

```

1  Ubuntu의 검색 명령어
2
3  -전통적인 리눅스와 유닉스 문서들은 모두 man pages 라고 하는 매뉴얼 페이지로 만들어져 있다.
4  -이후 GNU에 의해 좀더 정교한 문서화 방법이 제안되었다.
5
6  1. help 이용하기
7      1)리눅스 시스템의 거의 모든 명령어들은 자체적으로 대략적인 사용 정보를 담고 있다.
8      2)보통 명령어와 -h 또는 --help 옵션 사용
9          $ ls --help
10
11
12  2. man pages
13      1)man pages sections
14          -(1) : 실행 가능한 프로그램이나 셸 명령어
15          -(2) : 시스템 콜(커널에서 제공하는 기능)
16          -(3) : 라이브러리 콜(프로그램 라이브러리 기능)
17          -(4) : 특별한 파일들(일반적으로 /dev/ 디렉토리에 있는 파일)
18          -(5) : 파일 포맷과 /etc/passwd와 같은 파일 명명규칙
19          -(6) : 게임
20          -(7) : 그 외의 여러 가지(매크로 패키지와 명명규칙 등을 포함)
21          -(8) : 시스템 관리 명령어(일반적으로 root 유저를 위한 명령어)
22          -(9) : 커널 루틴(비 표준)
23
24      2)명령어 옵션
25          -$ man -a uname : uname에 대한 모든 맨 페이지 섹션을 보여줌
26          -$ man 2 uname : uname에 대한 섹션 2의 맨 페이지를 보여줌
27          -$ man uname -P more : uname 맨 페이지를 more 명령어를 사용하여 페이지 단위로 보여줌
28          -$ man -f uname : whatis 명령어 수행결과와 동일(키워드와 동일한 맨 페이지만 출력)
29          -$ man -k crontab : apropos 명령어 수행결과와 동일(키워드가 포함된 맨 페이지 출력)
30          -$ man -S 2 uname : 검색할 매뉴얼 섹션들을 콜론으로 구분하여 지정
31
32
33  3. info
34      1)경우에 따라 개발자들은 info 데이터베이스에 명령어, 파일 포맷, 디바이스 또는 다른 리눅스 컴퓨넌트에
35      대한 좀 더 자세한 설명, 예를 들어 온라인 매뉴얼 페이지에 대한 링크 정보 등을 입력해 놓는다.
36      2)단순히 info 명령어를 입력하거나, 특정 컴포넌트를 열어 info 데이터베이스에 접근할 수 있다(q를
37      사용하면 종료)
38      3)ls 명령어에 대한 info 정보 :
39          $ info ls
40
41      4)이동 : Page Up, Page Down 상,하,좌,우 방향키를 이용.
42      5)info 데이터베이스가 사용하는 파일 : /usr/share/info디렉토리에 저장.
43      6)info 화면에서 이동하기
44          -? : info 하에서 사용되는 기본적인 명령들을 보여줌
45          -Shift + I : 현재 화면에서 바로 이전 노드로 이동
46          -n, p, u : 순서대로 다음, 이전, 그리고 위 노드로 이동
47          -Enter : 커서가 있는 곳에 하이퍼링크를 따라 이동
48          -Shift + r : 참조 위치를 따라감
49          -q or Shift + q : info 유틸리티를 종료하고 빠져나감
50
51  4. 리눅스에서 명령을 입력하면 셸은 프로그램에서 사용하는 디렉토리를 검색해 처음 이름이 일치하는
52  프로그램을 실행한다.
53      1)이때 검색하는 디렉토리 목록을 검색 경로(search path)라고 한다.
54      2)검색 경로를 변경하는 방법은 PATH 환경변수를 변경하는 것이다.
55      3)사용자가 검색 경로를 변경하지 않으면 셸은 표준 디렉토리만 검색해 보고 검색 작업을 멈추게 된다.
56      4)물론 표준 디렉토리 외의 다른 디렉토리에도 유용한 유틸리티가 존재할 수 있다.
57
58  5. which : 유틸리티의 위치 검색

