

## 11. 리눅스 시스템 명령체계

리눅스 시스템은 쉘 프롬프트를 통하여 명령을 입력하고 실행하게 된다. 예를 들어, 현재 디렉토리 내의 파일 및 디렉토리의 목록을 출력하는 명령은 "ls"이다. 즉 우리는 프롬프트상에 "ls"를 입력하고 enter를 입력함으로써 파일 및 디렉토리의 목록을 출력하게 되는 것이다. 하지만 이 명령을 직접적으로 실행하는 것은 사용자가 아니다. 사용자가 입력한 명령은 쉘 스크립트에 의해 실행되어지는 것이다.

```
kihee@kihee-VirtualBox:/$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
boot dev  home lib32 libx32 media  opt  root  sbin  srv  tmp  var
```

또한 모든 명령은 파일 이름이기도 하다. "ls"는 명령이지만 이 명령의 실행은 실제로는 "/usr/bin/ls"라는 실행파일을 실행함으로써 작업이 이루어지게 되는 것이다. 그러므로 명령을 실행한다면 다음과 같이 실행하는 것이 올바른 것이다.

```
kihee@kihee-VirtualBox:/$ /usr/bin/ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
boot dev  home lib32 libx32 media  opt  root  sbin  srv  tmp  var
```

하지만 위의 경우 한가지 문제점이 발생된다. 그것은 실행하고자 하는 명령의 위치를 사용자가 전부 기억하고 있어야 한다는 것이다.

그렇다면 경로를 지정하지 않은 명령 만으로도 그 명령이 실행되는 이유는 무엇일까? 그것은 바로 PATH라는 환경변수 때문이다.

```
kihee@kihee-VirtualBox:/$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
```

환경변수란? 시스템을 운용하고 관리하기 위해 매크로화 시킨 값을 말한다. 그 중에서 PATH는 사용자가 명령을 실행할 때 해당 명령에 대한 실행파일을 검색하고 실행할 때 사용되는 변수이다. 사용자가 ls 라는 명령을 입력하면 쉘은 해당명령에 대한 실행파일인 ls 라는 파일을 검색하게 되는데 이때 사용되는 변수가 PATH이다.

PATH는 ":"를 구분문자로 하여 지정되어 있다. 즉 ":"로 구분하여 첫번째 경로로 부터 마지막 경로까지 해당 명령에 대한 실행파일이 존재하는지를 검색하고 존재하면 그 파일을 실행하게 된다. 만약 해당 명령에 일치하는 실행파일이 존재하지 않는다면 에러 메시지를 출력할 것이다.

```
kihee@kihee-VirtualBox:/$ whoareyou
whoareyou: 명령을 찾을 수 없습니다
```

리눅스에서의 명령은 다음과 같이 구성되어 있다.

명령 [옵션] [대상]

명령은 실행하고자 하는 파일을 명시함으로 반드시 명시하여야 하며 제일 먼저 명시되어야 한다. 이후 옵션

선과 대상은 명령에 따라 생략이 가능하다.

아래의 예에서는 단순히 명령만을 사용하였다.

```
kihee@kihee-VirtualBox:/$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
```

리눅스의 모든 명령이 명령 만으로 실행가능한 것은 아니다. 경우에 따라서는 하나 이상의 옵션 또는 하나 이상의 대상이 꼭 필요한 경우도 있다.

아래의 예는 명령과 옵션을 사용하였다.

```
kihee@kihee-VirtualBox:/$ ls -l -i
합계 76
12 lrwxrwxrwx 1 root root 7 8월 2 05:38 bin -> usr/bin
2 drwxr-xr-x 5 root root 4096 8월 7 13:29 boot
4587521 drwxrwxr-x 2 root root 4096 8월 2 05:41 cdrom
1 drwxr-xr-x 19 root root 4260 8월 7 17:29 dev
3407873 drwxr-xr-x 130 root root 12288 8월 7 16:23 etc
524289 drwxr-xr-x 4 root root 4096 8월 7 14:39 home
13 lrwxrwxrwx 1 root root 7 8월 2 05:38 lib -> usr/lib
14 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib32 -> usr/lib32
15 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib64 -> usr/lib64
16 lrwxrwxrwx 1 root root 10 8월 2 05:38 libx32 -> usr/libx32
11 drwx----- 2 root root 16384 8월 2 05:38 lost+found
```

