

---

# 유용한 명령어들

# 파일탐색(grep)

## ■ "grep"(get regular expression) 명령

- ◆ 파일 내에서 특정 문자열을 검색하기 위해 사용
- ◆ 문자열이 발견되면 해당 행을 표시
- ◆ 형식

grep	
일반형식	grep [옵션] 패턴 파일

### ◆ 옵션

옵션	의미
-v(invert)	검색 문자열을 포함하는 행을 제외한 모든 라인 출력
-n(number)	검색 문자열을 포함하는 행들의 번호 표시
-c(count)	검색 문자열을 포함하는 행을 출력하지 않도록 함
-l(listfile)	특정한 패턴을 찾기 위해 여러 개의 파일 검색
-y	대문자, 소문자 구분을 하지 않음
-w	완전한 하나의 단어만을 찾고싶을 때 지정

# 파일탐색(grep)

## ■ 정규식 사용하기

- ◆ 필요에 따라 약간 더 복잡한 검색 명령어를 사용할 필요로 하는 것을 정규식이라 함
- ◆ 메타문자는 문자들이 셸에서 사용하는 것과 다르고 특별한 의미를 갖음("^", "\$")
- ◆ "."(dot)는 임의의 한 문자와 일치
- ◆ 원래 "."의 의미를 제거하기 위해서는 앞에 "/"를 넣어 줌
- ◆ 문자열의 범위를 나타내기 위해서는 괄호"[]"와 "-"을 사용

## ■ 파이프 라인 이용

- ◆ **grep** 명령은 또한 파이프라인에 많이 사용
- ◆ 특정한 정보를 필터링
- ◆ 이 "**grep**" 명령은 단독으로 쓰이지 않고 다른 명령을 실행한 후 원하는 정보를 보기 위하여 파이프라인을 이용

# 파일탐색(grep)

## ■ 사용 예

A terminal window titled 'cprog2@seps1:~/temp' showing a series of commands and their outputs. The commands demonstrate various grep options: 'cat', 'grep', 'grep -n', 'grep -c', 'grep -l', and 'grep' with regular expressions. The output shows the contents of 'name.txt' and the results of the grep searches.

```
cprog2@seps1:~/temp
[cprog2@seps1 temp]$ cat name.txt
1 hong gil dong 896-1728 korea
2 tom cruise 500-5555 usa
3 jang gil san 895-0038 korea
4 sung ryong 111-1111 china
[cprog2@seps1 temp]$ grep korea name.txt
1 hong gil dong 896-1728 korea
3 jang gil san 895-0038 korea
[cprog2@seps1 temp]$ grep -n korea name.txt
1:1 hong gil dong 896-1728 korea
3:3 jang gil san 895-0038 korea
[cprog2@seps1 temp]$ grep -c gil name.txt
2
[cprog2@seps1 temp]$ grep -l korea *
name.txt
name1.txt
[cprog2@seps1 temp]$ grep "^4" name.txt
4 sung ryong 111-1111 china
[cprog2@seps1 temp]$ grep "[a-f]" name.txt
1 hong gil dong 896-1728 korea
2 tom cruise 500-5555 usa
3 jang gil san 895-0038 korea
4 sung ryong 111-1111 china
[cprog2@seps1 temp]$ 
[영어][완성][두벌식]
```

# 파일비교(diff, cmp, comm)

## ■ “diff” 명령

◆ 두 파일의 차이점을 출력

◆ 형식

diff	
일반형식	diff [옵션] from-file to-file

◆ 옵션

옵션	의미
-u	통일된 출력 형식을 사용
-N	비교하는 디렉토리에 파일이 하나만 있더라도 모두 있는 것처럼 처리
-r(recursive)	서브디렉토리까지 모두 비교

# 파일비교(diff, cmp, comm)

## ■ “diff” 명령

### ◆ 파일 변경을 표시하는 3가지 방법

옵션	의미
a(append, add)	2번째 파일에 나타난 결과를 얻기 위하여 첫 번째 파일에 첨가되는 라인을 표시
d(delete)	2번째 파일로부터 삭제되는 라인
c(change)	첫 번째와 두 번째 파일사이에 변경된 라인