```

- 1)which 유틸리티는 유틸리티 파일의 전체 경로를 보여줌으로써 유틸리티(명령어)의 위치를 알려준다.
- 2)셸은 검색 경로를 탐색해서 처음 나오는 명령을 실행한다.
- 3)which 는 이런 경우 셸이 어떤 프로그램을 실행할지를 알려준다.
- 4)which유틸리티는 명령어가 예상 외의 동작을 수행할 경우 유용하다.
- 5)which를 실행 해 보면 표준이 아닌 버전의 툴을 실행하고 있는지, 혹은 자신이 원하는 툴과는 전혀 다른 툴을 실행하고 있는지를 알 수 있게 될 것이다.
- 6)프로그램의 이름이 주어지면 which는 검색 경로에 있는 디렉토리를 차례대로 검색, 프로그램의 위치를 찾는다.
- 7)검색 경로에 해당 프로그램이 하나 이상 존재하는 경우에 which는 처음 나오는 명령에 대한 정보만 보여준다.

```
$ which ls
$ which tar
```

6. whereis : 유틸리티의 위치 검색

- 1)표준 디렉토리(standard directory) 목록을 통해 파일을 검색하며, 검색 경로와는 독립적으로 작동한다.
- 2)whereis는 사용자가 지정한 바이너리(실행)파일, 매뉴얼 페이지 및 프로그램 소스 코드의 위치 검색에 사용할 수 있는 데, 이 유틸리티는 which와 달리 모든 해당 파일에 대한 정보를 보여준다.

```
$ whereis ls
$ whereis tar
$ which diff vs whereis diff
```

3)실행 파일의 위치만

```
$ whereis -b diff
```

4)매뉴얼 페이지만 검색

```
$ whereis -m
```

5)Source code 파일에만

```
$ whereis -s
```

6)위치를 확인

```
$ whereis -l(소문자 l)
```

7)검색할 때 특정 위치로 제한 : -B옵션 사용

```
$ whereis perl
$ whereis -B /bini/ -f perl
-f 옵션은 마지막에 경로를 알리는데 사용
```

7. apropos: 키워드 검색

- 1)특정 작업을 수행하는 데 필요한 명령을 알지 못한다면 apropos 유틸리티로 키워드를 검색하면 된다.
- 2)apropos를 제대로 실행하려면 makewhatis로 whatis 데이터베이스를 설정하고 주기적으로 관리해야 한다.
 - 이 작업은 보통 cron으로 처리한다.
- 3)이 유틸리티는 man 페이지의 맨 위에 나오는 짧은 설명줄(description line)에서 키워드를 검색해 일치하는 항목이 있는 명령만 보여준다.
- 4)man 유틸리티를 -k(keyword) 옵션과 함께 사용하면 apropos를 실행한 것과 동일한 결과를 볼 수 있다.
- 5)다음 예제는 apropos를 who 키워드와 함께 호출한 결과를 보여준다.
 - 이 키워드와 관련된 다른 유용한 유틸리도 찾아서 보여준다.

```
$ apropos who
```

6)둘 이상의 검색어 사용

```
$ apropos chown chmod
```

8. locate: 파일 검색

- 1)locate유틸리티는 로컬 시스템에서 파일을 검색하는 작업을 수행한다.
- 2)locate는 updatedb 유틸리티로 생성/업데이트한 locate 데이터베이스가 있어야만 정확하게 검색할 수 있다.

3)일반적으로 이 데이터베이스는 cron 스크립트로 하루에 한번씩 업데이트 된다.

4)기본적으로 Ubuntu 20.04에는 locate가 설치되어 있지 않다.

```
- $ sudo apt install -y mlocate
```

5)Ubuntu에서는 locate와 mlocate가 상호 교환 가능하다.

```
- $ locate --version
```

```
mlocate 0.26
```

6)가장 큰 장점은 속도이다.

-find 명령은 파일 시스템 전체에서 검색을 수행한다.

-locate는 데이터베이스 검색을 수행하여 찾고 있는 것이 컴퓨터에 있는지 확인한다.

-데이터베이스를 최신 상태로 유지하는 것은 mlocate가 설치되어 있는 경우 보통 cron.daily. 매일 아침 실행되어 업데이트 된다.

