



Embedded Software

개발 환경 구축 실습

Young-woo Lee

System IC Design Technology Laboratory

1.1 Host 개발환경구축

1) 가상머신 설치 (1/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 가상머신(Virtual Machine)은 하나의 데스크톱에 두 개의 운영체제를 설치할 경우 사용
- 다운로드를 위해 해당 사이트로 접속 <https://www.vmware.com/kr/worldwide.html>



[그림] VMware 홈페이지

1.1 Host 개발환경구축

1) 가상머신 설치 (2/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 웹페이지 상단의 다운로드로 이동하여 'Workstation Player'를 선택하고 다운로드를 진행

<https://www.vmware.com/kr/products/workstation-player/workstation-player-evaluation.html>

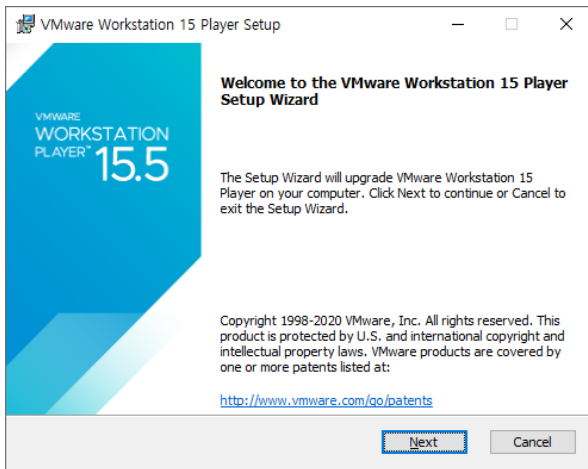


[그림] VMware 홈페이지

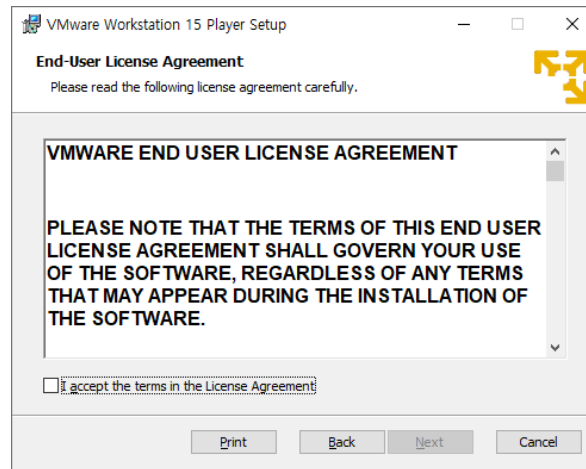
1.1 Host 개발환경구축

1) 가상머신 설치 (3/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- License등록 > Next
- 다운로드가 완료되면 설치를 진행한다.
- 설치파일 : VMware-player-15.xx.exe



[그림] VMware 설치-01

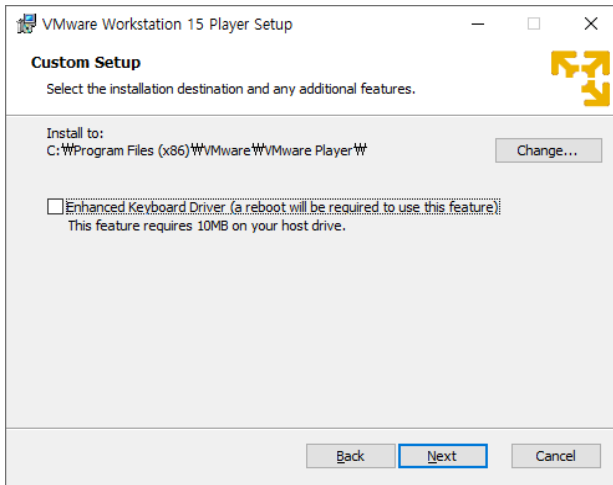


[그림] VMware 설치-02

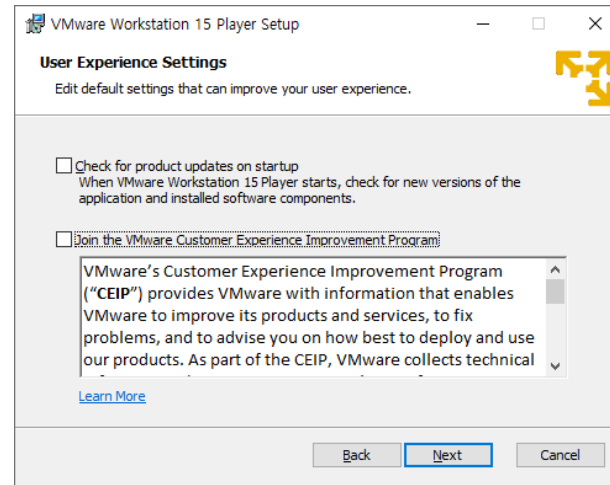
1.1 Host 개발환경구축

1) 가상머신 설치 (4/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 설치경로 > Next
- 사용자 기본 설정(업데이트) 체크 해제 > Next



[그림] VMware 설치-03

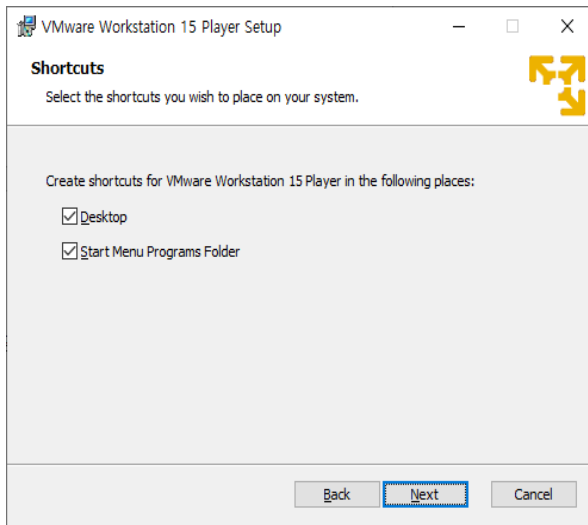


[그림] VMware 설치-04

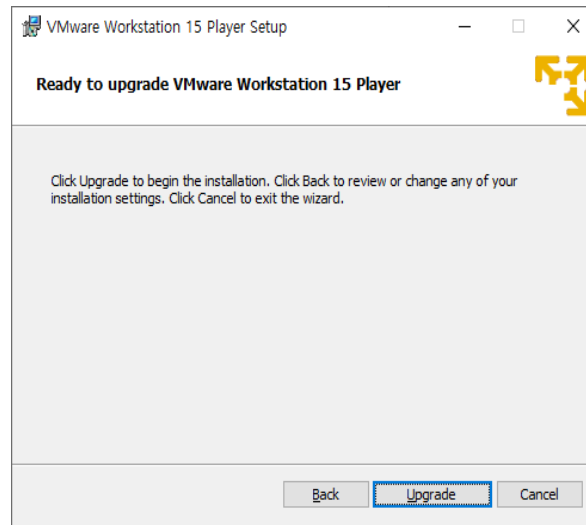
1.1 Host 개발환경구축

1) 가상머신 설치 (5/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 바로가기 만들기 > Next
- 업그레이드 > Upgrade



