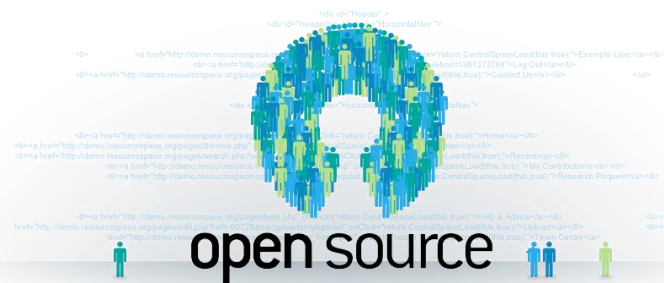




오픈소스 리눅스 실무

리눅스 기본 명령어2





디렉토리 내용

- ls (list)
 - 현재 또는 지정된 디렉토리의 내용을 리스트로 확인

```
ls [옵션] [ 디렉토리 ]
```

- ls 현재 디렉토리의 리스트를 출력

```
kdh@cs: /  
kdh@cs:/$ ls  
bin      initrd.img      media  run      tmp  
boot     initrd.img.old  mnt    sbin     usr  
dev      lib             opt    snap     var  
etc      lib64           proc   srv      vmlinuz  
home     lost+found      root   sys      vmlinuz.old  
kdh@cs:/$
```



디렉토리 내용

- ls (list)
 - 자주 사용하는 옵션

옵션	기능
-a	숨겨진 파일을 포함하여 모든 파일을 리스트
-s	파일의 크기를 k 바이트 단위로 출력
-l	파일의 상세 정보를 출력
-F	파일의 종류를 표시하여 출력
-R	모든 하위 디렉터리들을 리스트



디렉토리 내용

- ls (list)
 - 옵션 -a (all)
 - . 으로 시작하는 숨김 파일을 포함한 모든 파일과 디렉토리를 리스트
 - . 은 현재 디렉토리를 .. 은 부모 디렉토리를 나타냄

```
kdh@cs: /  
kdh@cs:/$ ls -a  
.      etc      lib64      proc      srv      vmlinuz  
..     home     lost+found root      sys      vmlinuz.old  
bin    initrd.img  media      run       tmp  
boot   initrd.img.old mnt        sbin     usr  
dev    lib        opt        snap     var  
kdh@cs:/$
```



디렉토리 내용

- ls (list)
 - 옵션 -s (size)
 - 디렉토리 내에 있는 파일의 크기를 킬로바이트(k byte)로 출력

```
kdh@cs: /  
kdh@cs:/$ ls -s  
합 계 96  
4 bin 16 lost+found 4 srv  
4 boot 4 media 0 sys  
0 dev 4 mnt 4 tmp  
12 etc 4 opt 4 usr  
4 home 0 proc 4 var  
0 initrd.img 4 root 0 vmlinuz  
0 initrd.img.old 0 run 0 vmlinuz.old  
4 lib 12 sbin  
4 lib64 4 snap  
kdh@cs:/$
```



디렉토리 내용

- ls (list)
 - 옵션 -l (long)
 - 디렉토리 내에 있는 파일을 속성을 포함해서 출력

```
kdh@cs: /  
kdh@cs:/$ clear  
kdh@cs:/$ ls -l  
합 계 96  
drwxr-xr-x  2 root root  4096  8월  28 16:55 bin  
drwxr-xr-x  3 root root  4096 11월  1 18:47 boot  
drwxr-xr-x 19 root root  4340  9월  6 19:04 dev  
drwxr-xr-x 119 root root 12288 11월  9 12:33 etc
```

접근권한, 링크 수 ,사용자 아이디,그룹 아이디,파일크기,최종 수정시간,파일명



디렉토리 내용

- ls (list)
 - 옵션 -F (file type)
 - 디렉토리 내에 있는 파일을 종류를 표시
 - *: 실행파일, /: 디렉터리, @:심볼릭 링크

```
kdh@cs: /  
kdh@cs:/$ ls -F  
bin/      home/      lib64/     opt/       sbin/      tmp/        vmlinuz.old@  
boot/     initrd.img@  lost+found/  proc/     snap/     usr/          
dev/      initrd.img.old@  media/      root/     srv/      var/          
etc/      lib/        mnt/        run/      sys/      vmlinuz@  
kdh@cs:/$
```



디렉토리 내용

- ls (list)
 - 옵션 -R (file type)
 - 디렉토리 내에 있는 하위 디렉토리도 포함해서 출력
- ls -R 현재 디렉토리에 포함된 하위 디렉토리를 포함한 모든 파일
- ls -R /home home 디렉토리에 포함된 하위 디렉토리를 포함한 모든 파일