-확인을 위해 다음의 명령을 수행한다.

```
$ ls /etc/cron.daily/*loc*  
/etc/cron.daily/mlocate
```

-수동으로 데이터베이스 업데이트 하려면

```
$ sudo updatedb
```

7)원하는 결과수를 알려준다.

- -n 옵션을 사용하면 결과수를 제한할 수 있다.

```
$ locate .html -n 10
```

-결과를 10개로 제한

8)일치하는 파일 수 계산

```
$ locate -c .html
```

```
7095
```

-컴퓨터에 .html 파일이 7095개 있다는 뜻.

9)대소문자 구별

```
$ locate -c .HTML
```

-이렇게 검색하면 결과가 0이 나옴

-대소문자 구별 무시하려면 -i 옵션 사용

```
$ locate -c -i .HTML
```

10)locate 데이터베이스 상태

-데이터베이스 상태를 보려면 -S(대문자)

-locate 데이터베이스의 크기와 내용에 관한 통계를 보여준다.

```
$ locate -S
```

```
Database /var/lib/mlocate/mlocate.db:
```

```
42,695 directories
```

```
480,374 files
```

```
33,377,394 bytes in file names
```

```
12,276,036 bytes used to store database
```

9. grep: 파일명 검색 또는 파일내 문자열 검색

1)which / whereis / apropos 등은 유틸리티 검색 하는데 사용되지만, grep 유틸리티는 하나 이상의 파일에서 행 단위로 간단한 문자열이나 정규 표현식으로 표현된 pattern을 찾는다.

2)-r 옵션은 하위디렉토까지 모두 검색한다.

3)-i 옵션은 파일내의 텍스트와 패턴이 매치되는 텍스트를 출력하지 말고 매칭된 파일 목록만 출력한다.

4)-L 옵션은 -i 옵션과 반대로 비 매칭된 파일 목록을 출력한다.

10. whatis

1)매뉴얼 페이지를 빠르게 검색하는데 사용

2)검색 요청한 용어에 대한 한 줄 요약 설명 제공

3)만일 man으로 검색할 때

```
$ whatis man
```

```

169     man (1)
170     man (7)
171
172 4)man 1로 검색
173     $ man 1 man
174
175 5)man 7로 검색
176     $ man 7 man
177
178 6)wild card 와 함께 사용
179     - -w 옵션 사용
180     -$ whatis -w char*
181
182
183 11. find
184     1)find는 리눅스에서 파일 및 디렉토리를 검색할 때 사용하는 명령
185     2)이름 그대로 리눅스에서 접근할 수 있는 파일 시스템에서, 파일 및 디렉토리를 찾는다.
186     3)그런데 리눅스 파일 시스템은 구조가 아주 복잡하다.
187     4)아주 많은 수의 파일과 디렉토리가 저장되어 있고, 이 중에는, 비슷한 이름의 파일과 동일한 확장자를 가진
188     파일, 그리고 종류가 다양한 파일들이 존재한다.
189     5)이에 따라, 복잡한 구조에서 원하는 파일을 가능한 정확히 찾을 수 있도록, find 명령에는 다양한 옵션이
190     존재한다.
191     6)명령어와 옵션
192         -$ find --help를 통해 확인
193
194     7)현재 디렉토리에 있는 파일 및 디렉토리 리스트 표시
195         -아무런 옵션없이 find 명령만 사용하면 현재 디렉토리(하위 디렉토리 포함)에 있는 모든 파일과
196         디렉토리를 표시.
197         -주로 디렉토리 파일 리스트를 다른 명령으로 전달하여 처리하고자 할 때 사용한다.
198         -$ find                # 현재 디렉토리의 파일 및 디렉토리 출력
199         -$ find -print0        # 현재 디렉토리의 모든 내용을 줄 바꿈 없이 출력.
200
201     8)대상 디렉토리에 있는 파일 및 디렉토리 리스트 표시
202         -find 명령 다음에 디렉토리를 지정하여, 대상 디렉토리에 있는 파일 및 디렉토리 리스트를 표시.
203         -$ find [PATH]        # PATH에 있는 파일 및 디렉토리 리스트 표시.
204
205     9)현재 디렉토리 아래 모든 파일 및 하위 디렉토리에서 파일 검색
206         -현재 디렉토리에 포함된 모든 하위 디렉토리 및 파일에서 지정된 파일을 검색하려면 아래 명령을
207         수행하면 된다.
208         -$ find . -name [NAME]    # 현재 디렉토리 아래 모든 파일 및 디렉토리 검색.
209
210     10)전체 시스템(루트 디렉토리)에서 파일 검색
211         -디렉토리 경로를 루트(/)로 지정하여 전체 시스템 중 특정 파일을 검색할 수 있다.
212         -$ find / -name [NAME]    # 루트 디렉토리에서 파일 이름으로 검색.
213
214     11)파일 이름이 특정 문자열로 시작하는 파일 검색
215         -찾을 파일 이름에 와일드 카드(*) 문자를 포함하여 특정 문자열로 시작하는 파일을 검색할 수 있다.
216         -$ find . -name "STR*"    # 지정된 문자열로 시작하는 파일 검색.
217
218     12)파일 이름에 특정 문자열이 포함된 파일 검색
219         -찾을 파일 이름 시작과 끝에 와일드 카드(*) 문자를 포함하여 특정 문자열이 포함된 파일을 검색할 수
220         있다.
221         -$ find . -name "*STR"    # 지정된 문자열이 포함된 파일 검색.
222
223     13)파일 이름이 특정 문자열로 끝나는 파일 검색 (파일 확장자로 검색)
224         -찾을 파일 이름 끝에 와일드 카드(*) 문자를 사용하여 특정 문자열로 끝나는 파일을 검색할 수 있다.
225         -확장자가 일치하는 파일을 검색할 때 유용합니다.
226         -$ find . -name "*STR"    # 지정된 문자열로 끝나는 파일 검색.
227
228     14)빈 디렉토리 또는 크기가 0인 파일 검색