[그림] VMware 설치-05

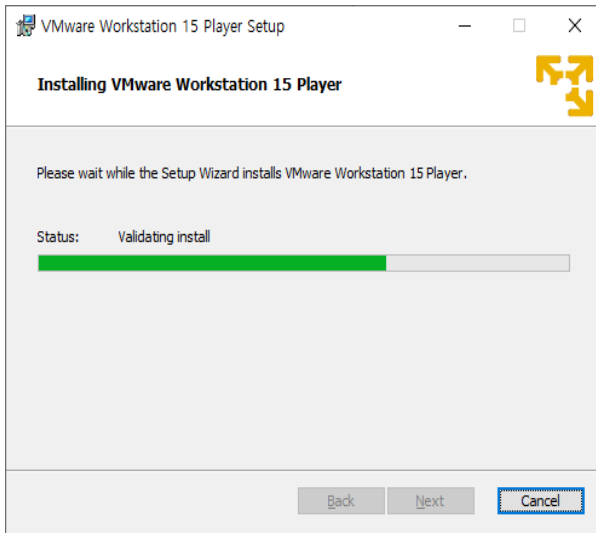


[그림] VMware 설치-06

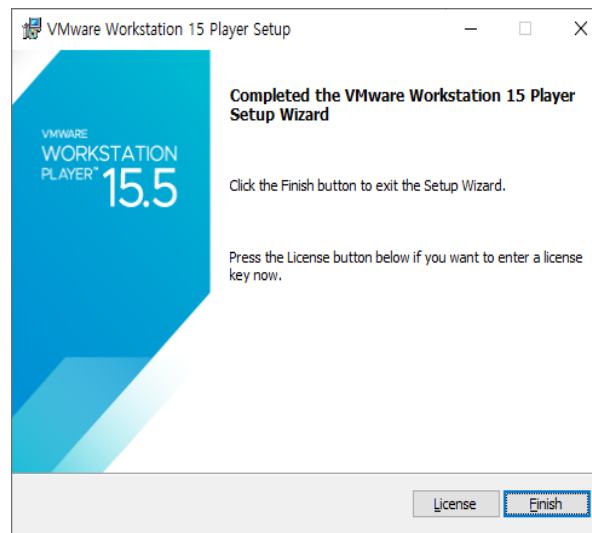
1.1 Host 개발환경구축

1) 가상머신 설치 (6/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 설치 > 설치완료 > Finish



[그림] VMware 설치-07

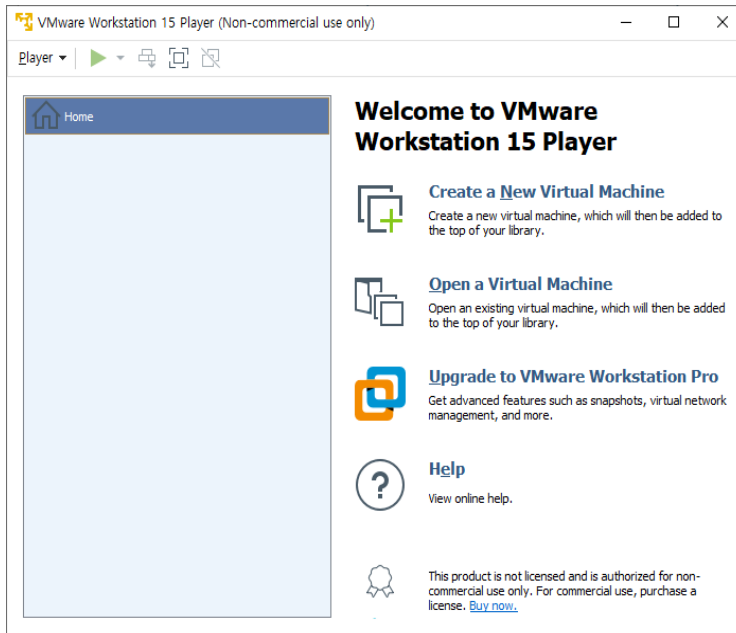


[그림] VMware 설치-08

1.1 Host 개발환경구축

1) 가상머신 설치 (7/7) [실습PC에 VMware가 설치되어 있다면, 생략할 것]

- 정상적으로 설치가 완료되면 프로그램을 실행한다.

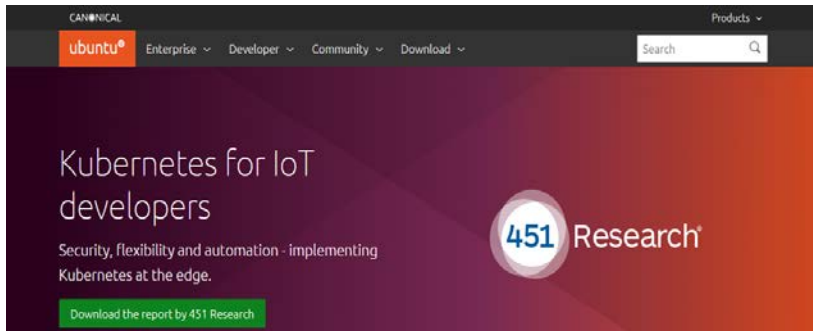


[그림] VMware 실행화면

1.1 Host 개발환경구축

2) 리눅스 설치 (1/8) [실습PC에 ubuntu가 download 되어 있다면, 생략할 것]

- 가상머신 설치가 완료되면 설치할 리눅스(우분투)운영체제가 필요하다. 우분투는 무료 배포판으로 공식 홈페이지를 통해 다운
- <https://ubuntu.com/>

