



## 1. 리눅스 기초 및 실습2

(주)주스  
박종원

[jason.park@juice.co.kr](mailto:jason.park@juice.co.kr)

## 1. 사용자 그룹관리

- 1) 사용자, 그룹, 권한관리
- 2) 그룹관리 명령어
- 3) 패스워드, 그룹관련 설정 파일

## 2. 권한의 이해 및 표시

- 1) 파일, 디렉토리의 소유자
- 2) 권한의 이해
- 3) 권한의 숫자 표기법
- 4) 권한설정

## 3. 링크파일

- 1) 하드링크
- 2) 심볼릭링크

## ● 1. 강의 들어가기

---

### ✓ 학습내용 소개

- 리눅스 서버 운영체제는 다양한 사용자가 접속하여 시스템을 이용하게 된다. 각각의 사용자는 서버 시스템을 사용할 수 있는 범위를 제한을 가지고 파일과 디렉토리등에 접근할 수 있도록 되어 있다. 시스템 관리자는 모든 권한을 가지고 시스템을 운영할 수 있으며, 일반 사용자는 자기의 영역을 사용하고 이 영역을 다른 사용자들이 사용하거나, 사용하지 못하게 조정할 수 도 있다. 이번 강의에서는 리눅스 권한관리 부분을 이해 후 실습하도록 한다.
- 앞에서 서버에 접속하기 위하여 간단하게 사용자 및 그룹의 개념에 대하여 학습하였다. 이러한 사용자와 그룹을 관리하는 방법과 설정파일에 대하여 하나 하나 배워보도록 한다.

### ✓ 학습목표 제시

- 사용자,그룹, 권한관리에 대하여 이해할 수 있다.
- 그룹을 관리 할 수 있다.
- 파일이나 디렉토리의 소유자의 개념을 이해할 수 있다.
- 권한의 개념을 이해하고 변경 할 수 있다.
- 링크파일의 개념을 이해할 수 있다.

## ● 2. 생각해볼 문제 및 용어

---

### ✓ 학습전 생각해볼 문제

- 윈도우에서 일반 사용자를 만들어 관리자(Administrator)와 다른 권한을 주는 방법을 알아봅니다.
- 윈도우의 탐색기를 통하여 파일 속성( [읽기전용] [숨김] 등)을 바꿔봅니다.
- 바탕화면의 아이콘과 실행 파일(\*.exe)의 차이를 알아봅니다.

### ✓ 용어 ( 강의 정리 시 필기 할 것)

- Link
- 권한, 관리자

## (1) 사용자 그룹관리

앞에서 서버에 접속하기 위하여 간단하게 사용자 및 그룹의 개념에 대하여 학습하였다. 이러한 사용자와 그룹을 관리하는 방법과 설정파일에 대하여 하나 하나 배워보도록 한다.

### 1) 사용자, 그룹, 권한관리

유닉스, 리눅스는 여러 사람이 사용하는 다중사용자 운영체제로 사용자 묶음의 그룹 개념이 존재한다

① 사용자는 여러 개의 그룹에 포함될 수 있음

② 사용자 및 그룹관리 명령어

·id : 현재의 사용자를 알아보는 명령

·groups : 현재의 그룹을 알아보는 명령

·adduser : 사용자를 등록하는 명령어

·addgroup: 그룹을 등록하는 명령어

·deluser: 그룹을 등록하는 명령어

·delgroup: 그룹을 등록하는 명령어

③ 사용자와 그룹은 시스템 내부에서 숫자로 표시됨

·uid: 사용자를 표시하는 숫자

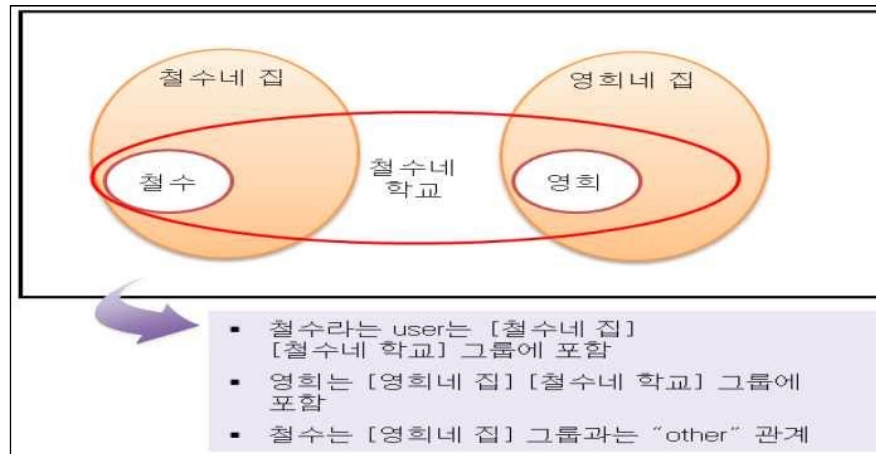
·gid: 그룹을 표시하는 숫자

·<그림 II-xx>에서 보면 kopoctc이라는 사용자는 uid(사용자id)가 1000이고 gid(그룹id)는 1000이며, 해당되는 그룹은 kopoctc이라는 그룹 한 개에만 속함

```
kopoctc@kopoctc:~$ id
uid=1000(kopoctc)                                gid=1000(kopoctc)
groups=1000(kopoctc),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),114(lpadmin)
,115(sambashare)
kopoctc@kopoctc:~$ groups
kopoctc adm cdrom sudo dip plugdev lpadmin sambashare
kopoctc@kopoctc:~$
```

<그림 II-1> id, groups 명령어

④ 사용자와 그룹관계를 나타내는 그림 <II-1> 참고



<그림 II-2> 사용자와 그룹의 관계

## 2) 그룹관리 명령어

파일을 탐색하기 위한 명령으로 `pwd`, `cd`, `ls`가 있다. <그림 I -35>

### ① 그룹조회

- 자기가 속한 그룹 : `groups` , `id`
- 서버 내 전체 정의되어 있는 그룹을 보려면 `/etc/group` 파일을 봄 <그림 II -3>

