



3. 파일 기본 사항

홍필두 교수
(리눅스프로그래밍)

1) 운영체제(Operating System, OS)

① 운영체제 : 컴퓨터의 하드웨어와 소프트웨어를 제어하여, 사용자가 컴퓨터를 쓸 수 있게 만들어주는 프로그램.

② 운영체제의 기능.

- 하드웨어와 응용프로그램간의 인터페이스 역할
- CPU, 주기억장치, 입출력장치 등의 컴퓨터 자원 관리
- 인간과 컴퓨터간의 상호작용을 제공함과 동시에 컴퓨터의 동작을 구동(Booting)
- 작업의 순서를 정하며 입출력 연산을 제어
- 프로그램의 실행을 제어
- 데이터와 파일의 저장을 관리

● 주요내용

1. 파일과 디렉토리 다루기

- 1)디렉토리, 절대/ 상대 경로
- 2)파일 탐색
- 3)파일내용 보기
- 4)파일 다루기
- 5)명령어 히스토리
- 6)디렉토리 관리
- 7) 파일의 문자수 세기

● 1. 강의 들어가기

✓ 학습내용 소개

- 리눅스 운영체계의 원조 격인 유닉스 운영체계는 현재 개인용PC의 설치되어 있는 윈도우 운영체계보다 먼저 출시되었으며 윈도우 운영체계의 사상의 근간이 된 부분이기도 하다. 윈도우에서 파일의 개념과 폴더(디렉토리)의 개념은 유닉스와 리눅스에도 그대로 존재한다. 또한 바탕화면의 바로가기 아이콘과 같이 파일을 단순 연결을 하여 사용하는 Link의 개념도 있다. 이번 강의에서는 리눅스에서 파일과 디렉토리 부분을 이해 후 실습하도록 한다.
- 리눅스서버 환경은 윈도우에서 파일 탐색기를 사용하는 것과는 다르게 GUI(Graphic User Interface) 환경이 아니며 하나하나 명령어 쉘 상태에서 파일과 디렉토리 관련 명령어를 입력하여야 한다. 이러한 명령어를 하나하나 배워보도록 한다.

✓ 학습목표 제시

- 디렉토리, 절대/상대경로에 대하여 이해할 수 있다.
- 파일을 탐색하고, 내용을 보거나 여러 가지 관리를 할 수 있다.
- 명령어가 사용된 기록을 찾아볼 수 있다.
- 디렉토리를 관리 할 수 있다.
- 파일의 문자의 개수를 알 수 있다.

● 2. 생각해볼 문제 및 용어

✓ 학습전 생각해볼 문제

- 윈도우 OS에서 “탐색기”의 기능을 자세히 조사.
- 윈도우의 명령어 쉘(실행창)에서 dir, cd명령 등 파일 및 디렉토리를 다루는 명령에 대하여 알아보고 실습하기

✓ 용어 (강의 정리 시 필기 할 것)

- 탐색기, shell, file system, bit, bite, 32bit/64bit

- 디렉토리, 폴더

- 파일명칭 규격 *확장자 마다 어떤 프로그램인지 알 수 있음*

- Windows file system, UNIX file system, Apple file system

*순명제 32/64 하이 64→32 용량
CPU가 한번에 처리하는 양 32→64 용량X*

*↓
8bit = 1bite*

● 3. 이해하기

(1) 파일과 디렉토리 다루기

리눅스서버 환경은 윈도우에서 파일 탐색기를 사용하는 것과는 다르게 GUI(Graphic User Interfac

e) 환경이 아니며 하나 하나 명령어 쉘 상태에서 파일과 디렉토리 관련 명령어를 입력하여야 한다. 이러한 명령어를 하나하나 배워보도록 한다.

서버버전에는 이우스가 없음

● 3. 이해하기

중요 1) 디렉토리, 절대/상대 경로

윈도우 OS의 파일을 폴더의 모아두는 개념과 마찬가지로 리눅스/유닉스에서는 디렉토리와 파일의 개념을 사용한다

① 디렉토리의 위치는 절대경로와 상대경로로 표시됨

② 윈도우에서 최상위 경로는 하드디스크 등 디바이스의 최상위 디렉토리(폴더)로 예를 들어 “C:”로 표시되며 이때 최 상위 경로는 슬래쉬 기호 “\”, “\”로 표시된다. 유닉스와 리눅스에서는 슬래쉬가 아닌 역 슬래쉬 “/” 기호가 최상위 디렉토리이며, 루트(root) 디렉토리 라고 읽는다.

