File System

디렉토리 와 파일

- 개요
  - ✓ 리눅스는 기본적으로 유닉스 계열의 운영체제이므로 유닉스의 많은 부분을 그대로 이어받아 유지
  - ✓ 유닉스에서는 시스템과 관련된 정보와 하드웨어 같은 장치를 모두 파일로 관리
  - ✓ 리눅스도 유닉스처럼 시스템 관련 정보와 장치를 관리하기 위해 파일을 사용
  - ✓ 파일은 관련 있는 정보들의 집합
  - ✓ 리눅스는 파일을 효과적으로 관리하기 위해 디렉토리를 사용하는데 디렉토리는 계층 구조를 가짐
  - ✓ 리눅스는 전체 파일을 용도에 따라 계층적인 디렉토리로 구분하여 관리
  - ✓ 디렉토리와 파일로 구성된 전체 집합을 파일 시스템이라고 함
  - ✓ 윈도에서는 물리 디스크가 2개 있다면 디렉토리 트리도 2개이지만 리눅스에서는 언제나 시스템 전체에 단 하나의 트리만 가짐
  - ✓ 우분투에서 관리하는 모든 데이터는 파일(현재 실행 중인 프로세스, 일반 파일, 하드웨어 등)로 인식되며 하나의 프로세스는 하나의 작업 디렉토리를 가지며 우분투는 일반 파일, 디렉토리 파일, 장치(특수) 파일, 심볼릭 링크를 가지며 파일은 사용자가 이용할 수 있는 데이터의 실체를 의미

디렉토리 와 파일

- ❖ 파일의 종류
  - ✓ 일반 파일
    - 데이터를 관리하는데 주로 사용되는 일상적인 파일을 의미
    - 실행 파일이나 텍스트 파일, 이미지 파일, 프로그램 소스 파일 등 우분투에서 사용하는 대부분의 파일을 의미하며 실행 파일이나 이미지 파일의 경우 데이터가 바이너리 형태로 저장
    - 실행 파일을 확인하기 위해서는 해당 파일의 내용을 확인할 수 있는 유틸리티 프로그램이 있어야 하고 텍스트 파일의 경우에는 문서 편집기를 사용하여 내용을 편집하거나 볼 수 있음
    - 일반 파일을 정규 파일 이라고도 하며 정규 파일은 표준 입출력 시스템의 호출을 통해 참조됨
    - 문서 파일(Text File)은 문자로 구성
    - 이진 파일
      - ◆ 이진 파일(Binary File)은 문자가 아닌 데이터 파일이 들어있으며 기계어인 0 과 1의 값으로 구성
      - ◆ 이진 파일은 프로그램을 코딩하여 컴파일을 수행하는 과정에서 생성되는 파일을 의미하며 실행 파일 이라고도 불림

디렉토리 와 파일

- 파일의 종류
  - ✓ 디렉토리 파일
    - 우분투에서는 디렉토리(Directory) 역시 파일로 취급
    - 디렉토리 파일은 해당 디렉토리에 저장된 파일이나 하위 디렉토리에 대한 정보를 소유
    - 디렉토리는 일반 파일과 마찬가지로 디스크에 저장되어 다른 파일을 조작하고 엑세스하는데 필요한 정보를 소유
    - 디렉토리 안에는 파일을 엑세스하는데 필요한 정보를 가지고 있으며 디렉토리라는 파일을 이용해서 전체 파일을 하나의 트리 구조로 만들어 관리

디렉토리 와 파일

- ❖ 파일의 종류
  - ✓ 링크 파일
    - 시스템 사용자에게 편리성을 제공해 주는 파일
    - 유닉스와 리눅스에서는 링크(Link)라는 개념을 도입해 여러 개의 파일 이름이 하나의 I-node에 연결되도록 수행하여 실제 파일이나 디렉토리 혹은 또 다른 링크를 가리킴
    - 하드 링크(Hard Link)
      - ◆ 원본 파일을 복사하여 원본 파일과 동일한 내용의 다른 사본 파일을 만드는 것
      - ◆ 하드 링크에서는 원본 파일과 링크 파일 두 개가 서로 다른 파일로 취급되므로 둘 중 어느 하나를 삭제하더라도 나머지 하나는 그대로 남아있음
      - ◆ 하드 링크에서는 원본 파일의 내용이 변경될 경우에는 복사된 링크 파일의 내용 또한 자동으로 변경되며 하드 링크는 동일한 I-node를 갖는 파일을 생성하는 것을 의미하며 파일의 실제 I-node의 정보를 공유하기 때문에 마치 동일한 파일이 여러 곳에 존재하는 것처럼 보이게 됨
    - 심볼릭 링크(Symbolic Link)
      - ◆ 다른 표현으로는 소프트 링크(Soft Link)
      - ◆ 심볼릭 링크 파일의 기능은 Windows 운영체제에서 사용되는 바로 가기 기능이나 단축 아이콘 과 같은 역할을 수행

디렉토리 와 파일

- 파일의 종류
  - ✓ 장치(특수) 파일
    - 리눅스에서는 하드디스크나 키보드 같은 각종 장치도 파일로 취급
    - 장치 파일은 리눅스 시스템에 부착된 장치들을 관리하기 위한 특수 파일
    - 리눅스 시스템에서 각종 장치를 관리하기 위해 시스템 관리자는 해당 장치 파일에 접근해야 함
    - 대부분의 장치 파일은 /dev 디렉토리 아래에 위치
    - Windows 운영체제에서 하드디스크(Hard Disk)가 C:. D:. E: 등으로 표시되지만 우분투에서는 /dev/hda1, .. /dev/hda3과 같이 사용

디렉토리 와 파일

- ❖ 파일의 종류 확인
  - ✓ file 명령으로 확인 가능
    - file .profile

    .profile: ASCII text
    - adam@adam:~$ file Downloads

    Downloads: directory
    - adam@adam:~$ file /bin/bash

    /bin/bash: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=be49fa95e8d5f9adc2d850e7967b3c3082466448, for GNU/Linux 3.7.0, stripped

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 개요
    - 우분투 파일 시스템은 파일과 디렉토리의 계층형 구조(Tree)로 구성
    - 최상위에 루트 디렉토리(/)를 하나 두고 그 아래에 디렉토리나 파일을 배치
    - 작업 디렉토리: 사용자가 우분투에 접속하여 현재 사용하고 있는 디렉토리를 의미하는 것으로 현재 작업 중인 디렉토리는 점(.)으로 표시되고 현재 작업 중인 디렉토리의 위치를 상세하게 확인하기 위해서는 pwd 명령을 사용
    - 상위 디렉토리: 자신을 포함하고 있는 디렉토리로 .. 으로 표시
    - 홈 디렉토리: 각 사용자에게 할당되는 디렉토리로 사용자 계정을 처음 만들 때 지정해 주는 것으로 각각의 사용자 별로 홈 디렉토리를 지정하는데 ~ 로 표시

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 기본적으로 제공되는 디렉토리
    - ls -F /

bin@ dev/ lib@ mnt/ root/ snap/ sys/ var/

boot/ etc/ lost+found/ opt/ run/ srv/ tmp/

cdrom/ home/ media/ proc/ sbin@ swap.img usr/

- ◆ 디렉토리 이름의 끝에 붙은 / 는 해당 파일이 디렉토리임을 @은 심볼릭 링크임을 의미
- ◆ bin, lib, sbin은 디렉토리가 아니라 심볼릭 링크이며 bin은 /usr/bin 디렉토리의 심볼릭 링크이고 lib는 /usr/lib 디렉토리의 심볼릭 링크
- ◆ bin과 lib는 유닉스 초기부터 있었던 디렉토리로 지금은 그 기능이 /usr/bin과 /usr/lib로 통합되었으나 호환성 유지를 위해 심볼릭 링크 형태로 남겨둔 것

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 기본적으로 제공되는 디렉토리
    - ● 루트 계정에 제공되는 디렉토리
      - ◆ /dev: 디바이스 파일이 배치되어 있는 디렉토리
      - ◆ /home
        - ▪ 사용자 별로 할당되는 홈 디렉토리가 배치되는 디렉토리
        - ▪ adam 이라는 사용자의 홈 디렉토리는 /home/adam
      - ◆ /media: DVD/CD 나 USB 와 같은 외부 장치를 연결하는 디렉토리
      - ◆ /etc: 리눅스 설정 파일이 위치하는 디렉토리
      - ◆ /opt: 추가 패키지가 설치되는 디렉토리
      - ◆ /root: root 계정의 홈 디렉토리
      - ◆ /sys: 리눅스 커널과 관련있는 파일이 저장된 디렉토리
      - ◆ /usr: 기본 실행 파일 과 라이브러리 파일, 헤더 파일 등이 저장되는 디렉토리
      - ◆ /boot: 부팅에 필요한 커널 파일이 저장된 디렉토리
      - ◆ /lost+found: 파일 시스템에 문제가 발생하여 복구할 경우 문제가 되는 파일이 저장되는 디렉토리

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 기본적으로 제공되는 디렉토리
    - ● 루트 계정에 제공되는 디렉토리
      - ◆ /mnt: 파일 시스템을 임시로 마운트하는 디렉토리
      - ◆ /proc: 프로세스 정보 등 커널 관련 정보가 저장되는 디렉토리
      - ◆ /run: 실행 중인 서비스와 관련된 파일이 저장되는 디렉토리
      - ◆ /srv: FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장되는 디렉토리
      - ◆ /tmp: 시스템 사용 중에 발생하는 임시 데이터가 저장되는 디렉토리로 이 디렉토리에 있는 파일은 재시작하면 모두 삭제됨
      - ◆ /var: 시스템 운영 중에 발생하는 데이터나 로그 등 내용이 자주 바뀌는 파일이 저장되는 디렉토리

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 기본적으로 제공되는 디렉토리
    - 사용자 홈 디렉토리에 제공되는 디렉토리
      - ls -F ~