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kopoctc,syslog
tty:x:5:
disk:x:6:
voice:x:22:
cdrom:x:24:kopoctc
floppy:x:25:
tape:x:26:
sudo:x:27:kopoctc
audio:x:29: dip:
x:30:kopoctc
"/etc/group" [readonly] 57L, 774C
```

<그림 II -3> group 정의 파일

### ② 그룹을 생성하기 위하여 `groupadd`를 사용

#### Tip

예) `groupadd -g 900 kopoctc` : kopoctc라는 그룹을 gid를 900번으로 생성

③ 그룹을 변경하기 위하여 **groupmod**를 사용

**Tip** 예) `groupmod -g 700 kopogroup`: kopogroup라는 그룹을 gid를 700번으로 변경

**Tip** 예) `groupmod -n newkopo kopogroup`: kopogroup라는 그룹을 명칭을 newkopo로 변경

③ 그룹을 삭제하기 위하여 **delgroup**를 사용

**Tip** 예) `delgroup newkopo`: newkopo라는 그룹을 삭제



### 3) 패스워드, 그룹관련 설정 파일

유닉스, 리눅스 시스템에는 패스워드, 사용자, 그룹에 관련되어 설정되어 있는 중요파일이 있다.

#### ① 사용자 정보 파일

· /etc/passwd 파일

· 사용자의 정보로 user, password, uid, pid 정보가 기록되며 해당 파일의 수정 삭제 등으로 사용자관련 설정 변경도 가능

#### ② 그룹 정보 파일

· /etc/group 파일

· 그룹의 정보가 기록되며 해당 파일의 수정 삭제 등으로 사용자 관련 설정 변경도 가능

· groupadd, groupmod, delgroup 의 명령어를 통하여 그룹관리를 하는 것이 일반적

#### ③ 패스워드관련 파일

· /etc/shadow 파일 : /etc/passwd 파일과 함께 사용자 패스워드를 저장

· 단 패스워드는 암호화 되어있는 문장으로 패스워드는 함부로 바꿀 수 없으며 이 파일에서 패스워드 필드를 고치면 시스템 오류가 발생

· passwd 파일 내부 형식

```
username:password:uid:gid:gecos:homedir:shell
```

· username : 사용자명

· password : 사용자 암호

· uid : 사용자아이디, 그룹아이디

·gid : 사용자아이디, 그룹아이디

·gecos : General Electric Comprehensive Operating System (예전 Unix서비스와 호환성을 갖추기 위하여 만든 필드, 처음 사용자 정보 넣은 값들이 저장)

·homedir : 해당 사용자의 기본 디렉토리

·shell : 해당 사용자가 사용하는 Unix shell의 종류

**Tip**

모든 시스템 정보파일은 root 사용자만 수정 및 쓰기 권한을 갖을 수 있도록 권한이 제한되어 있다. 그러므로 해당 파일을 조회할 때도 수정되지 않도록 주의를 기울인다.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,:/nonexistent:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
"/etc/passwd" [readonly] 29L, 1415C
```

<그림 11-4> 사용자 정보파일

## ● 4. 실습하기(1)

---

### 1) 사용자, 그룹 관리

- ① `/etc/passwd` 조회 (cat으로 조회후 캡처)
- ② 각 위치의 내용을 확인하고 앞 설명과 일치하는 지 확인(생략)
- ③ 사용자 생성, 삭제를 통한 `passwd`파일 변경확인(패스워드 변경전후 캡처)
- ④ 해당 id로 로그인 한 후 해당 id가 속하는 그룹과 홈 디렉토리 등을 확인 (두개이상ID로 로그인 후 기본디렉토리 캡처)
- ⑤ 그룹조회, 그룹생성, 그룹변경, 그룹삭제 과정 실습.
- ⑥ 각 절차 별로 `/etc/group`파일을 조회하여 변경유무를 확인
- ⑦ 사용자(user)가 속하는 그룹을 변경, 추가함에 따라 권한이 변경되는지 확인.
- ⑧ `/etc/shadow`파일 조회 캡처

## (2) 권한의 이해 및 표기

다중 사용자를 위한 운영체제인 유닉스, 리눅스에서는 파일이나 디렉토리에 대하여 권한을 주어서 관리한다. 이러한 권한을 표기하거나 관리하는 방법에 대하여 하나하나 배워보도록 한다.

### 1) 파일, 디렉토리의 소유자

파일이나 디렉토리는 소유자가 해당사용권한을 가지고 있으며, 해당 파일과 디렉토리의 사용자 및 관리자( `root`)는 이러한 권한을 바꿀 수 있다.

#### ① 파일이나 디렉토리의 소유자

·처음 파일이나 디렉토리를 생성한 User의 소유로 생성.

`kopoctc` 라는 사용자로 접속하여 파일을 하나 만들었다면, 해당 파일은 `kopoctc`가 소유권을 가짐

#### ② 파일, 디렉토리 소유 사용자 권한 변경

·`chown` : 파일 또는 디렉토리의 소유사용자를 바꿈

**Tip**

예) `chown kopoctc aa : aa`파일을 `kopoctc` 라는 사용자의 소유로 바꿈

#### ③ 파일, 디렉토리 소유 그룹 권한 변경

·`chgrp` : 파일 또는 디렉토리의 소유 그룹을 바꿈

**Tip**

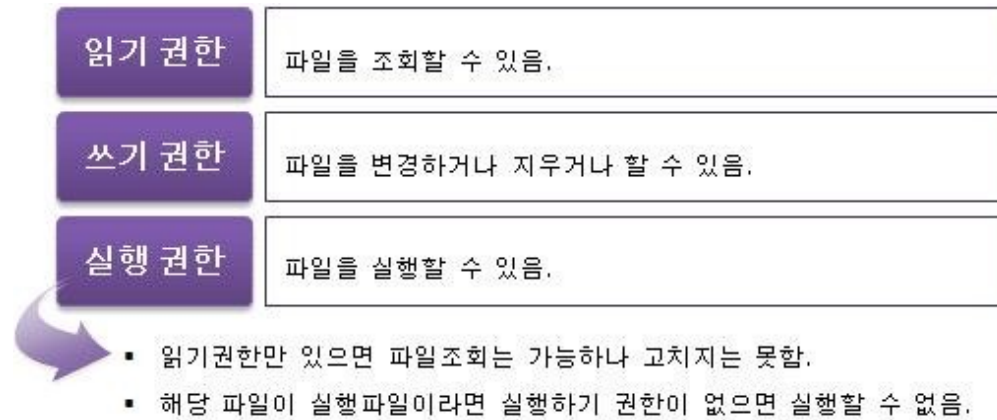
`chgrp kopoctc aa : aa`파일을 `kopoctc`라는 그룹의 소유로 바꿈

## 2) 권한의 이해

파일이나 디렉토리에 대하여 관련 사용자와 그룹의 권한(permission)을 표기하고 관리하는 방법에 대하여 알아본다.

### ① 3단계권한

·파일이나 디렉토리의 권한은 <그림 II-5> 와 같이 읽기, 쓰기, 실행 권한의 3단계로 나누어짐



<그림 II-5> 3단계 권한

·권한 부여는 소유자(user), 그룹(group), 다른 사용자(other)에 대하여 부여함.  
·즉 하나의 파일이나 디렉토리에 대하여 소유자, 같은 그룹의 사용자, 그 외에 사용자에게 대하여 파일 읽기, 쓰기, 실행하기의 권한을 지정

### ② ls -al 로 조회

·ls -al 명령으로 보여지는 내용으로 각각의 파일, 디렉토리의 권한을 알 수 있음

```
kopoetc@kopoetc:~$ ls -al
total 44
drwxr-xr-x 4 kopoetc kopoetc 4096 Jan 18 11:16 .
drwxr-xr-x 4 root    root    4096 Jan 18 11:09 ..
-rw----- 1 kopoetc kopoetc 577 Jan 18 11:11 .bash_history
-rw-r--r-- 1 kopoetc kopoetc 220 Dec 23 11:44 .bash_logout
-rw-r--r-- 1 kopoetc kopoetc 3486 Dec 23 11:44 .bashrc
drwx----- 2 kopoetc kopoetc 4096 Dec 23 11:45 .cache
-rw-r--r-- 1 root    root      7 Jan 18 10:50 help2.txt
-rw-r--r-- 1 root    root      7 Jan 18 10:50 help3.txt
drwxrwxr-x 2 kopoetc kopoetc 4096 Jan 18 11:06 mydir
-rw-r--r-- 1 kopoetc kopoetc 675 Dec 23 11:44 .profile
-rw----- 1 kopoetc kopoetc 3001 Jan 18 11:16 .viminfo
kopoetc@kopoetc:~$
```