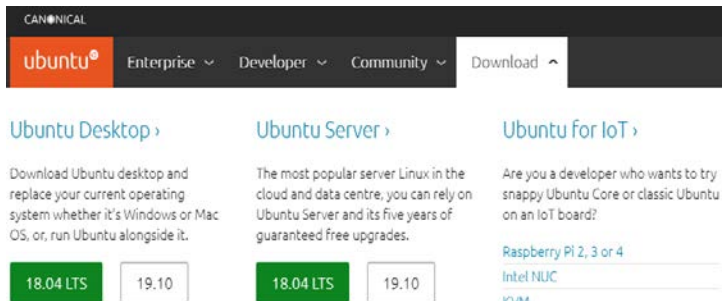


[그림] 우분투 공식 페이지

1.1 Host 개발환경구축

2) 리눅스 설치 (2/8) [실습PC에 ubuntu가 download 되어 있다면, 생략할 것]

- 우분투(Ubuntu)는 개인용, 데스크탑 환경에 최적화된 리눅스
- 무료 배포판으로 누구나 무료로 다운로드 가능
- 제품 버전은 LTS(Long Term Service)와 일반버전으로 나뉘며 LTS는 최대 5년간의 기능 및 보안 업데이트를 지원



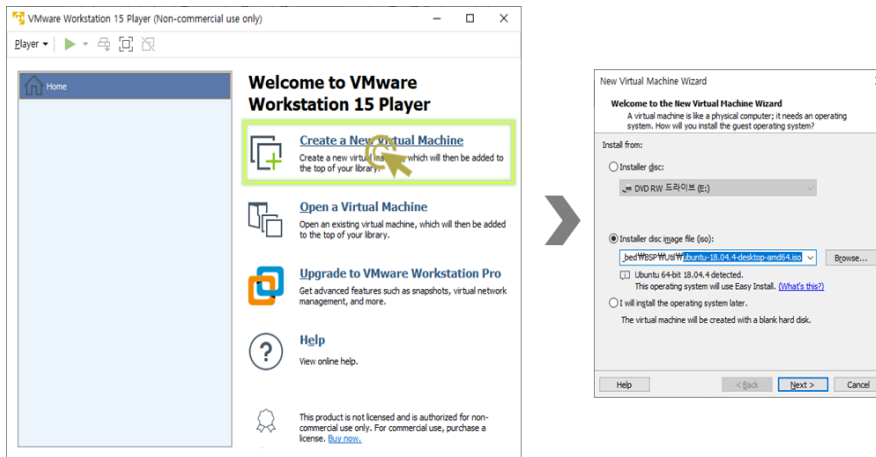
[그림] 우분투 다운로드 페이지



1.1 Host 개발환경구축

2) 리눅스 설치 (3/8)

- Ubuntu 리눅스 이미지 다운로드가 완료되면 VMware-Player를 실행
- 이미지 설치를 위한 과정을 진행
- 오른쪽 메뉴의 'Create a New Virtual Machine'을 선택하고 리눅스 이미지를 선택

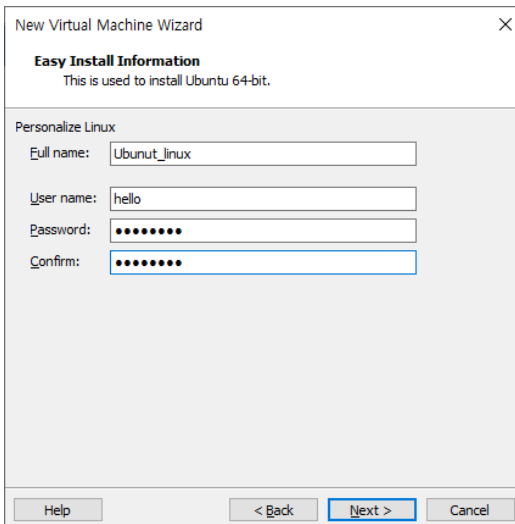


[그림] 우분투 다운로드 페이지

1.1 Host 개발환경구축

2) 리눅스 설치 (4/8)

- 이미지 선택이 완료되면 Next버튼을 클릭하고 기본정보를 입력
- User name과 Password는 가상머신에 로그인할 때 사용되는 정보
- Virtual Machine name을 정하고 설치경로를 설정



New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu 64-bit.

Personalize Linux

Full name:

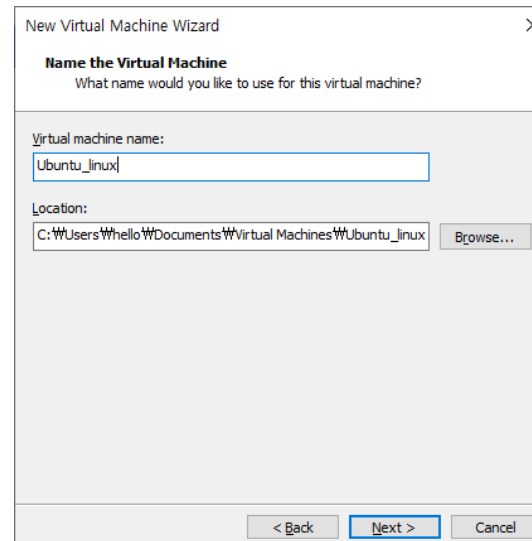
User name:

Password:

Confirm:

Help < Back **Next >** Cancel

Full name : Ubuntu_linux
User name : hello
Password : *****
Confirm : *****



New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:

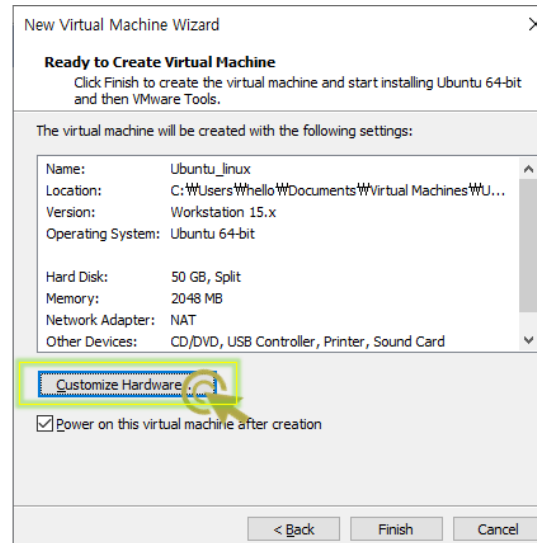
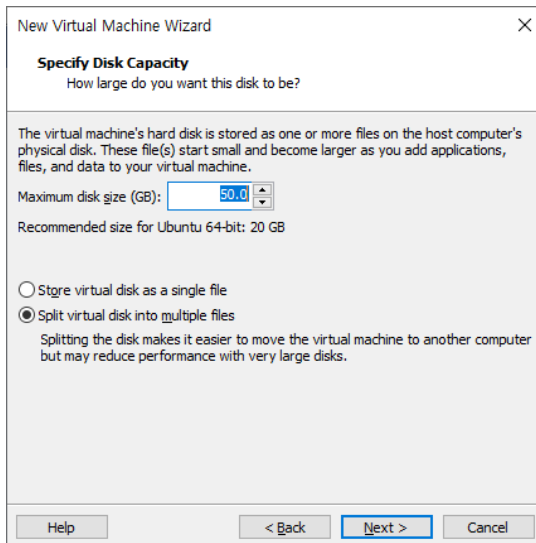
Location:
 Browse...