        Desktop/ Downloads/ Music/ practice/ Templates/

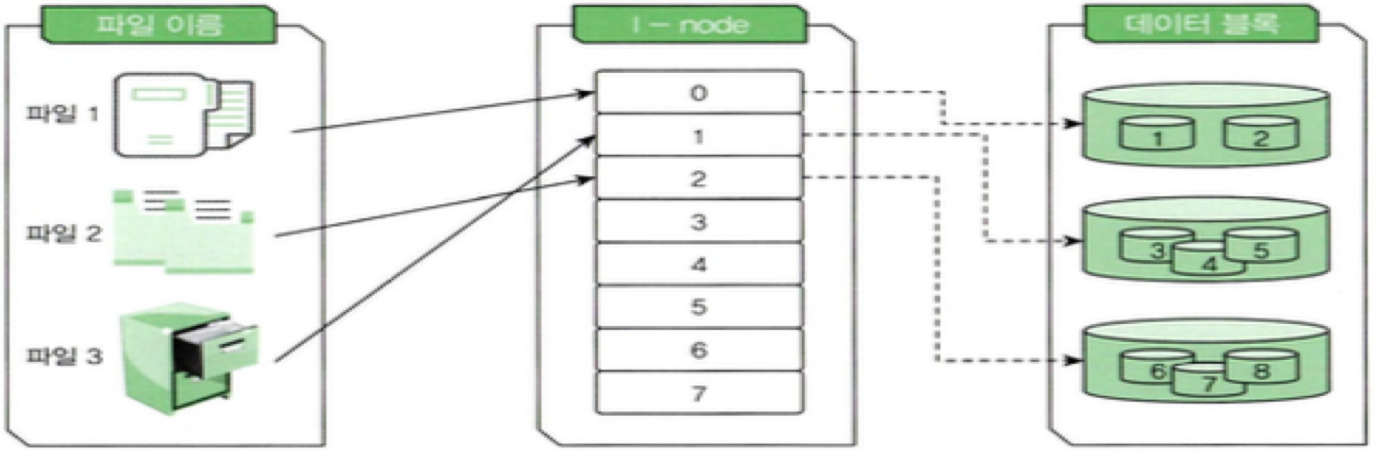
        Documents/ file1 Pictures/ Public/ Videos/

디렉토리 와 파일

- ❖ 디렉토리 와 파일 관리
  - ✓ 절대 경로 와 상대 경로
    - ● 절대 경로
      - ◆ 루트 디렉토리(/) 위치부터 시작하는 경로
      - ◆ 반드시 / 로 시작
      - ◆ 중간 단계의 모든 디렉토리를 표시
      - ◆ 특정 위치를 가리키는 절대 경로명은 항상 동일
    - ● 상대 경로
      - ◆ 현재 디렉토리를 기준으로 하는 경로로 / 이외의 문자로 시작
      - ◆ 현재 디렉토리에서 상위 디렉토리로 이동할 때는 ..
      - ◆ 현재 디렉토리에서 서브 디렉토리로 내려갈 경우 서브 디렉토리명을 추가
      - ◆ 현재 위치에 따라 위치가 경로명이 달라질 수 있음

디렉토리 와 파일

- 파일의 구성 요소
  - ✓ 우분투에서는 모든 처리 과정을 파일 단위로 처리하며 계층적인 구조의 특성을 지니고 있는데 우분투에서 다루고 있는 파일은 3개의 충족 요건인 파일 이름, I-node, 데이터 블록을 모두 충족해야만 존재 가치를 가질 수 있음



디렉토리 와 파일

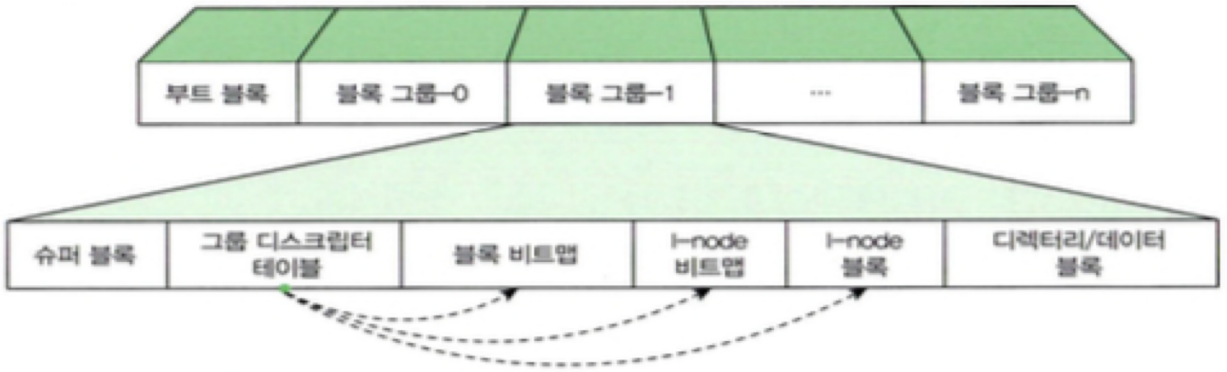
- ❖ 파일의 구성 요소
  - ✓ 파일 이름
    - 파일 이름은 사용자가 파일을 사용 목적에 맞도록 사용하고 해당 파일에 정확하게 접근할 수 있도록 구별할 수 있는 변별력을 제공
    - 변별력은 파일을 조작하기 위한 과정에서도 활용되는데 파일 이름을 정확히 파악할 수 없는 상태라면 실수로 다른 파일을 조작하게 되어 우분투 시스템에 예상치 못한 결과를 초래할 수도 있음
    - 우분투에서 확장자는 선택적 요소
    - 파일 과 디렉토리 이름 규칙
      - /는 파일명이나 디렉토리명에 사용할 수 없음
      - 파일과 디렉토리 이름에는 알파벳, 숫자, .(마침표) 만 사용할 수 있음
      - 공백. *, |, ', @, #, $, %, A & 등은 사용할 수 없음
      - 알파벳 대/소문자는 엄격하게 구별되어 다른 글자로 취급
      - 파일과 디렉토리명을 선언할 때 점(.)으로 시작하면 숨김 파일로 간주

디렉토리 와 파일

- ❖ 파일의 구성 요소
  - ✓ I-node
    - I-node는 Index Node라고 불리며 파일을 기술하는 디스크 상에서의 데이터 구조를 의미
    - I-node는 파일의 데이터 블록이 디스크 상의 어느 주소에 위치하고 있는가에 대한 정보를 기록하기 위해 사용
    - 파일을 생성하게 되면 I-node의 link가 0 인 위치에 I-node를 생성하고 정보를 저장한 다음 link의 값이 1의 위치로 이동
    - I-node에 저장된 정보
      - ◆ 파일의 종류
      - ◆ 파일의 소유권 - 사용자(소유자)와 그룹
      - ◆ 파일의 엑세스 모드
      - ◆ 파일의 타임스탬프(파일 갱신일)
    - 우분투에서 파일을 생성하기 위해서는 파일의 데이터를 저장하기 위한 디스크 내의 공간을 먼저 확보해야 하는데 이때 파일이 저장될 주소에 대한 정보를 기록하기 위해 I-node는 각각의 파티션마다 0 부터 시작하는 정수의 형태로 고유의 식별 번호를 가짐

디렉토리 와 파일

- ❖ 파일의 구성 요소
  - ✓ 데이터 블록
    - 데이터 블록(Data Block)은 파일에서 데이터를 저장하는 블록을 의미
    - 데이터 블록에는 일반 파일이나 디렉토리 파일의 데이터가 존재하며 디스크 장치에 파일을 저장할 때 실제 데이터는 특별한 구분없이 디스크에 저장
    - 파일 1과 파일 2가 디스크에 저장될 때 특별히 2개를 따로 따로 구분하지 않고 디스크에 차례대로 기록하는데 파일들의 실제 데이터는 디스크의 어느 한 곳에 차례대로 쌓이게 되는데 이 부분을 데이터 블록이라고 함
    - 파일을 저장할 때 어떠한 파일인지를 구분하지는 않기 때문에 파일들을 구분할 수 있는 부가 정보가 필요하며 우분투에서는 I-node를 이용해 부가 정보를 구별



파일 시스템 관련 명령어

- file
  - ✓ file 명령어는 지정된 파일의 종류(타입)을 확인하는 명령어
  - ✓ file 명령어는 /usr/share/file 디렉토리의 magic 파일을 참조하여 파일 종류를 표시
  - ✓ 형식

    file [옵션] [파일 및 디렉토리 경로]
  - ✓ 옵션
    - -C: 매직 파일의 포맷을 검사하는 옵션
    - -f 목록파일: 많은 파일들을 한번에 확인하기 위하여 파일 리스트인 목록 파일을 만들어서 그 안에 입력된 모든 파일을 한꺼번에 확인하는 옵션
    - -m 매직파일 : 지정된 매직 파일로 대상 파일을 확인

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ pwd
    - ● 현재 작업 중인 디렉토리를 절대 경로로 출력
    - ● 현재 디렉토리 확인
      - ◆ pwd

        /home/adam

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ cd
    - 현재 디렉토리에서 다른 디렉토리로 이동할 때 사용
    - 형식: cd [디렉토리]
    - 특별한 사용
      - ◆ 홈 디렉토리로 이동하고자 하는 경우는 cd 명령만 사용하거나 cd ~
      - ◆ 현재 디렉토리는 ./
      - ◆ 상위 디렉토리는 ../
      - ◆ 탭 키를 이용한 자동 완성 가능

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ cd
    - /usr로 이동
      - cd /usr
      - pwd

      /usr
    - 하위 디렉토리인인 lib 로 이동
      - cd lib
      - pwd

      /usr/lib
    - 상위 디렉토리로 이동
      - cd ..
      - pwd

      /usr
    - 사용자 홈 디렉토리로 이동
      - cd
      - pwd

      /home/adam

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - 디렉토리의 내용을 출력하는 명령
    - 형식: ls [옵션] [디렉토리(파일)]
    - 옵션
      - -a: 숨김 파일을 포함하여 모든 파일의 목록을 출력
      - -d: 디렉토리 자체의 정보를 출력
      - -i: 첫 번째 행에 I-node 번호를 출력
      - -l: 파일의 상세 정보를 출력
      - -A: . 와 .. 를 제외한 모든 파일 목록을 출력
      - -F: 파일의 종류를 표시(*: 실행 파일, /: 디렉토리, @: 심볼릭 링크)
      - -L: 심볼릭 링크 파일의 경우 원본 파일의 정보를 출력
      - -R: 하위 디렉토리의 목록까지 출력

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 현재 디렉토리 내용 확인
      - ◆ 숨김을 제외한 디렉토리 와 파일 표시: ls

