

Linux

- 자동마운트+링크파일+압축

INDEX

- 1. 자동마운트
- 2. 링크파일
- 3. 압축

1. 자동마운트

1. 자동마운트

▶ /etc/fstab

- 영구적인 마운트 경로 설정파일
- 리눅스 부팅시 자동 마운트경로에 대한 정보 저장

▶ /etc/fstab 확인

1)	2)	3)	4)	5)	6)
/dev/sdb1	/mp1	ext4	default	0	0

필드	설명
1) 디바이스명	파티셔닝된 디바이스명
2) 마운트포인트	마운트 포인트 경로 지정. 스왑파티션은 마운트 되지 않기 때문에 "none" 또는 "swap"으로 기술한다
3) 파일시스템	파일 시스템의 종류
4) 옵션	파일시스템 고유의 옵션 지정 default : 읽고 쓸수 있으며 비동기 액세스, 일반 유저에 의한 마운트의 금지 등 선택 가능 noauto : 명시적으로 지정하지 않는 한 마운트 할 수 없도록 설정 ro : 읽기 전용 user : 일반 유저에 의한 마운트 허가 usrquota : 일반 사용자에게 디스크 사용량을 제한하는 옵션
5) 시스템덤프 사용유무	dump 명령어가 그 파일 시스템을 덤프 할수 있는지 여부 지정
6) 파일시스템체크	fsck 명령어로 무결성의 체크 여부를 지정 0 : 파티션 체크 하지 않음 , 1: 숫자가 작은 순서로 체크(1이니 우선체크)

2. 링크파일

2. 링크파일

▶ 링크파일

- 원본과 연결되어 동기화 기능을 제공 하는 파일
- 윈도우의 바로가기 아이콘과 유사

▶ 링크파일 종류

- 심볼릭 링크

단순히 원본파일을 가리키도록 링크만 시켜둔 것으로 MS의 윈도우시스템에서 흔히 사용하는 '바로가기' 같은 것이며, 원본파일을 가리키고만 있으므로 원본파일의 크기와는 무관한다. 그리고 심볼릭링크에서는 원본파일이 삭제되어 존재하지 않을 경우에 링크파일은 깜박거리면서 링크파일의 원본파일이 없다는 것을 알려준다.

- 하드 링크

원본파일과 다른 이름으로 존재하는 동일한 파일이며 원본파일과 동일한 내용의 다른 파일이라고 할 수 있다. 그리고 하드링크에서는 원본파일과 링크파일 두개가 서로 다른 파일이기 때문에 둘 중 하나를 삭제하더라도 나머지 하나는 그대로 남아 있다. 또한 하드링크에서는 원본파일의 내용이 변경될 경우에는 링크파일의 내용 또한 자동으로 변경된다.

2. 링크파일

▶ ln : 링크파일 생성

ln은 Link 의 약어로서 리눅스 파일시스템에서 링크파일을 만드는 명령어이다

▶ ln 사용 형식

ln [옵션][원본경로][링크파일경로]

▶ ln 옵션

옵션	설명
--backup	대상파일이 이미 존재할 경우에 백업파일을 만든 후에 링크파일 생성
-b	링크파일 생성시에 대상파일이 이미 존재하면 백업파일을 만든 후에 링크파일 생성
-d	디렉토리에 대한 하드링크생성을 가능하게 함. 단 root권한으로 수행하더라도 시스템의 권한제한으로 인하여 실패할 가능성이 높음(-F 와 --directory 는 -d 와 동일함)
-f	대상파일이 존재할 경우에 대상파일을 지우고 링크파일 생성
-i	대상파일이 존재할 경우에 대상파일을 지울것인가를 확인 요청(--interactive 와 동일)
-s	심볼릭 링크파일을 생성
-S	백업파일 생성시에 원하는 접미사(suffix)를 지정할 수있음
-t, --target-directory=DIRECTORY	링크파일을 생성할 디렉토리를 지정

3. 압축

3. 압축

▶ gzip : 파일압축

GNU에서 배포하는 압축 유틸리티로 GNU 그룹에서 배포되는 유틸리티들은 모두 gzip으로 압축되어있음

▶ gzip 사용 형식

gzip [옵션][이름]