< Back **Next >** Cancel

1.1 Host 개발환경구축

2) 리눅스 설치 (5/8)

- 가상머신의 설치 저장 공간을 설정한다. (최소 권장크기는 50GB)
- 가상머신의 동작 환경을 나타내주며 메모리의 경우 기본 값(2048MB)을 4096Mb로 변경
- Host PC의 메모리가 기본 4Gb의 경우 2Gb로 설정

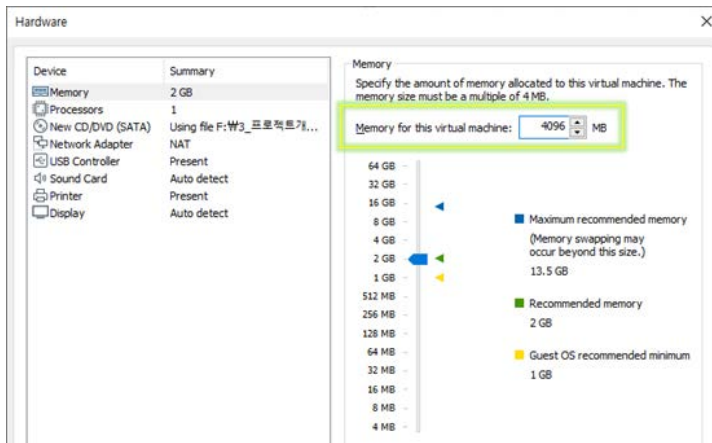


※안드로이드 빌드의 경우 150GB이상을 권장한다.(최소 100GB이상) ※안드로이드 빌드의 경우 10GB이상을 권장한다.(최소 8GB이상)

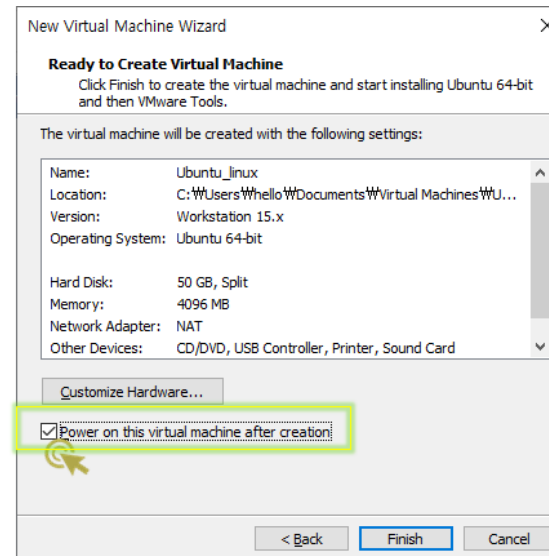
1.1 Host 개발환경구축

2) 리눅스 설치 (6/8)

- Customize Hardware 버튼을 클릭하고 메모리 값을 변경(4096Mb)
- 설정변경이 완료되면 'Power on this virtual machine after creation'박스를 체크하고 Finish 버튼을 클릭 (가상 머신 생성 후 실행)



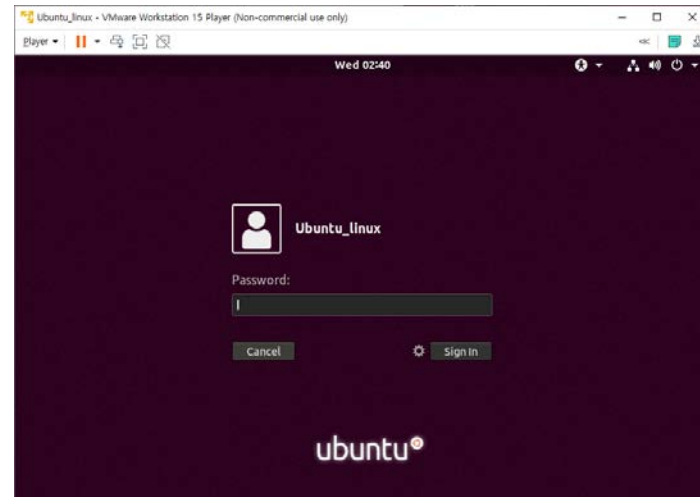
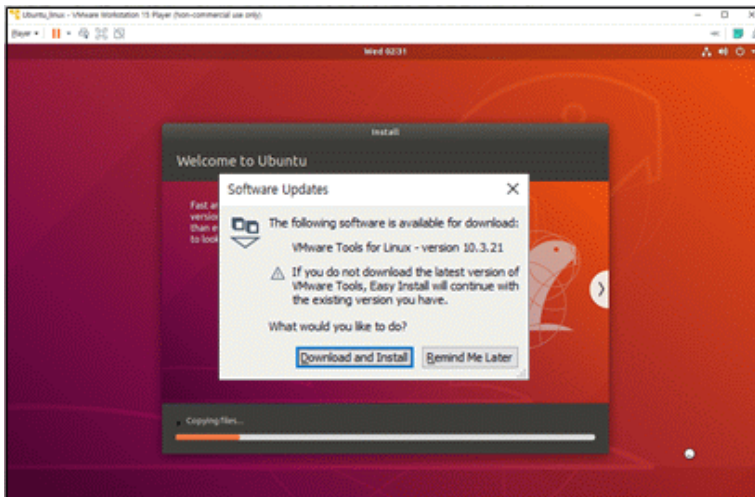
※안드로이드 빌드의 경우 10GB이상을 권장



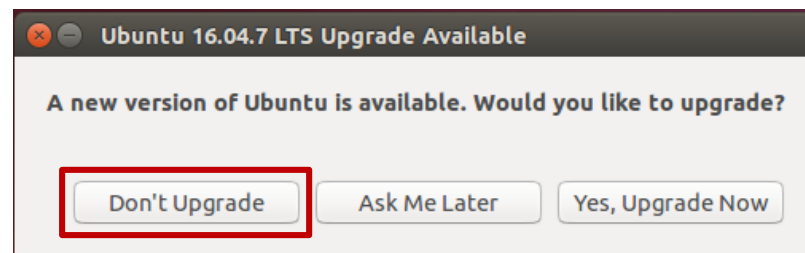
1.1 Host 개발환경구축

2) 리눅스 설치 (7/8)

