

사이버 보안과 블록체인

1주차: 실습 환경 설정 (Linux/Vim 사용법)

시스템보안연구실 박사과정 이광무



조교 소개 (1~3주)

이광무

gwangmu@snu.ac.kr

이유찬

yoochan1026@gmail.com

박성현

shpark95@snu.ac.kr

송수환

sshkeb96@snu.ac.kr



위치 제1공학관 (301동, 5511/5513 종점 근처) 615호
방문 전 반드시 이메일 연락 먼저!



Linux 기초 사용법



Vagrant 환경 시작

Vagrant 설치 후, 명령 프롬프트에서 **굵은** 명령어를 입력

```
C:\> vagrant up
```

설치된 Vagrant 환경(Linux) 시작

```
C:\> vagrant ssh
```

Vagrant 환경(Linux)에 접속

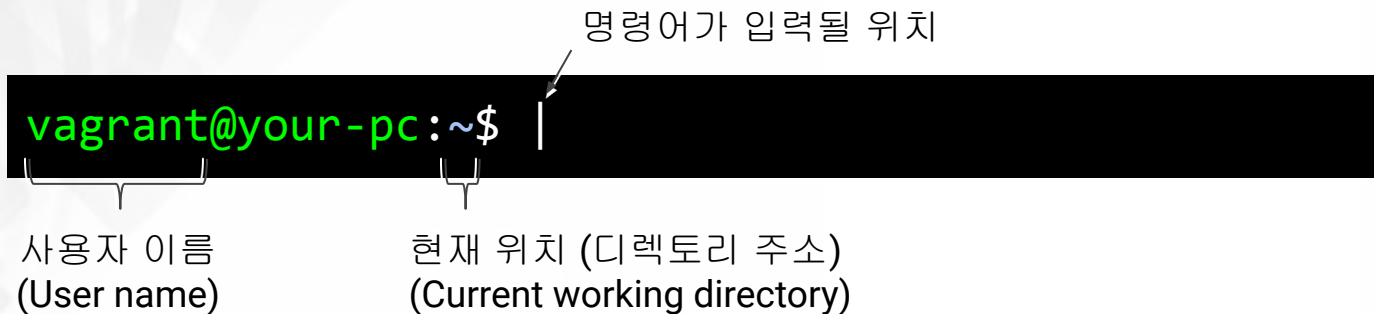
```
...
```

```
vagrant@your-pc:~$
```

Linux에 접속된 상태



Linux 환경 기본



Bash Shell: Linux의 기본 사용자 인터페이스 중 하나.

모든 Linux 사용자는 자신의 전용(Home) 디렉토리를 갖고 있음.

(~ = 현재 사용자의 Home을 의미)

자신의 Home에서부터 시작하여 다른 디렉토리로 이동할 수 있음.

Bash 기본 명령어

Bash 환경에서 **굵은** 명령어를 입력

```
vagrant@your-pc:~$ mkdir my_dir
```

my_dir 이라는 디렉토리를 만듦

```
vagrant@your-pc:~$ ls
```

현재 디렉토리 내용물을 열거

```
my_dir
```

```
vagrant@your-pc:~$ cd my_dir
```

my_dir 디렉토리로 현재 위치를 이동

```
vagrant@your-pc:~/my_dir$ pwd
```

현재 위치를 절대 경로로 출력

```
/home/vagrant/my_dir
```

```
vagrant@your-pc:~/my_dir$ touch my_file
```

my_file 빈 파일 만듦

```
vagrant@your-pc:~/my_dir$ ls
```

```
my_file
```



Bash 기본 명령어

Bash 환경에서 **굵은** 명령어를 입력

```
vagrant@your-pc:~/my_dir$ cp my_file ..
```

my_file을 하위
디렉토리(..)로 복사

```
vagrant@your-pc:~/my_dir$ cd ..
```

```
vagrant@your-pc:~$ ls
```

```
my_dir my_file
```

my_file을 my_dir에
my_file2로 이동

```
vagrant@your-pc:~$ mv my_file my_dir/my_file2
```

```
vagrant@your-pc:~$ ls
```

```
my_dir
```

```
vagrant@your-pc:~$ ls my_dir
```

my_dir 디렉토리의 내용물을 열거

```
my_file my_file2
```



Bash 기본 명령어

Bash 환경에서 **굵은** 명령어를 입력

```
vagrant@your-pc:~$ rm my_file
```

my_file 파일을 삭제

```
vagrant@your-pc:~$ ls
```

```
my_dir
```

```
vagrant@your-pc:~$ rm -r my_dir
```

my_dir 디렉토리와 내용물을 모두 삭제

```
vagrant@your-pc:~$ ls
```

```
vagrant@your-pc:~$ exit
```