```

```

224     "-empty" 표현식을 사용하여, 빈 디렉토리 또는 크기가 0인 파일을 검색할 수 있다.
225     -$ find . -empty # 빈 디렉토리 또는 크기가 0인 파일 검색.
226     -$ find . -name TMP -empty # TMP라는 이름의 빈 디렉토리 또는 크기가 0인 파일 검색.
227
228 15)특정 확장자를 가진 모든 파일 검색 후 삭제
229     "-delete" 표현식을 사용하여, 검색된 파일 및 디렉토리를 삭제할 수 있다.
230     -만약 검색 대상에 디렉토리가 포함된 경우, 디렉토리가 비어 있지 않으면 해당 디렉토리는 삭제되지
        않는다.
231     -$ find . -name "*.EXT" -delete # 확장자 검색 후 파일 삭제.
232
233 16)검색된 파일 리스트를 줄 바꿈 없이 이어서 출력하기
234     "-print0" 표현식을 사용하여, 검색 결과를 줄바꿈(newline) 없이 이어서 출력할 수 있다.
235     -$ find . -name [FILE] -print0 # 검색된 결과를 줄 바꿈 없이 출력.
236     -$ find . -name [FILE] -print # 검색된 결과를 줄 바꿈으로 구분하여 출력.
237
238 17)파일 또는 디렉토리만 검색하기
239     "-type" 표현식을 사용하여, 파일 종류를 지정하여 검색할 수 있다.
240     -b : block special
241     -c : character special
242     -d : directory
243     -f : regular file
244     -l : symbolic link
245     -p : FIFO
246     -s : socket
247     -$ find . -name build -type f # build라는 이름의 일반 파일 검색.
248     -$ find . -name PROJ -type d # PROJ라는 이름의 디렉토리만 검색
249
250 18)파일 크기를 사용하여 파일 검색
251     -파일 크기를 사용하여 파일을 찾으려면, "-size" 표현식을 사용한다.
252     -"-size" 표현식은 기본적으로 block 단위(512B)의 크기를 사용하게 되어 있는데, 파일 크기 값 뒤에
        'c' 또는 'k'를 붙임으로써 바이트 단위 또는 킬로바이트의 크기 값을 지정할 수 있다.
253     -(b:block, c:bytes, w:2bytes, k:kbytes, M:mbytes, G:gbytes)
254     -그리고 파일 크기를 비교할 때, "-size" 표현식 두 개를 조합하여 "초과" 또는 "미만"에 대한 내용도
        지정할 수 있다.
255     -이 때는 크기 값 앞에 +(초과) 또는 -(미만) 기호를 사용한다.
256     -예를 들어 1024 바이트보다 크고 2048바이트보다 작은 파일을 검색할 때, "-size +1024c -size
        -2048c" 표현식을 사용한다.
257     -$ find . -size 1024c # 파일 크기가 1024 바이트인 파일 검색.
258     -$ find . -size +1024c # 파일 크기가 1024 바이트를 초과하는 파일 검색.
259     -$ find . -size -1024c # 파일 크기가 1024 바이트 미만인 파일 검색.
260     -$ find . -size +1024c -size -2048c # 파일 크기가 1024 바이트 초과, 2048 바이트 미만인
        파일 검색
261
262 19)검색된 파일에 대한 상세 정보 출력. (find + ls)
263     "-exec" 표현식은 find 명령으로 검색된 결과를 사용해 다른 명령을 실행할 수 있게 만들어 준다.
264     -아래 예제는 find 명령으로 검색된 내용 ls -l 명령으로 전달하여 파일 상세 정보를 표시하게 만드는
        것이다.
265     -$ find . -name "*.c" -exec ls -l {} \;
266     --현재 디렉토리에서 c 확장자 파일 상세 정보 출력.
267
268 20)검색된 파일의 라인 수 출력. (find + wc)
269     -find 명령과 wc 명령(Word Count)을 조합하여 검색 파일의 문자 수 또는 라인 수를 계산하여 출력할
        수 있다.
270     -$ find . -name "*.c" -exec wc -l {} \;
271     --확장자가 c 인 파일의 라인 수 출력.
272
273 21)검색된 파일에서 문자열 검색. (find + grep)
274     -find 명령과 grep 명령을 조합하여, 검색된 파일로부터 특정 문자열을 찾을 수 있다.
275     -$ find . -name "*.c" -exec grep "main" {} \;
276     --.c 파일에 문자열 main이 있는지 검색.