디렉토리 내용

- ls (list)
 - 옵션의 조합
 - ls -al 숨김 파일(a)을 포함한 파일의 속성정보(l)를 출력
 - ls -asl 숨김 파일(a)을 포함하고 사이즈(s)와 파일의 속성정보(l)를 출력
 - ls -lF 파일의 속성정보(l)와 파일의 종류를 출력



디렉토리 내용

- dir
 - ls 명령의 심벌릭 링크
- vdir
 - ls -l 명령의 심벌릭 링크

```
kdh@cs: ~/linux
kdh@cs:~/linux$ dir
cs.txt
kdh@cs:~/linux$ vdir
합 계 0
-rw-rw-r-- 1 kdh kdh 0 11월 17 18:32 cs.txt
kdh@cs:~/linux$
```



cat 명령의 활용

- 입력된 내용을 그대로 출력하는 명령

```
kdh@cs: ~  
kdh@cs:~$ cat  
hi  
hi  
hello  
hello  
kdh@cs:~$
```

중지는 ctrl + d

- cat 명령 뒤에 파일명을 입력하면 파일의 내용을 입력 파일의 내용을 모니터에 출력

```
kdh@cs: ~  
kdh@cs:~$ cat hello.c  
#include <stdio.h>  
  
int main(int argc , char* argv[] ) {  
    printf("hello c\n");  
    return 0;  
}  
kdh@cs:~$
```



Linux

cat 명령의 활용

- cat 명령 뒤에 > 파일명을 입력하면 ctrl + d 가 입력될 때까지의 키보드 입력을 파일로 만듦

```
kdh@cs: ~  
kdh@cs:~$ cat > ex.txt  
hello  
linux  
kdh@cs:~$
```

중지는 ctrl + d

- 확인

```
kdh@cs: ~  
kdh@cs:~$ cat ex.txt  
hello  
linux  
kdh@cs:~$
```



Linux

빈 파일 생성

- touch
 - 파일 크기가 0인 이름만 있는 빈 파일을 만듦

kdh@cs: ~/linux

```
kdh@cs:~/linux$ touch cs.txt
```

```
kdh@cs:~/linux$ ls -s
```

```
합 계 0
```

```
0 cs.txt
```

```
kdh@cs:~/linux$
```



파일 내용 출력

- more 파일(들)
 - 파일(들)의 내용을 페이지 단위로 나누어 화면에 출력
 - Enter는 한줄 이동 , space는 한페이지 이동

```
kdh@cs: ~  
#include <stdio.h>  
#define MAXLINE 100  
void copy(char from[], char to[]);  
char line[MAXLINE];  
char longest[MAXLINE];  
  
main() {  
    int len;  
    int max;  
    max = 0;  
    while(gets(line)) {  
        len = strlen(line);  
        if(len == 0) break;  
        if(len > max) {  
            max = len;  
            copy(line, longest);  
        }  
    }  
    if(max > 0)  
    --More-- (68%)
```



파일 내용 출력

- head [-n] 파일
 - 파일의 앞부분 내용을 옵션으로 지정한 라인 수 만큼 화면에 출력

```
kdh@cs: ~  
kdh@cs:~$ head -10 longest.c longest.c 파일의 상단 10줄  
#include <stdio.h>  
#define MAXLINE 100  
void copy(char from[], char to[]);  
char line[MAXLINE];  
char longest[MAXLINE];  
  
main() {  
    int len;  
    int max;  
    max = 0;  
kdh@cs:~$
```



Linux

파일 내용 출력

- tail [-n] 파일
 - 파일의 뒷부분 내용을 옵션으로 지정한 라인 수 만큼 화면에 출력

```
kdh@cs: ~  
kdh@cs:~$ tail -10 longest.c longest.c 파일의 하단 10줄  
    return 0;  
  
}  
  
void copy(char from[], char to[]){  
    int i ;  
    i = 0;  
    while((to[i] = from[i]) != '\0')  
        i++;  
}  
kdh@cs:~$
```




파일 내용 출력

- `wc` `[-lwc]` 파일
 - 파일의 내용에 대한 줄(`l`), 단어(`w`), 문자(`c`)의 개수를 출력

```
kdh@cs: ~  
kdh@cs:~$ wc longest.c  
 30  66 432 longest.c  
kdh@cs:~$ wc -l longest.c  
30 longest.c  
kdh@cs:~$ wc -w longest.c  
66 longest.c  
kdh@cs:~$ wc -c longest.c  
432 longest.c  
kdh@cs:~$ █
```