③ 절대경로 루트부터 다 써주는 것

· 루트(root) 디렉토리부터 현재 파일이 위치한 디렉토리의 경로를 전체로 표기한 경로

cd /etc/ssh 는 절대경로

· 예 : /home/kopoctc/test/help.txt

④ 상대경로

· 명령어 쉘 상태에서 현재 위치로부터 파일이 있는 디렉토리를 표기한 경로

· 현재 위치를 점으로 나타내서 표기

· 현재 위치는 점을 한 개 “.”, 하위 상위 디렉토리 2개 “..”로 표시

cd /etc 는 루트 밑에 etc에 위치

cd /etc/ssh 상대경로

cd ./ssh 상대경로
cd . ssh
생각가능

Tip

예 : 현재 위치가 /home/kopoctc 라면

- 1) ../test/help.txt (상대경로 표시) -> /home/kopoctc/test/help.txt
- 2) ../../home/kopoctc/test/help.txt : ..은 상위 디렉토리를 의미하며, 결국 /home/kopoctc/test/help.txt 과 같은 위치

● 3. 이해하기

2) 파일 탐색

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

① `pwd` (print name of current/workinh directory)

· 현재 작업 디렉토리가 어디인지 출력

② `cd` (change directory)

· 현재 작업 디렉토리를 바꿈.

③ `ls` (list directory contents)

· 디렉토리에 파일 목록을 보여줌.

`cd ~`
`cd ~ /.. = /home`

홈디렉토리는

~은 홈디렉토리이다

```
root@kopoctc:~# pwd
/root
root@kopoctc:~# cd /home
root@kopoctc:/home# ls
aaa.log bbb.log ccc.log kopoctc kopoctcl
root@kopoctc:/home#
```

<그림 I -35> `pwd`, `cd`, `ls` 명령어

● 3. 이해하기

3) 파일내용보기

텍스트(TEXT)파일의 내용을 볼 수 있는 명령어로 **cat**, **page**, **head**, **tail**이 있으며, 파일의 내용이 많을 경우 화면 단위로 끊어서 보기위한 명령어로 **|more** 가 있다.

① cat

· **cat filename** : **filename**이라는 파일의 내용을 한 번의 출력 <그림 I -36> 참조

② more

· **|** (파이프)를 같이 사용하여 화면 단위로 출력

· 예 : “**cat filename | more**”

③ page

· **page filename** : **filename**의 파일을 화면 단위로 출력함.

· 멈춰진 현재 화면에서 아무키나 누르면 다음 페이지가 보여짐.

④ head

· 파일의 내용을 맨 앞을 기준으로 보여줌

· **head -n filename** : **filename**의 파일을 처음 **n**줄을 표시

· 예 : **head -10 abc.txt** : **abc.txt**파일을 처음부터 **10**줄을 보여줌

⑤ tail

· 파일의 내용을 맨 뒤를 기준으로 보여줌

· **tail -n filename** : **filename**의 파일을 뒤부터 **n**줄을 표시

● 3. 이해하기

· `tail -f filename` : 계속 작성중인 파일의 마지막을 계속적으로 표시

· 시스템관리 관련 로그저장 파일등을 실시간으로 감시할 때 많이 사용

Tip

모든 명령어는 기본적으로 사용되는 명령어를 소개하는 것을 원칙으로 한다. 하지만 명령어 대부분이 선택(option)값으로 인자를 “-A”와 같은 형식으로 넘겨서 사용한다. 옵션 값은 다양한 기능들이 있으니 명령어를 사용전 man 명령어로 명령어 사용 옵션을 확인하는 습관을 가지도록 한다

```
root@kopoctc:/etc/network# cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet static
    address 192.168.56.101
    netmask 255.255.255.0
    network 192.168.56.0
root@kopoctc:/etc/network#
```

<그림 1-36> cat 명령어 사용

● 4. 실습하기(1)

1) 파일탐색

- ① [pwd]
- ② [cd]
- ③ 상위 디렉토리 , 하위 디렉토리
- ④ 절대경로 , 상대 경로
- ⑤ . / .. * : 특별한 문자의 사용
- ⑥ [ls -adglFR]

2) 파일 내용보기

- ① [cat filename]
- ② "cat filename | more "
- ③ [page filename]
- ④ [head -n filename]
- ⑤ [tail -n filename]
- ⑥ [tail -f filename]