▶ gzip 옵션

옵션	설명
-c	압축/해제한 결과를 표준 출력으로 내보낸다. 원본 파일은 변형되지 않는다
-r	서브 디렉토리까지 압축/해제한다. -c옵션과 같이 사용한다
-d	압축을 풀 때 사용한다
-f	압축할 파일이 여러 링크를 가지고 있어도 그냥 압축한다
-l	압축 전 파일의 크기와 압축한 후의 크기, 압축률 등을 표시한다

3. 압축

▶ bzip2 : 파일압축

gzip 보다 높은 압축률을 가진 명령어

▶ bzip2 사용 형식

bzip2 [옵션][이름]

▶ bzip2 옵션

옵션	설명
-c	압축/해제한 결과를 표준 출력으로 내보낸다. 원본 파일은 변형되지 않는다
-r	서브 디렉토리까지 압축/해제 한다. -c 옵션과 같이 사용한다
-d	압축을 풀 때 사용한다. 강제 압축해제를 의미한다. bzip2, bunzip2 및 bzipcat 는 실제로 같은 프로그램이다
-f	출력 파일을 강제로 덮어 쓴다. 파일들에 대한 하드 링크를 깨도록 강제한다
-s	압축,해제와 테스트에 사용하는 메모리의 양을 줄인다. 파일들은 단위 바이트 당 오직 2.5바이트만을 필요로 하는 수정된 알고리즘을 이용해 압축&해제되고 검사된다
-1 부터 -9	압축할 때 블록 크기를 100k,200k,,,900k로 설정한다. 압축 해제 시에는 아무런 영향이 없다
-q	불필요한 경고 메시지를 무시한다. I/O 에러와 다른 치명적인 사건을 포함한 메시지는 무시되지 않을 것이다

3. 압축

▶ xz : 파일압축

데이터 압축툴. 확장자는 .xz형식이지만 구버전의 lzma형식과 로압축스트림(row compressed streams)도 지원

▶ xz 사용 형식

xz [옵션][이름]

▶ xz 옵션

옵션	설명
-z --compress	압축한다. 기본 운용 모드
-d --decompress, --uncompress	압축 해제한다
-t --test	압축 파일의 무결성을 검사한다. 어떤파일도 작성되지 않고 삭제되지 않는다
-l --list	압축 파일에 대한 정보를 보여준다. 어떤 압축해제 출력도 만들어지지 않고, 어떤 파일도 작성되거나 지워지지 않는다

3. 압축

▶ tar : 파일 및 디렉토리 묶기

여러개의 파일을 하나의 파일로 묶거나 풀어주는 명령어

▶ tar 사용 형식

tar [옵션][FILEDEST][FILESOURCE]

▶ tar 기능 옵션 : 반드시 들어가야 하는 옵션

옵션	설명
-A	tar 파일들을 아카이브에 추가한다
-c	새 tar 파일을 만든다
-d	tar 파일과 실제 파일 시스템과의 차이를 확인한다
-r	기존의 tar 파일에 새 파일들을 추가한다
-t	tar 파일의 내용을 리스팅한다
-u	tar 파일에 포함된 파일보다 새로운 파일들만 추가한다
-x	tar 파일에 포함된 파일들을 추출한다

3. 압축

▶ tar 부가 옵션 : 기능에 추가적으로 사용되어 tar의 동작을 결정

옵션	설명
-b	tar 파일을 생성할 때 블록 크기를 지정한다. 테이프와 같은 백업장비를 사용할 때, 기종마다 기본 블록 크기가 다른 경우가 있는데, 이러한 경우에 적절한 블록 크기를 설정해 주어야 한다
-C	tar 파일에 포함된 파일들을 풀어 놓을 디렉토리를 지정한다
-f	옵션 뒤에 지정된 파일 또는 장치를 사용한다(기본값으로 -가 사용되는데, 이것은 표준입력과 표준 출력(stdin/stdout)을 의미)
-p	사용자의 사용권한(퍼미션)을 변경하지 않고 보존한다
-v	현재 처리하고 있는 파일을 출력한다
-Z	compress 유틸리티로 압축한다
-z	gzip 유틸리티로 압축한다
-j	bzip2 유틸리티로 압축한다
-J	xz 유틸리티로 압축한다