<그림 11-6> ls -al 실행

- 파일 또는 디렉토리를 나타내는 해당 라인의 처음 필드 10자리는 다음과 같은 의미를 가짐

- ① 파일종류나 디렉토리 임을 표시(d또는 -)
- ② 사용자의 읽기 권한(r 또는 -)
- ③ 사용자의 쓰기 권한(w 또는 -)
- ④ 사용자의 실행 권한(x 또는 -)
- ⑤ 그룹의 읽기 권한(r 또는 -)
- ⑥ 그룹의 쓰기 권한(w 또는 -)
- ⑦ 그룹의 실행 권한(x 또는 -)
- ⑧ 다른 사람의 읽기 권한(r 또는 -)
- ⑨ 다른 사람의 쓰기 권한(w 또는 -)
- ⑩ 다른 사람의 실행 권한(x 또는 -)

<그림 11-7> 10자리 권한표시

**Ti  
p**

예) 처음 10개의 필드가 drwxr-r-- 로 표기되었다고 가정한다면  
ls에서 조회되는 처음 10개의 문자는 (1, 3, 3, 3)으로 나누어 보면(d rwx r--)

- a) 처음 비트는 파일이면 -, 디렉토리이면 d로 표시
- b) 다음 3자리는 소유자 (user)의 허가권
- c) 다음 3자리는 그룹의 허가권
- d) 다음 3자리는 소유자도 그룹도 아닌 자(other)의 허가권

그러므로 해당내용은 먼저 해당파일은 디렉토리이며, 소유자는 읽고 쓰고 실행하는 것 이 가능하고, 같은 그룹의 사용자는 읽기는 가능하지만 쓰거나 실행할 수 없고, 관련없 는 다른 사용자는 읽기는 가능하지만, 쓰거나 실행할 수 없음을 의미함

### 3) 권한의 숫자 표기법

앞서서 파일이나 디렉토리에 대한 권한(permission)을 10개의 자리로 표기하는 방법을 알아보았다. 권한을 관리하는데 이러한

10개의 표기법과 함께 이를 숫자로 나타내는 방법이 있는데 이에 대하여 알아보도록 한다.

#### ① 권한의 숫자 표기법

- 권한에 대한 부분을 대상자와 3단계 권한을 결합하여 다음과 같이 3자리 숫자로 나타냄
- 3자리 숫자는 순서대로 사용자, 같은그룹사용자, 다른사용자를 의미
- 읽기(4), 쓰기(2), 실행하기(1)로 보고 각각의 권한 부여를 더한 값을 사용<표 II-1>

파일 속성	소유자(User)			그룹(Group)			다른 사람(Other)		
	읽기	쓰기	실행	읽기	쓰기	실행	읽기	쓰기	실행
	r(4)	w(2)	x(1)	r(4)	w(2)	x(1)	r(4)	w(2)	x(1)

<표 II-1> 10자리 권한표시

Tip

예) 사용자가 rx의 권한, 그룹이 wx의 권한, 다른 사람은 아무 권한을 가지지 않는다면, 사용자 r(4) + x(1) = 5, 그룹 w(2) + x(1) = 3, 다른 사람 0으로 보고 그 파일은 530의 권한을 가지고 있다고 표현



#### 4) 권한설정

권한을 설정하는 방식은 상대모드와 절대모드가 있다.

##### ① 상대모드

- 현재 권한을 기준으로 권한을 제거하거나 부여하는 방식
- 상대모드 표현시 아래와 같은 방식으로 표기

Operator	의미	Access_class	의미
+	권한부여	u	사용자
-	권한제거	g	해당 그룹의 멤버들
=	권한유지	o	다른 사람
S	사용자와 그룹만 실행	a	사용자, 그룹, 다른 사람 모두 권한부여

<그림 11-8> 상대모드 권한표시

**Tip** 예) `chmod g-w aaa` : aaa파일에서 그룹의 쓰기권한을 제거

**Tip** 예) `chmod g+rw aaa` : aaa파일에서 그룹의 읽기 쓰기권한 부여

**Tip** 예) `chmod a+x aaa` : aaa파일에서 모두 실행권한을 부여

**Tip** 예) `chmod o-rwx aaa` : aaa파일은 다른 사람은 읽거나, 쓰거나, 실행하지도 못하도록 함.

## ② 절대모드

- 권한의 숫자표기법을 이용하여 권한을 표기
- 기존 부여된 권한은 무시되며 새롭게 지정된 권한으로 재 설정
- 앞에서 설명된 권한 숫자표기 방식대로 사용자 r(4) w(2) x(1), 그룹 r(4) w(2) x(1), 다른사람 r(4) w(2) x(1)으로 표시한값을 각 권한자 별로 더한 값으로 나타냄 <그림 II-xx>

파일 속성	소유자(User)			그룹(Group)			다른 사람(Other)		
	읽기 r(4)	쓰기 w(2)	실행 x(1)	읽기 r(4)	쓰기 w(2)	실행 x(1)	읽기 r(4)	쓰기 w(2)	실행 x(1)

<표 II-2> 10자리 권한표시

Ti  
p

예) 사용자가 읽고, 실행하고, 그룹 멤버는 쓰고 실행할 수 있는데, 다른 사람은 실행할 수 없게 하는 권한은?

• 사용자 r(4) + x(1) = 5 , 그룹 w(2) + x(1) = 3, 다른 사람 0 이므로 권한의 숫자 표현은 530임.

• 이런 권한을 aa파일에 부여한다면 [chmod 530 aa]로 명령

Ti  
p

• 시스템 설정 파일 등은 관리자 이외에는 읽기, 쓰기, 실행하기 등을 제한하여야 함

• 일반 사용자가 시스템 설정을 마음대로 바뀌서는 안 되며, 사용하다 실수로 바뀌어 서도 안 됨

## ● 4. 실습하기(2)

---

### 2) chown, chgrp 실습

- ① kopoctc 사용자로 접속하여 파일을 생성 : kopoctc의 소유권. (각자의 id)
- ② chown s1111111 aa : aa파일을 s1111111 라는 사용자의 소유로 바꿈 (id하나생성)
- ③ chgrp s1111111 aa : aa파일을 s1111111 라는 그룹의 소유로 바꿈
- ④ 앞 실습을 하며 ls -al 명령으로 각 파일의 권한을 조회하고 어떤 권한이 있는지 확인

### 3) 상대모드 실습

- ① chmod g-w aaa
- ② chmod g+rw aaa
- ③ chmod a+x aaa
- ④ chmod o-rwx aaa
- ⑤ 실행 후 ls -al로 권한설정 변경 확인

