

10-3. Linux 특수 문자

- 셸은 사용자가 더욱 편리하게 명령을 입력하고 실행할 수 있도록 다양한 특수 문자를 제공한다.
- 특수 문자의 종류와 사용법은 모든 셸에서 거의 비슷한데 여기서는 Bash Shell을 기준으로 설명한다.

특수문자 *

- *(별표)는 0개 이상의 문자를 대체한다.
- 명령을 실행할 때 파일 이름을 적어야 하는 자리에 *를 사용하면 모든 파일을 나타낸다.

ls *	현재 디렉터리의 모든 파일과 서브 디렉터리를 나열. 서브 디렉터리의 내용도 출력
cp * /tmp	현재 디렉터리의 모든 파일을 /tmp 디렉터리 아래로 복사
ls -F t*	t, tmp, temp와 같이 파일명이 t로 시작하는 모든 파일의 이름과 파일 종류를 출력. t도 해당
cp *.txt ../ch3	확장자가 txt인 모든 파일을 상위 디렉터리 밑의 ch3 디렉터리로 복사
ls -l h*d	파일명이 h로 시작하고 d로 끝나는 모든 파일의 상세정보를 출력. hd, had, hard, h12345d 등 이 조건에 맞는 모든 파일의 정보를 출력

특수 문자 ?와 []

- 모두 하나의 문자를 나타내는 데 사용된다.
- ?(물음표)는 길이가 1인 임의의 한 문자를, [](대괄호)는 괄호 안에 포함된 문자 중 하나를 나타낸다.
- []는 사용할 여러 문자를 나열하거나 범위를 지정할 수 있으며, 다른 특수 문자와 혼합하여 사용할 수 있다.

ls t?.txt	t 다음에 임의의 한 문자가 오고 파일 확장자가 txt인 모든 파일의 이름을 출력. t1.txt, t2.txt, ta.txt 등이 해당. 단 t.txt는 제외
ls -l tmp[135].txt	tmp 다음에 1, 3, 5 중 한 글자가 오고 파일 확장자가 txt인

	모든 파일의 이름을 출력. tmp1.txt, tmp3.txt, tmp5.txt 파일이 있으면 해당 파일의 상세 정보를 출력. 단 tmp.txt는 제외
ls -l tmp[1-3].txt	[1-3]은 1부터 3까지의 범위를 의미. 따라서 ls -l tmp[123].txt와 결과가 같다. tmp1.txt, tmp2.txt, tmp3.txt 파일이 있으면 해당 파일의 상세 정보를 출력
ls [0-9]*	파일명이 숫자로 시작하는 모든 파일 목록을 출력
ls [A-Za-z]*[0-9]	파일명이 영문으로 시작하고 숫자로 끝나는 모든 파일목록을 출력

특수 문자 ~와 -

- 모두 디렉터리를 나타내는 특수 문자이다.
- ~(물결표)만 사용하면 현재 작업 중인 사용자의 홈 디렉터리를 나타내고, 다른 사용자의 로그인 ID와 함께 사용하면 해당 사용자의 홈 디렉터리를 나타낸다.
- (볼임표)는 cd 명령으로 디렉터를 이전하기 직전의 작업 디렉터를 나타낸다.

cp *.txt ~/tmp	확장자가 txt인 모든 파일을 현재 작업 중인 사용자의 홈 디렉터리 아래 tmp 디렉터리로 복사
cp ~user2/linux.txt .	user2라는 사용자의 홈 디렉터리 아래에서 linux.txt 파일을 찾아 현재 디렉터리로 복사
cd -	이전 작업 디렉터리로 이동

특수 문자 ;과 |

- 명령과 명령을 연결한다.
- ;(쌍반점)은 연결된 명령을 왼쪽부터 차례로 실행하고, |(파이프)는 왼쪽 명령의 실행 결과를 오른쪽 명령의 입력으로 전달한다.

date; ls; pwd	날짜를 출력한 후 현재 디렉터리의 파일 목록을 출력하고, 마지막으로 현재 작업 디렉터리의 절대 경로를 나열
ls -al / more	루트 디렉터리에 있는 모든 파일의 상세 정보를 한 화면에 출력하나, ls -al / 명령의 결과가 more 명령의 입력으로 전달되어 페이지 단위로 출력

특수 문자 ‘ ‘와 “ “

- 문자를 감싸서 문자열로 만들어주고, 문자열 안에 사용된 특수 문자의 기능을 없앤다.
- ‘(작은 따옴표)는 모든 특수 문자를, “(큰 따옴표)는 \$, ", \를 제외한 모든 특수 문자를 일반문자로 간주하여 처리한다.

- 셸 변수에 저장된 값을 출력할 때 셸 변수 앞에 \$를 붙인다. \$를 문자 그대로 사용하려면 작은 따옴표를 쓰고, 셸 변수의 값으로 처리하려면 큰 따옴표를 써야 한다.

echo '\$SHELL'	\$SHELL 문자열이 화면에 출력
echo "\$SHELL"	셸 환경변수인 SHELL에 저장된 값인 현재 셸의 종류가 화면에 출력. 예를 들면 /bin/bash

특수 문자 ``

- `` (백쿼터, 백쿼테이션)은 키보드 상단의 숫자 1 키 앞에 있는 키를 사용한다.
- 셸은 ``로 감싸인 문자열을 명령으로 해석하여 명령의 실행 결과로 바뀌게 된다.

echo "Today is `date`"	'date'는 명령으로 해석되어 date 명령의 실행 결과로 바뀐다. 결과적으로 다음과 같이 화면에 출력 Today is 2018:10:29 (월) 18:19:50 KST
ls /platform/`uname -m`	uname -m 명령의 실행 결과를 문자열로 바꿔 디렉터리 이름으로 사용

특수 문자 \

- \ (역슬래시)는 특수 문자 바로 앞에 사용되는데, 해당 특수 문자의 효과를 없애고 일반 문자처럼 처리한다.

ls -l t*	t*라는 이름을 가진 파일의 상세 정보를 출력. \ 없이 t*를 입력하면 t로 시작하는 모든 파일의 상세 정보를 출력
echo \\$SHELL	\$SHELL을 화면에 출력. echo '\$SHELL'의 결과와 같다.

특수 문자 >, <, >>

- >, < (적외), >> (이중적외)는 입출력의 방향을 바꾸는 특수 문자이다.

ls -l > res	ls -l 명령의 실행 결과를 화면이 아닌 res 파일에 저장
-------------	------------------------------------