## ■ "cmp" 명령 - 두 파일이 서로 같은가를 결정하며, 텍스트 파일에서는 문자와 문자를 비교

## ■ "comm" - 두 파일의 같은 점을 찾음

# 파일비교(diff, cmp, comm)

## ■ 사용 예



```
cprog2@seps1:~/temp
[cprog2@seps1 temp]$ cat name1.txt
1 hong gil dong 896-1728 korea
2 tom cruise 500-5555 usa
3 chang gil san 895-0038 korea
4 sung ryong 111-1111 china
5 bruce lee 222-2222 china
[cprog2@seps1 temp]$ diff name.txt name1.txt
3c3
< 3 jang gil san 895-0038 korea
---
> 3 chang gil san 895-0038 korea
4a5
> 5 bruce lee 222-2222 china
[cprog2@seps1 temp]$ cmp name.txt name1.txt
name.txt name1.txt differ: byte 60, line 3
[cprog2@seps1 temp]$ comm name.txt name1.txt
      1 hong gil dong 896-1728 korea
      2 tom cruise 500-5555 usa
      3 chang gil san 895-0038 korea
3 jang gil san 895-0038 korea
      4 sung ryong 111-1111 china
      5 bruce lee 222-2222 china
[cprog2@seps1 temp]$
```

[영어][완성][두벌식]

