



## Unit 4 Basic Infomation



<http://cafe.daum.net/bsscolaris>

## **단원 목표**

---

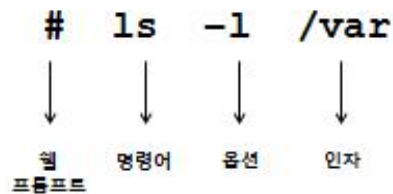
- 리눅스 명령어 형식
- 도움말과 암호 변경
- 시스템의 기본적인 정보 확인 명령어

---

<http://cafe.daum.net/bcsolaris>

## 리눅스 명령어 형식 (Linux Command Format)

### • 리눅스 명령어 형식



<http://cafe.daum.net/bsscolaris>

### ■ Linux CMD Format

리눅스에서는 GUI모드 보다는 TEXT모드를 위주로 지향하여 작업을 하기 때문에 명령어 사용법에 대해 익혀야 한다. 이때 명령어를 입력하게 되면 셸에서 이 명령어를 해석(전달)하여 작업을 하게 된다. 리눅스는 명령어 이름 외에도 옵션(Options)과 인자(Arguments)가 있다. 옵션은 명령어의 동작을 달리 할 때 사용하고 인자는 명령어의 인자나 혹은 옵션의 인자로서 사용이 될 수 있다.

#### (명령어 형식)

# Command [options] [Arguments]

[root@linux249 ~]#	ls	-l	/var
A	A	A	A
셸프롬프트	명령어	옵션	인자

명령어를 사용하는 다양한 형식은 다음과 같다.

- "# Command" 사용하는 경우
- "# Command Options" 사용하는 경우
- "# Command Arguments" 사용하는 경우
- "# Command options Arguments" 사용하는 경우
- 한 개의 명령어 라인에 여러 명령어 사용 하는 경우

(예) "명령어" 형식을 사용하는 경우

```
# ls
# cal
# uname
```

(예) "명령어 + 옵션" 형식을 사용하는 경우

```
# uname -sr
# uname -a
# ls -l
```

(예) "명령어 + 인자" 형식을 사용하는 경우

```
# cd /etc/sysconfig/network-scripts
# ls /var
# cal 2002
```

(예) "명령어 + 옵션 + 인자" 형식을 사용하는 경우

```
# ls -l /var
```

#### [참고] 옵션에 대한 설명

명령어의 옵션은 자리를 바꾸거나 혹은 합쳐서 사용해도 같은 의미로 동작한다.

```
# ls -a -l -F
# ls -al -F
# ls -a-lF
# ls -Fla
```

명령어의 옵션을 약자 형태로 쓰지 않기 위해서는 "--(double dash)"을 붙인다.

```
# ls --all    (# ls -a)
```

#### [참고] 옵션의 인자가 존재하는 경우의 예

명령어의 인자가 아닌 옵션의 인자 값인 경우에는 옵션의 위치가 변경이 될 때, 같이 따라다녀야 한다. 다음은 find 명령의 -name 옵션에 대한 인자가 존재하는 경우이다.

(예) # find / -name core -type f

```
(O) # cmd -i -f arg1
(O) # cmd -f arg1 -i
(O) # cmd -i f arg1
(X) # cmd -fi arg1
```

## 도움말과 암호변경

- man 명령어
  - # man ls
  - # man -k calendar
  - # whatis passwd
  - # man -s 5 passwd
  - # man ls | col -b > ls.txt
  - # info ls (# pinfo ls)
  - # which ls
  - # whatis ls
- passwd 명령어
  - # passwd
  - # passwd user01

<http://cafe.daum.net/bsscslaris>

## 1 man CMD

### 이름

man - 온라인 매뉴얼 페이지를 형식화하고 표시  
manpath - 맨 페이지를 위해 사용자의 검색 경로를 결정

### 사용법

man [-acdfHkKtW] [-m system] [-p string] [-C config\_file] [-M path]  
[-P pager] [-S section\_list] [section] name ...

### 설명

man 은 온라인 매뉴얼 페이지를 형식화하고 표시한다. 이 버전 은 MANPATH 와 (MAN)PAGER 환경 변수를 인식한다. 그러므로 자신의 개인적인 맨 페이지 집합을 가질 수 있고형식화된 페이지를 표시하기 위해 좋아하는 프로그램 을 선택할 수 있다. 만약 section 이 지정되면, man 은 단지 그 섹션에서만 매 뉴얼을 찾아 보여준다. 명령행 옵션이나 환경 변수를 통해서섹션 검색 순 서 와 소스 파일에 대해 어느 전처리기를 사용할 것인지 지정할 수 있다. 만약 name 이 / 을 포함하면, 이것은 파일명으로 먼저 처리된다. 그 래 서 man ./foo.5 혹은 man /cd/foo/bar.1.gz 처럼 지정 할 수 있다.

메뉴얼(manual) 페이지 확인, 명령어의 help 옵션과 비슷하다. MS 윈도우의 도움말(F1)과 비슷한 역할을 가진다.