파일 다루기 (파일 , 디렉토리 복사)



Linux

- cp
 - 파일이나 디렉토리를 복사

cp [옵션] 원본(원본 디렉토리) 사본(사본 디렉토리)

옵션 :

- i 파일이 이미 존재하면 덮어 쓸 것인가를 묻음
- r 하위 디렉토리가 존재하는 경우 디렉토리 포함

파일 다루기 (파일 , 디렉토리 복사)



Linux

- 파일 복사

- cp 원본파일 사본파일

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
cs.txt
kdh@cs:~/linux$ cp cs.txt cs.backup
kdh@cs:~/linux$ ls
cs.backup cs.txt
kdh@cs:~/linux$
```

- cp -i 원본파일 사본파일

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
cs.txt
kdh@cs:~/linux$ cp cs.txt cs.backup
kdh@cs:~/linux$ ls
cs.backup cs.txt
kdh@cs:~/linux$ cp -i cs.txt cs.backup
cp: 'cs.backup'를 덮어 쓸까요? y
kdh@cs:~/linux$
```

(복사하려는 이름이 이미 있는 경우)

y : 기존파일을 덮어쓰

n : 기존파일을 그대로 두고 복사하지 않음

파일 다루기 (파일 , 디렉토리 복사)



Linux

- 파일을 다른 디렉토리에 복사
 - cp 원본파일 디렉토리

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup  cs.backup  cs.txt
kdh@cs:~/linux$ cp cs.backup backup
kdh@cs:~/linux$ ls backup
cs.backup
kdh@cs:~/linux$
```

- 디렉토리를 다른 디렉토리로 복사
 - cp -r 디렉토리 디렉토리
 - r 옵션은 하위 디렉토리도 포함해서 복사

파일 다루기 (파일 , 디렉토리 복사)



Linux

- 여러 파일을 다른 디렉토리에 복사 (마지막은 반드시 디렉토리)
 - cp 원본파일1 원본파일2 디렉토리

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup  cs.backup  cs.txt
kdh@cs:~/linux$ cp cs.backup cs.txt backup
kdh@cs:~/linux$ ls backup
cs.backup  cs.txt
kdh@cs:~/linux$
```



파일 다루기 (파일 이동)

- mv
 - 원본 파일 또는 디렉토리를 다른 이름 또는 다른 위치로 이동

`mv` [옵션] 원본(원본 디렉토리) 이동파일(이동 디렉토리)

옵션 :

-i 파일이 이미 존재하면 덮어 쓸 것인가를 묻음

첫 번째 인자는 원본 파일명이나 디렉터리명을 지정하고 두 번째 인자는 목적지 파일명이나 디렉터리명을 지정



파일 다루기 (파일 이동)

- mv 원본파일 이동파일
 - 파일을 다른 파일로 이동 (이름 변경)

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup  cs.backup  cs.txt
kdh@cs:~/linux$ mv cs.backup cs.old
kdh@cs:~/linux$ ls
backup  cs.old  cs.txt
kdh@cs:~/linux$
```



파일 다루기 (파일 이동)

- mv 원본파일 디렉토리
 - 파일을 다른 디렉토리로 이동

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup  cs.old  cs.txt
kdh@cs:~/linux$ mv cs.old backup
kdh@cs:~/linux$ ls
backup  cs.txt  cs.old 가 없어짐
kdh@cs:~/linux$ ls backup
cs.backup  cs.old  cs.txt  cs.old 가 이동
kdh@cs:~/linux$
```




파일 다루기 (파일 이동)

- mv 파일1 파일2 디렉토리
 - 여러 파일을 다른 디렉토리로 이동

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup  cs.old  cs.txt
kdh@cs:~/linux$ mv cs.old cs.txt backup
kdh@cs:~/linux$ ls
backup  cs.old , cs.txt 가 없어짐
kdh@cs:~/linux$ ls backup
cs.backup  cs.old  cs.txt  cs.old , cs.txt 가 이동
kdh@cs:~/linux$
```



파일 다루기 (파일 이동)

- mv 원본디렉토리 이동디렉토리
 - 이동 디렉토리가 없는 경우 원본디렉토리의 이름 변경
 - 이동 디렉토리가 있는 경우 하위 디렉토리로 이동

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
backup
kdh@cs:~/linux$ mv backup old
kdh@cs:~/linux$ ls
old
kdh@cs:~/linux$
```



파일 다루기 (파일 삭제)

- rm
 - 지정한 파일 또는 디렉토리를 삭제

rm [옵션] 파일(디렉토리)

옵션 :