- 리눅스 설치가 자동으로 진행되며 다음과 같이 VMware tools 업데이트창이 나타나면 다운로드 및 설치를 진행
- 설치시간은 약 10분 내외의 시간이 소요되며 설치가 완료되면 다음과 같이 로그인 화면이 나타남
- 패스워드를 입력하여 로그인을 진행



- 최신버전 업데이트 팝업 창
→ Don't Upgrade 클릭하여 업데이트 거부



1.1 Host 개발환경구축

2) 리눅스 설치 (8/8)

- 기본 UI화면과 로그인을 관리하는 Display Manager가 존재하며 윈도우와 같은 개별 창들을 관리하는 Window Manger가 존재하며 그 외 GUI와 관련된 Desktop Environment로 이루어짐



[그림] 우분투 시작화면

1.2 리눅스 시작하기

1) 터미널(Terminal) (1/19)

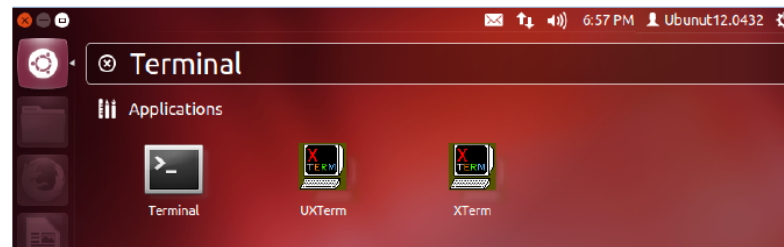
- 리눅스는 Shell기반의 Interface를 제공하는 터미널을 이용하여 주요 개발이 이루어짐
- 리눅스 커널에 내부적으로 동작하는 시스템 콘솔로써 커널과 다른 프로세스들이 텍스트를 사용자에게 출력하고 사용자로부터 텍스트 입력을 받을 수 있는 방법을 제공
- 사용자는 일반적으로 컴퓨터 키보드로 텍스트를 입력하고 컴퓨터 모니터로 텍스트를 출력
- 리눅스 커널은 가상 콘솔을 지원하는데, 이 콘솔은 논리적으로 분리되어 있지만 동일한 물리적 키보드와 디스플레이에 접근이 가능

1.2 리눅스 시작하기

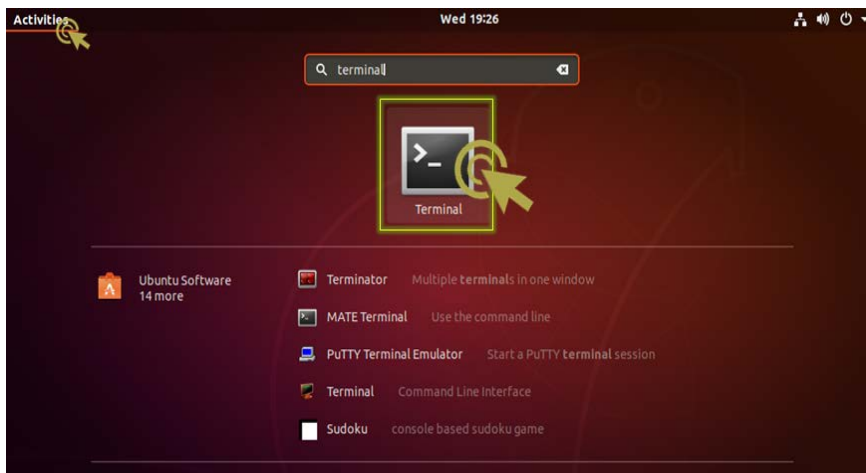
2) 터미널(Terminal) (2/19)

- 터미널 실행은 왼쪽 상단의 'Activities'를 클릭하고 'terminal'을 검색하여 프로그램을 실행

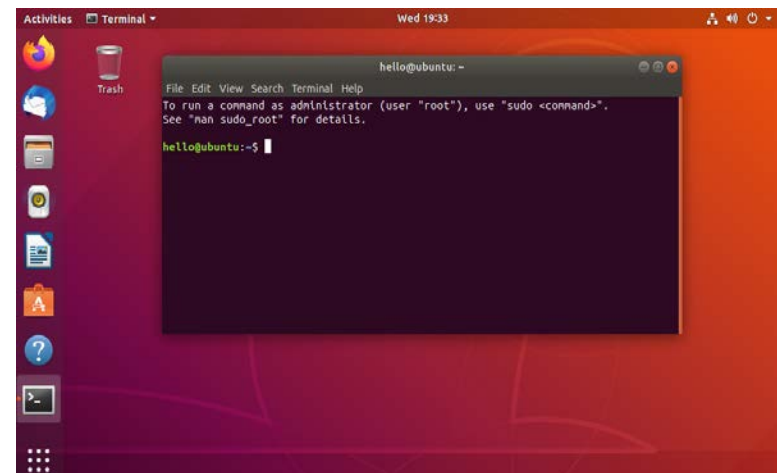
- Dash Home > Terminal 검색 > Terminal 실행
- ※ 단축키Ctrl + Alt + t



[그림] 터미널 실행



[그림] 터미널 실행



[그림] 터미널 실행화면

1.2 리눅스 시작하기

1) 터미널(Terminal) (3/19)

- 터미널 관련 단축키 정보

터미널 실행	Ctrl + Alt + T
터미널 종료	Ctrl + Alt + Q
터미널 검색	Ctrl + Shift + F
복사하기	Shift + Ctrl + C
붙여넣기	Shift + Ctrl + V
프로그램 중지	Ctrl + C
프로그램 일시정지	Ctrl + Z

1.2 리눅스 시작하기

1) 터미널(Terminal) (4/19)

- 터미널을 실행하고 root계정을 얻기 위해서 sudo 명령을 통해 권한을 얻음
- 권한을 얻지 않는 경우 일부 작업이 제한될 수 있음

▷root 계정

리눅스의 시스템 관리 계정으로 여러 사용자가 사용하는 컴퓨터에서 전체적으로 관리할 수 있는 총괄 권한을 가진 유일한 특별 계정.

