

## 제8장 파일 유틸리티

VEDA

I

### 8.1 파일 속성으로 파일 찾기

## find 명령어

---

- find 명령어
  - 파일 이름이나 속성을 이용하여 해당하는 파일을 찾는다.
- 사용법

```
$ find 디렉터리 [-옵션]
```

옵션의 검색 조건에 따라 지정된 디렉터리 아래에서 해당되는 파일들을 모두 찾아 출력한다<sup>2)</sup>



---

▶ 3

## find 명령어

---

- 예

```
$ find ~ -name src -print
/home/chang/linux/src

$ find ~ -name src -ls
89090 4 drwxrwxr-x 13 chang cs 4096 9월22 /home/chang/linux/src

$ find /usr -name *.c -print
```

---

▶ 4

## find 명령어: 검색 조건

검색 조건 및 처리 방법	설명
-name 파일명	파일명으로 찾는다.
-atime +n	접근 시간이 n일 이전인 파일을 찾는다.
-atime -n	접근 시간이 n일 이내인 파일을 찾는다.
-mtime +n	n일 이전에 수정된 파일을 찾는다.
-mtime -n	n일 이내에 수정된 파일을 찾는다.
-perm nnn	접근권한이 nnn인 파일을 찾는다.
-type x	파일 종류가 x인 파일들을 찾는다.
-size n	크기가 n 블록(512바이트)인 파일들을 찾는다.
-links n	링크 개수가 n인 파일들을 찾는다.
-user 사용자명	파일의 소유자가 사용자명인 파일을 찾는다.
-group 그룹명	그룹명을 갖는 그룹에 속한 파일을 찾는다.
-print	찾은 파일의 절대 경로명을 화면에 출력한다.
-ls	찾은 파일에 대해 ls -dils 명령어 실행 결과를 출력한다.
-exec cmd {};	찾은 파일들에 대해 cmd 명령어를 실행한다.

## find 명령어: 검색 조건

- 파일의 접근 권한(-perm)으로 검색

```
$ find . -perm 700 -ls
```

- 파일의 접근 시간(-atime) 혹은 수정 시간(-mtime)으로 검색

+n: 현재 시각을 기준으로 n일 이상 전

n: 현재 시각을 기준으로 n일 전

-n: 현재 시각을 기준으로 n일 이내

```
$ find . -atime +30 -print
```

```
$ find . -mtime -7 -print
```

## find 명령어: 검색 조건

---

- 파일의 소유자(-user)로 검색

```
$ find . -user chang -print
```

- 파일 크기(-size)로 검색

```
$ find . -size +1024 -print
```

- 파일 종류(-type)로 검색

d : 디렉터리	f: 일반 파일	l: 심볼릭 링크
b: 블록 장치 파일	c: 문자 장치 파일	s: 소켓 파일

```
$ find ~ -type d -print
```

---

▶ 7

## find 명령어: 검색 조건 조합

---

- find 명령어는 여러 검색 옵션을 조합해서 사용할 수 있다.

- 예

```
$ find . -type d -perm 700 -print
```

```
$ find . -name core -size +2048 -ls
```

---

▶ 8

## find 명령어: 검색된 파일 처리

---

- find 명령어의 -exec 옵션
  - 검색한 모든 파일을 대상으로 동일한 작업(명령어)을 수행
- 예

```
$ find . -name core -exec rm -i {} \;  
$ find . -name *.c -atime +30 -exec ls -l {} \;
```

