

리눅스 기초

1. 기초 이론

* 운영체제 (Operating System)

- 컴퓨터는 크게 하드웨어 (Hardware)와 소프트웨어 (Software)로 구분되어 있으며 이 두가지가 결합되어 동작한다.

- 하드웨어는 크게 4가지로 분류 된다.

- 1) 입력 장치 (Input Device) : 마우스, 키보드, 터치스크린 등
- 2) 처리 장치 (Processing Unit) : CPU
- 3) 기억 장치 (Storage Device) : 메모리, 디스크 등
- 4) 출력 장치 (Output Device) : 모니터, 스피커 등

- 소프트웨어는 크게 2가지로 분류 된다.

- 1) 시스템 소프트웨어 (System Software) : window , linux , unix

컴퓨터 내부적인 수행을 제어하는 프로그램들의 집합 (운영체제)
컴퓨터의 기본 기능과 응용 프로그램들이 실행 될 수 있는 환경을 제공

- 2) 응용 소프트웨어 (Application Software) : 한글, word 등

어느 응용 분야에 사용되기 위하여 특별히 제작된 소프트웨어
(워드 프로세서 , 스프레드시트 , 웹브라우저)

- 운영체제는 하드웨어와 사용자 사이에 위치하면서 하드웨어와 사용자간에 상호작용하기 위한 수단을 제공하고, 하드웨어를 제어한다.

* 서버 (Server)

- 네트워크에서 다른 컴퓨터나 소프트웨어와 같은 클라이언트에게 서비스를 제공하는 컴퓨터

서버	: 서비스를 제공하는 컴퓨터
클라이언트	: 서비스를 제공받는 컴퓨터

- 서버와 클라이언트를 하드웨어의 성능으로 구분하는 것이 아니라 어떤 역할의 프로그램을 설치하느냐에 따라 결정된다.

* 리눅스 (Linux)

- 공개형 오픈소스(Open Source)의 컴퓨터 운영체제(OS)
- 핀란드 리누스 토발즈에 의해 최초로 커널이 개발되어 세계의 프로그래머들에 의해 지속적으로 개발되어 가장 빠르게 성장하는 운영체제
- 윈도우는 GUI 도구를 사용하여 관리 또는 서비스 운영
리눅스는 주로 명령어를 사용하여 관리 또는 서비스 운영
- 윈도우는 GUI를 사용하기 때문에 자원관리가 비효율적
리눅스는 GUI를 사용하는 윈도우보다 자원관리가 효율적
- 윈도우는 사용자 친화적 시스템으로 성능보다 사용편이성이 우선
리눅스는 사용자 편이성을 고려한 시스템 아님 -> 시스템 / 서비스 중심형 시스템
- 유닉스와의 완벽한 호환기능
- 다중 사용자 (Multi-User) 환경 지원

* 관련인물

1) 리누스 토발즈

- 리눅스 운영체제의 아버지이자 현역 개발자

2) 리처드 스톨만

- GNU 프로젝트와 자유 소프트웨어(FSF) 재단의 설립자
- GPL 소프트웨어 라이선스의 개념을 도입

* GUI (Graphical User Interface) , CLI (Command Line Interface)

1) GUI

아이콘 기반으로 사용자가 그래픽 환경에서 마우스를 이용하여 프로그램을 사용하는 환경
쉽고 직관적인 조작방식으로 현재 가장 널리 사용되는 방식

2) CLI

운영체제 동작을 지시하는 명령어를 입력하여 이용하는 방법
일반 사용자들은 사용하기 어려운 단점

* 리눅스의 경로 이름 (path name)

- 디렉터리와 파일을 나타내는 방법
- 절대경로와 상대경로 2가지 방법이 있다.
- 리눅스는 하위경로를 '/'를 이용하여 표기한다.

1) 절대 경로

루트디렉터리부터 시작해서 명시된 경로. 전체 디렉터리 이름을 표시하는 방법

2) 상대 경로

현재 위치한 디렉터리부터 명시된 경로.