# 파일패치(patch)

## ■ “patch” 명령

◆ 변경된 파일에 대한 **diff** 파일을 이용하여 원래 파일을 패치하면 변경된 새로운 파일이 생성

### ◆ 형식

patch	
일반형식	patch [옵션] 원래파일 패치파일 patch -p숫자 < 패치파일

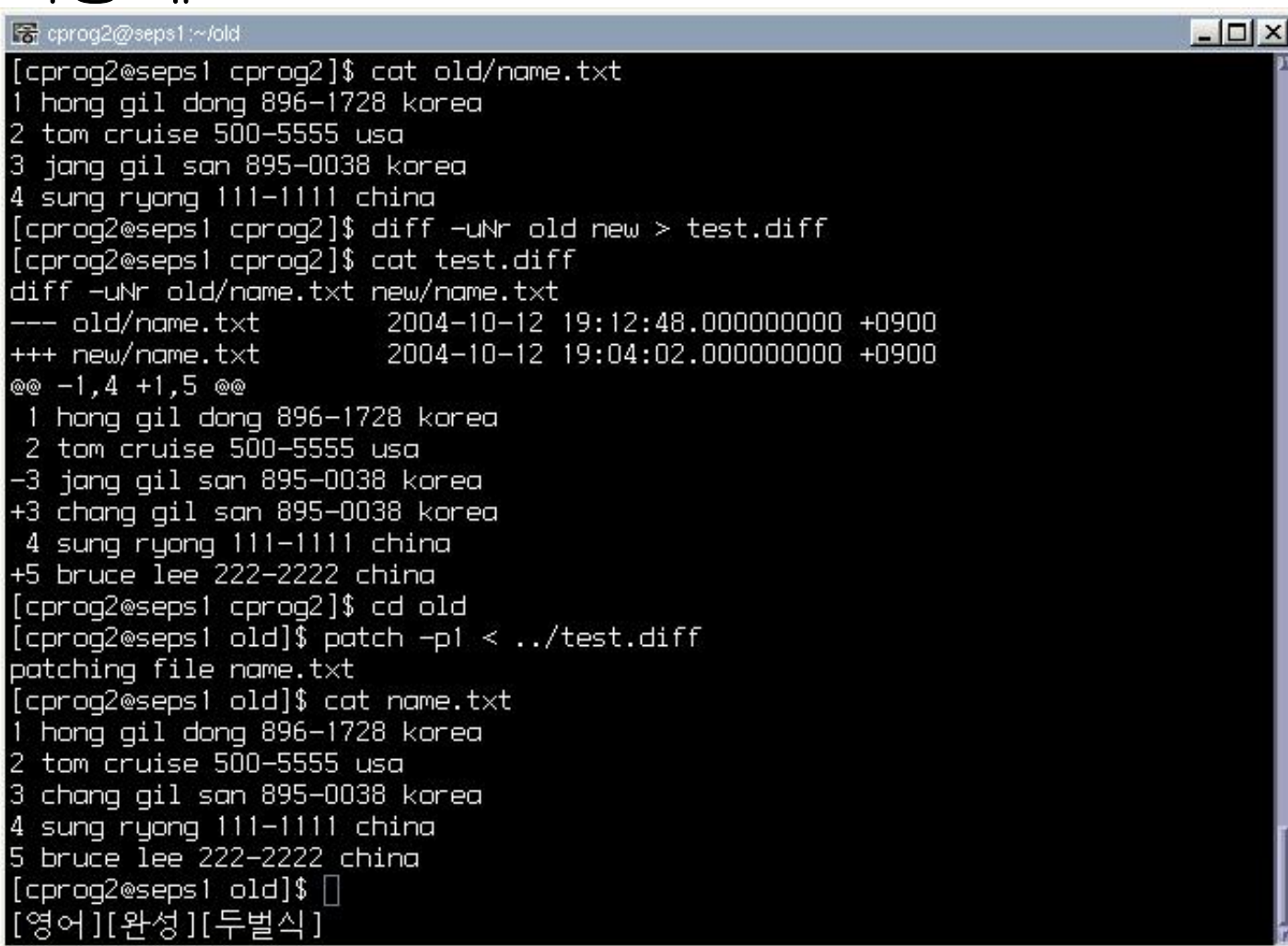
### ◆ 옵션

옵션	의미
-b	백업파일 생성
-R(--reverse)	패치를 역으로 적용
-p숫자	숫자 수준까지 경로명에서 prefix 를 제거함



# 파일패치(patch)

## ■ 사용 예



```
cprog2@seps1:~/old
[cprog2@seps1 cprog2]$ cat old/name.txt
1 hong gil dong 896-1728 korea
2 tom cruise 500-5555 usa
3 jang gil san 895-0038 korea
4 sung ryong 111-1111 china
[cprog2@seps1 cprog2]$ diff -uNr old new > test.diff
[cprog2@seps1 cprog2]$ cat test.diff
diff -uNr old/name.txt new/name.txt
--- old/name.txt      2004-10-12 19:12:48.000000000 +0900
+++ new/name.txt      2004-10-12 19:04:02.000000000 +0900
@@ -1,4 +1,5 @@
 1 hong gil dong 896-1728 korea
 2 tom cruise 500-5555 usa
-3 jang gil san 895-0038 korea
+3 chang gil san 895-0038 korea
 4 sung ryong 111-1111 china
+5 bruce lee 222-2222 china
[cprog2@seps1 cprog2]$ cd old
[cprog2@seps1 old]$ patch -p1 < ../test.diff
patching file name.txt
[cprog2@seps1 old]$ cat name.txt
1 hong gil dong 896-1728 korea
2 tom cruise 500-5555 usa
3 chang gil san 895-0038 korea
4 sung ryong 111-1111 china
5 bruce lee 222-2222 china
[cprog2@seps1 old]$
```

[영어][완성][두벌식]

# 파일검색(find)

## ■ “find” 명령

◆ 경로 이름에 지정된 디렉토리와 그 아래의 서브디렉토리를 검색하여 표현식에 지정된 파일을 찾음

◆ 형식

find	
일반형식	find [경로...] [표현식]

◆ 옵션

옵션	의미
-name 패턴	파일 이름(정규표현식) 으로 검색
-size 크기	일정 크기 이상의 파일 검색
-mtime	파일이 갱신한 시간으로 파일 검색
-exec 명령 ;	검색한 파일에 대하여 특정 명령을 수행
-ok 명령 ;	-exec 와 비슷하지만, 사용자에게 명령 수행에 대해 질의(y, n)
-print	전체 파일 이름을 표준 출력으로 출력

# 파일검색(find)

## ■ 기능

- ◆ 표현식에는 특정 파일의 이름이나 복수개의 파일을 지정하는 패턴 또는 파일의 속성(크기, 수정시간) 그리고 여러 가지 속성을 조합한 논리식이 지정될 수 있음
- ◆ 표현식과 일치하는 파일들에 대해 파일의 이름을 출력할 수도 있고 특정한 명령을 실행시킬 수도 있음
- ◆ 현재의 경로 이름을 지정하기 위하여 "{}"를 사용