유닉스 시스템의 루트(root)는 시스템 관리자인 운용관리자로서 사용자 계정을 생성하거나 소프트웨어 설치, 환경 및 설정을 변경하거나 시스템의 동작을 감시 및 제어할 수 있음

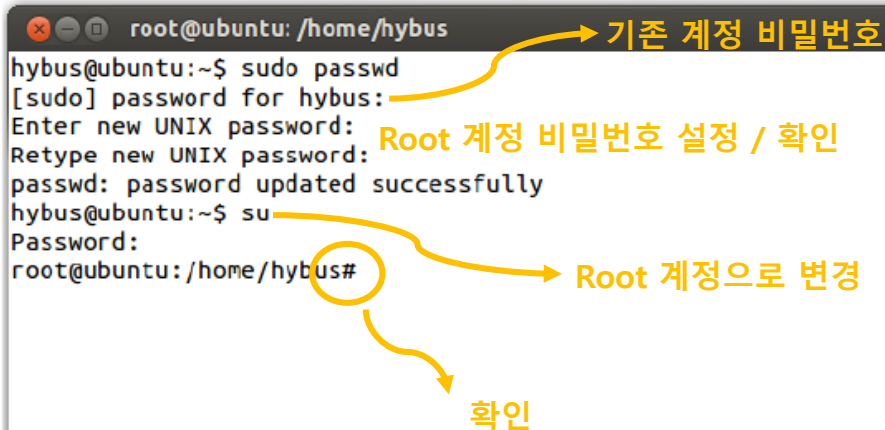
1.2 리눅스 시작하기

1) 터미널(Terminal) (5/19)

- `sudo passwd root`를 입력하고 자신의 계정 암호와 root계정의 암호를 정하여 입력
- 모든 입력을 마치고 `su` 명령을 통해 root권한을 얻음

Terminal

```
hello@ubuntu:~$ sudo passwd
[sudo] password for hello: [계정암호입력]
Enter new UNIX password: [root 계정암호 입력]
Retype new UNIX password: [root 계정암호 재입력]
passwd: password updated successfully
hello@ubuntu:~$ su
Password: [root 계정암호 입력]
root@ubuntu:/home/hello#
```



A terminal window showing the process of updating a password and switching to root. The window title is `root@ubuntu: /home/hybus`. The commands and output are as follows:

```
hybus@ubuntu:~$ sudo passwd
[sudo] password for hybus:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
hybus@ubuntu:~$ su
Password:
root@ubuntu:/home/hybus#
```

Annotations in yellow text with arrows point to specific parts of the terminal output:

- `기존 계정 비밀번호` points to the password prompt for `hybus`.
- `Root 계정 비밀번호 설정 / 확인` points to the new password and retype prompts.
- `Root 계정으로 변경` points to the `su` command and the resulting `root@ubuntu:/home/hybus#` prompt.
- `확인` points to the `hybus@ubuntu:~$ su` command.

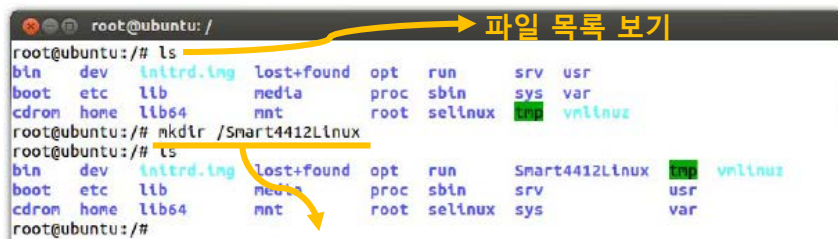
1.2 리눅스 시작하기

1) 터미널(Terminal) (6/19)

- 터미널은 Shell기반의 Interface를 제공하며 다양한 콘솔 명령어를 제공
- 파일정보 표시 ls -[option]
- 현재 디렉토리 내의 파일과 디렉토리 정보를 출력
- 윈도우 cmd 창의 'dir'과 같은 명령
- [option]

-l 파일에 대한 정보(파일 허용권, 소유자, 그룹, 크기, 날짜)를 표시
 -a 히든 파일을 포함한 모든 파일을 화면에 표시
 -t 최종 수정된 시간순으로 화면에 표시
 -u 최근에 액세스했던 파일들을 보여줌
 -F 파일의 특성을 보여줌
 디렉터리 : 이름/
 실행 파일 : 이름*
 심볼릭링크 : 이름@
 -R 하위 디렉터리의 파일도 모두 보여줌

mkdir /Smart4412Linux (제공하는 자료를 복사할 폴더를 생성)



```

root@ubuntu: /
root@ubuntu: /# ls
bin  dev  initrd.img  lost+found  opt  run  srv  usr
boot  etc  lib  media  proc  sbin  sys  var
cdrom  home  lib64  mnt  root  selinux  tmp  vmlinuz
root@ubuntu: /# mkdir /Smart4412Linux
root@ubuntu: /# ls
bin  dev  initrd.img  lost+found  opt  run  Smart4412Linux  tmp  vmlinuz
boot  etc  lib  media  proc  sbin  srv  usr
cdrom  home  lib64  mnt  root  selinux  sys  var
root@ubuntu: /#
    
```

Smart4412Linux 폴더 생성

1.2 리눅스 시작하기

1) 터미널(Terminal) (7/19)

- 디렉터리 이동 `cd -[option]`
- `[option]`

`cd`, `cd ~`, `cd ~/`, `cd $HOME` : 사용자의 홈 디렉토리로 이동
`cd .` : 현재 디렉토리로 이동
`cd /` : 최상위 디렉토리로 이동
`cd $변수명` : 변수에 지정된 경로로 이동
`cd ~계정명` : 입력한 사용자의 홈디렉토리로 이동
`cd -` : 이전 경로로 이동

`cd /media/My Disc` →
 # `cp -rf Development/ /Smart4412Linux/`

이클래스에서 **Development.zip** 파일
 download 후, 압축 해제 후 USB로 복사
 USB 삽입시 자동으로 가상머신으로
 잡히게 되는데 이름은 다를 수 있음