현재 위치하고 있는 디렉터리에서 서브(하위)디렉터리로 이동할 때 현재 시점에서
하위 디렉터리만 표기하는 방법

* 리눅스의 디렉터리 계층 구조

1) 최상위 루트 (root) 디렉터리

슬래쉬 '/'로 표기하며 모든 디렉터리의 상위 디렉터리이다.

2) 하위 or 서브 디렉터리

디렉터리 아래 위치한 디렉터리

3) 상위 or 부모 디렉터리

' .. ' 로 표시한다.

루트 디렉터를 제외한 모든 디렉터리는 부모 or 상위 디렉터리가 있다.

4) 작업 디렉터리

' . ' 으로 표시하며 현재 위치한 디렉터리

5) 홈 디렉터리

각 사용자에게 할당된 작업 영역

어느 사용자든 ' ~ '은 자신의 홈디렉터를 나타낸다.

Ex) root계정 ~ => /root/

user1계정 ~ => /home/user1

user2계정 ~ => /home/user2

cd 명령어 뒤에 인자를 아무것도 주지 않으면

어느 위치에 있던지 자신의 홈디렉터리로 이동한다.

```
$cd ~
```

```
$cd
```

* 파일 및 디렉터리 이름규칙

- 파일의 이름은 최대 255자까지 가능
- 파일과 디렉터리 이름에는 경로의 구분자로 사용하는' / ' 를 사용하지 않는다.
- 파일과 디렉터리 이름에는 알파벳 , 숫자 , 하이픈 , 밑줄 , 점만을 사용한다.
(절대적인 규칙은 아니나 , 이를 지키지 않을 경우 파일접근이 어려워진다.)
- 파일과 디렉터리 이름에 대소문자를 구별한다.
- 점 (.)으로 시작하는 파일은 숨김파일로 간주한다.

* 명령어의 형식

- 리눅스는 주로 명령어(command)를 통해서 리눅스를 제어한다.
- 일반 사용자의 프롬프트는 \$가 주어지고 관리자 계정에게는 #이 주어진다.
- 주로 명령어 대상 옵션 형태로 사용되어지며 띄어쓰기로 구분한다.
- 영어의 어순을 생각하면 명령어를 쉽게 이해할 수 있다.

한국어) A 명령어를 어떻게 실행해라.

영어) 실행해라 A 명령어를 어떻게

\$ command [-options] [arguments]

options : 명령에 특정한 기능을 사용

argument : 명령이 실행될 대상

Ex)

\$comand

\$comand -options

\$comand argument

\$comand -options argument

2. 기초 명령어

* pwd (Print Working Directory)

- 현재 작업중인 디렉터리 위치정보를 절대 경로로 표시

[사용법]

\$pwd

* clear

- 터미널의 텍스트 화면을 지운다.

[사용법]

\$clear

* cd (Change Directory)

- 디렉터리를 이동한다.

[사용법]

\$cd 경로

* ls (LiSt)

- 디렉터리의 목록을 출력한다.
- 특정파일을 지정하지 않을경우 현재 디렉터리의 목록을 출력한다.

[사용법]

\$ls 경로

\$ls

[옵션]

- a : 숨김파일을 포함한 모든 내용을 출력한다.
- l : 리스트 형태로 출력한다.

* mkdir (MaKe DIRectory)

- 빈 디렉터리를 생성한다.

[사용법]

\$mkdir 디렉터리명

* rm (ReMove)

- 파일과 디렉터리를 삭제한다.
- r 옵션을 지정하지 않으면 디렉터리를 삭제하지 못한다.
- f 옵션을 지정하지 않으면 지정된 대상을 하나씩 제거 할 때마다 메시지를 통해서 확인한다.

[사용법]

\$rm 파일

[옵션]

- r : 디렉터리 삭제시 하위디렉터리 , 파일을 포함하여 삭제
- f : 삭제시 확인메세지를 보여주지 않고 바로 삭제

* cat

- 텍스트 파일 내용을 출력한다.

