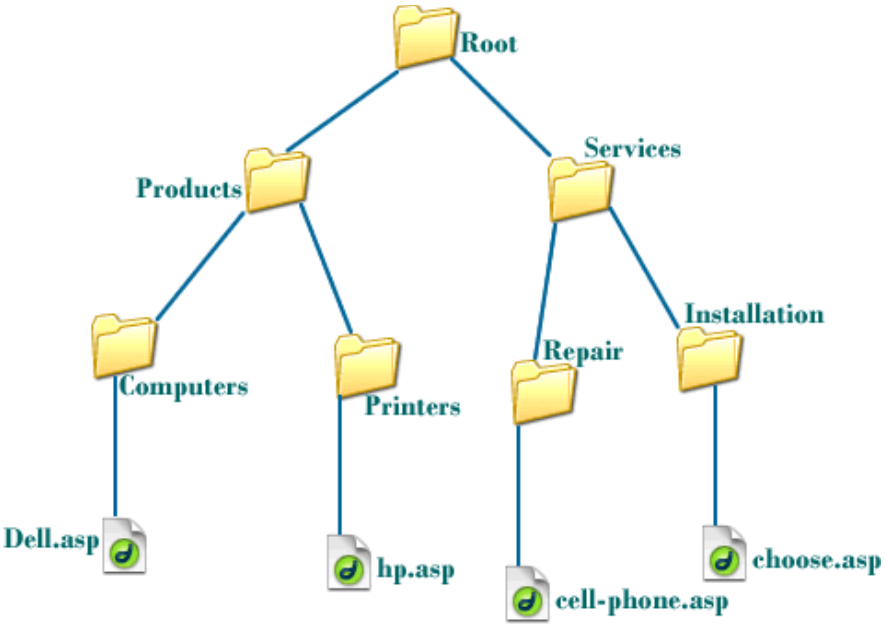
[이 문서는 Ubuntu 16.04 기준으로 작성 되었습니다.]

1주차 스터디 주요 내용

1. 리눅스 디렉터리 구조

- 리눅스는 계층형 디렉터리 구조를 갖는다. 계층형 디렉터리 구조란 최상위 폴더에 각 사용자 별로 폴더가 존재하고, 각 사용자는 자신의 폴더 아래에서 다양한 폴더를 만들어서 파일을 관리할 수 있는 구조를 말한다.



- 최상위 폴더(/)에 존재하는 폴더(꼭 알아야 되는 폴더만 소개하므로, 빠진 폴더가 있을 수 있습니다.)

  - /bin : 리눅스에 존재하는 모든 사용자들이 사용할 수 있는 명령어 실행파일이 존재하는 폴더. 폴더 내부에는 ls, mv, cp 등 자주 사용하는 명령어들이 들어있다.
  - /boot : 리눅스를 실제로 실행시키는 역할을 하는 부트로더가 들어있는 디렉토리이다.
  - /dev : 시스템에 설치된 장치들이 파일로 저장되어 있는 디렉토리이다. cdrom, HDD(sda)등을 찾아볼 수 있다. 리눅스는 이처럼 모든 시스템 자원을 파일로 관리한다.
  - /etc : 리눅스의 모든 설정파일들이 저장되어 있는 폴더이다.
  - /home : 리눅스의 일반 사용자들의 홈 디렉토리가 저장되어 있는 폴더이다.
  - /lib, /lib64 : 시스템 유틸리티(모듈)와 프로그램에 사용되는 라이브러리파일들이 저장되어 있는 폴더이다.

- /media : DVD, CD-ROM, USB와 같이 탈부착이 가능한 장치들의 마운트 포인트가 저장되는 폴더이다.
  예를 들어 리눅스에 외장하드를 꽂아 그곳에 파일을 저장하고 싶다고 할 때는 마운트 포인트 폴더에 저장을 하면 실제 하드디스크에 파일이 저장된다. 마운트 포인트는 그 폴더를 마운트 포인트로 하는 장치의 루트 디렉토리가 된다고 생각하면 된다.

- /mnt : media 폴더와 역할이 같다. 단, 마운트 포인트는 이렇게 정해진 폴더가 아니고 원하는 폴더로 설정할 수도 있는데 이는 나중에 설명할 예정이다.