      Desktop Documents Downloads Music Pictures Public Templates Videos
      - ◆ 숨김을 포함한 디렉토리와 파일 표시: ls -a

      . .cache .lesshst Public .viminfo
      .. .config .local .ssh
      .bash_history Desktop Music .sudo_as_admin_successful
      .bash_logout Documents Pictures Templates
      .bashrc Downloads .profile Videos
      - ◆ 형식을 같이 표시: ls -F

      Desktop/ Downloads/ Pictures/ Templates/
      Documents/ Music/ Public/ Videos/

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 특정 디렉토리 내용 확인
      - ◆ ls /tmp

snap-private-tmp

systemd-private-b2bed0cde1534991ab4929b320cff443-colord.service-7lUsRF

systemd-private-b2bed0cde1534991ab4929b320cff443-ModemManager.service-y1FT9V

systemd-private-b2bed0cde1534991ab4929b320cff443-spice-vdagentd.service-DS5vHp

systemd-private-b2bed0cde1534991ab4929b320cff443-systemd-logind.service-3bjPd2

systemd-private-b2bed0cde1534991ab4929b320cff443-systemd-resolved.service-3vA53n

systemd-private-b2bed0cde1534991ab4929b320cff443-systemd-timesyncd.service-3JL1UB

systemd-private-b2bed0cde1534991ab4929b320cff443-upower.service-i2PePE

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 상세 정보 출력
      - ◆ ls -l

```
total 32
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Desktop
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Documents
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Downloads
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Music
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Pictures
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Public
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Templates
drwxr-xr-x 2 adam adam 4096 May 29 07:53 Videos
```

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 상세 정보 출력

필드 번호	필드 값	의미
1	d	파일의 종류를 의미 -: 일반 파일. d: 디렉토리 l: 심볼릭 링크. b: 블록 장치 파일 c: 문자 장치 파일 p: 파이프 파일 s: 소켓 파일
2	rwxr-xr-x	소유자, 그룹, 기타 사용자에 대한 접근 권한
3	2	하드 링크의 개수
4	adam	파일 소유자
5	adam	파일이 속한 그룹
6	4096	파일 크기로 바이트 단위
7	May 29 07:53	파일이 마지막으로 수정된 시간
8	Desktop	이름

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 상세 정보 출력
      - ◆ 하드 링크 수는 연결된 디렉토리의 개수로 파일은 무조건 1이고 디렉토리는 기본적으로 2
      - ◆ 하위 디렉토리는 연결된 상위 디렉토리의 개수 만큼 링크 수 증가

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 디렉토리 자체의 정보를 알고자 할 때는 -dl 옵션 사용
      - ◆ adam@adam:~$ ls -dl

        drwxr-x--- 14 adam adam 4096 Jun 25 07:44 .
    - ● ls 명령을 이용해서 파일의 존재 여부를 확인할 수 있는데 파일이 존재하면 파일의 정보를 출력하고 없을 경우 없다는 메시지를 출력
      - ◆ ls .bashrc

        .bashrc
      - ◆ ls game

        ls: cannot access 'game': No such file or directory

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ ls
    - ● 유사한 명령으로 dir 과 vdir 명령이 있음
      - ◆ dir

        Desktop Documents Downloads Music Pictures Public Templates Videos
      - ◆ vdir

        total 32

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Desktop

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Documents

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Downloads

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Music

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Pictures

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Public

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Templates

        drwxr-xr-x 2 adam adam 4096 May 29 07:53 Videos

파일 시스템 관련 명령어

- 디렉토리 관련 명령
  - ✓ mkdir
    - 리눅스에서는 사용자가 필요에 따라 디렉토리를 생성할 수 있는데 아무 곳에나 만들 수 있는 것은 아니고 기본적으로 해당 계정의 홈 디렉토리에 만들 수 있음
    - /tmp 디렉토리와 같이 누구에게나 쓰기 권한이 있는 디렉토리에도 만들 수 있음
    - 다른 디렉토리의 소유자가 쓰기 권한을 부여해주면 그 디렉토리에도 파일이나 디렉토리를 생성할 수 있음
    - 형식: mkdir [옵션] [디렉토리]
    - 옵션은 –p가 있는데 하위 디렉토리를 계층적으로 생성할 때 중간 단계의 디렉토리가 없으면 자동으로 중간 단계 디렉토리를 생성하고 지정한 디렉토리를 생성

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ mkdir
    - ● 하나의 디렉토리 만들기
      - ◆ mkdir temp
      - ◆ ls temp
      - ◆ ls

        Desktop Documents Downloads Music Pictures Public temp Templates Videos
    - ● 여러 개의 디렉토리 만들기
      - ◆ mkdir tmp1 tmp2 tmp3
      - ◆ ls

        Desktop Downloads Pictures temp tmp1 tmp3

        Documents Music Public Templates tmp2 Videos

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ mkdir
    - ● 중간 디렉토리 만들기
      - ◆ mkdir temp/mid/han

        mkdir: cannot create directory ‘temp/mid/han’: No such file or directory
      - ◆ mkdir -p temp/mid/han
      - ◆ ls -R temp

        temp:

        mid

        temp/mid:

        han

        temp/mid/han:

파일 시스템 관련 명령어

- 디렉토리 관련 명령
  - ✓ rmdir
    - 디렉토리를 삭제하는 명령어
    - 형식: rmdir [옵션] [디렉토리]
    - 옵션은 -p 옵션을 이용해서 지정한 디렉토리를 삭제하고 그 디렉토리 와 부모 디렉토리가 빈 디렉토리일 경우 부모 디렉토리도 자동으로 삭제
    - 실습
      - rmdir tmp3
      - ls

        Desktop Downloads Pictures temp tmp1 Videos

        Documents Music Public Templates tmp2
      - rmdir temp

        rmdir: failed to remove 'temp': Directory not empty

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ 실습1
    - 홈 디렉토리로 이동
    - 홈 디렉토리에 linux_ex 디렉토리 생성
    - linux_ex 디렉토리 안에 autoever 디렉토리 생성
    - autoever 디렉토리 안에 one, two, three 디렉토리를 한 번에 생성
    - one 디렉토리 안에 tmp/test 디렉토리를 생성하는데 중간 디렉토리인 tmp는 자동 생성
    - rmdir 명령으로 one 디렉토리 삭제
    - two, three 디렉토리를 동시에 삭제
    - 홈 디렉토리로 이동

파일 시스템 관련 명령어

- ❖ 디렉토리 관련 명령
  - ✓ 실습2
    - 실습 디렉토리 아래의 autoever 디렉토리로 한 번에 이동
    - 현재 작업 디렉토리를 절대 경로로 출력
    - 현재 디렉토리의 숨김 파일을 포함하여 모든 파일을 출력
    - me 디렉토리를 생성
    - me 디렉토리 아래에 you, him 디렉토리를 동시에 생성
    - me 디렉토리를 삭제하는데 삭제되지 않는다면 이유는 무엇인가?
    - 실습을 마치고 홈 디렉토리로 이동

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● cat
      - ◆ 텍스트 형식의 파일을 출력
      - ◆ 텍스트 파일 전체를 열게 되는데 문서가 크면 스크롤을 제공
      - ◆ 매개변수로 여러 개의 파일을 지정해서 여러 개의 파일 출력 가능
      - ◆ -n 옵션을 이용해서 행 번호 출력 가능
      - ◆ 파일을 지정하지 않으면 키보드 입력 내용을 그대로 출력

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● cat
      - ◆ 실습
        - ▪ cat /etc/hostname

```
127.0.0.1 localhost
127.0.1.1 adam

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● cat
      - ◆ 실습
        - ▪ cat -n /etc/hosts

```
1       127.0.0.1 localhost
2       127.0.1.1 adam
3
4       # The following lines are desirable for IPv6 capable hosts
5       ::1     ip6-localhost ip6-loopback
6       fe00::0 ip6-localnet
7       ff00::0 ip6-mcastprefix
8       ff02::1 ip6-allnodes
9       ff02::2 ip6-allrouters
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● cat
      - ◆ 실습
        - ▪ cat /etc/crontab

          내용이 많아서 스크롤 기능을 제공

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● more
      - ◆ 텍스트 형식의 파일을 화면 단위로 출력하고 출력할 내용이 더 있으면 화면 하단에 --More--(0%) 와 같이 알려줌
      - ◆ spacebar를 누르면 다음 페이지를 출력하고 b를 누르면 이전 페이지를 출력하고 중단은 q
      - ◆ /문자열 입력하면 해당 문자열을 찾아 이동
      - ◆ +숫자 옵션을 이용해서 시작할 행 번호 설정 가능
      - ◆ 스크롤 기능이 없음
      - ◆ more /etc/services

```
qotd            17/tcp          quote
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp 21/tcp
--More--(5%)
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● less
      - ◆ less 명령은 more 명령을 약간 개선한 것
      - ◆ 원래 more 명령에서는 스크롤되어 지나간 내용은 다시 볼 수 없었음
      - ◆ 최근 리눅스 버전에서는 이 점을 개선하여 방향키로 이동이 가능하게 함
      - ◆ less 명령을 사용하면 파일 내용의 아래, 위로 방향키나 j, k 키를 이용하여 자유롭게 이동할 수 있다는 장점이 있음
      - ◆ 키와 동작
        - ▪ j, 방향키(아래): 한 행 씩 다음 행으로 스크롤
        - ▪ k, 방향키(위): 한 행 씩 이전 행으로 스크롤
        - ▪ Space Bar, CTRL + f: 다음 화면으로 이동
        - ▪ CTRL + b: 이전 화면으로 이동
        - ▪ q: 종료

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● less
      - ◆ less /etc/services

```
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-
numbers/service-names-port-numbers.xhtml .
#
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.
time            37/udp          timserver
whois           43/tcp          nicname
tacacs          49/tcp                          # Login Host Protocol (TACACS)
/etc/services
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● head/tail
      - ◆ 파일의 앞 부분이나 뒷 부분의 몇 개의 행을 출력하는 명령
      - ◆ 형식: 명령 [옵션] [파일]
      - ◆ tail 명령에서 -f 옵션을 이용하면 실시간으로 변경되는 내용을 출력할 수 있음
      - ◆ -숫자 옵션을 이용하면 숫자 만큼의 행을 출력하는데 숫자 옵션을 생략하면 10개의 행을 출력
      - ◆ tail /etc/services

```
sgi-cad         17004/tcp       # Cluster Admin daemon
binkp           24554/tcp       # binkp fidonet protocol
asp             27374/tcp       # Address Search Protocol
asp             27374/udp
csync2          30865/tcp       # cluster synchronization tool
dircproxy   57000/tcp           # Detachable IRC Proxy
tfido           60177/tcp       # fidonet EMSI over telnet
fido            60179/tcp       # fidonet EMSI over TCP

# Local services
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 내용 출력
    - ● head/tail
      - ◆ tail -7 /etc/services