## ■ 검색조건을 결합할 때 사용하는 논리연산자

논리연산자	의미
논리부 (NOT)	부정에는 "!" 연산자를 사용 예) ".c"로 끝나지 않는 모든파일 이름 -> <b>! -name "*.c"</b>
논리곱 (AND)	논리곱의 경우에는 별도의 연산자 없이 검색조건 나열 예) 크기가 10 블럭 이상이고 링크가 3개인 파일 -> <b>-size +10 link 3</b>
논리합 (OR)	-o 연산자를 사용 예) 크기가 10블럭을 넘거나 링크가 3개인 파일 -> <b>-size +10</b>

# 파일검색(find)

## ■ 사용 예

```
cprog2@seps1:~/temp
[cprog2@seps1 temp]$ ls -la
합계 84
drwxrwxr-x    2 cprog2  cprog2    4096 10월 12 18:36 .
drwx-----  11 cprog2  cprog2    4096 10월 12 18:32 ..
-rw-rw-r--    1 cprog2  cprog2     115 10월 12 18:26 name.txt
-rw-rw-r--    1 cprog2  cprog2     143 10월 12 18:32 name1.txt
-rw-rw-r--    1 cprog2  cprog2      42 9월 23 11:28 test.txt
-rw-rw-r--    1 cprog2  cprog2   58029 10월 12 18:36 utilities-1.jpg
[cprog2@seps1 temp]$ find . -name "*.txt" -print
./name.txt
./name1.txt
./test.txt
[cprog2@seps1 temp]$ find . -size +1 -print
.
./utilities-1.jpg
[cprog2@seps1 temp]$ find . -name "test*" -print -exec cat {} \;
./test.txt
안녕하세요 지금은 vi를 사용하고 있습니다.
[cprog2@seps1 temp]$ find . \( -name "*.txt" -a -name "name*" \) -print
./name.txt
./name1.txt
[cprog2@seps1 temp]$
```

[영어][완성][두벌식]

# 파일압축과 묶음

## ■ 압축과 풀기(gzip,gunzip,bzip2,bunzip2, bzip2, bzip2, bzip2)

### ◆ gzip 옵션

옵션	의미
-d	압축을 푼다
-l	현재 압축된 파일의 내용을 보여준다
-r	현재 디렉토리부터 하위 디렉토리까지 전부를 압축한다.
-t	압축 파일의 완전성을 검사한다.
-v	압축 진행상황을 보여준다.
-9	최대한 압축한다.

### ◆ gzip 사용 예

➤ **\$ gzip -d aaa.txt.gz**

➤ **\$ gunzip bbb.txt.gz**

# 파일압축과 묶음

## ■ bzip2 옵션

옵션	의미
-d	압축을 해제한다.
-z	파일을 압축한다.
-t	압축 파일의 완전성을 검사한다.
-v	압축 진행 상황을 보여준다.

## ■ 사용 예

◆ `$ bzip2 -z aaa.txt`

◆ `$ bzip2 -d aaa.txt.bz2`

# 파일압축과 묶음

## ■ 파일 묶기 (tar)

옵션	의미
-c	새로운 아카이브 파일을 작성한다.
-t	묶음파일의 내용을 보여준다.
-x	묶음 파일을 해제한다.
-r	묶음 파일에 새로운 파일을 추가한다.
--delete	묶음 파일에서 특정 파일을 삭제한다.
-u	묶음 파일보다 새로운 파일이면 업데이트 한다.
-v	묶음과정을 보여준다.
-f	묶음 파일명을 지정한다.
-z	묶음실행과 동시에 gzip으로 동시에 압축한다.

## ■ 사용 예

- ◆ `$ tar cvf total.tar aaa.txt bbb.txt`
- ◆ `$ tar cvf total.tar 20_jang`
- ◆ `$ tar tvf total.tar`
- ◆ `$ tar xvf total.tar`
- ◆ `$ tar cvfz total.tar.gz *`

# 파일압축과 묶음

## ■ 도스용 압축유틸리티(rar)

◆ rar 유틸리티는 명령어(Command)와 스위치(Switch) 로 구성

◆ 사용 예

➤ \$ ./rar a total \*

➤ \$ ./rar t total.rar

➤ \$ ./rar x total.rar

## ■ 기타 압축 유틸리티(compress)

◆ 사용 예

➤ \$ compress \*

➤ \$ zcat bbb.txt.Z

➤ \$ uncompress bbb\*