

Linux



Linux – Shell

Linux Shell : Linux Kernel 조작. 커널과 사용자간의 다리 역할로 사용자 명령을 해석하고 프로그램 실행

Linux Shell Types :

SH : 본 셸. 최초로 개발

BASH (Bourne-Again Shell) [/bin/bash]

리눅스에서 가장 흔하게 사용되는 셸

sh 본셸과 호환되기 때문에 대부분 sh와 bash에서 모두 작동

CSH (C Shell) [/bin/sh]

C 언어와 유사한 문법. 유닉스의 기본 셸

KSH (Korn Shell) [/bin/ksh]

유닉스에서 가장 많이 사용. 본 셸과 호환

TCSH [/bin/tcsh] : 일반적인 C 셸. 명령행 편집 기능

Linux - 디렉토리 구조 및 기능

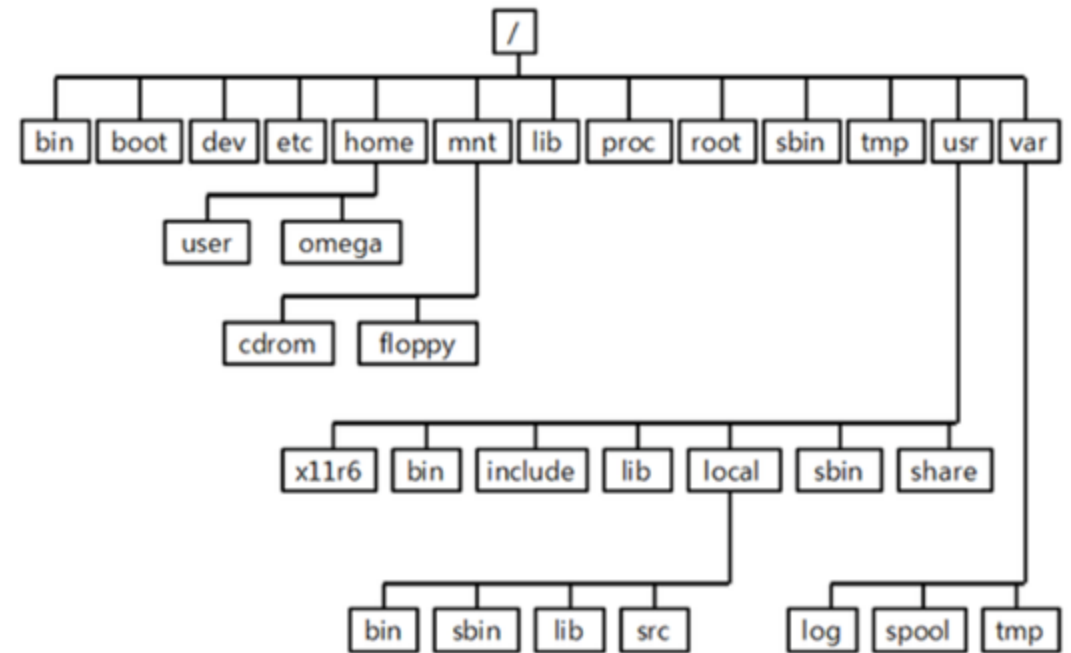
<https://inpa.tistory.com/entry/LINUX-%F0%9F%93%9A-%EB%A6%AC%EB%88%85%EC%8A%A4-%EB%94%94%EB%A0%89%ED%86%A0%EB%A6%AC-%EA%B5%AC%EC%A1%B0>

/

- 마운트 되는 리눅스 파일 시스템이 있는 최상위 디렉토리
- 시스템의 근간을 이루는 가장 중요한 디렉토리
- 파티션 설정 시 반드시 존재하여야 함
- 절대경로의 기준이 되는 디렉토리
절대경로 : / 디렉토리 기준 예) /usr/local
상대경로 : 현재 작업 디렉토리 기준 예) ./local