[사용법]

\$cat 파일이름

Ex)

\$cat /etc/services

* head

- 파일의 첫 10줄을 출력한다.
- 대용량 파일의 앞부분의 내용을 볼때 유용하다.
- 출력행수를 -n 옵션을 이용해서 지정해 줄 수 있다.

[사용법]

\$ head 파일이름

\$ head -n 라인수 파일이름

Ex)

\$head -n 20 /etc/services

* tail

- 파일의 마지막 행을 기준으로 10번째 행까지 출력한다.
- 로그파일과 같이 '마지막 행을 확인하고 싶을 때' 유용하다.
- 출력행수를 -n 옵션을 이용해서 지정이 가능하다.

[사용법]

\$ tail 파일이름

\$ tail -n 라인수 파일이름

[옵션]

-f : 실시간으로 확인한다.

Ex)

\$tail -n 20 /etc/services

3. 권한

* 소유권 & 허가권

- 시스템에 존재하는 모든 파일은 권한이 부여되어있다.
- 파일을 소유한 권한을 소유권 (Ownership) 이라고 한다.
- 접근을 제한 할 수 있는 권한을 (Permission)이라고 한다.

* 소유권 (Ownership)

- 소유권 확인 명령어

```
$ls -l
```

* 소유권 변경 - chown(Change OWNership)

1) 소유자만 변경

```
$chown [계정] [file or Dir]
```

2) 소유자와 그룹을 동시에 변경

```
$chown [계정].[그룹] [file or Dir]
```

[사용법]

```
$chown tomcat8 file_repo // file_repo 디렉터리의 소유주를 tomcat8로 변경
```

```
$chown tomcat8.tomcat8 pms_log // pms_log 디렉터리의 소유자와 소유그룹을  
tomcat8로 변경
```

[옵션]

-R : 해당 디렉터리 포함 하위의 모든 파일과 디렉터리의 소유자와 소유권한이 변경

```
$chown tomcat8 -R file_repo // file_repo 와 하위 모든 파일 및 디렉터리의  
소유주를 모두 tomcat8로 변경
```

```
$chown tomcat8.tomcat8 -R pms_log // pms_log와 하위 모든 파일 및 디렉터리의  
소유자와 소유그룹을 tomcat8로 변경
```

* 허가권 (Permission)

- 허가권 필드 확인 명령어

```
$ls -l
```

- 허가권 필드는 3개의 필드로 구성되어 있다.

소유자(User) , 그룹(Group) , 기타 사용자(Other)

u g o

- rwx의 의미

r (Read) : 읽기 권한 (파일 : 읽기 , 디렉터리 : 목록보기)
w (Write) : 쓰기 권한 (파일 : 편집하기 , 디렉터리 : 파일을 생성하기)
x (eXcute) : 실행 권한 (파일 : 명령파일 실행하기 , 디렉터리 : 이동하기)
- (bar) : 해당사항 없음

Ex)

```
-rw-r--r--
```

rw- : 소유자는 읽고 쓸 수 있는 권한이 있다.

r-- : 소유한 그룹에는 읽을 수 있는 권한이 있다.

r-- : 나머지 사용자는 읽을 수 있는 권한이 있다.

* 허가권 변경 - chmod

- 각각의 허가권 필드를 8진수로(0~7)로 표기하는 방법

@2진수 -> 8진수 진법 변환

2진수	8진수
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

	소유자(u)	그룹(g)	기타사용자(o)
기호	rwX	rwX	rwX
2진수	111	111	111
8진수	7	7	7

[연 습]