---

▶ 9

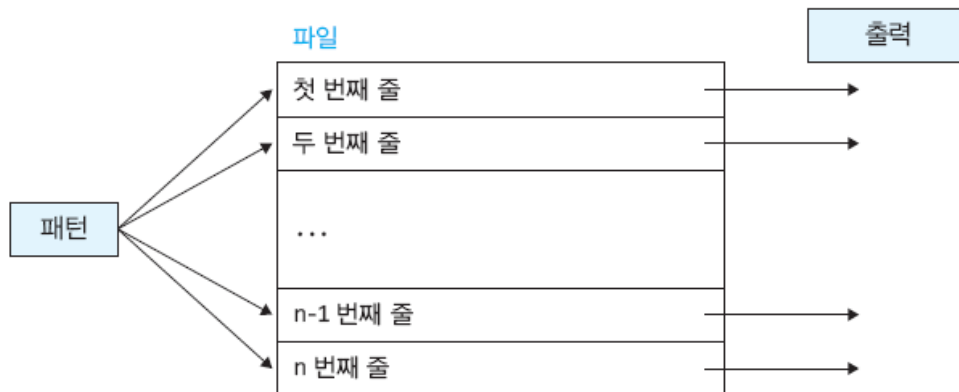
### 8.2 파일 필터링

## grep 명령어

- 사용법

```
$ grep 패턴 파일*
```

파일들을 대상으로 지정된 패턴의 문자열을 검색하고 해당 문자열을 포함하는 줄들을 출력한다



▶ 11

## grep 명령어

- \$ grep with you.txt

```
Until you come and sit awhile with me
There is no life - no life without its hunger;
But when you come and I am filled with wonder,
```

- \$grep -w with you.txt

```
Until you come and sit awhile with me
But when you come and I am filled with wonder,
```

- \$grep -n with you.txt

```
4:Until you come and sit awhile with me
15:There is no life - no life without its hunger;
17:But when you come and I am filled with wonder,
```

▶ 12

## grep 명령어의 옵션

옵션	기능
-i	대소문자를 무시하고 검색한다
-l	해당 패턴이 들어있는 파일 이름을 출력한다
-n	각 줄의 줄번호도 함께 출력한다
-v	명시된 패턴을 포함하지 않는 줄을 출력한다
-c	패턴과 일치하는 줄 수를 출력한다
-w	패턴이 하나의 단어로 된 것만 검색한다

▶ 13

## grep 명령어

- `$grep -i when you.txt`  
When I am down and, oh my soul, so weary  
When troubles come and my heart burdened be  
I am strong, when I am on your shoulders  
But when you come and I am filled with wonder,
- `$grep -v raise you.txt`  
When I am down and, oh my soul, so weary  
When troubles come and my heart burdened be  
Then, I am still and wait here in the silence  
Until you come and sit awhile with me  
I am strong, when I am on your shoulders  
There is no life - no life without its hunger;  
Each restless heart beats so imperfectly;  
But when you come and I am filled with wonder,  
Sometimes, I think I glimpse eternity

▶ 14





## 파이프와 함께 grep 명령어 사용

---

- 파이프와 함께 grep 명령어 사용
  - 어떤 명령어를 실행하고 그 실행 결과 중에서 원하는 단어 혹은 문자열 패턴을 찾고자 할 때 사용함.
- 예

```
$ ls -l | grep chang
$ ps -ef | grep chang
```

---

▶ 17

### 8.3 파일 정렬

## 정렬: sort 명령어

- 사용법

```
$ sort [-옵션] 파일*
```

텍스트 파일들의 내용을 줄 단위로 정렬한다. 옵션에 따라 다양한 형태로 정렬한다.

- 정렬 방법

- 정렬 필드를 기준으로 줄 단위로 오름차순으로 정렬한다.
- 기본적으로는 각 줄의 첫 번째 필드가 정렬 필드로 사용된다.
- -r 옵션을 사용하여 내림차순으로 정렬할 수 있다.