옵션을 여러 개 지정할 때에는 각각의 옵션을 지정할 수도 있지만 아래의 그림처럼 한꺼번에 지정할 수도 있다.

```
kihee@kihee-VirtualBox:/$ ls -li
합계 76
12 lrwxrwxrwx 1 root root 7 8월 2 05:38 bin -> usr/bin
2 drwxr-xr-x 5 root root 4096 8월 7 13:29 boot
4587521 drwxrwxr-x 2 root root 4096 8월 2 05:41 cdrom
1 drwxr-xr-x 19 root root 4260 8월 7 17:29 dev
3407873 drwxr-xr-x 130 root root 12288 8월 7 16:23 etc
524289 drwxr-xr-x 4 root root 4096 8월 7 14:39 home
13 lrwxrwxrwx 1 root root 7 8월 2 05:38 lib -> usr/lib
14 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib32 -> usr/lib32
15 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib64 -> usr/lib64
16 lrwxrwxrwx 1 root root 10 8월 2 05:38 libx32 -> usr/libx32
11 drwx----- 2 root root 16384 8월 2 05:38 lost+found
```

뿐만 아니라 리눅스 명령의 옵션에는 짧은 형태의 옵션과, 긴 형태의 옵션이 있다. 짧은 형태의 옵션은 "-옵션"으로 지정하며 긴 형태의 옵션은 "--옵션"의 형태로 지정한다.

```
kihee@kihee-VirtualBox:/$ ls --format=long --inode
합계 76
12 lrwxrwxrwx 1 root root 7 8월 2 05:38 bin -> usr/bin
2 drwxr-xr-x 5 root root 4096 8월 7 13:29 boot
4587521 drwxrwxr-x 2 root root 4096 8월 2 05:41 cdrom
1 drwxr-xr-x 19 root root 4260 8월 7 17:29 dev
3407873 drwxr-xr-x 130 root root 12288 8월 7 16:23 etc
524289 drwxr-xr-x 4 root root 4096 8월 7 14:39 home
13 lrwxrwxrwx 1 root root 7 8월 2 05:38 lib -> usr/lib
14 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib32 -> usr/lib32
15 lrwxrwxrwx 1 root root 9 8월 2 05:38 lib64 -> usr/lib64
```

위의 예에서 --format=long 옵션은 -l 옵션에 대한 긴 형태의 옵션이다. 또한 --inode 옵션은 -i 옵션에 대한 긴 형태의 옵션이다. 옵션은 경우에 따라서 그 옵션에 대한 값을 지정하여야 하는 경우도 있다.

```
kihee@kihee-VirtualBox:/$ mysql -ustudent -p1234 study
```

위의 예에서 -u 옵션은 유저를 지정하는 옵션이다. 그리고 -u 옵션 다음에 명시된 student가 -u 옵션에 대한 지정 값이다. 만약 긴 형태의 옵션을 사용한다면 아래의 그림과 같을 것이다.

```
kihee@kihee-VirtualBox:/$ mysql --user=student --password=1234 study
```

아래의 예에서는 명령, 옵션, 대상을 모두 지정하였다.

```
kihee@kihee-VirtualBox:/$ ls -il /home/kihee
합계 36
524295 drwx----- 3 kihee kihee 4096 8월 7 13:14 snap
524329 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 공개
524334 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 다운로드
524336 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 문서
524333 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 바탕화면
524339 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 비디오
```

대상을 지정하는데 있어서 모든 명령이 꼭 대상을 지정하여야 하는 것은 아니다. 만약 "ls"라는 명령을 실행 시, 대상이 지정하지 않았다면 기본값으로 현재의 디렉토리가 실행의 대상이 된다. 하지만 경우에 따라서는 반드시 하나 또는 그 이상의 대상이 필요한 경우도 있다. 예를 들어 파일 또는 디렉토리를 복사한다면 어떻게? 무엇을 어디에 복사할지, 복사할 원본과 복사할 위치 두개의 대상이 반드시 필요할 것이다.

아래의 예제는 "다운로드" 디렉토리를 "downloads"라는 이름으로 복사하는 명령 예로써 복사 대상과 복사할 이름이 필요하다.

```
kihee@kihee-VirtualBox:~$ cp -apfr 다운로드 / downloads
```

리눅스 시스템에서의 명령은, 도스의 명령에 비해 상당히 유연한 명령체계를 제공한다. 예를 들자면 아래와 같이 명령을 입력하여도 정상적으로 수행된다.

```
kihee@kihee-VirtualBox:~$ ls -l /home/kihee -i
합계 40
655553 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 downloads
524295 drwx----- 3 kihee kihee 4096 8월 7 13:14 snap
524329 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 공개
524334 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 다운로드
524336 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 문서
524333 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 바탕화면
524339 drwxr-xr-x 2 kihee kihee 4096 8월 7 13:14 비디오
```

위의 명령은 "명령 옵션 대상 옵션" 형식으로 구성이 되어있다. 그래도 리눅스에서는 정상적으로 작동이 된다. 만약 도스라면 명령형식이 잘못되었다는 에러 메시지를 출력하고 명령의 실행이 종료될 것이다. 하지만 모든 명령이 옵션과 대상의 위치를 무시해도 되는 것이 아니라는 것만은 알아 두자.