/bin

- binarise의 약어
- 리눅스의 기본 명령어(binary)들이 들어있는 디렉토리
- 기본적인 명령어들이 모여있는 디렉토리
- 부팅에 필요한 명령어들이 위치하며 부팅 후 시스템의 사용자들이 사용할 수 있는 일반적인 명령어들도 위치하고 있음



디렉토리 구조

/bin: 사용자가 사용하는 명령어 모음
/sbin: 관리자가 사용하는 명령어 모음
/etc: 프로그램 설정을 관리하는 디렉토리
/etc/init.d: daemon의 목적을 가진 프로그램들 있음.
/var: 내용이 바뀔 수 있는 파일들 모음
/tmp: 임시파일들. 컴퓨터가 꺼지면 날아간다.
/home: 사용자들의 파일들이 저장되는 디렉토리
/lib: /bin과 /sbin에 있는 프로그램들이 사용하는 라이브러리 모음
/usr: 유저가 다운받은 프로그램들 저장..

포트는 번호로 관리 : 0 ~ 65535번 까지 사용 ex http 80 cat /etc/services | more

[Linux] 명령어 정리 (Ubuntu 사용)

<https://seeuseeu.tistory.com/43>

<https://velog.io/@devyang97/Linux-%EB%AA%85%EB%A0%B9%EC%96%B4-%EC%A0%95%EB%A6%AC-Ubuntu-%EC%82%AC%EC%9A%A9>

<https://coding-factory.tistory.com/504>

https://www.mireene.com/webimg/linux_tip1.htm

필수 명령어들 요약

1. ls - 현재 위치의 파일 목록 조회
2. cd - 디렉터리 이동
3. touch - 0바이트 파일 생성, 파일의 날짜와 시간을 수정
4. mkdir - 디렉터리 생성
5. cp - 파일 복사
6. mv - 파일 이동
7. rm - 파일 삭제
8. cat - 파일의 내용을 화면에 출력, 리다이렉션 기호('>')를 사용하여 새로운 파일 생성
9. redirection - 화면의 출력 결과를 파일로 저장
echo "first line" > test1.txt, cp test1.txt test2.txt, echo "second line" >> test2.txt
10. alias - 자주 사용하는 명령어들을 별명으로 정의하여 쉽게 사용할 수 있도록 설정
11. top - 현재 내컴퓨터의 프로세스, 상태를 출력
12. find - 현재 폴더에서 들어있는 모든 파일을 검색한다. (<https://coding-factory.tistory.com/804>)
파일 : find ex -type f -name "*.txt", 디렉토리 : find . -type d -name "ex"
13. grep - 파일 내에서 지정한 패턴이나 문자열을 찾은 후에, 그 패턴을 포함하고 있는 모든 행을 표준 출력. find | grep tt : tt이름이 있는 파일의 경로를 찾아준다.
14. which - 명령어가 어디에서 비롯되는지 알려준다.
15. echo - 출력할 것

find 연습

현재 디렉토리에서 test가 포함되는 파일 찾기

```
find . -name "*test"
```

현재 디렉토리에서 .txt 확장자 모두 찾기

```
find . -name "*.txt"
```

현재 디렉토리에서 .txt 확장자 파일 검색 후 모두 삭제

```
find . -name "*.txt" -delete, find . -name "test2.*" -delete
```

현재 디렉토리에서 test로 시작되는 파일 찾기

```
find . -name "test*"
```

현재 디렉토리에서 test로 끝나는 파일 찾기

```
find . -name "*test"
```

find 연습

현재 디렉토리에서 모든 디렉토리 찾기

```
find . -type d
```

현재 디렉토리에서 test가 들어가는 디렉토리 찾기

```
find . -name "*test*" -type d
```

현재 디렉토리에서 모든 파일 찾기

```
find . -type f
```

현재 디렉토리에 "test"가 들어가는 파일을 찾아서 상세정보 출력

```
find . -name "*test*" -exec ls -l {} \;
```

현재 디렉토리에 있는 파일에서 "test"가 들어가는 내용 찾기

```
find . -type f -exec grep "test" {} \;
```