```
asp                     27374/udp
csync2                  30865/tcp               # cluster synchronization tool
dircproxy   57000/tcp                           # Detachable IRC Proxy
tfido                   60177/tcp               # fidonet EMSI over telnet
fido                    60179/tcp               # fidonet EMSI over TCP

# Local services
```

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 실습
    - less 명령으로 /etc/services 파일의 내용을 출력
    - spacebar를 눌러 다음 페이지로 이동
    - k 키를 네 번 눌러 위로 다시 이동
    - /TCP 를 입력하여 파일 내용 중에서 문자열 TCP가 있는 곳을 조회
    - n 키를 눌러 문자열 TCP를 계속 조회
    - q 키를 눌러 less 명령을 종료

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 실습
    - more 명령으로 /etc/services 파일을 출력
    - 파일 내용에서 HTTP를 검색
    - 다른 곳에도 HTTP 가 있는지 추가로 확인
    - more 명령을 종료

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 및 디렉토리 복사
    - 형식
      cp [옵션] [파일1 (디렉토리 1)] [파일2 (디렉토리 2)]
    - 옵션
      - ◆ -i: 파일2가 이미 존재하면 덮어쓸 것인지 물어봄
      - ◆ -r: 디렉토리를 복사할 때 지정
    - 경로가 존재하는 경우 덮어 씌우므로 주의
    - -i 옵션을 이용하면 동일한 파일이 존재하는 경우 대화형으로 작업 가능
    - 여러 개의 파일을 한꺼번에 복사가 가능한데 그 경우 맨 마지막은 복사될 디렉토리 경로(파일 경로는 안됨)
    - 실습
      - ◆ 파일 복사

        cp /etc/hosts text1
      - ◆ 파일을 디렉토리로 복사

        cp text1 temp

        ls temp

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 및 디렉토리 복사
    - ● 실습
      - ◆ 복사 실패

        cp text1 /etc
      - ◆ 여러 개 파일 복사

        cp /etc/hosts /etc/services temp

        ls temp
      - ◆ 덮어쓰기 전에 묻기

        cp -i /etc/hosts text1
      - ◆ 디렉토리 복사

        cp temp temp2

        cp -r temp temp2

        ls temp2

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 이동 과 파일 이름 변경
    - 형식
      mv [옵션] [파일1 (디렉토리 1)] [파일2 (디렉토리 2)]
    - 옵션
      - ◆ -i: 파일2가 이미 존재하면 덮어쓸 것인지 물어봄
    - 경로가 존재하는 경우 덮어 씌우므로 주의
    - mv 명령의 첫 번째 인자로는 원본 파일이나 디렉토리를 지정하고 두 번째 인자로는 목적지 파일이나 디렉토리를 지정
    - 파일 이름 바꾸기
      - ◆ mv text1 data1
      - ◆ ls

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 이동 과 파일 이름 변경
    - ● 다른 디렉토리로 이동
      - ◆ mv data1 temp
      - ◆ ls

        sample.txt temp temp2 tmp1 tmp2
      - ◆ ls temp

        data1 hosts mid services text1
    - ● 여러 파일 이동
      - ◆ mv temp/data1 temp/text1 .
      - ◆ ls

        data1 sample.txt temp temp2 text1 tmp1 tmp2
    - ● 디렉토리 이름 변경
      - ◆ mv temp2 temp3
      - ◆ ls

        data1 sample.txt temp temp3 text1 tmp1 tmp2

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 및 디렉토리 삭제
    - 형식
      - ◆ rm [옵션] [파일 도는 디렉토리]
    - 옵션
      - ◆ i: 파일을 정말 삭제할 것인지 확인
      - ◆ r: 디렉토리 삭제
    - 실습
      - ◆ data1 파일 삭제
        - ▪ ls

          data1 sample.txt temp temp3 text1 tmp1 tmp2
        - ▪ rm data1
        - ▪ ls

          sample.txt temp temp3 text1 tmp1 tmp2

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 파일 및 디렉토리 삭제
    - ● 실습
      - ◆ temp3 디렉토리 삭제
        - ▪ rm temp3

          rm: cannot remove 'temp3': Is a directory
        - ▪ rm -r temp3
        - ▪ ls

          sample.txt temp text1 tmp1 tmp2

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 실습
    - /etc/hosts 파일을 test.org로 복사
    - /test 디렉토리 생성
    - test.org 파일을 test 디렉토리로 복사
    - test 디렉토리의 test.org 파일을 test.bak로 이름 변경
    - test.org 파일 삭제
    - test 디렉토리의 test.bak 파일을 test.org로 복사
    - test 디렉토리 삭제

파일 시스템 관련 명령어

- ❖ 파일 관련 명령
  - ✓ 실습
    - test.org 파일을 test.txt로 복사
    - backup 디렉토리를 생성
    - text.txt 파일을 backup 디렉토리로 이동
    - backup 디렉토리의 이름을 work로 변경
    - rmdir로 work 디렉토리를 삭제
    - rm 명령으로 work 디렉토리를 삭제
    - 정말로 삭제할 것인지 물어보게 하려면 어떻게 해야 하는가?

파일 시스템 관련 명령어

- ln
  - ✓ 링크를 생성하는 명령
  - ✓ ln [옵션] <링크할 파일> <링크 이름>
  - ✓ 파일에 별명을 붙이는 것
  - ✓ 링크를 생성하는 이유
    - 긴 경로를 짧게 만들기 위해서
    - 여러 버전의 프로그램이 존재하는 경우 이름을 구분하기 위해서

파일 시스템 관련 명령어

- ln
  - ✓ 하드 링크
    - 하드 링크란 한 파일 원본에 이름을 여러 개 붙이는 기능
    - 양쪽이 모두 원본 파일
    - 실습
      - cp /etc/services data1
      - ls -l

```
total 40
-rw-r--r-- 1 adam adam 12813 Sep 11 04:53 data1
-rw-rw-r-- 1 adam adam    70 Sep  5 22:26 sample.txt
drwxrwxr-x 3 adam adam  4096 Sep 10 23:45 temp
-rw-r--r-- 1 adam adam   222 Sep 11 04:38 test.org
-rw-r--r-- 1 adam adam   222 Sep 10 23:35 text1
drwxrwxr-x 2 adam adam  4096 Sep  9 23:24 tmp1
drwxrwxr-x 2 adam adam  4096 Sep  9 23:24 tmp2
```

파일 시스템 관련 명령어

- ln
  - ✓ 하드 링크
    - ● 실습
      - ◆ ln data1 data1.ln
      - ◆ ls -l