Linux 환경에서 나가기

```
...
```

```
C:\> vagrant halt
```

Vagrant 환경 끄기

Bash 기본 명령어 정리 (1/2)

명령어	인자	수행 동작
ls		현재 디렉토리 내용물 출력
	<경로>	<경로>로 지정된 디렉토리의 내용물 출력
cp	<파일> 	<파일> 을 로 복사
	-r <디렉토리> 	<디렉토리> 를 로 복사 * 가 <목적 디렉토리/이름> 형식이면 해당 이름으로 복사
mv	<A> 	<A> 을 로 이동 * 이름을 변경할 때도 사용
rm	<파일>	<파일>을 삭제
	-r <디렉토리>	<디렉토리>와 그 안의 내용물을 모두 삭제



Bash 기본 명령어 정리 (2/2)

명령어	인자	수행 동작
mkdir	<이름>	<이름>으로 새 디렉토리를 만듦
rmdir	<경로>	<경로>로 지정된 디렉토리를 삭제 * 비어있을 때만 가능
touch	<이름>	<이름>이라는 빈 파일을 만듦
pwd		현재 위치의 절대 경로를 출력
exit		Linux 환경을 나감



실습 문제

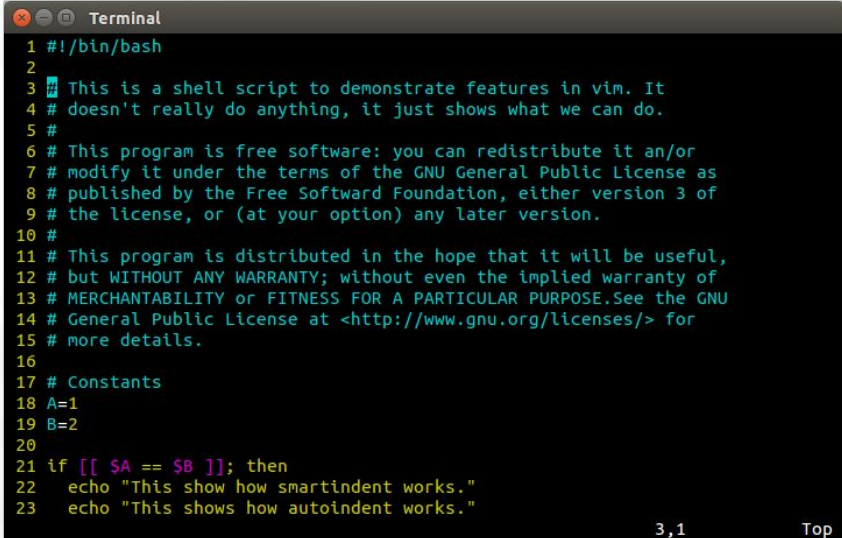
- Q1.** 자신의 영문 이름으로 **Home**에 디렉토리 만들기.
- Q2.** 자신의 학번을 이름으로 **Home**에 파일 하나 만들기.
- Q3.** Q2에서 만든 파일을 Q1의 디렉토리에 **quiz**라는 이름으로 이동시키기.
- Q4.** Q1의 디렉토리를 지우기.



Vim 기초 사용법



Vim 개요

A terminal window titled 'Terminal' with a dark background and light-colored text. It shows the execution of a Vim script. The script starts with a shebang line, followed by comments explaining its purpose and license. It then sets constants A=1 and B=2, and uses an if statement to print messages about smartindent and autoindent. The terminal output shows the script running successfully, with the cursor at the end of the last line. The bottom right corner of the terminal window shows '3,1' and 'Top'.

```
1 #!/bin/bash
2
3 # This is a shell script to demonstrate features in vim. It
4 # doesn't really do anything, it just shows what we can do.
5 #
6 # This program is free software: you can redistribute it an/or
7 # modify it under the terms of the GNU General Public License as
8 # published by the Free Software Foundation, either version 3 of
9 # the license, or (at your option) any later version.
10 #
11 # This program is distributed in the hope that it will be useful,
12 # but WITHOUT ANY WARRANTY; without even the implied warranty of
13 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
14 # General Public License at <http://www.gnu.org/licenses/> for
15 # more details.
16
17 # Constants
18 A=1
19 B=2
20
21 if [[ $A == $B ]]; then
22     echo "This show how smartindent works."
23     echo "This shows how autoindent works."
```

Vim: De-facto Linux 기본 에디터

다양한 용도로 사용됨.

