



파일 관리

zelratole@hanmail.net



파일 구성 요소 :

- 구성:파일이름, **I-node**, 데이터 블록
- 파일
 - 파일이름 변경, 수정, 삭제
 - .파일이름 : 숨어 있는 파일, 시스템 파일로 사용
 - **ls -a** 옵션.. 확인
- **I-node**
 - **Index-node**로 데이터 블록이 디스크 상의 어느 주소에 위치하고 있는가에 대한 정보를 기록
 - 하나의 파일이 생성되면 파일명과 **I-node**가 생성하며 번호가 부여
 - **ls -li : -i**로 링크의 갯수 확인..



- I-node에 저장된 정보
 - 파일의 종류
 - 파일의 소유권
 - 파일의 액세스 모드
 - 파일의 타임스탬프(파일의 갱신일)
- 데이터 블록
 - 파일을 저장할 때, 구분 없이 디스크에 저장되는 데, 실제 데이터는 디스크의 어느 한 쪽에 쌓여져 있다. 이것 데이터 블록



- 파일의 종류 확인 `ls -l`
 - 일반파일(-), 디렉토리파일(d), 링크파일(l), 디바이스 파일(b)
- 일반파일
 - `ls -l`, `ls -F`
- 디렉토리파일..
 - `./` : 현재, `../` : 상위, `/` : 최상위
- 링크 파일 : 파일명을 다르게 접근처리
 - 심볼릭 링크 : 저장 위치만 호출. 원본파일삭제시, 접근 불가
 - 하드 링크 : 사본을 만들어 호출. 원본파일삭제시도접근
 - ln 원본파일 하드링크파일



■ 디바이스 파일

- 리눅스 시스템에서 **HDD**, 프린터, 마우스, 키보드, 사운드 어댑터 등을 하나의 파일로 처리
- 블록 디바이스 파일
 - **hdd, fdd** 같은 보조 기억장치 설치 시 사용
 - 첫글자가 **b**로 설정
- 캐릭터 디바이스 파일..
 - 주로 터미널, 프린터, 플로터과 보조 기억장치등을 설치시 사용..
 - 첫글자가 **c**로 설정..
- `ls -al /dev/sda?`



■ ls [옵션] [디렉토리명]

- -a : 숨겨진 파일(.)과 디렉토리 출력
- -C : 여러 개의 줄을 세로로 정렬된 형태로 출력
- -d : 지정된 디렉토리 자체의 정보만을 출력
- -F : 파일의 종류를 알 수 있도록 파일 뒤에 유형출력
- -i : i-node에 index번호를 표시하여 출력
- -l : 파일과 디렉터리의 모든 속성 출력
- -m : 파일과 디렉터리의 내용을 쉼표로 구분하여 출력
- -n : 사용자 id대신 사용자와 그룹 소유주의 UID, GID 출력



- ls [옵션] [디렉토리명]
 - -p : 디렉토리에 / 붙여서 출력.
 - -r : 역순으로 출력
 - -R : 현재 디렉터리와 하위 디렉토리 내용까지 출력
 - -s : 앞에 512byte 블록 단위 숫자를 출력
 - -t : 가장 최신에 사용한 순서대로 파일을 정렬 출력
 - -x : 파일을 한줄로 정렬하여 출력
- touch [-option] 파일
 - 공백파일을 생성하거나, 현재 시간으로 update된 파일 처리.



- **cp [option] 원본파일명 복사파일명**
 - **-a** : 원본파일의 속성과 링크 정보를 유지 복사.
 - **-b** : 원본의 백업본을 생성..
 - **-d** : 원본 심볼릭 링크된 파일 그 자체까지 복사
 - **-f** : 파일 이름이 겹치더라도 강제 복사
 - **-i** : 파일 이름 겹치면 물어보는 프롬프트
 - **-p** : 원본 소유중, 그룹, 퍼미션, 시간을 동일 복사
 - **-r** : 디렉토리 안에 있는 모든 하위 디렉터리 복사
 - **-s** : 심볼릭 링크 생성
 - **-u** : 원본/변경 날짜가 최신 일때만 복사.
 - **-v** : 원복/복사 파일 이름 보여줌.