### [명령어 형식]

```
# man ls           /* 명령어나 파일의 이름(데몬)으로 검색하는 경우 */
# man -k calendar  /* keyword로 검색하는 경우 */
# man -f passwd     /* passwd 이름을 가진 모든 매뉴얼 목록 확인 */
# man -s 5 passwd   /* section 번호로 검색하는 경우 */
```

[실습] 매뉴얼 페이지

## ■ 사용시스템

- server 1

## ■ 실습 시나리오

- 명령어에 대한 man 페이지 확인
- 명령어를 알지 못하는 경우
- 섹션 별로 검색 할 경우
- 추가적인 정보 확인 명령어

[EX1] 명령어에 대한 man 페이지 확인

```
(간략하게 정보 확인) # ls --help      (# CMD --help)
(자세하게 정보 확인) # man ls         (# man CMD)
```

```
[root@server1 ~]# man ls
```

```

LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

..... (중략) .....

```

다음과 같은 추가적인 사용을 해 본다.

- \* 다음 줄  
    <Enter>
- \* 다음 페이지/이전 페이지  
    <Space>|<CTRL + F>      <--> <CTRL + B>
- \* 단어 검색  
    /a                              <--> n(정방향,다음), N(역방향,이전)      (n: next)  
    /OPTIONS  
    /EXAMPLES  
    /SEE ALSO
- \* 매뉴얼 명령 사용법  
    h (help)
- \* 매뉴얼 빠져나가기  
    q (quit)

[EX2] 명령어를 알지 못하는 경우

!!!! 달력(calendar)을 출력하는 명령어는? !!!!

[root@server1 ~]# man -k calendar

```
calendar: nothing appropriate
```

-> 에러메세지가 나오면 아래와 같이 mandb 명령어 수행

[root@server1 ~]# mandb

-> 약간 시간이 걸린다.(약 1분 정도)

-> CentOS 7 버전부터 mandb 명령어로 변경 되었다.

-> CentOS 5 이하버전에서는 makewhatis 명령어를 사용하였다.

[root@server1 ~]# man -k calendar

```
cal (1)          - display a calendar
cal (1p)         - print a calendar
difftime (3p)    - compute the difference between two calendar time values
```

\* 매뉴얼 section 1 -> 명령어, 데몬에 대한 매뉴얼

\* 매뉴얼 section 2,3 -> 라이브러리에 대한 매뉴얼

```
# man -k time
# man -k ext4
# man -k xfs
```

[EX3] 섹션(section) 별로 검색 할 경우

같은 매뉴얼 이름을 가진 매뉴얼 페이지가 존재하는 경우:

\* passwd : passwd 명령어 매뉴얼

\* passwd : /etc/passwd 파일에 대한 매뉴얼

[root@server1 ~]# man -f passwd

```
passwd (5)          - password file
openssl-passwd (1ssl) - compute password hashes
passwd (1)          - update user's authentication tokens
```

• 매뉴얼 section 1: passwd 명령어 매뉴얼

• 매뉴얼 section 5: /etc/passwd 파일에 대한 매뉴얼

[root@server1 ~]# man -s 1 passwd (# man 1 passwd) /\* passwd 명령어 매뉴얼 페이지 \*/

[root@server1 ~]# man -s 5 passwd (# man 5 passwd) /\* /etc/passwd 파일 매뉴얼 페이지 \*/

#### SEE ALSO

```
chmod(1), locate(1), ls(1), updatedb(1), xargs(1), lstat(2), stat(2), ctime(3)
fnmatch(3), printf(3), strftime(3), locatedb(5), regex(7)
```

```
Full documentation <https://www.gnu.org/software/findutils/find>
or available locally via: info find
```

FIND(1)

[root@server1 ~]# man -s 3 strftime

[root@server1 ~]# man -s 2 lstat

[EX4] 추가적인 정보 확인 명령어

다음과 같은 명령을 사용하여 추가적인 정보를 확인할 수 있다.

[root@server1 ~]# info ls

[root@server1 ~]# pinfo ls

[root@server1 ~]# apropos ls (# man -k calendar)

[root@server1 ~]# man -k ls

```
[root@server1 ~]# whatis ls (# man -f ls)
[root@server1 ~]# whereis ls
[root@server1 ~]# which ls
```



## [참고] 매뉴얼 섹션(Manual Section)

## ■ Linux Manual Section

[Section 1](#) : 사용자 명령(실행가능한 명령 및 셸 프로그램)  
 Section 2 : 시스템 호출(사용자 공간에서 호출된 커널 루틴)  
 Section 3 : 라이브러리 기능(프로그램 라이브러리에서 제공)  
 Section 4 : 특수 파일(예: 장치 파일)  
[Section 5](#) : 파일 형식(많은 구성 파일 및 구조의 경우)  
 Section 6 : 게임(오락 프로그램용 섹션)  
 Section 7 : 범례, 표준 및 기타(프로토콜, 파일시스템)  
[Section 8](#) : 시스템 관리 및 권한 명령(유지 보수 작업)  
 Section 9 : 리눅스 커널 API(내부 커널 호출)

**[실무예]** 시스템 관리자가 알아 두어야 하는 매뉴얼 섹션

- Section 1 : 명령어 및 데몬등에 대한 매뉴얼
- Section 5 : 운영체제 설정 파일 및 데몬의 설정 파일 매뉴얼
- Section 8 : 관리명령어 매뉴얼