## ● 4. 실습하기(1)

---

### 4)절대모드 실습

- ① `chmod 744 aaa` (어떤 명령일까요?)
- ② `chmod 553 aaa`
- ③ `chmod a-x aaa`를 절대모드로 변경하면?
- ④ `chmod o-rwx aaa` : 절대모드로 어떻게 변경해야 하나?  
(권한 조회를 하고 사용자와 그룹의 권한은 그대로 주어야 함)
- ⑤ `chmod -R 555 디렉토리명` : 하위 디렉토리에 모든 파일들의 권한을 바꿈

### (3) 링크 파일

시스템을 운영하다 보면 하나의 파일을 여러 디렉토리에 가져다 사용하는 경우가 있다. 윈도우 운영체제에서 예를 들면 아이콘이 이와 비슷한 개념이다. 바탕화면에 있는 윈도우 브라우저 아이콘은 바탕화면의 아이콘을 클릭하여 해당 프로그램이 실행 되지만 해당 파일은 바탕화면 디렉토리가 아닌, 윈도우의 프로그램의 익스플로러 디렉토리의 실행파일이 연결되어 있는 경우이다. 바탕화면의 아이콘을 삭제한다고 해당 파일이 지워지는 것은 아니며, 연결만 지워지는 경우이다. 유닉스와 리눅스에도 이러한 파일이 있는데 링크파일이 그러한 역할을 한다. 이번에는 링크파일에 대하여 하나하나 배워보도록 한다.

#### 1) 하드링크

하드링크(hard link)의 두 파일명은 같은 디스크에 위치한 같은 데이터를 가리키며 다음과 같은 특징이 있다.

##### ① 특징

- 하드링크의 두 파일명은 같은 디스크에 위치한 같은 데이터를 가리킴
- 하드링크는 원본파일과 완전히 동일하고, 부가적인 디스크 공간을 차지하지 않음
- 하드링크 파일은 원본과 동일하기 때문에 하드링크 파일을 지우면 원본도 삭제됨

윈도우운영체제에는 없는 개념

##### ② 하드링크로 연결하기

- 하드링크로 연결 ln 명령
- ln abc abc2: abc라는 파일을 abc2라는 하드링크파일로 연결, abc2 파일은 새로 생성됨
- 해당 파일상태를 보는 명령은 stat
- <그림 II-9>에서 ln abc l\_abc로 하드링크를 실행한 경우, abc와 l\_abc 는 동일한 파일이 연결되어 있음을 알 수 있다

```
kopoctc@kopoctc:~$ stat abc
  File: 'abc'
  Size: 7          Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d Inode: 261898      Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/ kopoctc)   Gid: ( 1000/ kopoctc)
Access: 2021-01-18 11:17:24.532815359 +0900
Modify: 2021-01-18 11:17:24.532815359 +0900
Change: 2021-01-18 11:17:32.516815359 +0900
 Birth: -
kopoctc@kopoctc:~$ stat l_abc
  File: 'l_abc'
  Size: 7          Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d Inode: 261898      Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/ kopoctc)   Gid: ( 1000/ kopoctc)
Access: 2021-01-18 11:17:24.532815359 +0900
Modify: 2021-01-18 11:17:24.532815359 +0900
Change: 2021-01-18 11:17:32.516815359 +0900
 Birth: - kopoctc
@kopoctc:~$
```

<그림 11-9> 하드링크 파일 상태조회

Tip

하드링크로 생성된 파일은 단순링크가 아니라 다른 이름의 동일한 파일이므로 삭제등 작업을 주의할 것

## 2) 심볼릭 링크

심볼릭 링크(symbolic link)는 원 파일을 연결해 주는 파일로 단순 링크된 파일이며 다음과 같은 특징이 있다.

### ① 특징

- 심볼릭 링크는 작은 파일로 존재하고 이 파일은 링크된 파일을 가리킴.
- 윈도우에서 바탕화면의 바로가기 아이콘 개념. 아이콘을 지운다고 해당 파일이 지워지지는 않음.

### ② 심볼릭 링크로 연결하기

- 하드링크로 연결 `ln -s` 명령
- `ln -s abc abc2`: abc라는 파일을 abc2라는 하드링크파일로 연결, abc2 파일은 새로 생성됨
- 해당 파일상태를 보는 명령은 `stat`
- <그림 II-10>에서 `ln -s efg l_efg`로 심볼릭 링크를 실행한 경우
- efg와 l\_efg 는 전혀 다른 파일임을 알 수 있다

```
kopoctc@kopoctc:~$ ln -s efg l_efg
kopoctc@kopoctc:~$ stat efg
  File: 'efg'
  Size: 7          Blocks: 8   De      IO Block: 4096   regular file
vice: fc00h/64512d   Inode: 263076   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ kopoctc)   Gid: ( 1000/ kopoctc)
Access: 2021-01-18 11:18:17.332815359 +0900
Modify: 2021-01-18 11:18:17.332815359 +0900
Change: 2021-01-18 11:18:17.332815359 +0900
Birth: -
kopoctc@kopoctc:~$ stat l_efg
  File: 'l_efg' -> 'efg'
  Size: 3          Blocks: 0   De      IO Block: 4096   symbolic link
vice: fc00h/64512d   Inode: 263078   Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/ kopoctc)   Gid: ( 1000/ kopoctc)
Access: 2021-01-18 11:18:28.132815359 +0900
Modify: 2021-01-18 11:18:28.132815359 +0900
Change: 2021-01-18 11:18:28.132815359 +0900
Birth: - kopoctc
@kopoctc:~$
```

<그림 11-10> 심볼릭 링크 파일 상태조회



## ● 4. 실습하기(3)

---

### 5) Hard Link

- ① `abc` : 파일을 생성
- ② `ln abc l_abc` : `abc`파일과 `l_abc`파일을 하드링크 함
- ③ `ls -al *`로 두 파일을 확인
- ④ `stat abc, stat l_abc`로 두 파일의 디스크 상황을 보고 하드링크를 확인

### 6) Symbolic Link

- ① `efg` : 파일을 생성
- ② `ln -s efg s_efg` : `efg`파일과 `s_efg`파일을 심볼릭 링크함
- ③ `ls -al *`로 두 파일을 확인
- ④ `stat efg , stat s_efg` 로 두 파일의 디스크 상황을 보고 하드링크를 확인

## 1. 시스템 측정 및 모니터링 명령

시스템 서버가 정상적으로 운영되고 있는지 점검을 위하여 수시로 모니터링을 수행하여야 한다. 이를 위하여 시스템을 모니터링 하는 방법에 대하여 하나하나 배워보도록 한다.

### 1) sar

sar(sar(System Activity Reporter) 명령어는 유닉스, 리눅스 등에서 유용하게 쓰는 시스템 모니터링 명령어이다.

#### ① sar 명령어 역할

- 모니터링 대상이 상당히 넓은 편이며 기본값은 CPU 활동에 대한 통계를 출력
- 각 종 활동에 대한 통계를 다른 프로그램(sysstat)을 이용하여 파일로 저장하고 통계치를 리포팅 하는 기능을 제공.

