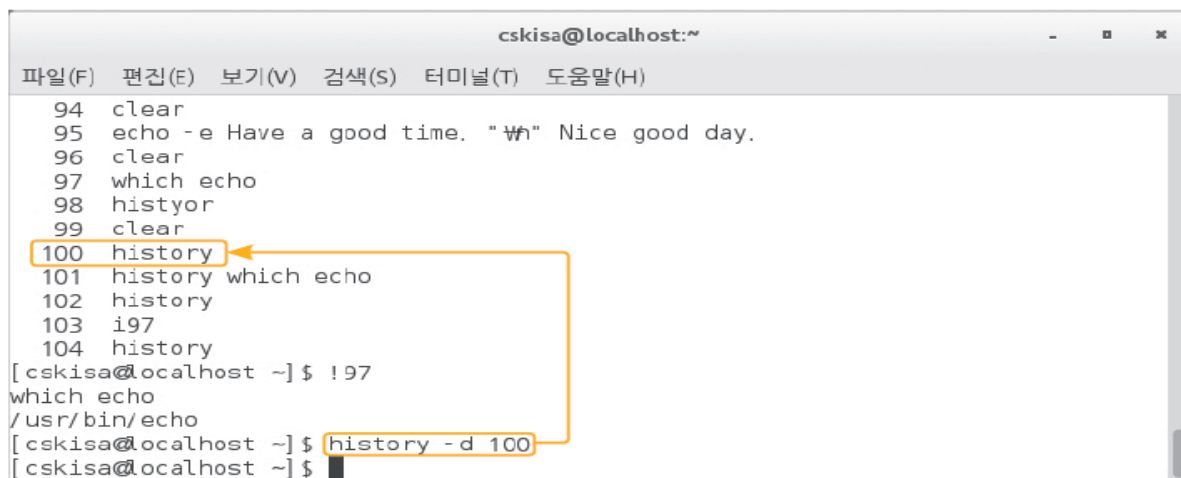
```
$ history -d 100
```

기능 지금까지 사용했던 모든 명령어 중에서 100번째 수행한 명령어를 다시 사용

형식 history [옵션] [번호] Enter

옵션 -d : 사용했던 명령어 중에서 라인번호를 지정하여 삭제

-c : 이전에 사용했던 모든 명령어를 삭제



```
cskisa@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
94 clear  
95 echo -e Have a good time. "ㄱ" Nice good day.  
96 clear  
97 which echo  
98 histyor  
99 clear  
100 history  
101 history which echo  
102 history  
103 i97  
104 history  
[cskisa@localhost ~]$ !97  
which echo  
/usr/bin/echo  
[cskisa@localhost ~]$ history -d 100  
[cskisa@localhost ~]$
```

[그림 2-21] history -d 100 명령어 실행

라인번호를 지정하여 명령 삭제 : history -[옵션] [라인번호]

| 예제 2-20 |

리눅스 시스템의 터미널 창에서 수행했던 명령들 중에서 특정 라인번호를 지정하여 선택적으로 수행했던 명령을 삭제합니다.

```
$ history -d 100
```

기능 지금까지 사용했던 모든 명령어 중에서 100번째 수행한 명령어를 다시 사용

형식 history [옵션] [번호] Enter

옵션 -d : 사용했던 명령어 중에서 라인번호를 지정하여 삭제

-c : 이전에 사용했던 모든 명령어를 삭제

```

cskisa@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
 94 clear
 95 echo -e Have a good time. "\n" Nice good day.
 96 clear
 97 which echo
 98 histyor
 99 clear
100 history
101 history which echo
102 history
103 i97
104 history
[cskisa@localhost ~]$ !97
which echo
/usr/bin/echo
[cskisa@localhost ~]$ history -d 100
[cskisa@localhost ~]$
  
```

[그림 2-21] history -d 100 명령어 실행

```

cskisa@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
 92 clear
 93 echo Have a good time.
 94 clear
 95 echo -e Have a good time. "\n" Nice good day.
 96 clear
 97 which echo
 98 histyor
 99 clear
100 history which echo
101 history
102 i97
103 history
104 which echo
105 history -d 100
106 history
[cskisa@localhost ~]$
  
```

[그림 2-22] history -d 100 명령어 실행 결과

사용했던 모든 명령들을 한꺼번에 삭제 : history -[옵션]

| 예제 2-21 |

리눅스 시스템의 터미널 창에서 수행했던 모든 명령들을 한꺼번에 삭제합니다.

```
$ history -c
```

기능 지금까지 사용했던 명령들을 모두 삭제

형식 history [옵션] [번호]

옵션 -d : 사용했던 명령들 중에서 삭제할 번호에 해당하는 명령 삭제

-c : 이전에 사용했던 명령들을 모두 삭제

```

cskisa@localhost ~$ history -c
92 clear
93 echo Have a good time.
94 clear
95 echo -e Have a good time. "\n" Nice
96 clear
97 which echo
98 histyor
99 clear
100 history which echo
101 history
102 i97
103 history
104 which echo
105 history -d 100
106 history
[cskisa@localhost ~]$ history -c

```

[그림 2-23] history -c 명령어 실행

```

cskisa@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
95 echo -e Have a good time. "\n" Nice good day.
96 clear
97 which echo
98 histyor
99 clear
100 history which echo
101 history
102 i97
103 history
104 which echo
105 history -d 100
106 history
[cskisa@localhost ~]$ history -c
[cskisa@localhost ~]$ history
1 history
[cskisa@localhost ~]$

```

[그림 2-24] history -c 명령어 실행 결과

실습 2-7 다음 항목에서 주어진 지시사항을 수행하시오.