```
total 56
-rw-r--r-- 2 adam adam 12813 Sep 11 04:53 data1
-rw-r--r-- 2 adam adam 12813 Sep 11 04:53 data1.ln
-rw-rw-r-- 1 adam adam    70 Sep  5 22:26 sample.txt
drwxrwxr-x 3 adam adam  4096 Sep 10 23:45 temp
-rw-r--r-- 1 adam adam   222 Sep 11 04:38 test.org
-rw-r--r-- 1 adam adam   222 Sep 10 23:35 text1
drwxrwxr-x 2 adam adam  4096 Sep  9 23:24 tmp1
drwxrwxr-x 2 adam adam  4096 Sep  9 23:24 tmp2
```

        - ▪ data1.ln 파일이 생성되고 하드 링크의 개수가 증가

파일 시스템 관련 명령어

- ln
  - ✓ 하드 링크
    - ● 실습
      - ◆ ls -i

524321 data1 524300 sample.txt 524317 test.org 524312 tmp1

524321 data1.ln 524311 temp 524318 text1 524313 tmp2

- data1.ln 파일 과 data1 파일의 i-node 가 동일
- rm 명령은 하드 링크의 개수를 1 줄여주는 명령
- 하드 링크의 값이 0이 되면 실제로 i-node 와 데이터 블록을 삭제

파일 시스템 관련 명령어

- ln
  - ✓ 심볼릭 링크
    - 디렉토리는 하드 링크할 수 없으며 하드 링크는 서로 다른 디스크에 걸쳐서 만들 수 없다는 제한이 있음
    - 심볼릭 링크는 제한없이 생성 가능
    - 심볼링 링크를 만들 때 –s 옵션을 추가해서 생성
    - 원본 파일에 대한 정보가 담긴 작은 특수 파일이며 하드 링크와 달리 원본과 구별
    - 심볼릭 링크를 삭제하는 것은 원본에는 아무런 영향을 주지 않지만 원본을 삭제하면 심볼릭 링크는 깨진 상태가 됨

파일 시스템 관련 명령어

- ln
  - ✓ 심볼릭 링크
    - ● 실습
      - ◆ ln -s data1 data1.sl
      - ◆ ls -i

        524321 data1 524322 data1.sl 524311 temp 524318 text1 524313 tmp2

        524321 data1.ln 524300 sample.txt 524317 test.org 524312 tmp1

        - ▪ i - node 가 다름
      - ◆ ls -l data1.sl

        lrwxrwxrwx 1 adam adam 5 Sep 11 08:14 data1.sl -> data1

        - ▪ 원본 파일을 확인할 수 있음

파일 시스템 관련 명령어

- ln
  - ✓ 심볼릭 링크
    - 특징
      - 파일의 종류가 l로 표시
      - 하드 링크의 개수가 변하지 않음
      - 원본 파일에 이름을 추가하는 것이 아님
      - 파일명 뒤에 원본 파일의 이름 표시
      - inode 번호가 원본 파일과 다름
      - 원본 파일과 심볼릭 링크 파일은 별개의 파일
      - 심볼릭 링크의 파일 내용은 원본 파일의 경로를 가지고 있는데 ls -1 datal.sl 명령을 실행한 결과에서 파일의 크기는 5로 나오는데 크기가 5라는 것은 datal.sl 파일의 내용이 5바이트라는 의미이며 이것은 원본 파일 경로의 길이
      - 원본 파일의 경로는 ln -s로 심볼릭 링크를 생성할 때 지정한 경로가 저장되지만 심볼릭 링크의 내용을 출력하면 원본 파일의 경로가 출력되는 것이 아니라 원본 파일의 내용이 출력되는데 이것이 심볼릭 링크의 특징
      - 심볼릭 링크를 열고 수정하면 원본 파일이 수정됨

파일 시스템 관련 명령어

- ln
  - ✓ 실습
    - test.org 파일의 하드 링크로 test.ln을 생성
    - test.org 파일의 하드 링크로 test.ln2를 생성
    - i-node 확인
    - test.org 와 test.ln, test.ln2의 하드 링크 개수를 확인
    - temp 디렉토리에 대한 심볼릭 링크로 tmp를 생성
    - temp와 tmp 디렉토리의 내용이 같다는 것을 확인
    - tmp 디렉토리 와 test.ln, test.ln2, test.org 파일을 모두 삭제

파일 시스템 관련 명령어

- ln
  - ✓ 실습
    - /etc/hosts 파일을 복사하여 work 파일을 생성
    - work 파일의 하드 링크로 work.ln 생성
    - work 파일의 심볼릭 링크로 work.sl 생성
    - work 파일을 복사하여 work.cp 파일을 생성
    - work.sl 파일을 편집기로 열어서 마지막 행을 삭제
    - work, work.ln, work.sl, work.cp 파일의 내용을 확인
    - 내용이 같지 않은 파일은 어느 것?
    - work, work.ln, work.sl, work.cp 파일을 삭제

파일 시스템 관련 명령어

- touch
  - ✓ 빈 파일을 생성하는 명령
  - ✓ 형식

    touch [-acm] [-r ref_file I -t time] [파일]
  - ✓ 옵션
    - -a: 접근 시간만 변경
    - -m: 수정 시간만 변경
    - -t [[CC]YY]MMDDhhmm[.ss]: 시간을 직접 입력

파일 시스템 관련 명령어

- ❖ touch
  - ✓ 실습
    - touch test
    - ls -l test

      -rw-rw-r-- 1 adam adam 0 Sep 11 22:28 test
    - ls -l data1

      -rw-r--r-- 2 adam adam 12813 Sep 11 04:53 data1
    - date

      Wed Sep 11 10:28:39 PM UTC 2024
    - touch data1
    - ls -l data1

      -rw-r--r-- 2 adam adam 12813 Sep 11 22:28 data1

파일 시스템 관련 명령어

- touch
  - ✓ 시간 표시
    - 형식: [[CC]YY]MMDDhhmm[.ss]
      - CC: 연도의 첫 두 자리
      - YY: 연도의 마지막 두 자리
      - MM: 달(01~12 범위 내 지정)
      - DD: 날짜(01~31 범위 내 지정)
      - hh: 시간(00~23 범위 내 지정)
      - mm: 분(00~59 범위 내 지정)
      - ss: 초(00~59 범위 내 지정)
      - 년도 지정
        - YY를 69-99를 설정하면 CC는 19가 설정되고 나머지 경우는 CC가 20으로 설정됨

파일 시스템 관련 명령어

- touch
  - ✓ 시간 설정
    - ls -l test

      -rw-rw-r-- 1 adam adam 0 Sep 11 22:28 test
    - touch -t 12311200 test
    - ls -l test

      -rw-rw-r-- 1 adam adam 0 Dec 31 2024 test

파일 시스템 관련 명령어

- 파일 내용 검색
  - ✓ grep 명령은 텍스트에서 특정 문자열을 검색할 때 사용하는 명령으로 정규 표현식을 이용하여 복잡한 검색도 가능
  - ✓ 형식

    grep [옵션] [패턴] [파일]
  - ✓ 옵션
    - i: 대문자 및 소문자를 모두 검색
    - l: 지정한 패턴이 포함된 파일명을 출력
    - n: 행 번호를 출력
  - ✓ 실습
    - cp /etc/services data
    - grep NNTP data

      nntps 563/tcp snntp # NNTP over SSL
    - grep -n NNTP data

      121:nntps 563/tcp snntp # NNTP over SSL

파일 시스템 관련 명령어

- find
  - ✓ find 명령은 리눅스의 디렉토리 계층 구조에서 특정 파일이 어느 디렉토리에 있는지 찾아주는 명령으로 파일의 생성 일자와 이름, 소유자 등 다양한 조건에 맞는 파일을 검색
  - ✓ 형식

    find [경로] [검색 조건] [동작]
  - ✓ 검색 조건
    - -name filename: 파일명으로 검색
    - -type 파일 종류: 파일 종류로 검색
    - -user loginID: 지정한 사용자가 소유한 모든 파일을 검색
    - -group 그룹이름: 그룹을 기준으로 검색
    - -perm 접근 권한: 지정한 사용 권한과 일치하는 파일을 검색
    - -size: 크기로 검색
    - -newer: 수정 시간을 기준으로 특정 파일보다 더 최근에 수정된 파일을 검색
  - ✓ 동작
    - -exec 명령 {} ₩;: 검색된 파일에 명령을 실행
    - -ok 명령 {} ₩;: 사용자의 확인을 받아서 명령을 실행
    - -print: 검색된 파일의 절대 경로명을 화면에 출력 – 기본 동작
    - -ls: 검색 결과를 긴 목록 형식으로 출력

파일 시스템 관련 명령어

- find
  - ✓ 이름으로 찾기
    - -name은 대소문자를 구별하고 –iname은 대소문자를 구별하지 않음
    - 와일드카드 문자를 사용하는 것이 가능
    - 실습
      - mkdir -p dir1/doc
      - touch file-1.txt dir1/doc/file-1.txt
      - find . -name file-1.txt -print

        ./file-1.txt

        ./dir1/doc/file-1.txt
      - find . -name '*.txt' -print

        ./file-1.txt

        ./dir1/doc/file-1.txt

파일 시스템 관련 명령어

- find
  - ✓ 타입으로 찾기
    - 파일 형식
      - -type f
      - -type d
      - -type l
    - 실습
      - find . -type d -print

        .

        ./dir1

        ./dir1/doc

      - find . -type f -name '*.txt' -print

        ./file-1.txt

        ./dir1/doc/file-1.txt

파일 시스템 관련 명령어

- find
  - ✓ 시간으로 찾기
    - 참조 파일보다 더 최근에 수정된 파일 검색
      - find . -newer reference.txt
    - 특정 날짜 이후에 수정된 파일 검색
      - find . -newermt "2023-01-01"
    - 파일 접근 시간 기준으로 검색
      - find . -neweraa reference.txt
    - 두 파일 사이에 수정된 파일 검색
      - find . -newer reference1.txt ! -newer reference2.txt

파일 시스템 관련 명령어

- find
  - ✓ 명령 실행
    - find /tmp -user user1 -exec rm {} ₩;
    - find /tmp -user user1 -ok rm {} ₩;

파일 시스템 관련 명령어

- find
  - ✓ 기타
    - 1M 이상 파일 찾기: find /usr/share -size 1M
    - /usr/bin 디렉토리 하위에서 파일 크기가 10KB~100KB인 파일 찾기: find /usr/bin -size +10k -size -100k
    - /home에서 user가 adam 인 파일을 list 형식으로 출력: find /home -user adam -ls
    - /home에서 user가 adam, itstudy 인 파일을 list 형식으로 출력: find /home -user adam -or -user itstudy -ls
    - /etc에서 group이 ntp 인 파일을 list 형식으로 출력: find /home -group ntp –ls
    - 현재 사용자의 홈 디렉토리 하위에서 허가권이 64인 파일 찾기: find ~ -perm 644
    - /etc에서 수정된 지 10분 미만인 파일을 출력: find /etc/ -mmin -10
    - /bin, /usr/bin, /sbin, /usr/sbin에서 3일 동안 수정된 적이 있는 파일들을 출력: find /bin /usr/bin /sbin /usr/sbin -ctime -3
    - /etc에서 iptables가 담긴 파일들의 경로를 모두 찾아 I found를 붙여 출력: find /etc -iname iptables -exec echo "I found {}" ₩;
    - 현재 사용자의 홈 디렉토리 하위에서 파일 크기가 0인 파일의 목록을 상세히 출력: find ~ -size 0k -exec ls -l { } ₩;
    - /home 홈 디렉토리 하위에서 확장명이 *.swp인 파일 삭제: find /home -name "*.swp" -exec rm { } ₩;

파일 시스템 관련 명령어

- ❖ locate
  - ✓ 개요
    - ● 리눅스 시스템 내 존재하는 특정 파일 및 디렉토리를 검색하는 명령어
  - ✓ find 와 차이

	locate	find
특징	빠른 검색 속도 한정적인 검색 옵션 파일이나 디렉토리를 데이터베이스화 해서 파일을 검색 이미 생성된 DB 파일을 이용(updated 명령어를 통한 갱신 필요)	느림 검색 속도 다양한 검색 옵션 Real System에서 파일을 검색해서 경로를 출력 Always up-to-date 접근 제한 파일 및 디렉토리는 검색 불가
이용 사례	파일이 존재하는 것은 알고 있으나 파일 위치가 어디인지 모를 경우	검색과 더불어 다른 작업을 수행할 때(퍼미션 변경, 찾은 파일 모두 삭제 등)

파일 시스템 관련 명령어

- ❖ locate
  - ✓ 설치
    - 확인: locate –version
    - 설치: sudo apt-get install plocate
  - ✓ 데이터베이스 구축: sudo updatedb
    - locate 명령어는 이미 생성된 데이터베이스를 통해 검색을 진행하기에 파일 검색 측면에서 빠른 속도를 보이지만 파일이나 디렉토리를 데이터베이스화 하여 사용하므로 정보를 갱신하지 않는다면 부정확한 데이터를 출력할 가능성이 존재
    - 사용하기 전에 sudo updatedb 명령어를 실행 후 검색을 하는 것을 권장
    - NFS를 이용중인 경우 update 명령어로 인해 마운트된 영역이 느려질 수 있고 최악의 경우(거의 일어나진 않지만) eviction(축출)까지 이루어 질 수 때문에 NFS 환경을 이용하는 경우 updatedb 명령어를 지양하는 것을 권장
  - ✓ 파일 검색

    locate [Option] <Pattern>

파일 시스템 관련 명령어

- locate
  - ✓ 실습
    - bash 라는 문자열이 포함된 경로 검색: locate bash
    - 확장자가 sed인 파일 검색: locate "*.sed"
    - 대소문자 구분없이 검색: locate –i notes
    - 파일 이름만으로 검색: locate -b python
    - 여러 개의 패턴 모두 일치하는 경우 검색: locate -A bash docs
    - 정규식을 이용한 검색: locate --regex -i "(₩.mp4|₩.txt)"

파일 시스템 관련 명령어

- wc
  - ✓ 개요
    - wc 명령은 파일 또는 표준 입력에서 단어 수, 줄 수, 문자 수 등을 셀 때 사용하는 명령어
    - wc는 word count의 약자
    - 파일의 크기나 텍스트의 분포를 확인할 때 유용하게 사용
    - 기본 사용
      - wc [옵션] [파일...]
    - 옵션
      - -l: 줄 수(line count)
      - -w: 단어 수(word count)
      - -c: 바이트 수(byte count)
      - -m: 문자 수(character count)
      - -L: 가장 긴 줄의 길이(maximum line length)
    - 출력
      - <줄 수> <단어 수> <문자 수> <파일 이름>
      - 여러 파일을 입력하면 각 파일에 대해 개별적으로 결과가 출력되며 마지막에 전체 합계를 표시

파일 시스템 관련 명령어

- wc
  - ✓ 실습
    - 파일의 줄 수, 단어 수, 문자 수 출력: wc file.txt
    - 파일의 줄 수만 출력: wc -l file.txt
    - 파일의 단어 수만 출력: wc -w file.txt
    - 파일의 문자 수만 출력: wc -c file.txt
    - 파일의 가장 긴 줄 길이만 출력: wc -L file.txt
    - 여러 파일에 대한 통계 출력: wc file1.txt file2.txt