**[실무예]** 개발자가 알아 두어야 하는 매뉴얼 섹션

- Section 2 : 시스템 콜 함수 매뉴얼
- Section 3 : 일반 함수 매뉴얼
- Section 9 : 리눅스 커널 API 매뉴얼

## ■ Linux Manual Page Common Subject

NAME	: 간략한 설명
SYNOPSIS	: 명령 구문 요약
DESCRIPTION	: 주제에 대한 기본적인 이해를 위한 자세한 설명
OPTIONS	: 명령 실행 옵션과 설명
EXAMPLES	: 명령, 함수 또는 파일을 사용하는 방법에 대한 예제와 설명
FILES	: 도움말 페이지와 관련된 파일 및 디렉토리 목록
SEE ALSO	: 관련 정보로 일반적으로 다른 도움말 페이지 주제 표시
BUGS	: 소프트웨어의 알려진 버그
AUTHORS	: 주제 개발에 기여한 사람에 대한 정보

## 2

## passwd CMD

```

NAME
    passwd - update user's authentication tokens

SYNOPSIS
    passwd [-k] [-l] [-u [-f]] [-d] [-n mindays] [-x maxdays] [-w warndays]
    [-i inactivedays] [-S] [--stdin] [username]

DESCRIPTION
    The passwd utility is used to update user's authentication token(s).

    This task is achieved through calls to the Linux-PAM and Libuser API.
    Essentially, it initializes itself as a "passwd" service with Linux-PAM
    and utilizes configured password modules to authenticate and then
    update a user's password.

    A simple entry in the global Linux-PAM configuration file for this service
    would be:

        #
        # passwd service entry that does strength checking of
        # a proposed password before updating it.
        #
        passwd password requisite pam_cracklib.so retry=3
        passwd password required pam_unix.so use_authtok
        #

OPTIONS
    --stdin
        This option is used to indicate that passwd should read the new
        password from standard input, which can be a pipe.

FILES
    /etc/pam.d/passwd - the Linux-PAM configuration file

```

root 사용자가 일반사용자의 암호를 변경하는 경우 사용자의 암호를 물어 보지 않고 바로 변경이 가능하다. 하지만 일반 사용자가 자신의 암호를 변경하는 경우에는 이전 암호를 반드시 맞추어야만 새로운 암호로 변경 할 수 있고, 이때 **사용자 암호 변경 규칙이 적용**된다.

#### 암호 변경 권한 차이

- root 사용자 : 모든 사용자의 암호를 변경 가능
- 일반 사용자 : 자신의 암호만 변경 가능

#### 암호 변경 시 이전 암호 입력 여부

- root 사용자 : 암호 변경시 이전 암호 물어보지 않고 변경 가능
- 일반 사용자 : 이전 암호를 반드시 맞추어야만 새로운 암호 입력 가능

#### 암호 변경 규칙(ex: 일반 사용자가 암호를 변경하는 경우)

- 암호는 8글자 이상이어야 한다.
- 사용자이름(ex: user01)과 암호(ex: user01)는 동일할 수 없다.
- [이전암호]와 [새로운 암호]는 3글자이상은 틀려야 한다.
- 암호안에는 2글자 문자와 1글자의 숫자 또는 특수 기호가 포함되어 있어야 한다.
- 사전파일에 등록된 암호는 안된다.(ex: apple)
- 암호는 순서대로 사용하지 못한다.(ex: 1234abcd)

#### [명령어 형식]

```

# passwd          /* 사용자 이름이 생략되면 자신 */
# passwd fedora   /* fedora 사용자 암호 변경 */

```

[실습] passwd 명령어 실습

#### ■ 사용시스템

- server1

#### ■ 실습 시나리오

- [관리자]가 자신의 암호 변경하는 경우,
- 또는 [사용자]가 암호를 변경하는 경우를 확인해 본다.

[EX1] 관리자가 자신의 암호 변경하는 경우

- 다음 실습은 2개의 터미널을 띄워서 작업을 진행한다.

① (server1) root 사용자가 자신의 암호를 변경하는 경우

- 첫번째 터미널에서 root 사용자가 자신의 암호를 변경하는 방법을 실습해 본다.

[TERM1] root 사용자 터미널

[root@server1 ~]# passwd

```
Changing password for user root.
New UNIX password: (fedora)
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: (fedora)
passwd: all authentication tokens updated successfully.
```

[root@server1 ~]# passwd root

```
Changing password for user root.
New UNIX password: (centos)
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: (centos)
passwd: all authentication tokens updated successfully.
```

② (server1) root 사용자가 일반사용자(fedora) 암호를 변경하는 경우

[root@server1 ~]# passwd fedora

```
Changing password for user fedora.
New UNIX password: (123user)
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: (123user)
passwd: all authentication tokens updated successfully.
```

③ (server1) 변경된 사용자 암호 확인

- 일반사용자의 변경된 암호를 확인하기 위해서 ssh 명령을 사용하여 자신의 서버로 접속하면서 변경된 암호를 입력한다. 정상적으로 변경이 되었다면 로그인이 가능할 것이다.