- 파일 이동과 파일이름 변경 : mv
- mv [-option] [기본파일이름] [이동할 디렉토리/변경할파일명]
 - -f : 이동할 위치에 동일한 파일 있어도 강제 이동
 - -u : 최신 파일만 이동
 - -v : 이동과정..



- **rm** : 파일 삭제
 - **rm [-option]** 삭제할 파일이름
 - **-f, -i, -r, -v**



■ find

- 시스템에서 원하는 조건에 맞는 파일을 찾고자 할 때 사용하는 명령어..
- 형식

- find 경로명 조건옵션 파일이름(*,??활용)
동작옵션

■ 경로명

- /, /특정폴드, ../상위폴드, ./현재폴드,
/폴드1/폴드2

■ 조건옵션

- -empty : 비어있는 파일 찾기



■ 조건 옵션

- **-group** : 지정한 그룹의 소유하고 있는 파일
- **-name** : 지정한 파일 찾기('*', '?', '[']) 활용
- **-newer** : 지정한 파일보다 최근에 생성된 파일 찾기
- **-size** : 지정한 파일 크기 n으로 찾기
 - c:byte/k:kbyte/w:word
 - +n : 보다 큰 파일, n:정확한 파일크기, -n:보다 작은 파일
- **-type** : 찾고자 하는 파일의 타입 지정
 - b:블록타입, c:캐릭터타입, d:디렉토리, f:파일, l:심볼릭 링크



- 조건옵션
 - **-user**: 지정한 사용자가 소유하고 있는 파일 찾기
- 동작옵션 : **find**로 검색된 파일에 대한 처리..
 - **-print** : 찾은 파일 리스트를 보여줌..
 - **-prune** : 현재 디렉토리의 하위 디렉토리로는 가지 않음.
 - **-exec** [명령] {} \;
 - **-ok** [명령] {} \; 명령을 적용하기 전 물어봄.



확인예제 :

- /home/vagrant/**findexp**를 기준으로 동작명령으로 수행하세요..
 - 해당파일 a01.txt 파일을 생성.
 - `find /home/vagrant/* -type d -name 'find*' -exec touch {} /a01.txt \;`
 - 폴드 a01work 폴드 생성..
 - `find /home/vagrant/* -type d -name 'find*' -exec mkdir {} /a01work \;`
 - a01.txt을 a01work폴드에 복사
 - findexp의 모든 폴드와 파일을 현재시간으로 수정으로 변경
 - findexp폴드 하위에 모든 파일을 삭제처리.



확인예제 :

- /home/vagrant/findexp를 기준으로 동작명령으로 수행하세요..
 - a01.txt을 a01work폴드에 복사
 - `find /home/vagrant/* -type d -name 'find*' -exec cp a01.txt {} /a01work \;`
 - findexp의 모든 폴드와 파일을 **현재시간으로** 수정으로 변경
 - `find /home/vagrant/* -type d -name 'find*' -exec touch {} \;`
 - findexp폴드 하위에 모든 파일을 **삭제처리**. `find /home/vagrant/* -type d -name 'find*' -exec rm -rf {} \;`



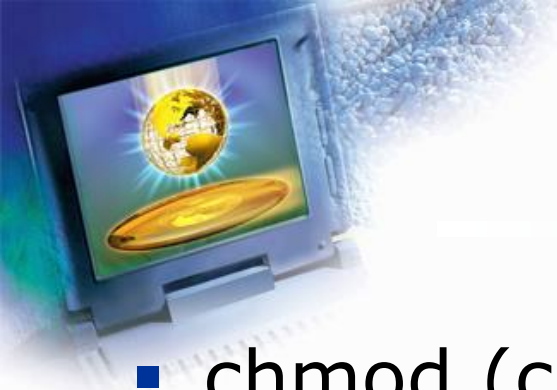
확인예제 :