1. 지금까지 사용한 모든 명령들의 목록 출력하기(없으면 명령어 5개 실행)
2. 수행했던 라인번호 3의 명령을 다시 실행하기
3. 사용한 명령들 중에서 라인번호 5번만 삭제하기
4. 지금까지 수행했던 모든 명령들을 한꺼번에 삭제하기

\$ history

\$!3

\$ history -d 5

\$ history -c

1. 다음 글을 읽고 빈 괄호를 채우시오.

- (1) 운영체제에서 제공하는 시각은 (**마더보드**)에서 배터리 전력으로 구동되는 하드웨어 시계인 RTC와 (**커널**)에 의해 관리되는 소프트웨어의 두 가지 형태가 있습니다.
- (2) 소프트웨어 시계는 시스템 부팅 시 (**RTC**)로부터 설정되어 있는 시각을 받아오며 초기 설정 이후에는 RTC와 상관없이 독자적으로 구동됩니다. 그리고 리눅스 시스템 시간은 (**UTC**)형태로 저장됩니다.

2. 다음 주어진 문장을 읽고 해당하는 명령어를 <보기>에서 찾아서 입력하십시오.

date	timedatectl	cal	clear
logname	users	who	whoami
who am I	uname	hostname	arch
env	echo	which	history

(1) 시스템의 현재 시각과 날짜를 출력	(date)
(2) 시스템에 대한 정보 확인	(uname)
(3) 하드웨어와 소프트웨어, 시스템 시각 및 타임존, 시간 동기화 출력	(timedatectl)
(4) 오늘 날짜에 해당하는 달력 출력	(cal)
(5) 명령어가 어느 디렉터리에 존재하는지 확인	(which)
(6) 터미널 창에서 수행했던 모든 명령어와 실행결과를 화면에서 지움	(clear)
(7) 시스템에 접속한 사용자의 상세한 정보 확인	(who am i)
(8) 지정한 문자열 및 환경변수 표준 출력	(echo)
(9) 시스템에 접속하여 사용 중인 로그인 네임 확인	(logname)
(10) 시스템에 접속하여 현재 사용 중인 사용자들의 ID 확인	(users)
(11) 지금까지 사용했던 모든 명령어들을 보여줌	(history)
(12) 시스템에 로그인 한 모든 사용자의 계정과 터미널, 로그인 시간을 확인	(who)
(13) 시스템의 호스트네임 정보 확인	(hostname)
(14) 현재 사용 중인 컴퓨터의 CPU에 대한 정보 확인	(arch)
(15) 시스템에 동시에 로그인 한 사용자 확인	(whoami)
(16) 환경 변수에 대한 정보 확인	(env)

3. echo 명령어에서 [옵션]으로 사용되는 이스케이프 문자의 의미에 해당하는 이스케이프 문자를 입력하시오.

이스케이프 문자	의미
(“Wa”)	백~ 경고음
(“Wb”)	백스페이스 적용
(“Wc”)	마지막에 줄 바꿈 문자 출력 안 함
(“Wf”)	폼피드 형식으로 출력 (프린터에서 용지 변경)
(“Wn”)	줄 바꿈 문자 출력
(“Wr”)	다음 행의 처음으로 커서 이동 (개리지 리턴)
(“Wt”)	수평 탭 정렬
(“Wv”)	수직 탭 정렬
(“WW”)	역슬래시 (또는 W) 출력
(“Wnnn”)	아스키코드가 8진수(nnn)인 문자 출력

4. 다음에 주어진 문장을 보고 옵션을 사용하여 터미널 창에서 표준으로 출력되도록 수행 하시오.

<출력 예시>

```
We are the korea student.  
Today we will learn Linux system.  
I am very happy.
```

```
$ echo -e We are the korea student. "Wn"Today we will learn Linux system.  
"Wn"I am very happy.
```

5. 2025년도 달력이 출력되도록 수행하시오.

```
$ cal 2025
```

6. 본인이 사용하고 있는 PC의 하드웨어와 소프트웨어, 시스템 시각, 타임존, 시간동기화 정보가 출력되도록 수행하시오.

```
$ timedatectl
```

7. date 명령이 어느 디렉터리에 있는지 확인할 수 있도록 수행하시오.

```
$ which date
```

8. 아래 주어진 예시와 같이 명령어를 수행하고 지시사항에 따라 수행결과가 출력되도록 하시오.

<수행 예시>

```
$ who am I  
$ uname -m  
$ timedatectl
```

(1) 수행 예시에서 수행한 모든 명령어를 출력하시오.

```
$ history
```

(2) 문(1)에서 출력된 명령어 리스트를 보고 두 번째로 수행했던 명령어를 다시 실행하시오.

```
$ ![uname -m 명령을 수행했던 행 번호 입력]
```

(3) 문(2)까지 수행한 명령어 중에서 세 번째로 수행했던 명령어만 삭제하시오.

```
$ history -d [timedatectl 명령을 수행한 행 번호 입력]
```

(4) 지금까지 수행했던 명령어를 한꺼번에 모두 삭제하고 삭제된 결과를 확인하시오.

```
$ history -c  
$ history
```

Chapter 02

Master

최상의 노력에 따른 인고의 가치는 반드시 증명될 수 있습니다!

수고 하셨습니다.