- i 파일을 정말 삭제할지를 묻음
- r 디렉토리 삭제



파일 다루기 (파일 삭제)

- `rm -i`

```
kdh@cs: ~/linux/old
kdh@cs:~/linux/old$ ls
cs.backup cs.old cs.txt
kdh@cs:~/linux/old$ rm -i cs.old
rm: 일반 빈 파일 'cs.old'를 제거 할 까 요 ? y
kdh@cs:~/linux/old$ ls
cs.backup cs.txt
kdh@cs:~/linux/old$
```

파일을 삭제할지를 확인

- `rm -r`

```
kdh@cs: ~/linux
kdh@cs:~/linux$ rm -r old
kdh@cs:~/linux$ ls
kdh@cs:~/linux$
```

디렉토리를 삭제

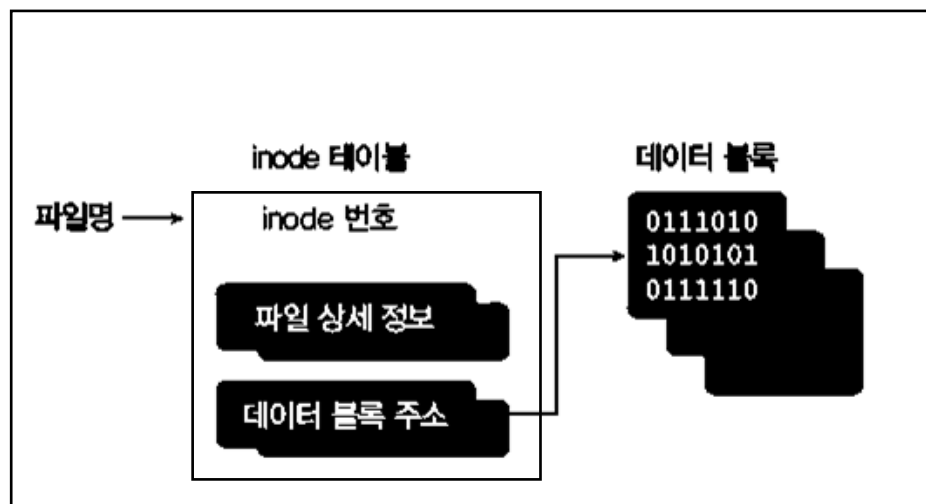


파일 다루기 (파일 링크)

- 파일링크

- 기존에 있는 파일에 대해 새로운 파일명을 만드는 것
 - 하드링크 : 기존 파일과 동일한 데이터 블록을 가리키는 파일 생성
 - 소프트링크 : 심벌릭 링크 - 기존 파일을 가리키는 링크파일 생성

파일 시스템





파일 다루기 (파일 링크)

- ln
 - 원본 파일에 대한 링크 파일을 만듦

ln [옵션] 원본파일 링크파일

옵션 :

-s 심벌릭 링크 생성



파일 다루기 (파일 링크)

- ln exam exam.link

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
exam
kdh@cs:~/linux$ ln exam exam.link
kdh@cs:~/linux$ ls
exam exam.link
kdh@cs:~/linux$ ls -li
40370278 exam 40370278 exam.link
kdh@cs:~/linux$
```

- exam 파일에 대해 exam.link 파일을 만듦
- ls -li 로 inode에 대한 정보를 출력해 보면 두 파일 모두 inode 가 같음 (동일한 파일)
- 원본파일이 삭제되도 링크파일로 파일 접근 가능



파일 다루기 (파일 링크)

- `ln -s exam exam.softlink`

```
kdh@cs: ~/linux
kdh@cs:~/linux$ ls
exam exam.link
kdh@cs:~/linux$ ln -s exam exam.softlink
kdh@cs:~/linux$ ls -l
합 계 8
-rw-rw-r-- 2 kdh kdh 13 11월 19 09:10 exam
-rw-rw-r-- 2 kdh kdh 13 11월 19 09:10 exam.link
lrwxrwxrwx 1 kdh kdh 4 11월 19 09:26 exam.softlink -> exam
kdh@cs:~/linux$ ls -li
40370278 exam 40370278 exam.link 40370279 exam.softlink
kdh@cs:~/linux$
```

- exam 파일에 대해 exam.softlink 파일을 만듦
- 링크파일 -> 원본파일 형태로 표시
- `ls -li` 로 inode에 대한 정보를 출력해 보면 두 파일은 서로 다른 inode 를 가짐 (원본 파일에 대한 바로가기 파일)
- 원본파일이 삭제되면 링크파일은 더 이상 파일 참조 불가능