- /home/vagrant를 기준 폴드 동작명령어 이용!!
 - test15폴드 생성
 - `sudo find /home -type d -name 'vagrant' -exec mkdir {} /test15 \;`
 - test1가 포함된 폴드에 모두가 exp01.txt 파일 생성
 - `sudo find /home/vagrant -type d -name 'test1*' -exec touch {} /exp01.txt \;`



확인예제 :

- /home/vagrant를 기준 폴드 동작명령어 이용!!
 - 위 파일의 모두 exp10.log 파일로 변경
 - `sudo find /home/vagrant -type d -name 'test1*' -exec mv {} /exp01.txt {} /exp10.log \;`
 - exp10.log 파일 모두 삭제
 - `sudo find /home/vagrant -type d -name 'test1*' -exec rm -f {} /exp10.log \;`



- **chmod (change mode)**
 - 파일과 사용자 권한 변경처리..
 - 파일 권한
 - 소유자, 그룹, 그외사용자
 - `read(4), write(2), execute(1) : 7`
 - `-(0)`
 - `644`

종류	사용자			사용자소속그룹			그외 사용		
	r	w	x	r	w	x	r	w	x
-	4	2	-	4	-	-	4	-	-



- 권한의 변경 연습.
 - `chmod @@@ 파일명`
 - 1) 생성자는 모든 권한(`rwX -421 ==> 7`)
 - 2) 생성자 소속 그룹 모든 권한(`rwX -421 -> 7`)
 - 3) 그외사람은 read만 설정 (`r-- --> 400 -> 4`)
 - `chmod 774 exp01.txt`
 -



확인예제 :

확인예제

/home/vagrant 하위 폴드에 authorexp폴드 생성

하위에 다음과 같은 파일 및 폴드를 생성하고 권한을 설정하세요.

	사용자	그룹	그외
1. a01.txt	읽기	실행	없음
2. a02.txt	쓰기	쓰기	실행
3. a03(폴드)	읽기	읽기	쓰기
	쓰기		
4. a03.sh	읽기	실행	실행
	쓰기		
	실행		



- 심볼릭 모드로 사용자 권한 변경
 - 문자와 기호를 이용해서 사용 권한 설정.
 - 사용되는 문자
 - **u**(user) : 사용자 권한, 소유자, 파일 생성자
 - **g**(group) : 사용자와 동일한 그룹
 - **o**(other) : 타인 권한
 - 사용되는 기호 - 권한내용(r, w, x)
 - **+** : 권한 추가
 - ex) u+w : 소유자에게 쓰기권한 추가
 - **-** : 권한 제거 ex) g-x : 그룹의 실행 권한 삭제
 - **=** : 명시된 권한을 할당 (두 사용자 권한 동일)



심볼릭 권한 수정 연습예제 :

- /home/vagrant/authorexp2 폴드에
 - one.txt 파일을 기본으로 생성
 - 1. 모든 권한(사용자, 그룹, 그외)을 삭제처리
 - `chmod ugo-rwx one.txt`
 - 2. 사용자에게 수정, 실행, 읽기 권한 부여
 - `chmod u+rwx one.txt`
 - 3. 그룹에게 실행과 읽기 권한 부여
 - `chmod g+rw one.txt`
 - 4. 그외 사람에게 읽기 권한 부여
 - `chmod o+r one.txt`
 - exp01 폴드 생성
 - 위 권한 연습 처리..