현재 디렉토리에 ".txt" 확장자를 찾아서 모두 삭제

```
find . -name "*.txt" -exec rm {} \;
```

명령어 옵션과 설명

1. ls (List segments) : 현재 위치의 파일 목록 조회

ls -l : 파일의 상세정보

ls -a : 숨김 파일 표시

ls -t : 파일들을 생성시간순(제일 최신 것부터)으로 표시

ls -rt : 파일들을 생성시간순(제일 오래된 것부터)으로 표시

ls -f : 파일 표시 시 마지막 유형에 나타내는 파일명을 끝에 표시

('/' : 디렉터리, '*' : 실행파일, '@' : 링크 등등,,)

2. cd (Change directory) : 디렉터리 이동

cd [디렉터리 경로] : 이동하려는 디렉터리로 이동 (경로 입력 시 '[', ']' 부분은 빼고 입력!)

cd ~ : 홈 디렉터리로 이동

cd / : 최상위 디렉터리로 이동

cd . : 현재 디렉터리

cd .. : 상위 디렉터리로 이동

cd - : 이전 경로로 이동

명령어 옵션과 설명

3. touch : 0바이트 파일 생성, 파일의 날짜와 시간을 수정

touch filename : filename의 파일을 생성

touch -c filename : filename의 시간을 현재시간으로 갱신

touch -t 202110291608 filename : filename의 시간을 날짜 정보(YYYYMMDDhhmm)로 갱신
(20211029160 => 2021.10.29.16:08)

touch -r oldfile newfile : newfile의 날짜 정보를 oldfile의 날짜 정보와 동일하게 변경

4. mkdir (Make directory) : 디렉터리 생성

mkdir dirname : dirname이라는 디렉터리 생성

mkdir dir1 dir2: 한 번에 여러 개의 디렉터리 생성

mkdir -p dirname/sub_dirname : dirname이라는 디렉터리 생성, sub_dirname이라는 하위 디렉터리도 생성

mkdir -m 700 dirname : 특정 퍼미션(권한)을 갖는 디렉터리 생성

명령어 옵션과 설명

5. cp (Copy) : 파일 복사

cp file1 file2 : file1을 file2라는 이름으로 복사

cp -f file1 file2 : 강제 복사(file2라는 파일이 이미 있을 경우 강제로 기존 file2를 지우고 복사 진행)

cp -r dir1 dir2 : 디렉터리 복사. 폴더 안의 모든 하위 경로와 파일들을 복사

6. mv (Move) : 파일 이동

mv file1 file2 : file1 파일을 file2 파일로 변경

mv file1 dir : file1 파일을 dir 디렉터리로 이동

mv file1 file2 dir : 여러 개의 파일을 dir 디렉터리로 이동

mv dir1 dir2 : dir1 디렉터를 dir2 디렉터리로 이름 변경

7. rm (Remove) : 파일 삭제

rm file1 : file1을 삭제

rm -f file1 : file1을 강제 삭제

rm -r dir : dir 디렉터리 삭제 (디렉터리는 -r 옵션 없이 삭제 불가)

명령어 옵션과 설명

8. cat (Catenate) : 파일의 내용을 화면에 출력, 리다이렉션 기호('>')를 사용하여 새로운 파일 생성

cat file1 : file1의 내용을 출력

cat file1 file2 : file1과 file2의 내용을 출력

cat file1 file2 | more : file1과 file2의 내용을 페이지별로 출력

cat file1 file2 | head : file1과 file2의 내용을 처음부터 10번째 줄까지만 출력

cat file1 file2 | tail : file1과 file2의 내용을 끝에서부터 10번째 줄까지만 출력

9. redirection ('>', '>>') : 화면의 출력 결과를 파일로 저장

'>' 기호 : 기존에 있는 파일 내용을 지우고 저장

'>>' 기호 : 기존 파일 내용 뒤에 덧붙여서 저장

'<' 기호 : 파일의 데이터를 명령에 입력

cat file1 file2 > file3 : file1, file2의 명령 결과를 합쳐서 file3라는 파일에 저장

cat file4 >> file3 : file3에 file4의 내용 추가

cat < file1 : file1의 결과 출력

cat < file1 > file2 : file1의 출력 결과를 file2에 저장

10. alias : 자주 사용하는 명령어들을 별명으로 정의하여 쉽게 사용할 수 있도록 설정

명령어 옵션과 설명

12. 권한 부여

```
chmod 777 ex2  
chmod 777 test1.txt
```

권한 번호는 0에서 7 사이의 3자리 또는 4자리 숫자로 구성될 수 있습니다.