[참고] 원격 접속 명령(ssh CMD)

■ 원격접속 명령어 : telnet CMD, ssh CMD

■ ssh 명령어 사용법 : ssh <사용자명>@<아이피> (# ssh root@192.168.10.10)

[TERM2] fedora 사용자 터미널

[root@server1 ~]# ssh fedora@localhost (# ssh fedora@127.0.0.1)

```
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 3b:25:5b:9a:ae:0f:ba:fc:85:66:73:d4:fc:e2:78:c6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
fedora@localhost's password: (123user)
```

[fedora@server1 ~]\$ id

```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

④ 사용자가 자신의 암호를 변경하는 경우

[fedora@server1 ~]\$ passwd

```
Changing password for user fedora.
Changing password for fedora
Current password: (123user) /* 이전 비밀번호 확인 */
New password: (acbd1324) /* 새로운 비밀번호 입력 */
Retype new UNIX password: (acbd1324)
passwd: all authentication tokens updated successfully.
```

## ⑤ 작업 종료

```
[fedora@server1 ~]$ exit  
[root@server1 ~]# exit
```

## ⑥ 복원 작업

- fedora 사용자의 암호를 fedora로 변경한다.

```
[root@server1 ~]# passwd fedora
```

-> 일반사용자는 아이디와 암호를 동일하게 설정할수 없다.

## [참고] 특수기호 영어 표현

■ 특수기호 영어표현(#, !, @, #, \$, %, ^, &, \*, (, ), [, ], :, ;, ', ', ~) 기타 자료  
- <http://achicgo.blogspot.com/2013/05/blog-post.html>

- **#** : Crosshatch (크로스해치) / Pound (파운드) / **Sharp(샤)** / **Hash(해쉬)** / TicTocToe(틱탁토)  
읽는 법에 문제가 많은 부호입니다. 특이하게 유독 이 기호만 ASCII코드를 만들 때부터 정식 명칭이 없는 것으로 알려져 있습니다. 국내에서는 "Sharp(샤)"으로 많이 부르고 외국에서는 천차만별인 것 같습니다만 요즘은 MS C# 덕분에 "Sharp(샤)"으로 통일화 되가는 추세인 것 같습니다.
- **^** : Circumflex (서큘플렉스) / **Caret (캐릿)**  
이 기호 이름을 모르는 사람이 아주 많을 것입니다. 한국어로는 삿갓머리, 윗꼭쇠라고 하는데요. 문장부호로는 "써큘플렉스"라고 읽고 일반적으로 "캐릿"이라고 부르는 것으로 되어 있습니다.
- **(** : Left Parenthesis(**레프트 퍼렌씨시스**)  
소괄호 열기
- **)** : Right Parenthesis(**라이트 퍼렌씨시스**)  
소괄호 닫기
- **{** : Left Brace (**레프트 브레이스**)  
중괄호 열기, 활괄호 열기. 영문명칭으로 말하는 사람을 별로 본적이 없네요.
- **}** : Right Brace (**라이트 브레이스**)  
중괄호 닫기, 활괄호 닫기.
- **[** : Left Bracket (**레프트 브라켓**)  
대괄호 열기, 꺾쇠괄호 열기
- **]** : Right Bracket (**라이트 브라켓**)  
대괄호 닫기, 꺾쇠괄호 닫기
- **!** : Exclamation Point (**익스클레메이션 포인트**), 느낌표  
문장부호의 느낌표를 나타내는 기호입니다.
- **"** : Quotation Mark (쿼테이션 마크) / Double Quotation (**더블 쿼테이션**)  
큰따옴표, 쌍따옴표로 부르면 되겠죠. 영문명칭은 제 경험으로는 볼 때 후자의 것을 쓰는 사람이 더 많은 것 같습니다.
- **'** : Quotation Mark (쿼테이션 마크) / Single Quotation (**싱글 쿼테이션**) / Apostrophe(어퍼스트로피) / Acute Accent(어큐트 엑센트)  
작은 따옴표로 부르면 되겠지요. 명칭이 여러개 있는 문자입니다. 어퍼스트로피로 쓰인다는 점을 알아 둘 필요가 있습니다. 다음에 나오는 기호와 함께 어떤 것이 어퍼스트로피인지 헷갈려 하시는 분들이 많더군요.
- **\$** : Dollar Sign (**달러사인**)  
설명할 필요가 없을 듯. 국가 코드에 따라 다른 표기로 바뀌는 경우도 있다고 하네요.
- **%** : Percent Sign (**퍼센트사인**)  
역시 설명이 필요 없을 듯
- **@** : At Sign (엣 사인, **엣**)  
국내에서는 골뱅이라고 부릅니다만 "At(엣)"이라고 불러야 하며 a자 곁에 동그라미로 둘러 싸거나 a의 꼬리가 길게 늘어나 글자를 감싸는 모양입니다. 의미상으로 주로 위치나 소속을 나타낼 때 씁니다. 때문에 전자우편에서 사용자계정과 소속도메인을 구분하는 기호로도 씁니다.
- **&** : Ampersand (**앰퍼샌드**)  
And(앤드)라고도 많이 부릅니다. "그리고"의 뜻이 강하지요.
- **\*** : Asterisk (**아스테리스크**)  