```
 5      15    100 file1.txt
 8      25    150 file2.txt
13      40    250 total
```

파일 시스템 관련 명령어

- sort
  - ✓ 개요
    - 데이터를 정렬하는 데 사용되는 명령어
    - 기본적으로 오름차순으로 정렬하지만 다양한 옵션을 통해 내림차순 정렬, 숫자 정렬 등 다양한 방식으로 데이터를 정렬할 수 있음
    - 기본 사용법
      - sort [옵션] [파일...]
    - 옵션
      - -r: 내림차순으로 정렬(reverse)
      - -n: 숫자 크기 순으로 정렬(numeric sort)
      - -k: 특정 필드를 기준으로 정렬(key)
      - -u: 중복된 행을 제거(unique)
      - -t: 필드 구분자를 지정(field separator)
      - -M: 월 이름을 기준으로 정렬(month sort)
      - -b: 공백을 무시하고 정렬(ignore leading spaces)

파일 시스템 관련 명령어

- sort
  - ✓ 실습
    - file.txt

      park,17

      kim,2

      Lee,52

      choi,9

      Lee,52

파일 시스템 관련 명령어

- ❖ sort
  - ✓ 실습
    - sort file.txt

      choi,9

      kim,2

      Lee,52

      Lee,52

      park,17
    - sort -r file.txt

      park,17

      Lee,52

      Lee,52

      kim,2

      choi,9

파일 시스템 관련 명령어

- ❖ sort
  - ✓ 실습
    - sort -t ',' -k 2 file.txt

      park,17
      kim,2
      Lee,52
      Lee,52
      choi,9
    - sort -t ',' -k 2 -n file.txt

      kim,2
      choi,9
      park,17
      Lee,52
      Lee,52

파일 시스템 관련 명령어

- sort
  - ✓ 실습
    - sort -t ',' -k 2 -nu file.txt

      kim,2

      choi,9

      park,17

      Lee,52

파일 시스템 관련 명령어

- awk
  - ✓ 개요
    - 텍스트 처리 도구
    - 패턴을 기반으로 텍스트 파일, 스트림 및 기타 데이터 형식을 조작하고 처리
    - 기본 형식

      awk 옵션 'pattern { action }' file
    - 옵션

      -F 구분자

파일 시스템 관련 명령어

- awk
  - ✓ 실습
    - file.txt

      홍길동 25 서울

      이순신 30 부산

      김유신 27 대전
    - 이름 과 나이만 출력

      awk '{ print $1, $2 }' file.txt

      홍길동 25

      이순신 30

      김유신 27

파일 시스템 관련 명령어

- awk
  - ✓ 실습
    - 특정 패턴으로 조회

      awk '/이/ { print $0 }' file.txt

      이순신 30 부산
    - 인코딩 확인

      file -i file.txt

      file.txt: text/plain; charset=utf-8
    - 인코딩 변경

      iconv -f 기존_인코딩 -t UTF-8 기존파일.txt > 새파일.txt

파일 시스템 관련 명령어

- sed
  - ✓ 개요
    - 스트림 편집기(stream editor)로 파일이나 입력 스트림의 텍스트를 변환하거나 조작하는 데 사용
    - 파일 수정, 텍스트 치환, 라인 삭제 등 다양한 작업에 유용
    - 기본 형식

      sed [옵션] 스크립트 [파일]
    - 옵션
      - -n: 기본 출력을 억제
      - -i: 파일을 직접 수정
      - -r: 확장된 정규 표현식을 활성화
      - -e: 여러 sed 명령을 순차적으로 실행
      - -f FILE: 스크립트를 파일에서 읽어오기

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● file.txt

      how old are you

      Old old

      old

      Hello

      hello

      12

      2

      3

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 각 줄에서 첫 번째로 등장하는 "old"를 "new"로 대체

      sed 's/old/new/' file.txt

      how new are you
      Old new
      new
      Hello
      hello
      12
      2
      3

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 각 줄에서 등장하는 "old"를 "new"로 모두 대체

```
sed 's/old/new/g' file.txt
```

```
how new are you
Old new
new
Hello
hello
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 패턴이 존재하는 라인 삭제

      sed '/old/d' file.txt

      Hello
      hello
      12
      2
      3

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 특정 줄 만 출력

      sed -n '2,4p' file.txt

      Old old

      old

      Hello
    - ● 빈 줄 삭제

      sed '/^$/d' file.txt

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 패턴 앞에 텍스트 삽입

```
sed '/old/i Text to insert' file.txt
```

```
Text to insert
how old are you
Text to insert
Old old
Text to insert
old
Hello
hello
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 패턴 뒤에 텍스트 삽입

```
sed '/old/a Text to insert' file.txt
```

```
how old are you
Text to insert
Old old
Text to insert
old
Text to insert
Hello
hello
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 3번째 줄 만 치환

```
sed '3s/old/new/' file.txt
```

```
how old are you
Old old
new
Hello
hello
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 정규 표현식 사용

```
sed -r 's/[0-9]+/NUMBER/g' file.txt
```

```
how old are you
Old old
old
Hello
hello
NUMBER
NUMBER
NUMBER
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 대소문자 무시

```
sed 's/hello/hi/I' file.txt
```

```
how old are you
Old old
old
hi
hi
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 파일에 직접 수정

```
sed -i 's/old/new/g' file.txt
cat file.txt

how new are you
Old new
new
Hello
hello
12
2
3
```

파일 시스템 관련 명령어

- ❖ sed
  - ✓ 실습
    - ● 파이프와 함께 사용