```

```
277 -$ find . -name "*.java" -exec grep -n "class" {} \;  
278 --.java 파일에 문자열 class가 있는 라인 표시.  
279  
280 22)파일 검색 결과를 파일로 저장. (find, redirection)  
281 -find 명령의 결과 내용을 표준 출력이 아닌, 파일로 저장하려면 redirection(>)을 사용.  
282 -$ find . -name "*.c" > source-list.txt  
283 --모든 c 파일 검색 결과를 source-list.txt에 저장.  
284  
285 23)검색 중 에러 메시지 출력하지 않기. (find, redirection)  
286 -find 명령을 실행할 때, 실행 결과와 별개로 에러 메시지가 표시되는 경우가 있다.  
287 -특히 디렉토리 접근 권한 문제로 인해 "Permission denied" 에러 메시지가 화면을 가득채우게 되면,  
실제 검색 결과를 확인하기가 어렵게 된다.  
288 -이럴 때 redirection을 통해 에러 메시지를 화면에 표시되지 않게 만들 수 있다.  
289 -표준 에러(2)를 "/dev/null"로 redirection 시켜버리는 것이다.  
290 -$ find / -name "*.c" 2> /dev/null  
291 -- 에러 메시지를 표시하지 않음.  
292  
293 24)하위 디렉토리 검색하지 않기  
294 -기본적으로 find 명령은 하위 디렉토리까지 모두 탐색한다.  
295 -하지만 "-maxdepth" 표현식을 사용하면 검색할 하위 디렉토리의 깊이를 제한할 수 있다.  
296 -참고로 "-maxdepth"는 다른 표현식보다 앞에 사용해야 한다.  
297 -$ find / -maxdepth 1 -name "sys"  
298 --sys라는 파일을 루트(/) 디렉토리에서만 검색.  
299  
300 25)검색된 파일 복사 (find + cp)  
301 -find 명령과 cp 명령을 조합하여, 검색된 파일을 지정된 디렉토리로 복사할 수 있다.  
302 -$ find . -name "*.tar.gz" -exec cp {} /mnt/usb/ \;  
303 --tar.gz 파일을 /mnt/usb/ 에 복사.
```