#### ② sar 설치

- sar 명령어가 실행되지 않는 경우 sysstat를 설치하여야 함

```
root@kopoctc:~# sar 1 100
The program 'sar' can be found in the following packages:
 * sysstat
 * atsar
Try: apt-get install <selected package>
root@kopoctc:~# apt install sysstat
```

<그림 IV-25> sysstat 설치

#### ④ sar 보여지는 항목

- %user : user level (application) 이 사용하는 cpu의 양 (%)
- %nice : nice priority(우선순위가 지정된 job)를 가진 user level이 사용하는 cpu의 양 (%)
- %system : system level(kernel)이 사용하는 cpu양 (%)

·%iowait : 디스크등의 i/o의 속도차로 인하여 cpu가 쉬게 되는 시간 비율(%)

·%steal : 프로세스(cpu상 하나의 실행단위)가 cpu등을 바꾸거나, 잡 우선순위에 의하여 다른 잡이 들어오거나 하는 순간 때 문에 사용된 시간 비율(%)

·%idle : cpu가 널널한 비율 (100- (%user+%nice+%system))

#### ④ sar 사용

·sar 1 5 : cpu 사용량을 1초 간격으로 5번 조사

```
root@kopoctc:~# sar 1 5
Linux 3.13.0-74-generic (kopoctc)      01/13/2025      _i686_ (1 CPU)

04:57:35 PM   CPU   %user   %nice   %system   %iowait   %steal   %idle
04:57:36 PM   all    0.00    0.00    0.00    0.00    0.00   100.00
04:57:37 PM   all    0.00    0.00    0.00    0.00    0.00   100.00
04:57:38 PM   all    0.00    0.00    1.00    0.00    0.00    99.00
04:57:39 PM   all    0.00    0.00    0.00    0.00    0.00   100.00
04:57:40 PM   all    0.00    0.00    0.00    0.00    0.00   100.00
Average:     all    0.00    0.00    0.20    0.00    0.00    99.80
root@kopoctc:~#
```

<그림 IV-26> sar 1 5

·sar -d 1 1 : disk io정보를 1초 간격으로 1번 조사

```

root@kopoctc:~# sar -d 1 1
Linux 3.13.0-74-generic (kopoctc)      01/13/2022      _i686_ (1 CPU)

  05:02:38 PM          DEV      tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz   await
svctm    %util
  05:02:39 PM      dev8-0      0.00      0.00      0.00      0.00      0.00      0.00
0.00      0.00
  05:02:39 PM      dev252-0      0.00      0.00      0.00      0.00      0.00      0.00
0.00      0.00
  05:02:39 PM      dev252-1      0.00      0.00      0.00      0.00      0.00      0.00
0.00      0.00

Average:  svc      DEV      tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz   await
tm    %util
Average      dev8-0      0.00      0.00      0.00      0.00      0.00      0.00
: 0.00 0.00
Average      dev252-      0.00      0.00      0.00      0.00      0.00      0.00
: 0.00 0.00      0
Average      dev252-      0.00      0.00      0.00      0.00      0.00      0.00
: 0.00 0.00      1
root@kopoctc:~#

```

<그림 IV-27> sar -d 1 5

## 2) top

top 명령어는 유닉스, 리눅스 등에서 유용하게 쓰는 시스템 모니터링 명령어로 다시 복습해 본다.

- ① top 명령 : 현재 cpu에 수행되고 있는 프로세스상황을 cpu자원을 많이 사용하는 순서대로 자동으로 계속 보여줌
- ② 메모리상황 , swap상황, 실행 우선순위, 버추얼 메모리, 리소스, 등을 보여줌
- ③ 가장 중요한 것은 처리시간을 얼마나 사용하고 있는가를 모니터링 하는 것임
- ④ 만일 계속하여 처리시간을 잡아먹고 있는 프로세스가 있다면, 그 프로세스는 hang인 상태인 경우(무한루프)일 수 있으며 이 프로세스는 kill -9 job#로 처리해주는 경우가 있을 수 있음

```
top - 17:08:57 up 26 min, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 80 total, 1 running, 79 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 766352 total, 354120 used, 412232 free, 32000 buffers
KiB Swap: 782332 total, 0 used, 782332 free. 193332 cached Mem
```

```
260 root      20   0   0   0   0 S  0.3  0.0  0:01.36 kworker/0:2
1162 tomcat7   20   0 281920 63960 9648 S  0.3  8.3  0:04.62 java
1386 kopoctc   20   0 11196 1696  948 S  0.3  0.2  0:00.16 sshd
  1 root      20   0  4324  2328 1392 S  0.0  0.3  0:00.88 init
  2 root      20   0   0   0   0 S  0.0  0.0  0:00.00 kthreadd
  3 root      20   0   0   0   0 S  0.0  0.0  0:00.01 ksoftirqd/0
  4 root      20   0   0   0   0 S  0.0  0.0  0:00.00 kworker/0:0
  5 root       0 -20   0   0   0 S  0.0  0.0  0:00.00 kworker/0:0H
  6 root      20   0   0   0   0 S  0.0  0.0  0:00.16 kworker/u2:0
  7 root      20   0   0   0   0 S  0.0  0.0  0:00.16 rcu_sched
  8 root      20   0   0   0   0 S  0.0  0.0  0:00.00 rcu_bh
  9 root      rt    0   0   0   0 S  0.0  0.0  0:00.00 migration/0
10 root      rt    0   0   0   0 S  0.0  0.0  0:00.01 watchdog/0
11 root       0 -20   0   0   0 S  0.0  0.0  0:00.00 khelper
12 root      20   0   0   0   0 S  0.0  0.0  0:00.00 kdevtmpfs
13 root       0 -20   0   0   0 S  0.0  0.0  0:00.00 netns
14 root       0 -20   0   0   0 S  0.0  0.0  0:00.00 writeback
```

<그림 IV-28> top 명령어

### 3) fsck, du

fsck, du 명령어는 유닉스, 리눅스 등에서 유용하게 쓰는 파일 관련 모니터링 명령어이다.

#### ① fsck

- fsck : file system check
- 파일시스템의 에러나 크래시(파일 시스템의 부서짐)를 복구.
- 각 파일시스템을 **umount**하고 단일사용자 모드에서 하는 것이 효과적( 장애 때 )
- 디스크 작업 시 본 명령 실행 시 디스크 크래시 주의 (디스크가 사용되고 있는 상황에서 본 명령이 실행되는 경우 파일 시스템의 복구 불능 장애 발생)

#### ② du

- du(disk usage): 파일 스페이스의 사용량을 보여줌
- du -a 현재 디렉토리 하위 모든 파일의 스페이스 사용량을 보여줌. ls에서 보이는 실크기와 디스크에 할당된 크기는 차이가 있음
- 파일 시스템이 1024단위로 할당되지 않은 경우 du -k로 실행

```
root@kopoctc:/home# du
4      ./kopoctc/.cache
32     ./kopoctc
48     .
root@kopoctc:/home# du -k
4      ./kopoctc/.cache
32     ./kopoctc
48     .
root@kopoctc:/home#
```

<그림 IV-29> du 명령어

#### 4) 명령어 총정리

아래 명령어는 시스템에 따라 옵션이 조금씩 차이가 있을 수 있다.