```
cat file.txt | sed 's/new/old/g'
```

```
how old are you
Old old
old
Hello
hello
12
2
3
```

파일 접근 권한 관리

- 개요
  - ✓ 리눅스와 같은 다중 사용자 시스템은 사용자의 파일에 마음대로 접근할 수 없도록 보안 기능을 제공
  - ✓ 사용자는 자신의 파일과 디렉토리 중에서 다른 사용자가 접근해도 되는 것과 그렇지 않은 것을 구분하여 접근 권한을 제한할 수 있음
  - ✓ 접근 권한은 파일이 가지고 있는 속성 중 하나
  - ✓ 파일의 접근 권한을 확인하는 명령은 ls –l
    - adam@adam:/home$ ls -l /etc/hosts

      -rw-r--r-- 1 root root 219 Sep 25 06:37 /etc/hosts
      - 파일의 속성
        - - : 파일의 종류(-는 일반 파일이고 d는 디렉토리)
        - rw-r--r-- : 파일을 읽고 쓸 수 있는 접근 권한 표시
        - 1 : 하드 링크 개수
        - root : 파일 소유자의 로그인 ID
        - adm : 파일이 속한 그룹 이름
        - 219 : 파일의 크기
        - Sep 25 06:37 : 파일이 마지막으로 수정된 날짜
        - /etc/hosts : 파일 이름

파일 접근 권한 관리

- ❖ 개요
  - ✓ 파일이 속한 그룹 이름
    - 리눅스에서 사용자는 기본적으로 하나 이상의 그룹에 속해 있는데 ls –l 명령에서 출력되는 그룹명은 파일이 속한 그룹의 이름
    - 그룹에 속한 사용자들에게 권한을 부여하여 파일을 공유할 수 있음
    - 사용자가 속한 기본 그룹은 시스템 관리자가 사용자를 등록할 때 결정
    - 사용자가 임의로 자신의 그룹을 바꿀 수 없으며 반드시 시스템 관리자에게 변경을 요청
    - 그룹이 정의된 파일은 /etc/group으로 시스템 관리자만 수정할 수 있음
    - 사용자가 속한 그룹을 알려주는 명령은 groups
      - ◆ groups [사용자 명]
        - adam@itstudy:~$ groups

          adam adm cdrom sudo dip plugdev lxd

파일 접근 권한 관리

- ❖ 파일 접근 권한
  - ✓ 권한 종류
    - 읽기 권한: 파일의 내용을 읽거나 복사할 수 있으며 디렉토리의 경우는 ls 명령으로 디렉토리의 목록은 볼 수 있지만 옵션은 실행 권한이 있어야 함
    - 쓰기 권한: 파일을 수정, 이동, 삭제할 수 있으며 디렉토리 안에 파일을 생성하거나 삭제할 수 있음
    - 실행 권한: 파일을 실행할 수 있으며 디렉토리의 경우는 cd 명령을 사용할 수 있음

파일 접근 권한 관리

- 파일 접근 권한
  - ✓ 표기 방법
    - 읽기 권한은 r, 쓰기 권한은 w, 실행 권한은 x로 나타내며 해당 권한이 없는 경우에는 - 로 표기
    - 사용자 카테고리 별로 세 가지 권한의 부여 여부를 rwx 세 문자를 묶어서 표기하는데 사용자 카테고리가 세 개이고 권한도 세 개이므로 총 아홉 개의 문자로 표현
    - adam@itstudy:~$ ls -l /etc/hosts

      -rw-r--r-- 1 root root 222 Sep 5 07:55 /etc/hosts
      - 맨 앞의 - 은 파일의 종류를 나타내는 부분으로 접근 권한과 관계가 없으며 실제 접근 권한은 - 을 제외한 rw-r--r--
      - 이는 사용자 카테고리별로 세 문자씩 한 묶음으로 표기한 것
      - 소유자의 권한은 rw-이고 그룹의 권한은 중간의 r-- 이며 기타 사용자의 권한은 마지막의 r--
      - 소유자가 읽기와 쓰기 권한을 가지고 있고 그룹과 기타 사용자는 읽기 권한만 가지고 있음을 나타냄

파일 접근 권한 관리

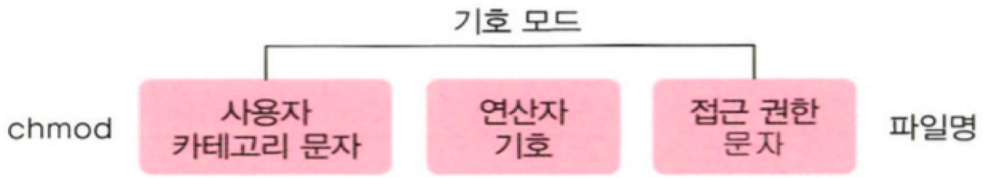
- 파일 접근 권한 변경 명령
  - ✓ 형식: chmod [옵션] 권한 [파일 또는 디렉토리 경로]
  - ✓ 옵션은 -R 이 있는데 하위 디렉토리까지 모두 변경
  - ✓ 접근 권한 설정 방법
    - 기호 모드: 접근 권한을 변경하기 위해서 기호를 사용
    - 숫자 모드: 접근 권한을 변경하기 위해서 숫자를 사용

파일 접근 권한 관리

❖ 파일 접근 권한 변경 명령

✓ 기호를 이용한 파일 접근 권한 변경

● 형식



● 문자 와 기호

사용자 카테고리 문자	u	파일 소유자
	g	파일 소유 그룹
	o	소유자와 그룹 이외의 기타 사용자
	a	전체 사용자
연산자 기호	+	권한 부여
	−	권한 제거
	=	접근 권한 설정
접근 권한 문자	r	읽기 권한
	w	쓰기 권한
	x	실행 권한

파일 접근 권한 관리

❖ 파일 접근 권한 변경 명령

✓ 기호를 이용한 파일 접근 권한 변경

● 설정 예시

u+w	소유자(u)에게 쓰기(w) 권한 부여(+)
u-x	소유자(u)의 실행(x) 권한 제거(-)
g+w	그룹(g)에 쓰기(w) 권한 부여(+)
o-r	기타 사용자(o)의 읽기(r) 권한 제거(-)
g+wx	그룹(g)에 쓰기(w)와 실행(x) 권한 부여(+)
+wx	모든 사용자에게 쓰기(w)와 실행(x) 권한 부여(+)
a+rwx	모든 사용자에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(+)
u=rwx	소유자(u)에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(=)
go+w	그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)
u+x,go+w	소유자(u)에게 실행(x) 권한을 부여하고(+) 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 기호를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 실습 디렉토리 및 파일 복사

```
adam@adam:~$ mkdir ex
adam@adam:~$ cd ex
adam@adam:~/ex$ cp /etc/hosts test.txt
```

      - ◆ 접근 권한 확인

```
adam@adam:~$ ls -l
        total 4
        -rw-r--r-- 1 adam adam 219 Sep 26 09:23 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 기호를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 그룹에 쓰기 와 실행 권한 부여

```
adam@adam:~/ex$ chmod g+wx test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rw-rwxr-- 1 adam adam 222 Sep 17 05:57 test.txt
```

      - ◆ 기타 사용자에게 실행 권한을 부여

```
adam@adam:~/ex$ chmod o+x test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rw-rwxr-x 1 adam adam 222 Sep 17 05:57 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 기호를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 그룹 과 기타 사용자의 실행 권한을 제거

```
adam@adam:~/ex$ chmod go-x test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rw-rw-r-- 1 adam adam 222 Sep 17 05:57 test.txt
```

      - ◆ 모두에게 실행 권한을 부여

```
adam@adam:~/ex$ chmod a+x test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rwxrwxr-x 1 adam adam 222 Sep 17 05:57 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 기호를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거

```
adam@adam:~/ex$ chmod u+w,g-w test.txt
adam@adam:~/ex$ ls -l
    total 4
    -rwxr-xr-x 1 adam adam 219 Sep 26 09:23 test.txt
```

파일 접근 권한 관리

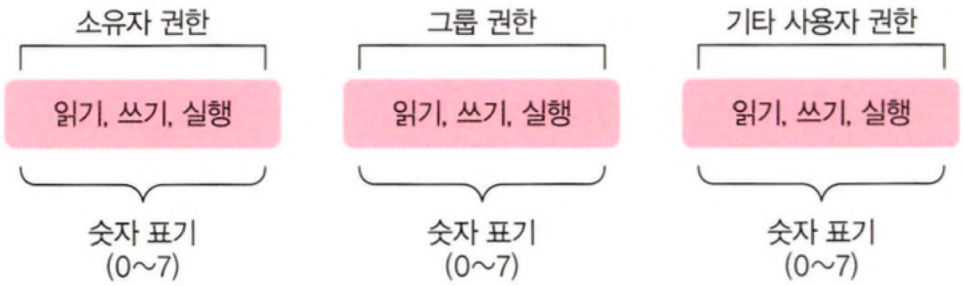
- ❖ 파일 접근 권한 변경 명령
  - ✓ 연습문제
    - test.txt 파일을 현재 접근 권한 상태에서 그대로 사용
    - 모든 사용자의 실행 권한을 한 번에 제거
    - 그룹과 기타 사용자에게 쓰기 권한을 한 번에 부여
    - 그룹과 기타 사용자의 읽기, 쓰기 권한을 한 번에 모두 제거
    - 소유자의 쓰기 권한을 제거
    - 소유자의 읽기 권한을 제거
    - 소유자에게는 읽기와 쓰기 권한을, 그룹과 기타 사용자에게는 읽기 권한을 한 번에 부여

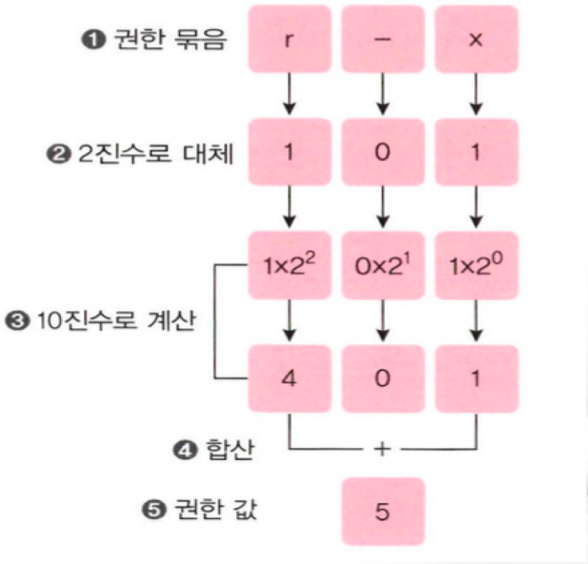
파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - chmod 명령은 문자와 기호를 사용하는 기호 모드 외에 숫자를 사용하여 접근 권한을 변경할 수 있는 숫자 모드를 제공
    - 소유자, 그룹, 기타 사용자 별로 권한을 부여하거나 제거할 때는 기호 모드가 편리하지만 전체적으로 권한을 조정할 때는 문자의 조합이 복잡
    - 숫자 모드로 chmod 명령을 사용하면 소유자, 그룹, 기타 사용자의 권한을 한 번에 원하는 대로 변경할 수 있어서 매우 편리

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 숫자 환산





파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 대응 관계

접근 권한	환산	숫자	의미
rwx	111 → 4+2+1	7	읽기, 쓰기, 실행
rw-	110 → 4+2+0	6	읽기, 쓰기
r-x	101 → 4+0+1	5	읽기, 실행
r--	100 → 4+0+0	4	읽기
-wx	011 → 0+2+1	3	쓰기, 실행
-w-	010 → 0+2+0	2	쓰기
--x	001 → 0+0+1	1	실행
---	000 → 0+0+0	0	권한이 없음

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 숫자로 표현한 접근 권한의 예

접근 권한	숫자 모드	접근 권한	숫자 모드
rwxrwxrwx	777	rw-r--r--	644
rwxr-xr-x	755	rwx------	700
rw-rw-rw	666	rw-r-----	640
r-xr-xr-x	555	r--------	400

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 모든 사용자에게 읽기 권한만 부여

```
adam@adam:~/ex$ chmod 444 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -r--r--r-- 1 adam adam 219 Sep 26 09:23 test.txt
```

      - ◆ 그룹에 쓰기 와 실행 권한 부여

```
adam@adam:~/ex$ chmod 474 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -r--rwxr-- 1 adam adam 219 Sep 26 09:23 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 기타 사용자에게 실행 권한 부여

```
adam@adam:~/ex$ chmod 475 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -r--rwxr-x 1 adam adam 219 Sep 26 09:23 test.txt
```

      - ◆ 그룹 과 기타 사용자의 실행 권한을 제거

```
adam@adam:~/ex$ chmod 464 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -r--rw-r-- 1 adam adam 219 Sep 26 09:23 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 모두에게 실행 권한 부여

```
adam@adam:~/ex$ chmod 575 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -r-xrwxr-x 1 adam adam 219 Sep 26 09:23 test.txt
```

      - ◆ 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거

```
adam@adam:~/ex$ chmod 755 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rwxr-xr-x 1 adam adam 222 Sep 17 05:57 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 숫자를 이용한 파일 접근 권한 변경
    - ● 실습
      - ◆ 유자의 권한만 남기고 나머지 사용자의 권한은 모두 제거

