

# 용어정리

## 1. Link

사전정의미로는 연결하는 것, 사물의 고리 정도 찾아지는데 이 의미처럼 리눅스나 모놀리식, 여타 운영체제에서 링크와 연결성을 위한 원본을 비로 부르는 다른 그것과 연결해주는 방법과 같음. 물론 결과가 복잡하여 접근의 편의성을 가져다 주기 위해 쓰는 것을 우리는 '링크'라고 부른다.

## 2. 권한관리자

- 1) 권한 : 리눅스에는 파일이나 디렉토리 등에 접근(read 읽기), 수정(write 쓰기), 실행(exe) 할 수 있는 말종의 2가지 권한으로 두어 각 사용자와 그룹별로 그 권한을 달리하여 관리할 수 있도록 해두었다.
- 2) 관리자 : 위 권한은 최고사용자(super user)인 root 계정은 모든 파일에 대한 권한이 가능하며, 일반 파일을 권한으로 받은 사람이 만든 파일이나 디렉토리에 한해 제한은 논리 혹은 권한 양도가 가능하다.

# 정리하기

## 1-1) 리눅스에는 여러 사용자가 함께 사용하는 시스템으로 사용자들은 모든 그룹도 존재한다.

그리고 각 사용자에게는 읽고 쓰기 실행(rwx) 하는 권한을 가지며 그 권한을 다른 사용자에게 넘겨거나 관리할 수 있다.

## 1-2) addgroup groupname : 그룹추가 (=groupadd)

addgroup groupname username : 그룹에 사용자 추가  
groups : 현재 사용자가 속해있는 그룹 조회  
cat /etc/group : 현재 etc 디렉토리에 존재하는 모든 그룹조회  
groupadd -n : 그룹의 이름변경  
" -g : 그룹의 gid 변경  
delgroup : 그룹 삭제

## 1-3) /etc/group : 그룹의 정보조회

/etc/passwd : 사용자 정보파일 user.uid.gid.homedir 등이 나열되며 pw는 x로 표시됨  
/etc/shadow : 사용자의 pw를 저장하는 파일 암호화되어 저장되어있다.

## 2-1) 리눅스 시스템에서 권한은 어떤 파일이나 디렉토리 등에 접근하는 수를 실행할 수 있는 자격이나 권한이다.

권한은 user group other 으로 나뉘며 rwx rwx rwx (read write exe)와 같이 나타내거나 2진수의 합으로 777 664 와 같이 나타내기도 한다.

## 2-2)

권한은 user group other  
rwx rwx rwx 각각 주어져 이진수로 111 111 111 과 같이 나타낼 수 있다.  
수자로 표현한 시에는 합으로 나타낸다.  
ex) rw-rw-r-- : 664 / rwx rw- -- : 760  
주석(주석) 주석 주석 주석 주석

## 2-3)

권한 설정은 chmod 라는 명령어로 변경이 가능하다.

설정방법 : chmod a-x file(dir)name 해당 파일 혹은 디렉토리에 대한 실행권한을 모두 제거해버림  
(현재 기준으로) o+rwx '1' 해당 파일 디렉토리에 대한 다른사람의 읽는 권한을 부여함  
권한 제거(부여) g-w '1' 그룹원에게서 쓰기권한을 빼앗음  
u-x '1' user에게서 실행권한을 빼앗음

정확한 방법 : chmod 664 file(dir)name 해당 file(dir)에 대한 권한을  
수치표기법 사용 rw-rw-r--로 설정함  
(현재 권한과 능력없이 재설정)

## 3-1) Hard link : ln filename linkfilename 의 명령어를 사용하여 link 걸 file이 있으면 명령어 실행시 이중 생성된다. 리눅스에는 많은 개념으로 원본파일과 Hard link 된 파일은 명만 다른 동일한 파일로 구분되는 2가지의 개념인 Inode 또한 조화하면 똑같이 나뉜다. 권한또한 원본과 동일하다.

Symbolic link : ln -s filename linkfilename 의 명령어를 사용하여 link 걸 file이 있으면 명령어 실행시 이중 생성된다. 리눅스에는 많은 개념으로 원본파일과 Hard link 된 파일은 명만 다른 동일한 파일로 구분되는 2가지의 개념인 Inode 또한 조화하면 똑같이 나뉜다. 권한또한 원본과 동일하다.

Hard link와 달리 링크 파일은 원본파일과 별개의 파일로 '링크를 수행하지않는' 파일이라 파일 자체도 굉장히 작은 원본파일이 사라져도 실행할 수 있는 Hard link와 달리 원본파일이 없으면 실행시 파일을 찾을 수 없다는 에러 메시지가 나올 수 있다.