별표, 백설표로 불립니다. 컴 계열에서는 곱하기 기호로 사용되지요.
- **-** : Hyphen (**하이픈**) / Dash (**대쉬**)  
Minus(마이너스), Line(라인) 이라고도 부릅니다.
- **\_** : Under Line(언더라인), Underbar (**언더바**)  
우리말로 하면 밑줄이라고 해야겠죠. 언더바라고 부르는 사람이 많더군요.
- **.** : Period (피리어드) / Point (**포인트**)/Dot (**닷**)  
마침표. 설명이 필요없죠.
- **/** : Slash (**슬래시**)  
- 우리말로 "사선"이라고 하면 되겠지만 그렇게 쓰는 사람은 거의 없는 듯 합니다. 보다 유식해 보이려면 "슬래쉬" 발음을 연습해서 써먹으시다.
- **\** (W) : Back Slash (**백 슬래쉬**)  
버티칼 바인 "|" 문자와 같은 키에 위치하고 있으며 한글모드에서는 원화기호로 표기가 대체 됩니다. 영문 모드에서는 슬래쉬와 방향이 반대인 역사선 기호로 표기됩니다. 한국어자판에는 원화기호로 대신 되어 있는 것이 일반적입니다. 초보분들은 백슬래시라고 하면 원화기호를 말한다는 것을 기억해 두세요.
- **:** : Colon (**콜론**)  
한국말로 땡땡이라고 하나요? (—;) 어쨌든 이것은 "콜론"입니다.
- **;** : Semicolon (**세미콜론**)  
한국어로는 대체할 만한 말이 없죠. 우리 나라 문장기호에는 없는 까닭에 프로그래머가 아니면 한국사람들에게는 별로 친숙하지 않은 문자입니다.
- **|** : Vertical Bar (**버티컬바**)  
세로선, 도스나 유닉스에서의 명령프롬프트에서 쓰이는 기능 때문에 **pipe(파이프)**라고 부르기도 합니다.
- **~** : Tilde (**틸드**)  
물결기호입니다.
- **+** : Plus(**플러스**)  
말이 필요 없을 것 같군요. 그런데 이걸 "더하기" 라고 부르는 사람을 거의 못 본 것 같습니다. 왜일까요?
- **=** : Equal(**이퀄**)
- **?** : Question Mark (**퀘스천 마크**)

## 시스템 기본 정보 확인

- `uname` 명령어
  - # `uname -a`
  - # `cat /etc/redhat-release`
- `date` 명령어
  - # `date`
  - # `date +%m%d`
  - # `rdate -p time.bora.net`
  - # `rdate -s time.bora.net`
- `cal` 명령어
  - # `cal`
  - # `cal 2002`
  - # `cal 6 2002`

<http://cafe.daum.net/bcsolaris>

## 1

### uname CMD

#### 이름

`uname` - 시스템 정보를 출력한다

#### 개요

`uname [-snrvma] [--sysname] [--nodename] [--release]`  
`[--machine] [--all] [--help] [--version]`

#### 설명

이 맨페이지는 GNU 버전의 `uname` 을 설명한다. `uname` 은 현재 작동 중인 머신과 운영체제에 대한 정보를 출력한다. 아무런 옵션도 주어지지 않으면, `uname` 은 마치 `-s` 옵션이 주어진 것처럼 동작한다. 여러 개의 옵션이 주어지거나 `-a` 옵션이 주어지면 각 항목마다 스페이스 문자로 구분하여 'snrvm' 순서대로 선택된 정보를 출력한다.

#### 옵션

- `-m, --machine`  
머신(하드웨어) 타입을 출력한다.
- `-n, --nodename`  
머신의 네트워크 노드 호스트명을 출력한다.
- `-r, --release`  
운영체제 릴리즈 넘버를 출력한다.
- `-s, --sysname`  
운영체제의 이름을 출력한다.
- `-v`  
운영체제의 버전을 출력한다.
- `-a, --all`  
위 모든 정보를 출력한다.

운영체제에 맞는 애플리케이션(Application)을 설치하려면 운영체제의 종류와 버전에 대한 정보등 확인 할 수 있어야 한다. 또한 운영체제의 시간을 확인하고 서버와 동기화를 시키는 작업등을 할 수 있어야 한다.

`uname` 명령어를 사용하는 경우

- 운영체제의 종류(예: Linux)와 버전 확인(예: 5.14.0)
- 커널 버전 확인(예: 5.14.0-479.el9.x86\_64)
- 운영체제의 커널 비트 확인(예: x86\_64)

**[명령어 형식]**

# uname [OPTIONS]

**[명령어 옵션]**

옵션	설 명
-a	모든 정보
-s	운영체제 종류 (예: Linux)
-r	운영체제 릴리즈 (예: 2.6.18-164.11.1.el5)
-n	호스트 이름 (예: linux200)
-i	머신아키텍처 (예: i386)
-p	프로세스 종류 (예: athlon)

**[실습] uname 명령어 사용**

## ■ 사용시스템

- server1

## ■ 실습 시나리오

- uname 명령어 출력 결과 확인 및 해석
- 설치된 운영체제 종류 확인
- 문서(Documentation) 사이트(운영체제 버전별 정보 확인)

**[EX1] uname 명령어 출력 결과 확인 및 해석****[root@server1 ~]# uname -a**

```

❶Linux ❷server1.example.com ❸5.14.0-452.el9.x86_64 ❹#1 SMP PREEMPT_DYNAMIC Sat May 18 20:39:48 UTC
2024 ❺x86_64 ❻x86_64 ❼x86_64 ⓉGNU/Linux

```

## ■ uname 명령어의 출력 결과 해석