(코딩, 논문 작성, 디스어셈블리, ...)

Mode-based 작동 방식

처음 Normal Mode로 시작.

모든 Mode의 시작점이 되는 모드.

편집을 위해 Insert Mode로 전환.

블록 설정을 위해선 Visual Mode로.

Vim 기본 사용법

Bash 환경에서 **굵은** 명령어를 입력

vim 설치

```
vagrant@your-pc:~$ sudo apt install vim  
vagrant@your-pc:~$ vim my_doc
```

vim에서 my_doc 편집을 시작
(my_doc 파일이 없다면 새로 만듦)

문서 편집 (Normal Mode에서 편집 시작)

키 입력	수행 동작
<방향키>	커서를 움직임
i	현재 커서 위치에서부터 입력 시작 (→ Insert Mode)
a	현재 커서 위치 다음부터 입력 시작 (→ Insert Mode)
<Shift>+r	현재 커서 위치부터 덮어쓰기 시작 (→ Replace Mode) * Vim은 입력의 대소문자를 구분! Caps Lock 확인.
o	현재 커서 위치 아랫줄부터 입력 시작 (→ Insert Mode)
<Shift>+o	현재 커서 위치 윗줄부터 입력 시작 (→ Insert Mode)

* 연한 글자들은 참고용



문서 편집 (Insert Mode에서 편집)

키 입력	수행 동작
<방향키>	커서를 움직임
<문자 키>	해당 문자를 입력
Esc	Normal Mode로 되돌아가기



문서 편집 (Normal Mode에서 블록 설정 시작)

키 입력	수행 동작
<방향키>	커서를 움직임
v	커서 위치부터 블록 설정을 시작 (→ Visual Mode)
<Shift>+v	줄 단위의 블록 설정을 시작 (→ Visual Mode)
<Ctrl>+v	직사각형 모양의 블록 설정을 시작 (→ Visual Mode)

* 연한 글자들은 참고용



문서 편집 (Visual Mode를 이용한 블록 편집)

키 입력	수행 동작
<방향키>	블록 영역을 설정
y	복사 (→ Normal Mode)
x or d	잘라내기 (→ Normal Mode)
Esc	Normal Mode로 되돌아가기

* Vim은 특별한 지우기 조작이 따로 없음!



문서 편집 (Normal Mode에서 붙여넣기)

키 입력	수행 동작
<방향키>	커서 위치를 이동
p	커서 위치 다음에 붙여넣기 (→ Insert Mode)



문서 편집 (Normal Mode에서 바로 복사)

키 입력	수행 동작
<방향키>	커서 위치를 이동
yy	현재 줄을 복사
dd	현재 줄을 잘라내기
x	문자 하나를 잘라내기
y^/d^	커서부터 줄의 맨 앞까지 복사/잘라내기
y\$/d\$	커서부터 줄의 맨 뒤까지 복사/잘라내기
p	커서 위치 다음에 붙여넣기 (→ Insert Mode)

* 연한 글자들은 참고용



문서 편집 되돌리기 (Normal Mode에서)

키 입력	수행 동작
u	되돌리기
<Ctrl>+r	되돌리기 취소



문서 탐색 (Normal Mode에서)

키 입력	수행 동작
<방향키>	커서 위치를 이동 * Page Up / Page Down / Home / End 버튼도 사용 가능
:<줄 번호>	<줄 번호>로 커서를 이동
/<문자열>	<문자열>을 문서에서 검색 * 대소문자 구분
:%s,<A>,,g	문서 전체에서 <A>를 로 바꾸기
:%s,<A>,,gI	대소문자 구분 없이 <A>를 로 바꾸기
(Visual Mode에서 블록 설정 후 : 입력하고) s,<A>,,g	블록 영역 내에서 <A>를 로 바꾸기

* 연한 글자들은 참고용



문서 저장 및 나가기 (Normal Mode에서)

키 입력	수행 동작
:w	파일 저장 * 선택적으로 파일 이름도 넣어줄 수 있음 (e.g., :w filename)
:q	나가기
:wq	저장하고 나가기
:q!	변경 내용을 버리고 강제로 나가기

실습 문제

Q1. Home에 아래 내용을 가진 파일 `my_first_doc`를 만들기.

```
Linux is not Unix
```

Q2. Home에서 `my_first_doc`을 `my_second_doc`으로 복사하기.

Q3. `my_second_doc`의 내용을 아래와 같이 바꿔 저장하기.

```
Linux is not Unix  
Linux is not Unix  
Linux is not Unix
```