구분	명령어	설명
프로세스확인	jobs	현재 id,prompt에서 수행되는 job확인
	ps	각종 프로세스 상황확인
	pstree	트리구조로 프로세스 상황 확인
	top	프로세스가 cpu를 차지하는 순서로 확인
	ulimit	ulimit -a 로 프로세스 자원한도 확인
	/proc디렉토리	현재 수행중인 프로세스는 /proc디렉토리에 정보파일로 존재함( 대부분 리눅스, 유닉스는 아님)
CPU상황확인	vmstat	Cpu활동상황 확인
	sar	순간적 cpu활당률, 디스크(sar -d)i/o상 황을 보여줌
	vi /proc/cpuinfo	현재 설치된 cpu의 자세한 상황을 보여줌
Memory 확인	top	메모리 상황도 top에서 보임
	free	메모리 상황보기



Disk 확인	df -k	파일 시스템 별 마운트 상황, 배분 크기가 보 임
	du -a	파일별 스페이스 사용량 보임
	iostat	현재 io발생상황을 보여줌
네트워크 관련	ifconfig	랜카드등 인터페이스 상황을 보여줌
	ping	네트워크연결설정을 보는 기본 명령
	netstat-na	Tcp/ip어플리케이션차원의 서비스 상태를 보 여줌 <표 IV-5> 주요 모니터링 명령어

### Tip

아래 명령어 들은 시스템 모니터링을 위하여 엄선된 명령어이다.  
 옵션을 잘 활용함에 따라 모니터링의 활용도가 높아진다.  
 반드시 man을 통하여 명령어를 찾아보자

## ● 4. 실습하기(3)

---

### 4) sar명령 실습

- ① sysstat 설치
- ② sar 1 100 실행후 보여지는 항목 조사
- ③ sar -d 1 100실행으로 디스크 io정보 조사

### 5) top명령 실습

- ① top명령 실행후 보여지는 항목 조사

### 6) fsck, du 실습

- ① fsck실행
- ② du -k 실행
- ③ du -a 실행

## ● 4. 실습하기(4)

---

### 7) 프로세스확인 실습

- ① jobs 실습
- ② ps 실습
- ③ pstree 실습
- ④ top 실습
- ⑤ ulimit 실습
- ⑥ /proc 디렉토리 확인

### 8) CPU 상황확인 실습

- ① vmstat 실습
- ② sar 실습
- ③ vi /proc/cpuinfo 확인

### 9) 메모리 확인 실습

- ① top 실습
- ② free 실습

## ● 4. 실습하기(5)

---

### 10) 디스크 확인 실습

- ① `df -k` 실습
- ② `du -a` 실습
- ③ `iostat` 실습

### 11) 네트워크 관련 확인 실습

- ① `ifconfig` 실습
- ② `ping` 실습
- ③ `netstat -na` 실습

## 1. 네트워크 관리

기업의 정보시스템을 위한 리눅스 서버를 운영하는 경우, 시스템 관리를 위하여 네트워크 상황을 상시 모니터링하고 관리하는 업무는 중요한 사항의 하나이다. 리눅스서버가 웹서버로 이용되거나 리눅스 운영체제로 일반 사용자 PC나 스마트폰의 안드로이드 운영체제로 이용되는 경우를 상상하더라도, 네트워크의 역할은 비중이 큰 부분이다. 이번 강의에서는 이러한 네트워크 분야에 대하여 기본적 이론, 환경설정 및 네트워크 관리방안에 대하여 이해 후 실습하도록 한다.

### Tip

학습 전 인터넷 검색을 통하여 “TCP/IP”라는 용어에 대하여 미리 조사하여 학습 준비도를 높이도록 한다.

### (1) 네트워크 이론

본 과목 처음 설치부분에 서버로 접속하기 위하여 네트워크 설정 부분이 있었으나 무따기(무조건 따라하기)로 진행했던 부분이 있었다. 여러분이 그동안 본 과목을 진행함에 있어서, 해당 내용들을 이해할 만한 내공이 충분한 상황이라고 생각된다. 이번 강의에는 네트워크에 대한 기초 이론에 대하여 하나하나 배워보도록 한다.

#### 1) TCP/IP 기초

이제는 수많은 서버, 수많은 PC 및 기타 수많은 장비들이 서로 통신을 위하여 네트워크로 연결되어 다양한 서비스 지원하는 것이 일반적이다.

##### ① Internetworking

- 인터넷 등 네트워크를 이용하여 통신을 하는데 그 세부 사항들이 어떻게 이루어지는지 고민 해 봄
- TCP/IP 프로토콜 사용

##### ② Internetworking Protocol

- TCP/IP (transfer control protocol/Internet protocol)
- 네트워크 전송 프로토콜
- 서로 다른 운영체제를 쓰는 컴퓨터 간에도 데이터를 전송할 수 있어 인터넷에서 정보전송을 위한 표준 프로토콜로 쓰임.
- TCP : 전송 데이터를 일정 단위로 나누고 포장하는 것에 관한 규약
- IP: 직접 데이터를 주고 받는 것에 관한 규약

#### Tip

프로토콜(Protocol): 한 컴퓨터에서 다른 컴퓨터로 데이터를 완전하게 보내기 위한 전송 규정이나 장치 등을 총칭

#### Tip

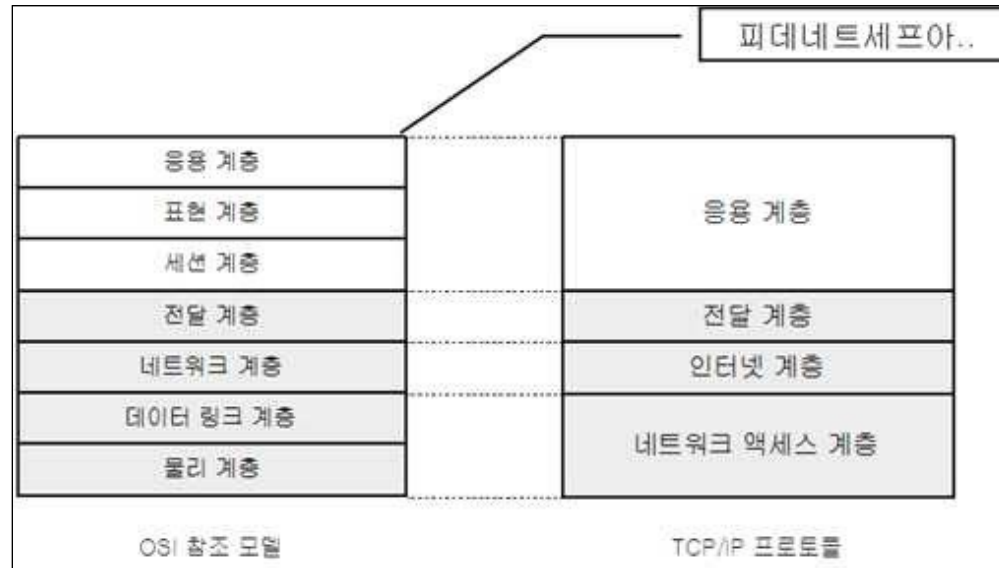
TCP/IP 전송 이해

- IP는 출발과 도착 주소를 가지는 헤더를 참조해 네트워크를 통해 데이터를 목적지에 전달
- TCP는 전달 과정에서 데이터가 잘못 전달되거나 파손될 수도 있고, 보내진 순서대로 데이터를 전송하지 못하는 경우를 방지.
- IP가 데이터 패킷을 분실하거나 파손하면 TCP는 문제가 발생했다는 것을 알리고 데이터를 알맞게 재전송.
- TCP가 에러를 복구하는 과정에서 데이터를 수신하는 호스트에서는 언제나 에러 없이 순서대로 데이터를 전송 받음.