```

❶Linux          (-s) 커널 이름(kernel name)
❷server1.example.com (-n) 호스트 이름(Hostname)
❸5.14.0-452.el9.x86_64 (-r) 커널 릴리즈
❹#1 SMP PREEMPT_DYNAMIC Sat May 18 20:39:48 UTC 2024 (-v) 커널 버전
❺x86_64         (-m) 머신 하드웨어 이름
❻x86_64         (-p) CPU 종류
❼x86_64         (-i) 하드웨어 구현 플랫폼
ⓉGNU/Linux     (-o) 운영체제 종류

```

```

# uname -s      /* 커널 이름 */
# uname -n      /* 호스트 이름 */
# uname -r      /* 커널 릴리즈 */
# uname -sr     /* 커널 이름과 릴리즈 */
# uname -v      /* 커널 버전 */
# uname -m      /* 머신 하드웨어 이름 */
# uname -p      /* CPU 종류 */
# uname -i      /* 하드웨어 구현 플랫폼 */
# uname -o      /* 운영체제 종류 */

```

[EX2] 설치된 리눅스 운영체제 배포판의 종류 확인

- `uname -a` 명령 출력 결과에는 리눅스 배포판 버전에 대한 정보는 없다. 따라서 배포판 버전을 별도로 확인해야 한다. 이런 경우 `/etc/` 디렉토리에 `*release*` 파일이 일반적으로 존재한다. 이 파일의 내용을 확인하면 된다.

[root@server1 ~]# `uname -a`

```
Linux server1.example.com 5.14.0-603.el9.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Jul 31 19:01:25 UTC 2025
x86_64 x86_64 x86_64 GNU/Linux
```

[root@server1 ~]# `ls /etc/*release`

```
/etc/centos-release /etc/redhat-release /etc/system-release-cpe
/etc/os-release /etc/system-release
```

[root@server1 ~]# `cat /etc/centos-release`

```
CentOS Stream release 9
```

[EX3] 배포판 버전에 맞는 문서(Documentation) 사이트(운영체제 버전별 정보 확인)

<http://docs.redhat.com/>

([https://docs.redhat.com/ko/documentation/red\\_hat\\_enterprise\\_linux/9](https://docs.redhat.com/ko/documentation/red_hat_enterprise_linux/9))

<http://www.centos.org/docs/>

(<https://docs.centos.org/en-US/docs/>)

#### [참고] RHEL(RedHat Enterprise Linux)와 CentOS 리눅스

##### ■ RHEL/CentOS 리눅스 변천사

RHEL		CentOS		커널버전
발표일	RHEL 버전	발표일	CentOS 버전	
2002.05	2.1	2002.05	2.1	커널 2.4.9
2003.10	3.1	2004.03	3.1	커널 2.4.2.1
2005.02	4.0	2005.03	4.0	커널 2.6.9
2007.03	5.0	2007.04	5.0	커널 2.6.18
2010.11	6.0	2011.07	6.0	커널 2.6.32
2014.06	7.0	2014.07	7.0	커널 3.10.0



## 2 date CMD

### 이름

date - 시스템 시간과 날짜를 출력/설정한다.

### 개요

```
date [-u] [-d datestr] [-s datestr] [--utc] [--universal]
[--date=datestr] [--set=datestr] [--help] [--version]
[+FORMAT] [MMDDhhmm[[CC]YY][.ss]]
```

### 개요

이 맨페이지는 GNU 버전의 date 를 설명한다. 아무 인수없이 실행하면 date 는 현재 시간과 날짜(아래에서 설명할 '%c' 지시자의 형태대로)를 출력한다. '+' 로 시작하는 옵션이 있으면 현재 시간과 날짜를 주어진 인수가 지시하는 형태로 출력한다. 이는 'strftime' 함수에 주어지는 형식 문자열과 같은 형태이다. '%' 로 시작하는 지시자(directive)를 제외하고 문자열에 포함된 모든 문자가 그대로 출력된다.

### 예

이틀 전의 날짜를 출력하기 위해서는

```
date --date '2 days ago'
```

3 개월하고 하루 지난 날짜를 출력하기 위해서는

```
date --date '3 months 1 day'
```

올해 크리스마스 날짜를 출력하기 위해서는

```
date --date '25 Dec' +%j
```

오늘 날짜를 완전한 월 이름, 월 중 일을 포함하는 형식으로 출력하기 위해서는

```
date '+%B %d'
```

하지만 처음 9 일에 대하여 '%d' 는 2 자리를 맞추기 위하여 0을 채우기 때문에 예를 들어 'date -d 1-may '+%B %d' ' 라고 하면 'May 01' 라고 출력하므로 원하는 결과가 아닐 수도 있다.

한 자리 수 날짜에 대하여 앞에 0 이 안나타도록 하기 위해서는 비표준인 '-' 변형자를 사용하여 채움을 막는다.