● 3. 이해하기

4) 파일 다루기

파일을 복사하고 파일 이름을 바꾸는 명령으로 **mv**, **cp**가 있다. <그림 I -37>

① mv

·파일을 이동 (잘라내서 복사)하는 명령

·**mv oldfilename newfilename** : oldfile의 파일을 newfile로 이동함. 결국 파일명을 바꾸는 명령 (예: **mv a.txt b.txt** : a.txt파일명을 b.txt로 바꿈)

·**mv filename dirName** : 파일을 해당 디렉토리로 보냄. 뒤의 인자가 파일명이 아니라 디렉토리명인 경우 해당 디렉토리 에 파일을 옮기는 결과가 됨 (예: **mv a.txt adir** : a.txt파일을 adir의 디렉토리로 옮기는 명령이며 결국 **mv a.txt adir/a.txt**와 같은 명령으로 뒤 인자의 a.txt가 생략된 것으로 인식된 것)

·**mv oldDirName newDirName** : 두인자의 명칭이 디렉토리 인 경우, oldDirname의 모든 파일을 newDirname이라는 디렉토리를 만든후 옮기는 명령이 됨. 결국 결과는 디렉토리 명을 바꾸는 명령 임

② cp

·파일을 복사(기존 파일은 남음)하는 명령

·**cp oldfilename newfilename** : oldfile의 파일을 newfile로 복사함.

Tip

cp a.txt b.txt : a.txt파일을 b.txt로 복사하는 명령으로 실행후 a.txt와 b.txt파일 두 개가 존재

·**cp filename dirName** : 파일을 해당 디렉토리로 보냄. 뒤의 인자가 파일명이 아니라 디렉토리명인 경우 해당 디렉토리 에 파일을 복사하는 결과가 됨

● 3. 이해하기

Tip

`cp a.txt adir` : a.txt파일을 adir의 디렉토리로 복사하는 명령이며 결국 `cp a.txt adir/a.txt` 같 명령으로 뒤 인자의 a.txt가 생략된 것으로 인식된 것
와 은

• `cp oldDirName newDirName` : 두인자의 명칭이 디렉토리 인 경우, `oldDirName`의 모든 파일을 `newDirName`이라는 디렉토리를 만든 후 복사하는 명령이 됨.

• 실제로는 `cp -R oldDirName newDirName` 명령이 많이 사용됨. -R 옵션은 하위디렉토리까지 모두 복사하는 명령

③ rm

• 파일을 지우는 명령

• `rm filename`: 파일을 지움 (예: `rm a.txt a.txt`파일을 지움)

• `rm -i filename`: 파일을 지움, 단 확인 메시지가 나와서 지울것인지 묻고 **yes**를 선택시만 지움

• `rm -r` : 해당 위치의 파일 및 디렉토리를 지움

• `rm -d` : 빈 디렉토리를 지움

• `rm -f` : 파일을 지울 때 확인 물음을 하지 말고 강제로 지움

• 다음 명령어 사용은 주의하여야 함 [`rm *`] [`rm -F`] [`rm -rF`]

• `rm *` : 모든 파일을 다 지우는 명령

• `rm -F`: 지우는 과정을 묻게 되있더라도 이를 무시하고 묻지않고 지움

• `rm -rF`: 해당 위치의 모든 파일 및 디렉토리를 묻지도 말고 무조건 지움. 만일 최상위 루트(**root**)디렉토리에서 해당 명령을 실행시 서버의 모든 파일이 포맷(**format**)수준으로 지워짐. 회사 운영자가 이 명령을 잘못 사용하다간 사표쓰는 경우가 생김.

● 3. 이해하기

```
root@kopoctc:/home/kopoctc# ls
help.txt
root@kopoctc:/home/kopoctc# cp help.txt help2.txt
root@kopoctc:/home/kopoctc# ls
help2.txt  help.txt
root@kopoctc:/home/kopoctc# mv help.txt hepl3.txt
root@kopoctc:/home/kopoctc# ls
help2.txt  hepl3.txt  root@
kopoctc:/home/kopoctc#
```

<그림 1-37> cp, mv 명령어 사용

● 4. 실습하기(2)

3) 파일 다루기

- ① `[mv oldfilename newfilename]`
- ② `[mv filename dirName]`
- ③ `[mv oldDirName newDirName]`
- ④ `[cp filename {path/}filename]`
- ⑤ `[cp filename path]`
- ⑥ `[cp DirName newDirName]`

4) 파일 지우기

- ① `[rm filename]`
- ② `[rm -i filename]`

주의 `[rm *]` `[rm -F]` `[rm -rF]`

● 3. 이해하기

5) 명령어 히스토리

리눅스 명령은 윈도우 시스템과 달리 매번 명령어를 입력하기 때문에 불편할 수 있다. 하지만 이전에 입력한 명령어를 다 시 찾아 명령할 수 있는 기능이 있는데, 이 명령어 히스토리 기능을 이용하면 명령어 사용이 편리해 진다. <그림 I -38>

① [r 또는 !]