### ③OSI 7 Layer

- 국제표준화기구(ISO)가 1977년에 정의한 국제 통신 표준 규약.
- 통신의 접속에서부터 완료까지의 과정을 7단계로 구분 정의한 통신 규약
- 현재 모든 통신 규약의 지침으로 따름

·OSI7Layer 의 규약에 따라 대응되는 TCP/IP계층은 <그림 IV-xx>과 같음



<그림 IV-1> id, groups 명령어

#### ④계층의 이해

- 서로 간 통신을 위하여 여러 계층별로 많은 구성요소 및 작용들이 일어남
- TCP/IP 체계의 흐름을 아래의 계층으로 분류하여 이해하면 쉬움

계층	이해
응용계층	PC의 인터넷 브라우저를 실행, telnet,ftp등을 실행함, 통신을 이용한 프로그램을 제작, 이때 응용계층을 집중하여 이해하는 것

	만으로도 본연의 목적수행이 가능
전달계층	Windows OS의 TCP커널, 소켓 라이브러리 등이 통신프로그램과 실제 네트워크 부분과 연결전달 역할을 함
인터넷계층	인터넷IP체계, 라우팅 동작 원리 등은 시스템 하드웨어를 보다 논리적으로 이해하여 사용할 수 있도록 해줌
네트워크 액세스 계층	UTP케이블 스펙, 랜카드, 라우터, 허브 등 통신을 위한 물리적, 전기적 장비 및 이론 필요

<그림 IV-2> TCP/IP 계층의 이해



## 2) IP 주소 체계

인터넷 통신을 수행하기 위하여 각 장비들은 서로 간 통신을 위하여 출발지와 목적지의 위치를 알 수 있는 방법이 필요한데 TCP/IP체계에서는 IP주소를 사용한다.

### ① 물리적 주소

- 하나의 물리적 하드웨어를 나타내는 주소
- 처음 하드웨어 제조사에서 하드웨어기기에 고정하여 생산
- 예 : Mac Address, device ID, IRQ

### ② 논리적 주소

- 통신을 위한 상대는 반드시 물리적인 장치가 아니라, 가상의 소프트웨어적인 장치일 수 있음.
- 즉 하나의 PC에 대하여 여러 개의 네트워크 카드(랜카드등)나 버추얼 머신을 이용한 다수의 통신개체가 존재할 수 있음
- 즉 물리적 기기와 IP주소가 1대1로 대응되는 것이 아니고 논리적 단위로 대응됨
- 예 : IP Address , Url (DNS...)

### ③ 내부(서버) 프로토콜 주소

- 하나의 논리적 장치에 대하여 내부 적용할 서비스를 지정
- 예 : 프로토콜체계(TCP,UDP), 포트 번호



<그림 IV-3> 물리적주소와 논리적주소

#### ④ IP주소 체계

- 네트워크 식별자와 호스트 식별자로 구성
- 네트워크 식별자 : 네트워크 구분, 다른 단말등과 이 부분이 같으면 동일한 네트워크(허브 등으로 단순 연결), 아니면 다른 네트워크(라우팅을 통한 다른 네트워크로 접근)
- 호스트 식별자 : 각각의 단말, 서버, 장비, 랜 카드 별로 유일한 한 개의 IP부여
- IP주소 체계는 2진수를 10진 표기로 사용 <그림 IV-4>

호스트 이름 : [www.naver.com](http://www.naver.com)

IP 주소 : 11010010 01110011 10101010 01100101

210          115          170          101



210.115.170.101

<그림 IV-4> IP주소의 10진수 표기

•네트워크 식별자와 호스트 식별자를 나타내는 비트 크기에 따라 IP주소의 클래스를 구분



<그림 IV-5> 네트워크 식별자와 호스트 식별자

·IP주소의 클래스는 A,B,C클래스로 구분됨

클래스	주소 범위	네트워크(*)	단말(0)	넷마스크
A	1.0.0.0 – 127.255.255.255	*.0.0.0	*.0.0.0	255.0.0.0
B	128.1.0.0 – 191.254.255.255	*.*.0.0	*.*.0.0	255.255.0.0
C	192.0.1.0 – 223.255.254.255	*.*.*.0	*.*.*.0	255.255.255.0

<표 IV-6> IP주소 클래스

## DHCP

```
network:
  version: 2
  renderer: networkd
  ethernets:
    wlp5s0:
      dhcp4: yes
```

## 경로

cd /etc/netplan

ls 이용하여 파일 확인

## static

```
network:
  ethernets:
    eno1:
      dhcp4: no
    eno2:
      dhcp4: no
    enp7s0:
      dhcp4: no
      dhcp6: no
      addresses: [10.1.1.253/24]
      gateway4: 10.1.1.1
      nameservers:
        addresses: [168.126.63.1, 8.8.8.8]
  version: 2
```

## ● 4. 실습하기(1)

---

### 1) 리눅스 서버에서 네트워크 설정 실습

- ① Static방식 IP주소 설정
- ② DHCP방식 IP주소 설정

### (3) 네트워크 관리

이제 네트워크의 대한 기본적 사항에 대하여 이해할 수 있게 되었다. 이제부터 실제 리눅스 서버에서의 네트워크 환경을 설정하는 방법에 대하여 하나하나 배워보도록 한다.