```
date -d 1-may '+%B %-d'
```

date를 이용해서 시간을 확인한다. 만약 시간이 잘못 설정되어 있으면 date나 rdate명령어를 통해 시간을 재설정해준다. 이때 rdate명령어를 사용하는 경우 서버로부터 시간을 동기화하여 사용할 수 있으므로 시스템의 시간을 정확하게 설정 할 수 있다.

일반 사용자(EX: fedora)는 date 명령어를 통해 시간 확인만 가능하고, 관리자(EX: root)는 date 명령어를 통해 시간 확인이나 변경이 가능하다.

- root 사용자 : 현재 시간을 확인 및 설정을 할 수 있다.
- 일반 사용자 : 현재 시간만 확인 할 수 있다. (즉, 관리자가 아니면 시간을 변경 할 수 없다.)

### [명령어 형식]

```
# date /* 시스템 시간 출력 */
# date 07241300 /* 시스템 시간 변경 */
# date +%m%d /* 시간 출력 형태 변경(magic cookie - %m: month, %d: day) */
```

[참고] NTP(Network Time Protocol) : (예전) rdate CMD -> (현재) NTP Client 구성  
CentOS 8.X/9.X NTP 지원 프로그램

- chronyd 서비스(패키지 이름: chrony)

## [실습] 운영체제 시간 설정

## ■ 사용시스템

- server1

## ■ 실습 시나리오

- 운영체제 시간 확인 및 변경
- 운영체제 시간을 원격서버(NTP 서버)에 동기화 하기
- date 명령어의 시간 출력 형식 변경하여 출력하기

## [EX1] 운영체제 시간 확인 및 변경

## ■ 운영체제 시간 변경하기

- date CMD 사용하여 변경(# date 08161300.00)
- timedatectl CMD 사용하여 변경(# timedatectl set-time 2021-02-08 13:00:00)

[root@server1 ~]# date /\* 현재 운영체제에 설정되어 있는 시간 출력 \*/

```
Tue Jan 26 05:16:44 KST 2010
```

[root@server1 ~]# date 08161300 /\* 8월 16일 13:00 로 설정 \*/

```
Wed Aug 16 13:00:00 KST 2017
```

[root@server1 ~]# date 081613002009 /\* 2009년 8월 16일 13:00 \*/

```
Sun Aug 16 13:00:00 KST 2009
```

## [실무 예] 라이선스 있는 프로그램의 데몬 버전 사용시

- 데몬 버전 라이선스 기간이 지난 경우 -> date CMD 시간을 임시적으로 옛날 시간으로 변경
- data/timedatectl CMD 명령어를 이용하여 시간을 변경하는 경우에는 반드시 chronyd.service를 비활성하고 시간을 변경한 이후에 다시 chronyd.service를 활성화 한다.

## [EX2] 운영체제 시간을 원격 서버(NTP 서버)에 동기화 하기

- CentOS 7.X/8.X/9.X 버전에서 NTP 서버에 시간을 동기화 하기 위해서는 chrony 패키지를 사용하여 NTP 클라언트 로 구성해야 한다.
- 다음 목록은 일반적으로 실무에서 많이 사용하는 NTP 서버(한국기준)이다.
  - 원격 타임 서버 목록(NTP 서버) - <https://www.ntppool.org/zone/kr>
  - time.kriss.re.kr : 한국표준과학연구원 타임서버
  - time.bora.net : 보라넷 타임서버
  - time-a.nist.gov : 네트워크 타임 프로토콜(ntp.org)타임서버
  - kr.pool.ntp.org

## ① 패키지 설치 확인

[root@server1 ~]# yum list chrony

```
Last metadata expiration check: 0:29:53 ago on Sat 06 Feb 2021 10:34:16 PM KST.
Installed Packages
chrony.x86_64                               3.5-1.el8                                @anaconda
```

## ② NTP 서버 지정

[root@server1 ~]# gedit /etc/chrony.conf

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
[수정전]
pool 2.centos.pool.ntp.org iburst
[수정후]
# pool 2.centos.pool.ntp.org iburst
server kr.pool.ntp.org iburst
```

- \* pool .... 라인은 주석처리(#)하고
- \* server .... 라인을 추가한다.

## [참고] NTP 서버 목록

- \* <https://www.ntppool.org/zone/kr>
- 1.asia.pool.ntp.org
- 2.kr.pool.ntp.org

- 3.asia.pool.ntp.org

③ 서비스 재기동

```
[root@server1 ~]# systemctl stop chronyd  
[root@server1 ~]# systemctl start chronyd  
[root@server1 ~]# systemctl enable chronyd
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/chronyd.service →  
/usr/lib/systemd/system/chronyd.service.
```

-> 시간이 동기화 되는데 시간이 걸릴수도 있다.

-> timedatectl

-> chronyc sources -v

[EX3] date 명령어의 시간 출력 형식 변경하여 출력하기

- 대부분의 프로그램은 하루에 한번 씩 로그 파일을 만들도록 하는 경우가 많이 존재한다.
- 이런 경우 date 명령어를 아래와 같이 응용하여 사용하는 경우가 많다.