- 유닉스 계열 (korn shell)에서는 “r”을 입력하고 리눅스 계열(c shell)에서는 “!” (느낌표)를 입력하면 기존 사용했던 명령어를 찾을 수 있음

- 예를 들어 “!c”이라고 명령하면 기존 사용한 명령어 중 c로 시작하는 명령을 실행한다

② [화살표 위,아래]

- 지금까지 사용한 명령어를 순차적으로 보여줌.

```
root@kopoctc:/home/kopoctc# ls
help2.txt  hepl3.txt
root@kopoctc:/home/kopoctc# !c
cp help.txt help2.txt
cp: cannot stat 'help.txt' : No such file or directory
root@kopoctc:/home/kopoctc#
```

<그림 I -38> 명령어 히스토리 사용

③ 명령어 히스토리 저장 파일

- 지금까지 사용한 명령어는 해당 사용자의 기본 디렉토리 내에 “.bash_history” (리눅스 bash)또는, “.history”(일반적 유닉스) 라는 파일안에 저장되어 있음. <그림 I -39>

● 3. 이해하기

- 어떠한 사용자가 어떤 명령어로 시스템에 접근하였는지 근거파일로 사용됨.
- 기업용 시스템에서는 해당 파일을 실시간으로 중앙통제장치로 전송하여 보관함으로써 불순한 의도의 접근을 감시하는 용도로도 사용됨.

```
kopectc@kopectc:~$ pwd
/home/kopectc
kopectc@kopectc:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  help2.txt  hepl3.txt
kopectc@kopectc:~$ tail -n5 .bash_history
apt update
su -
ls
who
su -
kopectc@kopectc:~$
```

<그림 1-39> 명령어 히스토리 저장 파일

● 3. 이해하기

6) 디렉토리 관리

디렉토리를 관리하기 위하여 디렉토리를 만들거나, 디렉토리를 지우는 작업을 하는데 이때 `mkdir`, `rmdir` 명령을 사용한다. <그림 I -40>

① `mkdir`

·[`mkdir dirname`] : 새로운 디렉토리를 만들 `mkdir -p aaa/bbb/c`
aaa 안에 bbb 안에 c 디렉토리 만들

② `rmdir`

·[`rmdir dirname`] : 해당 디렉토리를 지움, 단 해당 디렉토리에 파일이 있으면 지울 수 없음

·해당 디렉토리에 파일이 있는 경우엔 모든 파일 및 해당 디렉토리를 지우기 위하여 `rm` 명령어를 응용하여 사용 (예: `abc` 디렉토리 및 디렉토리 하단 파일까지 지우기 위하여 `rm -rf abc`를 사용)

● 3. 이해하기

```
kopectc@kopectc:~$ pwd
/home/kopectc
kopectc@kopectc:~$ mkdir mydir
kopectc@kopectc:~$ ls -a
.  .bash_history  .bashrc  help2.txt  mydir  .viminfo
.. .bash_logout  .cache  help3.txt  .profile
kopectc@kopectc:~$ cd mydir
kopectc@kopectc:~/mydir$ pwd
/home/kopectc/mydir
kopectc@kopectc:~/mydir$ touch myfile
kopectc@kopectc:~/mydir$ ls
myfile
kopectc@kopectc:~/mydir$ cd ../
kopectc@kopectc:~$ rm mydir
rm: cannot remove 'mydir' : Is a directory
kopectc@kopectc:~$
```

<그림 1-40> mkdir, rmdir 명령어 사용

● 3. 이해하기

7) 파일의 문자수 세기

파일의 글자 수를 세는 명령어로 **wc**가 있는데, 일반적으로 파일의 크기를 글자 수로 알고 싶을 때 나, 셸 프로그래밍에서 응용하여 많이 사용된다. <그림 I -41>