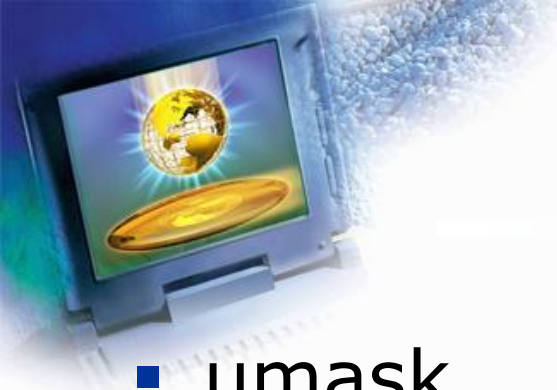
■ 기본 권한 설정 : **umask**

- 파일과 디렉터리를 초기 생성시, 기본 권한 처리에 대한 설정..
- 기본권한 읽기 : **umask**
- 기본권한 변경 : **umask @@@**

■ 확인

- **umask**를 통해서 기본 권한을 확인하고, **umask**를 사용자는 전체권한, 그룹은 읽기와 실행, 다른사용자는 읽기

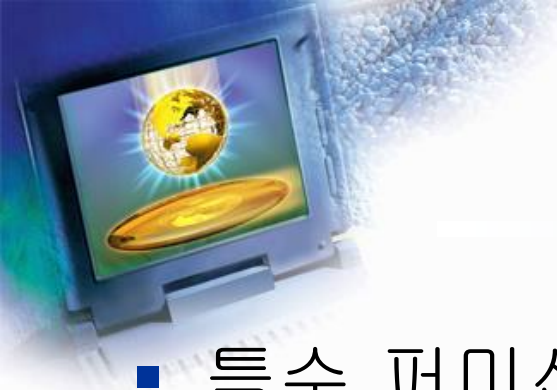
권한구분	SetUid	User	Group	Others
umask bit	0~7	0	0	2



■ umask

	000	001	002	022
파일	666	666	664	644
디렉터리	777	776	775	755

- 초기 권한 변경
 - umask 0000
 - 파일생성 권한 확인
 - umask @@@



- 특수 퍼미션 : set**U**id, set**G**id, Sticky bit
 - 프로그램이 파일과 프로세스에 접근하는 것을 허용하는 기능..(가장 왼쪽 **bit**를 통해)
 - 권한 설정..
 - **setUid** : 4000 - 파일과 명령어가 실행되는 동안에 실행한 사용자X, 파일소유자의 권한을 부여하는 퍼미션 값..
 - **setGid** : 2000 - 파일과 명령어가 실행되는 동안에 그룹 소유자의 권한을 갖게 되는 퍼미션 값. 실행파일 뿐만 아니라 디렉토리에도 명시 가능
 - **Sticky bit** : 1000- 누구나 파일을 쓰고, 실행이 가능 하지만 삭제는 오직 소유자에게 허용



- 특수퍼미션과 권한 처리 예제..
 - 여러 사람이 동시에 사용하는 os. 톰캣의 실행 파일을 사용자의 권한이 아닌 파일 소유자의 권한으로 톰캣을 구동 할 수 있도록 설정..
 - $4700(4000+400+200+100)$



- 리눅스의 처리 과정을 파일 단위로 구성
- 계층적인 구조의 특성
- 파일은 **파일이름, I-node, 데이터블록** 세가지 요소
- 파일의 종류는 일반파일, 디렉터리 파일, 링크 파일, 디바이스 파일, 파이프 파일, 소켓 파일 등이 있다.



연습문제 :

1. 파일의 구성 요소에 대하여 상세하기 기술하세요.
2. ls -l 을 통해서 나열한 파일 상세 내용을 기술하세요.

drwxr-x---	2	root	root	4096	Dec 18	12:31	aa
-rw-----	1	root	root	2019	Dec 4	01:38	ab.txt

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

3. 해당 내용에 대한 명령어를 기술
 1. 파일 목록 보기 :
 2. 파일의 타임스탬프(시간) 변경 :



4. 파일 복사
5. 파일 삭제
6. 파일 찾기
7. chmod ?
8. 파일과 디렉토리의 기본 권한 변경?



Thank You !

www.themegallery.com