시간 정보 출력 포맷을 변경하기

```
# date +%m%d          /* %m : month, %d : day */
# date +%m%d_%H%M     /* %H : hours, %M : Minutes, %S : Seconds */
```

실무에서 많이 사용되는 [웹 로그] 파일 이름 형식 : file\_0807.log

```
[root@server1 ~]# mkdir -p /test
[root@server1 ~]# cd /test
[root@server1 ~]# touch file_`date +%m%d`.log
[root@server1 ~]# ls -l
```

```
-rw-r--r-- 1 root root 0 Feb 1 02:40 file_0201.log
```

```
# date -d '1 day ago' +%m%d (# date -d '-1 days' +%m%d)
# date -d '2 day ago' +%m%d (# date -d '-2 days' +%m%d)
```

#### [참고] 시간의 종류(Clock Type)

시간(Clock)

- Software Clock : 운영체제의 시간
- Hardware Clock : 하드웨어(펌웨어) 시간

##### ■ 시스템 시간(Software clock)

# date

```
Mon Feb 1 01:46:01 KST 2010
```

##### ■ 하드웨어 시간(Hardware Clock)

# hwclock --show

```
2013년 07월 24일 (수) 오후 02시 19분 34초 -0.720006 seconds
```

```
# hwclock --systohc /* System Time -> Hardware Clock */
# hwclock --hctosys /* Hardware Clock -> System Time */
```

(OS 시간을 변경하는 예제)

- 문제 원인: OS 시간이 잘못된 경우가 발생이 되었다.
- 해결 방법: NTP 서버에 시간 동기화해야 한다.
- 전제 조건: NTP 서비스가 실행되어 있는 상태이다.

# systemctl stop chronyd

(⌵) timezone 설정(timedatectl set-timezone Asia/Seoul)

(⌵) 시간 동기화(gedit /etc/chrony.conf)

# systemctl start chronyd

# hwclock --systohc

### 3 cal CMD

**NAME**  
cal - 달력 출력

**SYNOPSIS**  
cal [-jy] [month [year]]

**DESCRIPTION**  
Cal 명령은 간단한 달력을 보여준다. 아무런 명령행 인자가 없으면, 시스템의 현재 달을 보여준다. 사용할 수 있는 옵션은 다음과 같다: The options are as follows:

- j      율리우스력(구태양력)으로 출력.
- y      현재 년도 모든 달을 모두 보여준다.

사용할 수 있는 연도는 1-9999 이며, 년도를 지정할 경우에는 세기가 생략되어서는 안된다. "cal 89" 명령이면 1989년 달력을 보여주지 않고 89년 달력을 보여준다. 연도 다음에 오는 숫자는 달로 인식한다. 그렇기 때문에, 1-12까지만의 숫자만 사용할 수 있다. 이런 명령행 인자가 아무것도 없으면, 현재 달의 달력만 보여준다.

즉, 명령행 인자로 사용할 수 있는 가장 작은 값은 1 1 (서기 1년 1월)이다.

그레고리오력(지금의 태양력)은 1752년에 9월에 세번째 개정된 것을사용한다. 이번 버전에서는 1900년 이전의 년도들에 대해서 제대로 인식한다. 재 개정되면서 1900년 이전 년도들의 날짜 수가 10일 삭제되었다. 그래서, 1900년 이전의 달력들을 볼 때 조금 이상하게 생각할 수도 있다.

달력(calendar)을 출력하는 명령어이다. 시스템에서 달력을 출력하는 경우는 드물고, 보통 윈도우 환경에서 오른쪽 하단의 시간 아이콘을 클릭하여 확인 하는 경우가 대부분이다. 하지만 환경만 사용이 가능하고, 스케줄러(예:crontab)를 사용하는 경우라면 특정 날짜와 요일을 확인 할 때 유용하게 사용된다.

#### [명령어 형식]

```
# cal
# cal 2002
# cal 6 2002
```

#### [실습] cal 명령어 실습

- 사용시스템
  - server1
- 작업 시나리오
  - cal 명령어 간단한 실습
  - 자신의 태어난 요일 확인

#### [EX1] cal 명령어 간단한 실습

- 다음과 같은 정보를 확인한다.
  - \* 현재 월 달력
  - \* 2002년 월드컵 달력 확인(월드컵 개막식: 2002년 5월31일, 폐막식: 2002년 6월30일)
  - \* 2002년 월드컵이 열렸던 달(요일 확인)

```
# cal                           /* 현재 월의 달력 표시 */
```

```
January 2010
Su Mo Tu We Th Fr Sa
                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
```

```
# cal 2002          /* 2007년 달력 표시 */
-> 출력 내용 생략
```

```
# cal 6 2002        /* 2004년 8월 달력 표시 */
```

```
      June 2002
Su Mo Tu We Th Fr Sa
                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
```

[EX2] 자신이 태어난 요일 확인

- 다음과 같은 정보를 확인한다.
  - \* 자신은 무슨 요일에 태어났나요?

```
# cal 6 2002
```

```
-> 생년/월/일
```

```
-> 나는 무슨 요일에 태어났나요?
```

[질문 & 답변]

[질문 1] 현재 사용하고 있는 OS의 **커널 버전**과 **배포판 종류**는 어떻게 되나요?

[답변 1]

[질문 2] 배포판에 맞는 적절한 매뉴얼은 어디에서 찾을 수 있나요?

[답변 2]

[질문 3] 스케줄링 작업(정기적인 잡 실행)을 하려고 하는데 2번째 주 토요일은 며칠인가?

[답변 3]

[질문 4] 스크립트 제작 시 다음과 같은 로그 파일을 명령어로 생성하려고 하는데 어떻게 사용하나요?

[조 건] 파일명: file\_0407.log

[답변 4]