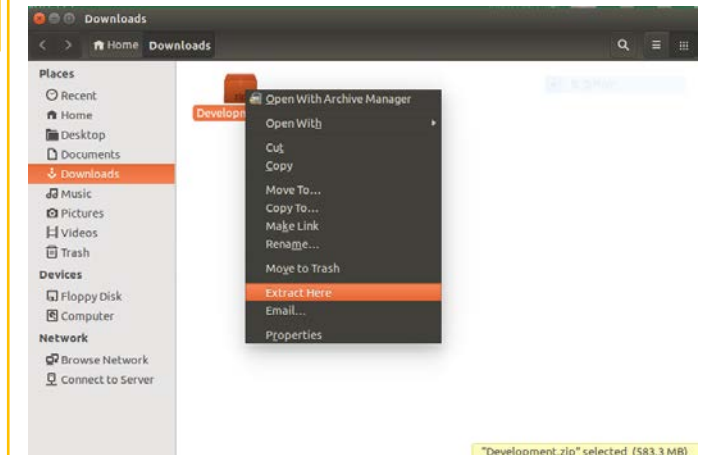
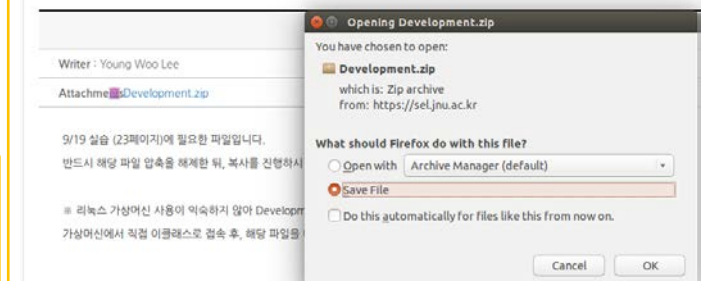
```

root@ubuntu: /Smart4412Linux/Development
root@ubuntu: /# ls
bin  dev  initrd.img  lost+found  opt  run  Smart4412Linux  vmlinuz
boot  etc  lib  media  proc  sbin  srv  usr
cdrom  home  lib64  mnt  root  selinux  sys  var

root@ubuntu: /# cd media/My Disc/
root@ubuntu: /media/My Disc# ls
Application  Development  Document  Util
root@ubuntu: /media/My Disc# cp -rf Development/ /Smart4412Linux/
root@ubuntu: /media/My Disc# cd /Smart4412Linux/
root@ubuntu: /Smart4412Linux# ls
Development
root@ubuntu: /Smart4412Linux# cd Development/
root@ubuntu: /Smart4412Linux/Development# ls
Image  Source  Toolchain
root@ubuntu: /Smart4412Linux/Development#
  
```

이클래스 접속 후 **Development.zip** 파일을 다운로드
 하여 압축 해제 후 **/smart4412linux**에 복사하여도 가능

자료실



1.2 리눅스 시작하기

1) 터미널(Terminal) (8/19)

- 패스워드 등록/변경 `passwd -[option]`
- 사용자 계정의 패스워드를 등록하거나 변경하는 경우 이용하는 명령어
- 슈퍼 유저는 다른 유저의 패스워드도 변경 가능
- [option]

-S : 계정의 상태를 표시한다.

(PS : 정상, NP : 패스워드가 없음, LK : Lock상태 혹은 NP 상태)

-d : 계정의 패스워드를 삭제, 패스워드 없이 로그인 가능

-l : 계정을 Lock 상태로 변경

-u : 계정의 Lock 상태를 해제

1.2 리눅스 시작하기

1) 터미널(Terminal) (9/19)

- 디렉터리 삭제 `rmdir -[option][디렉토리]`
- 디렉터리 삭제
- `[option]`

`-p` : 상위 경로도 삭제

1.2 리눅스 시작하기

1) 터미널(Terminal) (10/19)

- 파일 복사 `cp -[option] [대상파일] [디렉토리]`
- 파일 및 디렉터리 복사 명령어
- [option]

-i : 파일 복사 시, 동일한 파일명이 있을 경우 사용자에게 덮어 쓸 것인지 물어봄

-f : 동일한 파일명이 있을 경우에도 강제로 지우고 복사

-p : 원본 파일 소유주, 그룹, 권한, 시간 정보들이 그대로 보존되어 복사

-r : [원본파일명]이 경로일 경우, 그 경로에 있는 모든 하위 디렉토리들을 포함하여 모두 복사한다. 일반 파일일 경우는 그냥 복사

-u : 복사 대상이 이미 존재하며 파일의 날짜가 같거나 더 최신이면 복사하지 않음

1.2 리눅스 시작하기

1) 터미널(Terminal) (11/19)

- 파일삭제 `rm -[option] [대상파일]`
- 파일 및 디렉터리 삭제 명령어
- [option]

-i : 파일이나 디렉토리가 삭제될 때마다 확인(y : 삭제, n : 취소)
-f : 사용자에게 확인하지 않고 삭제
-v : 각각의 파일 지우는 정보를 자세하게 모두 보여준다.
-r : 해당 디렉토리의 하위 디렉토리까지 모두 삭제

1.2 리눅스 시작하기

1) 터미널(Terminal) (12/19)

- 파일의 이름이나 위치 변경 `mv -[option] [대상파일] [이동디렉터리]`
- `[option]`

`-f` : 이동할 위치에 파일 이름이 겹치면 강제로 이동

1.2 리눅스 시작하기

1) 터미널(Terminal) (13/19)

- 파일검색 find [대상파일] [option]
- 파일을 이름, 수정 날짜, 소유주, 허가 등을 기준으로 검색하게 해준다.
- [option]

-empty : 비어있는 파일을 찾을
-user [사용자명] : 지정한 사용자가 소유하고 있는 파일을 찾을
-group [그룹명] : 지정한 그룹이 소유하고 있는 파일을 찾을
-nouser : 소유주가 없는 파일을 찾을(사용자 계정을 삭제했을 경우 사용자가 생성한 파일들을 남기는 경우가 있음).
-nogroup : 소유그룹이 없는 파일을 검색
-name [파일명] : 지정한 패턴을 갖는 파일을 찾을(파일명 또는 확장명을 기준으로 검색).
-newer [파일명] : [파일명]보다 최근에 생성된 파일을 찾을
-anewer [파일명] : [파일명]보다 최근에 읽기(접근)가 수행된 파일을 찾을
-cnewer [파일명] : [파일명]보다 최근에 상태가 변경(수정)된 파일을 찾을
-perm [권한] : 권한과 일치하는 파일을 찾을
-type [파일타입] : [파일타입]에 맞는 파일을 찾을
[파일타입]
d : 디렉토리(Directory)
c : 캐릭터 특수 파일(Character Device)
f : 일반파일(File)
b : 블록 특수 파일(Block Device)
l : 심볼릭 링크(Link)
p : 파이프(Pipe)
s : 소켓(Socket)

1.2 리눅스 시작하기

1) 터미널(Terminal) (14/19)

- 파일 내용 출력 `cat [option] [대상파일]`
- 파일의 내용을 화면으로 출력한다. 파일 내용을 확인하거나, 여러 파일을 하나의 파일로 만들 수도 있다. 또한, 키보드 등 표준 입력으로부터 입력받은 내용을 파일로 만들어 저장할 수 있다.