#### 1) ifconfig

네트워크 인터페이스의 상황을 알거나, 설정을 하거나, 재가동을 하기 위한 명령어

##### ①ifconfig -a

- 모든 인터페이스의 상황을 볼 수 있음
- 보통 eth0, eth1, eth2 ...로 NIC(일명 랜카드)의 상태가 보임.
- 다른 명칭으로 랜카드를 볼수 있음

##### ②ifconfig eth0

- 하나의 네트워크 인터페이스 카드 (NIC, 일명 랜카드)의 상태를 보요줌
- IP, MAC을 확인
- Network 설정을 확인
- 네트워크 카드상의 패킷 전송상황(에러, 성공,실패) 을 확인 :TX,RX

##### ③ifconfig 명령으로 네트워크 설정도 할 수 있음

- ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
- route add -net 192.168.1.0 network 255.255.255.0 eth0
- route add default gw 192.168.1.1 eth0

##### ④ifconfig 명령으로 보이는 내용

- inet addr : IP주소
- Bcast : 브로드 캐스트 주소.
- Mask : 넷마스크(Netmask)값
- UP : 인터페이스가 활성화되어 있음을 나타냄
- BROADCAST : 브로드 캐스트를 사용함
- RUNNING : 동작중임을 나타냄
- MULTICAST : 멀티 캐스트 사용
- MTU : Maximum Transmission Unit, 한 번에 전송할 수 있는 최대패킷의 크기
- Metric : 라우팅할 때 참조되는 거리로 로컬인 경우 값이 1임
- RX/TX : 받은 패킷/전송한 패킷의 총 개수(packets)
- errors : 에러가 발생한 패킷의 수
- dropped : 버려진 패킷의 수
- overruns : 손실된 패킷의 수
- collisions : 충돌이 발생한 패킷의 수



```
kopoctc@ubuntu:~$ ifconfig eth0
eth0      Link encap:Ethernet HWaddr 08:00:27:bc:46:b1
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:febc:46b1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0  T
          X packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5469 (5.4 KB) TX bytes:4715 (4.7KB)

kopoctc@ubuntu:~$
```

<그림 IV-10> ifconfig 명령 수행

#### Tip

ifconfig 명령은 많이 사용되는 명령이므로 man ifconfig를 통하여 다양한 기능을 알아 둘 것

## 2) ping

대상 장비가 네트워크상에서 응답하는지 테스트 하는 명령

① 대상컴퓨터 ICMP[Internet Control Message Protocol] 에코 패킷을 보낸 후 에코 응답 패킷 을 수신하여 대상 컴퓨터와의 연결을 확인 가능하며 송신한 패킷의 수를 되돌려줌

② 수많은 사람이 반복적으로 ping을 조회하는 것도 대상 컴퓨터에게는 치명적인 공격이 되기 때문에 ping응답을 막아두기도 함.

③ 그러므로 ping응답이 없다고 반드시 네트워크가 절단 난 것은 아님

```
kopoctc@ubuntu:~$ ping -a www.kopo.ac.kr
PING www.kopo.ac.kr (112.76.2.31) 56(84) bytes of data.
```

<그림 IV-11> ping 명령 수행

④ ping 주요 사용예  
제

명령어 예시	설명
ping -t 172.17.245.1	ctrl+c로 중단할 때 까지 네트워크 연결 테스트를 계속함
ping -a <a href="http://www.kopo.ac.kr">www.kopo.ac.kr</a>	IP주소를 알아보기 위하여 많이 사용(DNS의 반응 테스트 등)
ping -n 8 172.17.245.1	ping 명령어를 8회 수행
ping -l 64000 172.17.245.1	PING테스트를 위한 테스트 패킷의 크기를 지정 하여 실행 (64K까지 가능)
ping -r 10 <a href="http://www.naver.com">www.naver.com</a>	지정한 숫자만큼 라우팅 경로를 보여줌

```
ping -s 10 www.naver.com
```

지정한 숫자만큼 라우팅 홉(hop)수의 타임 스템  
프를 인쇄함

<표 IV-2> ping 명령 예시

**Tip**

routing hop count(라우팅 홉 수) 또는 routing metrix count(라우팅 매트릭스 수) 는 네 트워  
크 구간 통신시 거치는 라우팅 개수 즉 거쳐 가는 다른 네트워크 수를 의미한다

**Tip**

ping 명령은 많이 사용되는 명령이므로 man ping를 통하여 다양한 기능을 알아둘 것

### 3) netstat

현재의 컴퓨터와 연결되었거나 연결될 목록을 프로토콜과 함께 보여주는 명령어

#### ① netstat 내용

- 각 리눅스 유닉스 버전에 따라 보이는 필드가 약간 다르지만 의미는 비슷함

#### Tip

netstat 명령의 내용을 좀더 깊게 알기 함 위해서는 socket programming을 공부하여야  
본 과목에서는 여러분의 검색에 맡김

#### ② netstat 주요 사용

예제

명령어 예시	설명
netstat	현재 서버와 연결된 목록만 보여줌
netstat -a	현재 서버와 연결되거나 연결될 목록을 프로토콜과 함께 모두 보여줌. -a 옵션을 주면 연결되었거나 연결을 기다리는 목록을 모두 보여줌
netstat -n	통상 연결 목록이 서버이름이 나오는데, -n 옵션을 주면 서버이름대신 ip주소가 보임
netstat -an or -na	서버와 연결되었던 혹은 연결을 기다리는 목록을 IP주소로 바꾸어서 보임.

<표 IV-3> netstat 명령 예시

```

Active Internet connections (servers and established) Proto
Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 192.168.56.101:22       192.168.56.1:62004      ESTABLISHED
tcp6       0      0 :::22                  :::*                     LISTEN
tcp6       0      0 127.0.0.1:8005          :::*                     LISTEN
tcp6       0      0 :::8080                 :::*                     LISTEN
tcp6       0      0 :::80                   :::*                     LISTEN
udp        0      0 0.0.0.0:36828           0.0.0.0:*               *
udp        0      0 0.0.0.0:68              0.0.0.0:*               *
udp6       0      0 :::59756                :::*                     *

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags               Type               State         I-Nod   Path
unix   5      [ ]                 DGRAM             89          /dev/log
unix   2      [ ]                 STREAM            LISTENING      69          /var/run/dbus/system_bus_socket
unix   2      [ ACC ]              SEQPACKET         LISTENING      7761        /run/udev/control
unix   2      [ ACC ]              STREAM            LISTENING      7440        @/com/ubuntu/upstart
unix   2      [ ACC ]              STREAM            LISTENING      9435        /var/run/acpid.socket
unix   2      [ ACC ]              STREAM            LISTENING      9957
/var/run/mysqld/mysqld.sock
unix   3      [ ]                 STREAM            CONNECTED      894
/var/run/dbus/system_bus_socket
unix   3      [ ]                 STREAM            CONNECTED      8851        @/com/ubuntu/upstart
unix   3      [ ]                 STREAM            CONNECTED      9037        @/com/ubuntu/upstart
--More--

```

<그림 IV-12> netstat 명령 수행

**Tip**

netstat 명령은 많이 사용되는 명령이므로 man netstat를 통하여 다양한 기능을 알아둘 것

## ● 4. 실습하기(2)

---

### 3) ifconfig 명령어 실습

- ① ifconfig -a
- ② ifconfig eth0
- ③ ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
- ④ route add -net 192.168.1.0 network 255.255.255.0 eth0
- ⑤ route add default gw 192.168.1.1 eth0

### 4) ping 명령어 실습

- ① ping -t 172.17.245.1
- ② ping -a www.kopo.ac.kr
- ③ ping -n 8 172.17.245.1
- ④ ping -l 64000 172.17.245.1
- ⑤ ping -r 10 www.naver.com
- ⑥ ping -s 10 www.naver.com

## ● 4. 실습하기(3)

---

### 5) netstat 명령어 실습

- ① netstat
- ② netstat -a
- ③ netstat -n
- ④ netstat -an or -na