▶ 19

## sort 명령어 예

```
$ sort you.txt
```

```
But when you come and I am filled with wonder,  
Each restless heart beats so imperfectly;  
I am strong, when I am on your shoulders  
Sometimes, I think I glimpse eternity  
Then, I am still and wait here in the silence  
There is no life - no life without its hunger;  
Until you come and sit awhile with me  
When I am down and, oh my soul, so weary  
When troubles come and my heart burdened be  
You raise me up, so I can stand on mountains  
You raise me up, to more than I can be  
You raise me up, to walk on stormy seas
```

▶ 20

## sort 명령어 예

```
$ sort -r you.txt
```

```
You raise me up, to walk on stormy seas
You raise me up, to more than I can be
You raise me up, so I can stand on mountains
When troubles come and my heart burdened be
When I am down and, oh my soul, so weary
Until you come and sit awhile with me
There is no life - no life without its hunger;
Then, I am still and wait here in the silence
Sometimes, I think I glimpse eternity
I am strong, when I am on your shoulders
Each restless heart beats so imperfectly;
But when you come and I am filled with wonder,
```

▶ 21

## 정렬 필드 지정

필드 지정	기능
-k 필드번호	필드번호에 해당하는 필드를 기준으로 정렬한다 <sup>2)</sup> 이 옵션에서 필드번호는 1부터 시작된다 <sup>2)</sup>
+시작필드 -종료필드	시작필드부터 종료필드-1까지의 필드들을 기준으로 정렬한다 <sup>2)</sup> 이 때 필드 번호는 1부터 시작된다 <sup>2)</sup>

▶ 22

## 정렬 필드 지정 예

```
$ sort -k 3 you.txt 혹은 sort +2 -3 you.txt
```

Then, I am still and wait here in the silence

When I am down and, oh my soul, so weary

Until you come and sit awhile with me

When troubles come and my heart burdened be

Each restless heart beats so imperfectly;

You raise me up, so I can stand on mountains

You raise me up, to more than I can be

You raise me up, to walk on stormy seas

There is no life - no life without its hunger;

I am strong, when I am on your shoulders

Sometimes, I think I glimpse eternity

But when you come and I am filled with wonder,

▶ 23

## sort 명령어의 옵션

옵션	기능
-b	앞에 붙는 공백은 무시한다
-c	정렬이 되지 않은 상태로 출력한다
-d	숫자문자공백만 비교하여 사전식 순서로 정렬한다
-f	대소문자를 구분하지 않고 정렬한다
-n	숫자 문자열의 숫자 값에 따라 비교하여 정렬한다
-r	역순내림차순으로 정렬한다
-t 문자	지정한 문자를 필드 구분자로 사용한다

▶ 24

## sort 명령어의 옵션 예

---

- -o 출력파일 옵션

- 정렬된 내용을 지정된 파일에 저장할 수 있다.

```
$ sort -o sort.txt you.txt
```

- -n 옵션

- 숫자 문자열의 경우에 숫자가 나타내는 값의 크기에 따라 비교하여 정렬할 수 있다.
- 예: "49"와 "100"

---

▶ 25

## 필드 구분 문자 지정

---

```
$ sort -t: -k 3 -n /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
...
```

---

▶ 26

## 8.4 파일 비교

27

### 파일 비교: cmp 명령어

- 사용법

```
$ cmp 파일1 파일2  
파일1과 파일2가 같은지 비교한다
```

- 출력

- 두 파일이 같으면 아무 것도 출력하지 않음.
- 두 파일이 서로 다르면 서로 달라지는 위치 출력

- 예

```
$ cmp you.txt me.txt  
you.txt me.txt 다름: 340 자, 10 행
```

## 파일 비교: diff

- 사용법

```
$ diff [-i] 파일1 파일2
```

파일1과 파일2를 줄 단위로 비교하여 그 차이점을 출력한다

-i 옵션은 대소문자를 무시하여 비교한다

- 출력

- 첫 번째 파일을 두 번째 파일 내용과 같도록 바꿀 수 있는 편집 명령어 형태

▶ 29

## diff 출력: 편집 명령어

- 추가(a)

첫 번째 파일의 줄 n1 이후에 두 번째 파일의 n3부터 n4까지의 줄들을 추가하면 두 파일은 서로 같다.

```
n1 a n3,n4
```

> 추가할 두 번째 파일의 줄들

- 예

```
$ diff you.txt me.txt
```

```
9a10,13
```

>

> You raise me up, so I can stand on mountains

> You raise me up, to walk on stormy seas

> I am strong, when I am on your shoulders

▶ 30

## diff 출력: 편집 명령어

---

- 삭제(d)

첫 번째 파일의 n1부터 n2까지의 줄들을 삭제하면 두 번째 파일의 줄 n3 이후와 서로 같다.