3자리 숫자를 사용하는 경우 첫 번째 숫자는 파일 소유자의 권한, 두 번째 숫자는 파일 그룹, 마지막 숫자는 다른 모든 사용자의 권한을 나타냅니다.

쓰기, 읽기 및 실행 권한에는 다음과 같은 숫자 값이 있습니다.

r (읽기) = 4
w (쓰기) = 2
x (실행) = 1
권한 없음 = 0

Linux – 실습

- man man : help man, q로 나감
- clear

[Navigating file system]

- pwd : 현재 경로
- ls : list의 약자
ls dir1, ls -l, ls -a, ls -la
- cd : [이동 할 디렉토리 경로]
cd dir1, cd ..(상위 폴더), cd ~(home dir), cd -(이전 경로)
cd ex) cd / : root로 이동
ex) cd project : 현재 디렉토리 내부에 있는 project 디렉토리로 이동
ex) cd /user/jtaewu : /user/jtaewu 경로의 디렉토리로 이동
- find . -type file -name "*.txt", find . -type file -name "*2", find . -type directory -name "*2"
- which : 명령어의 경로, which find

[Create and manage file]

- touch new_file.txt
- cat new_file.txt
- echo "first line"> new_file1.txt, echo "second line">> new_file1.txt, echo "Hello World"> new_file2.txt

[Directory]

- mkdir
- mkdir -p dir1/subdir1/subdir2
- cp file1.txt dir1/
- mv file2.txt dir1/, mv file1.txt file2.txt
- rm file2.txt, rm dir2, rRm -r dir2

Linux – 실습

[Global Regular Expression Print]

- `grep "world" *.txt` : 키워드를 모든 파일에서 검색
- `grep -n "world" *.txt` : 몇 번째인지 검색
- `grep -ni "world" *.txt` : 대소문자 구분없이 검색
- `grep -nir "world" .` : 현재 경로와 하위 모든 경로에서 검색

[Work with environment variables]

- `export MY_DIR="dir1"` (환경변수 설정), `cd $MY_DIR`(환경변수 사용), `unset MY_DIR`(환경변수 삭제)

Linux – 실습

현재 디렉토리의 모든 파일과 디렉토리를 tar로 묶기

\$ tar cvf 파일명.tar *

대상 디렉토리를 포함한 모든 파일과 디렉토리를 tar로 묶기

\$ tar cvf 파일명.tar [PATH]

파일을 지정하여 tar 아카이브로 묶기

\$ tar cvf 파일명.tar [FILE_1] [FILE_2]

tar 아카이브의 내용 확인하기

\$ tar tvf 파일명.tar

tar 아카이브를 현재 디렉토리에 풀기

\$ tar xvf 파일명.tar

tar 아카이브를 지정된 디렉토리에 풀기

\$ tar xvf 파일명.tar -C [PATH]

tar 아카이브 묶거나 풀 때 파일 별 진행 여부 확인하기

\$ tar cvfw 파일명.tar *

Linux – 실습

현재 디렉토리를 tar로 묶고 gzip으로 압축하기

\$ tar zcvf 파일명.tar.gz *

gzip으로 압축된 tar 아카이브를 현재 디렉토리에 풀기

\$ tar zxvf 파일명.tar.gz

현재 디렉토리를 tar로 묶고 bzip2로 압축하기

\$ tar jcvf 파일명.tar.bz2 *

bzip2로 압축된 tar 아카이브를 현재 디렉토리에 풀기

\$ tar jxvf 파일명.tar.bz2