- /proc : 리눅스(운영체제)는 컴퓨터가 꺼져 있을 때는 하드디스크(또는 SSD)에 저장 되어있지만, 이를 부팅이라는 과정을 통해 우리가 사용할 수 있도록 하려면 리눅스 커널을 메모리(RAM)에 올려야 한다.
  마찬가지로 프로그램도 실행되기 위해서는 RAM에 자신의 코드를 올려야 하는데, proc폴더는 그 RAM에 있는 내용을 저장하여 체계적으로 보여준다고 생각하면 된다. Windows에서 Ctrl+Alt+Del키로 접근할 수 있는 작업관리자 정보도 리눅스에서는 이 proc 폴더에 존재한다.

- /root : 사용자의 개인 폴더는 /home의 아래에 존재하지만, root(관리자)의 개인 폴더는 /root로 존재한다.

- /sbin : 시스템 관리자(root)가 사용하는 시스템 관리 명령어 파일이 저장 되어있다.

- /tmp : 모든 사용자들이 사용할 수 있는 임시 폴더이다. 인터넷 접속 기록과 같은 임시 파일도 이 폴더에 저장된다.

- /usr : 일반 사용자들을 위한 디렉터리이다. /usr/bin에는 일반 사용자가 사용할 수 있는 많은 명령어들이 있다.

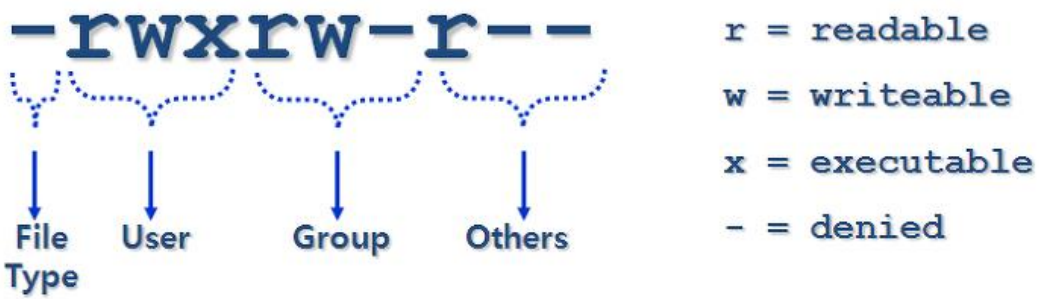
- /var : 그 크기가 자주 변하는 파일들을 일시적으로 저장하기 위한 폴더이다. 예를 들면 로그파일, DNS의 설정파일, 받은 메일, 예약 작업 등이 이 폴더에 저장되어 있다.

- /lost+found : 리눅스는 관리자의 명령어 실행 또는 기존에 설정된 대로 주기적으로 시스템을 점검하고 문제가 있으면 복구를 진행하게 되는데, 복구를 하다가 자체적으로 복구를 할 수 없는 파일이 등장하면 이 폴더에 담는다. 관리자는 이 파일들을 수동으로 복구할 수 있다.

2. 리눅스 파일 및 폴더의 권한

   - 리눅스는 파일의 접근 권한을 관리할 때 파일의 소유자(User), 파일의 소유그룹

(Group), 그 외의 타인(Others)으로 권한을 부여할 주체를 설정한 다음, 각 주체 마다 읽기(read), 쓰기(write), 실행(execute) 권한을 부여한다.



- 임의의 디렉토리에서 ls -al 명령어를 실행하면 각 파일별로 자세한 정보를 확인할 수 있는데, 제일 첫번째 필드는 파일의 종류이고, 그 다음에 나오는 것이 파일에 부여된 권한이다.
- 권한의 종류
  - r(read)
    - 파일의 read : 파일의 내용을 읽는 것을 말한다(cat 등으로 출력).
    - 디렉터리의 read : 디렉터리 내부에 어떤 파일이 있는지를 보는 것을 말한다(ls 명령으로 디렉토리 내용 출력).
  - w(write)
    - 파일의 write : 파일을 열어서 수정하는 것을 말한다(vim 등으로 편집)
    - 디렉터리의 write : 디렉토리 내부에서 파일이나 디렉토리를 생성하고 지우는 것을 말한다.
  - x(execute)
    - 파일의 execute : 파일을 실행할 수 있는 지의 여부를 말한다.
    - 디렉터리의 execute : 디렉터리에 접근할 수 있는 지의 여부를 말한다(cd를 통해서 해당 디렉터리로 들어갈 수 있는지).
- 권한 주체의 종류
  - u(user) : 파일이나 폴더를 소유한 개인을 말한다.
  - g(group) : 파일이나 폴더를 소유한 그룹을 말한다.
  - o(others) : 그 파일을 소유하지 않은 나머지를 말한다.
- 권한의 표현

■ 문자에 의한 표현

◆ 필드를 3개로 분할하여 첫 번째에는 user의 권한, 두 번째에는 group의 권한, 세 번째에는 타인의 권한을 표시한다

◆ 예를 들어, icewall이라는 파일에 대해 소유자에게는 읽기 쓰기 실행 권한, 소유 그룹에게는 읽기 실행 권한, 그 밖의 타인에게는 읽기 실행 권한을 주고 싶다면 다음과 같이 적으면 된다.

rwxr-xr-x

■ 숫자에 의한 표현

◆ 문자로 나열된 권한을 수로 변환한다. 권한이 있으면 1, 없으면 0으로 하여 적고 이를 2진수로 읽는다.