-rwxr-x--x	-> rwX r-X --X	-> 111 101 001	-> 7 5 1	-> 751
-r-----wx	-> r-- --- -wx	-> 100 000 011	-> 4 0 3	-> 403
--wx-w-rwx	-> -wx -w- rwx	-> 101 010 111	-> 5 2 7	-> 527
-r--r--r--	-> r-- r-- r--	-> 100 100 100	-> 4 4 4	-> 444
-----	-> --- --- ---	-> 000 000 000	-> 0 0 0	-> 000
-rw-rw-rw-	-> rw- rw- rw-	-> 110 110 110	-> 6 6 6	-> 666
-r-xr-xr-x	-> r-x r-x r-x	-> 101 101 101	-> 5 5 5	-> 555
-rwx-----	-> rwX --- ---	-> 111 000 000	-> 7 0 0	-> 700
-rwxr-----	-> rwX r-- ---	-> 111 100 000	-> 7 4 0	-> 740

4. 편집기

* VI / VIM Editor

- Visual Editor의 약자로 UNIX계열 시스템에 사용되는 대표적인 편집기
- 우분투 리눅스에서는 `$sudo apt-get install -y vim` 설치 후 vim 에디터를 사용하는 것이 초심자가 사용하기에 편리하다.

[사용방법]

- 1)\$vim 기존 파일이름 // 기존파일을 연다.
- 2)\$vim 새로운 파일이름 // 새로운 빈 파일을 연다.
- 3)\$vim // 새로운 빈파일을 만들고 저장할때 파일명을 입력한다.

1) 편집 모드

- 편집에 사용되는 모드
- 편집 모드인지 아닌지를 확인하고 편집 작업 시작

i : 현재 커서 위치부터 편집을 시작
esc : 다른모드로 전환

2) 실행 모드

```
:set number // 라인을 보여준다.  
:set nu // 라인을 보여준다.  
:set nonumber // 라인을 보여주지 않는다.  
:set nonu // 라인을 보여주지 않는다.  
:w // 저장  
:q // 저장하지 않고 종료  
:q! // 저장하지 않고 강제종료  
:wq // 저장후 종료  
:wq! // 강제로 저장후 종료
```

5. 소프트웨어 설치

* apt-get

- 패키지 정보와 파일을 가지고 있는 저장소 (Repository)를 이용한다.
- rpm 기반의 패키지 관리도구로 의존성 문제를 자동으로 해결한다.
- 서비스 구축은 보통 아래의 과정을 거쳐서 진행된다.
 - 1) 소프트웨어 설치
 - 2) 설정 파일 편집 및 설정
 - 3) 방화벽 설정
 - 4) 서비스 오픈

* apt-get 명령어

- 관리자용 명령어로 대부분 sudo와 같이 사용한다.

[형식]

\$apt-get [옵션] [명령어] [패키지명]

\$sudo apt-get install [패키지명]	// 패키지 설치
\$sudo apt-get upgrade	// 패키지 저장소 업그레이드
\$sudo apt-get update	// 패키지 저장소 업데이트

[옵션]

-y : 설치시 확인 메시지를 사용하지 않으므로 편의를 위해 많이 사용되어진다.

\$sudo apt-get install 패키지명 - y

* 웹서버스 구축 예시

1) 패키지 설치

\$sudo apt-get install openjdk-8-jdk -y	// 자바 설치
\$sudo apt-get install tomcat8 -y	// 톰캣 설치

2) 서비스 시작

```
$sudo service tomcat8 restart // 톰캣 서비스 시작(restart도 시작에 포함)
$sudo service tomcat8 status // 톰캣 서비스 상태 확인 (running)이면 OK!
```

ctrl + c를 눌러서 현재 확인을 취소하고 커맨드 프롬프트로 진입

```
$ ip addr // 아이피 주소 확인 후 웹서비스 접근
Ex) 웹브라우저에서 100.100.100.100:8080로 접근 //아이피:포트번호/파일명
```

[실습]

```
$cd /var/lib/tomcat8/webapps/ROOT
```

```
$sudo vim test.html
```

- > 웹브라우저
- > 개발자도구
- > 소스 복사 붙여 넣기

```
:wq
```

```
$ ip addr // 아이피 주소 확인 후 웹서비스 접근
```

Ex) 웹브라우저에서 100.100.100.100:8080/test.html로 접근 //아이피:포트번호/파일명