```
adam@adam:~/ex$ chmod 700 test.txt
adam@adam:~/ex$ ls -l
        total 4
        -rwx------ 1 adam adam 222 Sep 17 05:57 test.txt
```

파일 접근 권한 관리

- ❖ 파일 접근 권한 변경 명령
  - ✓ 연습문제
    - test.txt 파일을 현재 접근 권한 상태에서 그대로 사용
    - 소유자의 권한을 읽기와 쓰기로 변경
    - 그룹과 기타 사용자도 읽고 쓰기가 가능하도록 변경
    - 기타 사용자의 쓰기 권한을 제거
    - 그룹의 쓰기 권한을 제거
    - 소유자의 읽기 권한만 남겨놓고 모든 권한을 제거
    - 파일의 모든 권한을 제거

파일 접근 권한 관리

❖ 기본 접근 권한

✓ 개요

- 리눅스에서는 파일이나 디렉토리를 생성할 때 기본 접근 권한이 자동으로 설정되는데 일반 파일의 경우 소유자와 그룹에는 읽기와 쓰기 권한이 설정되고 기타 사용자에게는 읽기 권한만 설정되며 디렉토리의 경우 소유자와 그룹에는 읽기, 쓰기, 실행 권한이 설정되고 기타 사용자에게는 읽기, 실행 권한만 설정됨
- 확인

```
adam@adam:~/ex$ touch adam.txt
adam@adam:~/ex$ mkdir temp
adam@adam:~/ex$ ls -l
        total 8
        -rw-rw-r-- 1 adam adam     0 Sep 26 09:50 adam.txt
        drwxrwxr-x 2 adam adam 4096 Sep 26 09:50 temp
        -r--rw-r-- 1 adam adam  219 Sep 26 09:23 test.txt
```

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - umask [옵션] [마스크 값]
    - 옵션은 –S로 마스크 값을 문자로 출력
    - 확인

```
adam@adam:~/ex$ umask
0002
```

◆ 의미: mask는 설정하지 않을 권한으로 0002 는 첫번째 숫자를 제외하면 ----------w- 가 되므로 기타 사용자에게 w 권한을 부여하지 않겠다는 의미가 됨

```
adam@adam:~/ex$ umask -S
u=rwx,g=rwx,o=rx
```

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - ● 마스크 값 변경

```
adam@adam:~/ex$ umask 077
adam@adam:~/ex$ umask
0077
```

    - ● 확인

```
adam@adam:~/ex$ touch linux.txt
adam@adam:~/ex$ ls -l
total 8
-rw-rw-r-- 1 adam adam     0 Sep 26 09:50 adam.txt
-rw------- 1 adam adam     0 Sep 26 09:57 linux.txt
drwxrwxr-x 2 adam adam 4096 Sep 26 09:50 temp
-r--rw-r-- 1 adam adam  219 Sep 26 09:23 test.txt
```

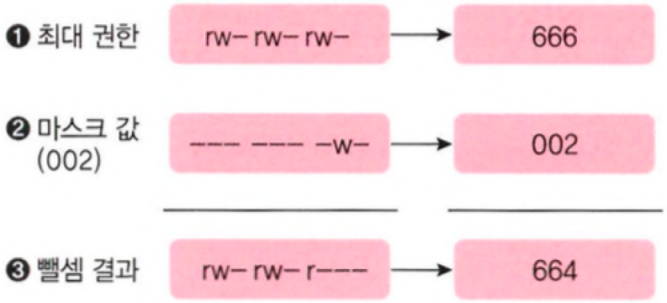
파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - ● 적용 과정
      - ◆ 마스크 값은 파일이 생성될 때마다 적용되는데 마스크 값을 비트로 표시했을 때 그 값이 1인 경우 대응하는 권한은 제외

요청 권한	1	1	0	0
마스크	1	0	1	0
부여된 권한	0	1	0	0

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - ● 적용 과정



파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - ● 적용 과정
      - ◆ 마스크 값 계산 방법
        - ▪ 리눅스의 기본 설정을 적용하는 일반적인 사용자 환경에서 마스크 값을 사용자 입장에서 쉽게 이해하는 방법은 최대 접근 권한에서 마스크 값을 빼는 것으로 마스크 값이 002일 경우 이를 접근 권한으로 해석해 표시하면-------- w-이 되는데 이를 최대 접근 권한인 rw-rw-rw-에서 빼면 파일의 기본 접근 권한은 rw-rw-r--이 되고 숫자로 계산할 수도 있음(666-002=664)
        - ▪ 일반 파일이 가질 수 있는 최대 접근 권한은 666인데 현재 마스크 값이 002일 때 666과 002를 마스킹하면 결과가 664
        - ▪ 마스크 값이 002일 때 일반 파일을 생성하면 기본 접근 권한이 664
        - ▪ 디렉토리의 경우에는 최대 접근 권한이 777이므로 여기에 002를 마스킹하면 기본 접근 권한이 775
        - ▪ 실행 파일도 기본적으로 실행 권한을 가져야 하므로 디렉토리와 같은 형태로 권한 부여

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - ● 여러 가지 마스크 값

마스크 값	일반 파일	디렉터리	의미
022	644	755	그룹과 기타 사용자는 읽기만 가능하다.
077	600	700	그룹과 기타 사용자의 접근 권한을 모두 제거한다.
027	640	750	그룹은 읽기와 실행만 가능하고 기타 사용자의 접근 권한을 모두 제거한다.

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 기본 접근 권한 확인 및 변경
    - 현재 기본 접근 권한 확인

      adam@adam:~/ex$ umask

      0077
    - 그룹과 기타 사용자가 쓰기와 실행을 할 수 없도록 기본 접근 권한을 변경

      adam@adam:~/ex$ umask 022

      adam@adam:~/ex$ umask

      0022
    - 확인

      adam@adam:~/ex$ touch mask.txt

      adam@adam:~/ex$ ls -l mask.txt

      -rw-r--r-- 1 adam adam 0 Sep 26 10:07 mask.txt

파일 접근 권한 관리

- ❖ 기본 접근 권한
  - ✓ 연습문제
    - 그룹에는 읽기와 쓰기 권한을 부여하고 기타 사용자에게는 아무 권한이 없도록 마스크 값을 변경
    - mask2.txt 파일과 tmp 디렉토리를 생성하여 기본 접근 권한이 변경되었는지 확인
    - 기본 접근 권한을 원래의 기본 마스크 값(002)으로 변경

파일 접근 권한 관리

- ❖ 특수 접근 권한
  - ✓ 개요
    - umask 맨 앞 자리는 특수 접근 권한
    - 맨 앞자리의 숫자가 0이면 일반적인 접근 권한이지만 1, 2, 4이면 특수 접근 권한을 설정
    - 특수 접근 권한의 종류
      - SetUID: 맨 앞자리가 4
      - SetGID: 맨 앞자리가 2
      - sticky bit: 맨 앞자리가 1

파일 접근 권한 관리

❖ 특수 접근 권한

✓ SetUID

- SetUID가 설정된 파일을 실행하면 해당 파일이 실행되는 동안에는 파일을 실행한 사용자의 권한이 아니라 파일 소유자의 권한이 적용
- 실습

```
adam@adam:~/ex$ touch set.exe
adam@adam:~/ex$ chmod 755 set.exe
adam@adam:~/ex$ ls -l set.exe
        -rwxr-xr-x 1 adam adam 0 Sep 26 10:15 set.exe
adam@adam:~/ex$ chmod 4755 set.exe
adam@adam:~/ex$ ls -l set.exe
        -rwsr-xr-x 1 adam adam 0 Sep 26 10:15 set.exe
```

◆ SetUID가 설정되면 소유자의 실행 권한에 s가 표시되는데 set.exe 파일을 실행하면 항상 adam의 권한을 가진다는 의미

파일 접근 권한 관리

- ❖ 특수 접근 권한
  - ✓ SetUID
    - ● 실습

      adam@adam:~/ex$ ls -l /usr/bin/passwd

      -rwsr-xr-x 1 root root 55544 Nov 24 2022 /usr/bin/passwd

      - ◆ passwd 명령은 사용자 계정의 암호를 바꾸는 명령인데 계정의 암호가 저장된 /etc/shadow 파일은 root 계정으로만 수정이 가능
      - ◆ 일반 사용자가 passwd 명령으로 암호를 바꾸려고 할 때 본인의 권한으로 실행하면 암호를 바꿀 수 없는데 /etc/shadow 파일을 수정할 수 없기 때문
      - ◆ passwd 명령에는 SetUID가 설정되어 있기 때문에 소유자인 root 권한으로 실행되어 /etc/shadow 파일을 수정해 암호를 바꿀 수 있음
      - ◆ SetUID 는 일반 사용자가 root 권한으로 실행할 필요가 있는 일부 명령에 설정하여 사용

파일 접근 권한 관리

- ❖ 특수 접근 권한
  - ✓ SetGID
    - SetGID는 SetUID와 거의 동일
    - SetGID가 설정된 파일을 실행하면 해당 파일이 실행 되는 동안에는 파일 소유 그룹의 권한으로 실행
    - SetGID는 접근 권한의 맨 앞자리에 2를 설정
    - 실습

```
adam@adam:~/ex$ ls -l set.exe
        -rwxr-sr-x 1 adam adam 0 Sep 26 10:15 set.exe
```

파일 접근 권한 관리

- ❖ 특수 접근 권한
  - ✓ sticky bit
    - 스티키 비트는 디렉토리에 설정하며 디렉토리에 스티키 비트가 설정되어 있으면 이 디렉토리에는 누구나 파일을 생성 가능
    - 파일은 파일을 생성한 계정으로 소유자가 설정되고 다른 사용자가 생성한 파일은 삭제할 수 없음
    - 이런 종류의 디렉토리 중 하나가 /tmp
    - 스티키 비트가 설정되면 기타 사용자의 실행 권한에 t가 표시
    - /tmp 디렉토리의 권한을 확인

      adam@adam:~/ex$ ls -ld /tmp

      drwxrwxrwt 17 root root 4096 Sep 26 07:35 /tmp
    - ex 디렉토리에 스티키 비트 설정

      adam@adam:~$ chmod 1755 ex

      adam@adam:~$ ls -ld ex

      drwxr-xr-t 3 adam adam 4096 Sep 26 10:15 ex