◆ 위의 예

rwx r-x r-x는 먼저 이를 수로 변환하면

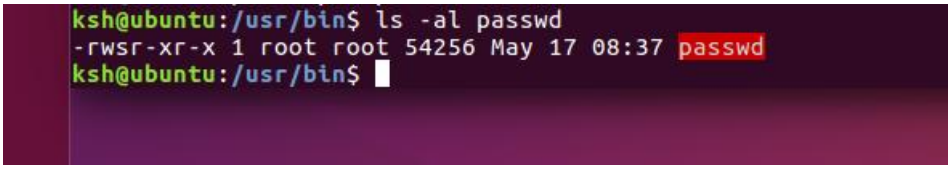
111 101 101이고 이를 각각 2진수로 읽으면

7 5 5 이다.

따라서 파일의 권한은 755가 된다.

- 특수 권한

■ 어떤 경우에는 일반 사용자가 사용할 수 없는 파일에서 자기 자신의 부분만은 변경해야 할 경우가 있다. 예를 들어, passwd 명령이 있는데



이 명령어는 비밀번호를 변경할 수 있는 명령어로 일반 사용자도 자신의 비밀번호는 변경할 수 있다. 그런데, 비밀번호가 저장되어 있는 파일은 root만이 접근할 수 있다.

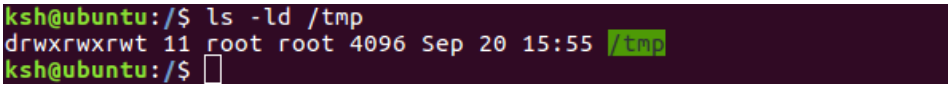
◆ 위의 그림에서 user의 권한 필드의 x자리에 s가 적혀 있는 것을 확인할 수 있다. 이 s라는 권한이 이를 가능하게 해준다.

◆ SetUID : user의 권한 필드의 실행 자리에 s가 들어가 있는 것으로, 이 실행 파일이 실행 중일 때 만큼은 root의 권한을 갖는다.

◆ SetGID : group의 권한 필드의 실행 자리에 s가 들어가 있는 것으로, 이 실행 파일이 실행 중일 때 만큼은 root의 권한을 갖는다.

◆ 참고로 실행파일이 아닌 경우에 SetUID, SetGID가 설정되면 S로 표기된다.

■ 어떤 폴더는 모든 사람이 읽고 쓸 수 있어야 한다. 그렇기 위해서는 폴더의 권한 설정에 있어 others에게 읽기, 쓰기, 실행 권한을 모두 부여해야 한다(그 이유는 위의 권한 종류의 설명을 읽어보면 알 수 있다). 그런데 아이러니하게도 이러한 권한이 설정되면, 이 폴더에 A라는 사람이 파일을 만들어 자신이 소유자 더라도, A와 전혀 관련 없는 B가 A가 만든 파일을 지울 수가 있다. 이 이해가 되지 않는 현상을 방지하기 위해 나온 권한이 t(sticky bit)이다.



◆ /tmp 폴더는 모든 사용자가 사용할 수 있는 임시 폴더이다. 따라서 위와 같은 문제가 발생할 수 있다. tmp폴더의 권한 필드를 잘 보면 others의 x자리에 t가 적혀 있는 것을 확인할 수 있다.

◆ sticky bit : others의 권한 필드의 실행 자리에 t가 적혀 있는 것으로서 해당 폴더 내부에서 만들어진 파일은 소유자(user)만이 지울 수 있다.

◆ 이 권한도 마찬가지로 실행파일이 아닌데 sticky bit가 설정되어 있으면 T로 나타난다.

■ 위의 특수 권한들을 숫자로 표현하는 방법은 다음과 같다.
4 = SetUID, 2 = SetGID, 1 = Sticky bit로 하고
원래 권한 앞에 부여된 권한을 적는다. 예를 들어 위의 tmp 폴더 같은 경우 권한은 1777이다.