### ● 3. 이해하기

1) 그룹관리 명령어

파일을 탐색하기 위한 명령으로 pwd, cd, ls가 있다. <그림 1-35>

### ①그룹조화

- 자기가속한 그룹 : groups, id
- 서버 내 전체 정의되어있는 그룹을 보려면 /etc/group 파일을 봄 <그림 11-3>

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

<그림 II-3> group 정의 파일

②그룹을 생성하기위하여 groupadd를 사용

Tip

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

### ● 3. 이해하기

### 3) 패스워드, 그룹관련 설정 파일

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

### ①사용자정보 파일

·/etc/passwd파일 - 실제 pw /etc/shadow

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

②그룹 정보 파일

·/etc/group파일

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

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

### ③패스워드관련 파일

·/etc/shadow 파일: /etc/passwd파일과 함께 사용자패스워드를 저장

· 단 패스워드는 암호화되어있는 문장으로 패스워드는 함부로 바꿀 수 없으며 파일에서 패스워드 필드를 고치면 시스템 오류가 발생

- passwd 파일 내부 형식

```
username:password:uid:gid:gecos:homed
```

- username : 사용자명

·password : 사용자암호

- uid: 사용자아이디, 그룹아이디

### ● 3. 이해하기

- gid : 사용자아이디, 그룹아이디
- gecos : General Electric Comprehensive Operating System (예전 Unix서비스와 호환성을 갖추기 위하여 만든 필드, 처음 사용자정보 넣은 값들이 저장)
- homedir : 해당 사용자의기본 디렉토리
- shell : 해당 사용자가사용하는Unix 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 로그인 X.
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 passwd파일 변경확인(패스워드 변경전후 캡처)
- ④ 해당 id로 로그인 한 후 해당 id가 속하는 그룹과 홈 디렉토리 등을 확인 (두개이상ID로 로그인후 기본디렉토리 캡처)
- ⑤ 그룹조회, 그룹생성, 그룹변경, 그룹삭제 과정 실습.
- ⑥ 각 절차 별로 /etc/group파일을 조회하여 변경유무를 확인 ( ) 그룹은 수도 확인 가능
- ⑦ 사용자(user)가 속하는 그룹을 변경, 추가함에 따라 권한이 변경되는지 확인.
- ⑧ /etc/shadow파일 조회 캡처

## ● 3. 이해하기

### (2) 권한의 이해 및 표기

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

#### 1) 파일, 디렉토리의 소유자

파일이나 디렉토리는 소유자가 해당 사용권한을 가지고 있으며, 해당 파일과 디렉토리의 사용자 및 관리자( root )는 이러한 권한 한을 바꿀 수 있다.

##### ① 파일이나 디렉토리의 소유자

· 처음 파일이나 디렉토리를 생성한 User의 소유로 생성.

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

##### ② 파일 디렉토리 소유 사용자 권한 변경

chown <sup>소유자</sup> <sup>그룹</sup> kopoctc: kopoctc aa

· chown : 파일 또는 디렉토리의 소유 사용자를 바꿈

↳ ① 사용자 변경 ② 파일 명. 디렉 명

file directory  
소유자 그룹 권한  
rwx rwx rwx  
소유자 그룹 other

#### Tip

예) chown kopoctc aa : aa파일을 kopoctc 라는 사용자의 소유로 바꿈

##### ③ 파일 디렉토리 소유 그룹 권한 변경

· chgrp : 파일 또는 디렉토리의 소유 그룹을 바꿈

↳ ① 그룹명 ② 파일 명. 디렉 명

#### Tip

chgrp kopoctc aa : aa파일을 kopoctc라는 그룹의 소유로 바꿈

## ● 3. 이해하기

```
kopoctc@kopoctc:~$ ls -al
total 44
drwxr-xr-x 4 kopoctc kopoctc 4096 Jan 18 11:16 .
drwxr-xr-x 4 root root 4096 Jan 18 11:09 ..
-rw-r----- 1 kopoctc kopoctc 577 Jan 18 11:11 .bash_history
-rw-r--r-- 1 kopoctc kopoctc 220 Dec 23 11:44 .bash_logout
-rw-r--r-- 1 kopoctc kopoctc 3486 Dec 23 11:44 .bashrc
drwx----- 2 kopoctc kopoctc 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 kopoctc kopoctc 4096 Jan 18 11:06 mydir
-rw-r--r-- 1 kopoctc kopoctc 675 Dec 23 11:44 .profile
-rw-r----- 1 kopoctc kopoctc 3001 Jan 18 11:16 .viminfo kopoctc@kopoctc:~$
```

<그림 11-6> ls -al 실행

· 파일 또는 디렉토리를 나타내는 해당 라인의 처음 필드 10자리 는 다음과 같은 의미를 가짐

## 4. 실습하기(2)

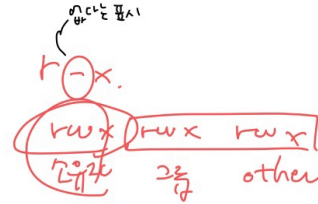
### 3. 이해하기

#### 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



## 3. 이해하기

### (3) 링크 파일

시스템을 운영하다 보면 하나의 파일을 여러 디렉토리에 가져다 사용하는 경우가 있다. 윈도우 운영체제에서 예를 들면 아이 콘이 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 경우가 다. 바탕화면의 아이콘을 삭제한다고 해당 파일이 지워지는 것은 아니며, 연결만 지워지는 경우이다. 유닉스와 리눅스에도 이와 한 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 11-9>에서 ln abc l\_abc로 하드링크를 실행한 경우, abc와 l\_abc는 동일한 파일이 연결되어 있음을 알 수 있다

## 3. 이해하기

### 2) 심볼릭 링크는 바로가기

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 지워지는 않음.

##### ② 심볼릭 링크로 연결하기

- ~~하드링크~~로 연결 ln -s 명령
- ln -s abc abc2: abc라는 파일을 abc2라는 ~~하드링크~~ 파일로 연결, abc2 파일은 새로 생성됨
- 해당 파일상태를 보는 명령은 stat
- <그림 11-10>에서 ln -s cfig l\_cfig로 심볼릭 링크를 실행한 경우
- cfig와 l\_cfig는 전혀 다른 파일임을 알 수 있다