**n1,n2 d n3**

< 삭제할 첫 번째 파일의 줄들

- 예

```
$ diff me.txt you.txt
```

**10,13d9**

<

< You raise me up, so I can stand on mountains

< You raise me up, to walk on stormy seas

< I am strong, when I am on your shoulders

---

▶ 31

## diff 출력: 편집 명령어

---

- 변경(c)

첫 번째 파일의 n1부터 n2까지의 줄들을 두 번째 파일의 n3부터 n4까지의 줄들로 대치하면 두 파일은 서로 같다.

**n1,n2 c n3,n4**

< 첫 번째 파일의 대치될 줄들

--

> 두 번째 파일의 대치할 줄들

- 예

```
$ diff 파일1 파일2
```

**1 c 1**

< This is the first file

--

> This is the second file.

---

▶ 32



## 8.5 기타 파일 조작

33

### 파일 분할 : split

- 사용법

```
$ split [-l n] 입력파일 [출력파일]
```

하나의 입력파일을 일정한 크기의 여러 개 작은 파일들로 분할한다. 옵션을 이용하여 분할할 줄 수를 지정할 수 있다.

- 1000줄씩 분할하여 xaa, xab, ... 형태의 파일명으로 저장

- 예

```
$ split -l 10 you.txt
```

```
$ ls -l
```

```
-rw-r--r-- 1 chang faculty 341 2월 16일 14:36 xaa
```

```
-rw-r--r-- 1 chang faculty 177 2월 16일 14:36 xab
```

```
-rw-r--r-- 1 chang faculty 518 2월 15일 19:33 you.txt
```

## 파일 합병: cat

- cat 명령어를 이용한 파일 합병

```
$ cat 파일1 파일2 > 파일3
```

파일<sub>1</sub>과 파일<sub>2</sub>의 내용을 붙여서 새로운 파일<sub>3</sub>을 만들어 준다<sub>??</sub>

- 예

```
$ cat xaa xab > xmerge
```

▶ 35

## 파일 합병: paste

- paste 명령어를 이용한 줄 단위 파일 합병

```
$ paste [ -s ] [ -d구분문자 ] 파일*
```

여러 파일들을 줄 단위로 합병하여 하나의 파일을<sub>?</sub>만들어 준다<sub>??</sub>

<sub>????</sub>한 파일 끝에 다른 파일 내용을 덧붙인다<sub>?</sub>

- 예

```
$ paste -s xaa xab > xmerge
```

▶ 36

## 파일 합병: paste 예

---

- `line.txt`  
line 1:  
line 2:  
...  
line 13:  
line 14:
- `$ paste line.txt you.txt`  
line 1: When I am down and, oh my soul, so weary  
line 2: When troubles come and my heart burdened be  
...  
line 13: But when you come and I am filled with wonder,  
line 14: Sometimes, I think I glimpse eternity
- `$ paste line.txt you.txt > lineyou.txt`

---

▶ 37

## 핵심 개념

---

- `find` 명령어는 파일 이름이나 속성을 이용하여 해당하는 파일을 찾는 데 사용된다.
- `grep` 명령어는 파일들을 대상으로 지정된 패턴의 문자열을 검색하고, 해당 문자열을 포함하는 줄들을 출력한다
- `sort` 명령어는 텍스트 파일을 줄 단위로 정렬한다.
- `cmp` 명령어는 두 파일이 같은지 비교한다.
- `diff` 명령어는 두 파일이 서로 다른지 비교한다.

---

▶ 38