① wc

· 파일 내부의 글자 수 및 줄 수를 보여줌

· **[wc]** : 출력되는 순서는 파일의 줄 수[newline], 단어 수[word], 글자 수[byte]로 보여줌

```
kopoetc@kopoetc:~/mydir$ ls -al
total 12
drwxrwxr-x 2 kopoetc kopoetc 4096 Jan 18 10:56 .
drwxr-xr-x 4 kopoetc kopoetc 4096 Jan 18 10:55 ..
-rw-rw-r-- 1 kopoetc kopoetc  27 Jan 18 10:55 myfile
kopoetc@kopoetc:~/mydir$ cat myfile
You are a good guy
Me too.
kopoetc@kopoetc:~/mydir$ wc myfile
 2  7 27 myfile
```

<그림 I -41> wc 명령어 사용

● 4. 실습하기(2)

3) 파일 다루기

- ① `[mv oldfilename newfilename]`
- ② `[mv filename dirName]`
- ③ `[mv oldDirName newDirName]`
- ④ `[cp filename {path/}filename]`
- ⑤ `[cp filename path]`
- ⑥ `[cp DirName newDirName]`

4) 파일 지우기

- ① `[rm filename]`
- ② `[rm -i filename]`

주의 `[rm *]` `[rm -F]` `[rm -rF]`

● 4. 실습하기(3)

5) 명령어 히스토리

- ① [r , !]
- ② [화살표, "ctrl-[" 후 "ctrl-k" 과 "ctrl-k"]
- ③ [.history .bash_history]

6) 디렉토리 관리, 글자 수 세기

- ① [mkdir dirname]
- ② [rmdir dirname]
- ③ [wc]

● 4. 실습하기 (4 – Jump Up)

- 1) 디렉토리, cp, mv등에서 *, ~, ? 에 대하여 실습
- 2) 금일 배운 명령어를 man으로 찾아보고 각종 옵션에 대하여 조사 후 실습

● 5. 퀴즈

문제	보기	정답 및 해설
1) 리눅스 유닉스에서 최상위 디렉토리 (root 디렉토리)를 의미하는 것은?	가) c:W 나) W 다) / 라) *	다) 유닉스 리눅스에서 역 슬레쉬(/)가 최상위 디렉토리를 의미합니다.
2) 파일 맨끝 5줄을 보기를 원 할때 사용하는 명령어는?	가)tail -5 filename 나)tail -n5 filename 다)head -5 filename 라)head -n5 filename	나) 파일 끝부분을 출력하는 명령어는 tail이며 출력 줄 수를 옵션으로 지정하는 경우는 n을 사용합니다.
3) 사용자가 어떤 명령어를 사용했는지 조사하기 위하여 적당한 파일은?	가) logfile 나).bash_history 다)xinetd.conf 라)rootfile	나) bash_history파일은 명령어 히스토리가 담겨있는 파일입니다.
4) [!c]라고 명령어를 사용한 경우 실행되는 명령어는?	가) error 나) clear 다) move 라) mkdir	나) [!c]라고 명령하면 지금까지 실행한 명령어 중 c로 시작하는 명령어를 실행합니다.

● 6. 정리하기

✓ 다음 제시된 내용을 자필로 작성하여 제출 하시오 (상단 학번, 이름 기입)

1. 파일과 디렉토리 다루기

- 1)디렉토리, 절대/ 상대 경로에 대하여 설명하라
- 2)파일 탐색 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라
- 3)파일내용 보기 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라
- 4)파일 다루기 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라
- 5)명령어 히스토리 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라
- 6)디렉토리 관리 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라
- 7) 파일의 문자수 세기 주요명령어를 나열하고 자주 사용하는 옵션을 설명하라

● 7. 차시 예고

✓ 차시 학습내용

- 파일 필터를 이해하고 사용하는 방법을 알 수 있다.
- 파일을 비교하고, 정렬하고, 필요한 파일을 찾을 수 있다.
- 파일 여러 개를 하나의 파일로 묶어서 보관할 수 있다.
- 파일의 크기를 줄이기 위한 압축과 복원 명령을 실행시킬 수 있다.

✓ 차시 준비

- 윈도우에서 파일을 다루는 유틸리티 프로그램이나 워드, 아래아 한글 등에서 기능 등을 사용하여 두 개의 파일의 내용을 비교하는 방법을 알아봅니다.
- 윈도우 엑셀 등의 프로그램에서 행을 정렬(sort)하는 방법에 대하여 알아봅니다.