- [option]

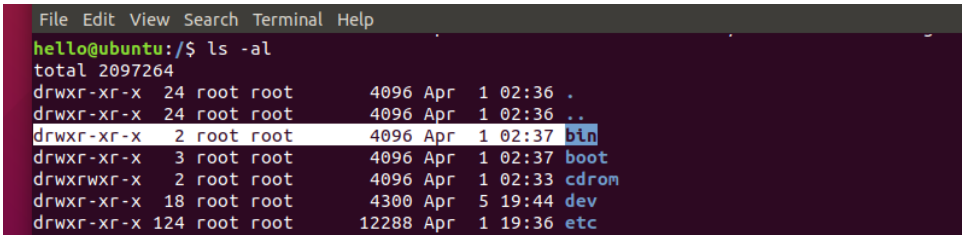
- `cat > [파일이름]` 형태로 명령어를 작성했을 경우, 파일 생성 및 데이터 입력도 가능
* 파일 저장 : `Ctrl + d`, 파일 종료 : `Ctrl + z`
- `cat [파일명] | more` : 엔터키를 입력할 때마다 한 줄씩 내려가면서 확인이 가능
- `cat [파일명] | less` : 화살표 위, 아래키로 페이지 올림, 내림이 가능
- `-n` : 파일을 출력할 때, 라인에 번호를 붙여 출력
- `-b` : 공백 외의 글자가 있는 라인에 번호를 붙여 출력

1.2 리눅스 시작하기

1) 터미널(Terminal) (15/19)

- 리눅스는 다중 사용자가 시스템 접근이 가능하며 사용자는 하나 또는 여러 그룹에 속함
- 파일에 사용자와 그룹의 접근 권한을 부여하여 접근을 제한하며 파일의 접근 권한은 ls 명령어로 확인할 수 있음

```
hello@ubuntu:~$ ls -al
```



권한표시	파일 수	소유자	그룹	크기	변경 일자	파일명
drwxr-xr-x	2	root	root	4096	Apr 1 02:37	bin

권한 표시에 해당하는 정보의 의미

d	rwX	r-X	r-X
(1)	(2)	(3)	(4)

1.2 리눅스 시작하기

1) 터미널(Terminal) (16/19)

- (1) 파일 Type : d(디렉토리), l(링크파일), -(일반파일)
- (2)(3)(4) 파일 소유자 권한 : r(읽기권한), w(쓰기권한), x(실행권한)
- (2) 파일 소유자 권한
- (3) 그룹 권한
- (4) 모든 사용자 권한
- 권한 표시 : r(4), w(2), x(1)
- ex) 모든 권한의 경우 111,111,111(십진수 7로 표시하면 777)
- ex) \$ chmod 777 filename
- 파일소유자는 모든 권한, 그룹과 기타사용자는 읽기와 실행 권한만 부여

1.2 리눅스 시작하기

1) 터미널(Terminal) (17/19)

- 프로세서의 정보 및 메모리, 네트워크 관리 기능 명령어
- 시스템관리 명령어

프로세서에 관한 리스트를 보여줌

```
ps
```

현재 프로세서를 종료하고 빠져나감

```
exit
```

실행 중인 프로세서 ID를 종료

```
Kill [프로세서 ID]
```

명령어를 단순화하여 자주 쓰는 명령어를 쓰기 편하게 바꾸는 명령, 일종의 명령어의 별명

```
alias
```

mv를 mv -i로 별명을 하나 더 만들 mv를 mv -i로 사용하여, mv만 입력해도 겹쳐 쓸지 여부를 확인

```
alias mv = 'mv -i'
```

현재 시각 출력

```
date
```

```
$date
```

```
Sun Apr 5 20:07:05 PDT 2020
```

패키지 인덱스 정보를 업데이트

```
apt-get update
```

1.2 리눅스 시작하기

1) 터미널(Terminal) (18/19)

- 프로세서의 정보 및 메모리, 네트워크 관리 기능 명령어
- 시스템관리 명령어

설치되어 있는 패키지를 모두 새 버전으로 업그레이드

```
apt-get upgrade
```

패키지를 설치한다

```
apt-get install [패키지 이름]
```

설치한 패키지를 지움 (설정 파일은 지우지 않음)

```
apt-get remove [패키지 이름]
```

설치한 패키지를 설정 파일을 포함하여 지움

```
apt-get --purge remove [패키지 이름]
```

파일 시스템 구조와 용량 등을 화면에 출력

```
df
```

메모리의 사용 상태와 버퍼의 상태 등을 보여줌

```
free
```

```
hello@ubuntu:/$ free
              total        used        free      shared  buff/cache   available
Mem:           4002252      1237628       800012         9988       1964612       2465336
Swap:          2097148           268       2096880
```

네트워크 호스트명 출력

```
hostname
```

1.2 리눅스 시작하기

1) 터미널(Terminal) (19/19)

- 프로세서의 정보 및 메모리, 네트워크 관리 기능 명령어
- 시스템관리 명령어

네트워크 확인 명령

```
ping [옵션][IP주소 또는 호스트 이름]
```

LAN 카드 상태 확인

```
ifconfig [인터페이스] [주소] [up/down] [옵션]
```

네트워크 상태 확인

```
Netstat [옵션][address family options]
```

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (1/6)

- 리눅스의 명령 줄 인터페이스(CLI, Command Line Interface)환경에서 코드를 작성하기 위해서는 에디터를 사용
- vi는 vi(sual) editor의 약자로 키보드 입력만으로 모든 기능을 사용할 수 있게 제공하는 터미널 에디터 프로그램
- vi 에디터는 크게 명령 모드와 입력 모드로 나눌 수 있고, 명령 모드에는 vi 명령 모드와 ex 명령 모드가 있음
- vi 시작시, vi 명령 모드로 시작하며, 특정키를 눌러, 입력 모드 및 ex 명령 모드로 전환 가능
- vi 명령 모드로 돌아오기 위해서는, Esc(Escape)키를 누르면 돌아옴

version 1.1

April 1st, 06

Esc

normal mode

vi / vim graphical cheat sheet

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	% next ident	(begin sentence) end sentence	"soft" bol	+ next line
↵ goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	w next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	* misc	.	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	. ex cmd line	" reg. 1	bol/ goto col	
a append	s subst char	d delete	f find char	g extra cmds	h ←	j ↓	k ↑	l →	* repeat 1	' goto mk. bol	\ not used!	
Z quit	X back-space	C change to col	V visual lines	B prev WORD	N prev (find)	M screen mid	< un-indent	> indent	? find (rev.)			
Z extra cmds	X delete char	c change	v visual mode	b prev word	n next (find)	m set mark	reverse	repeat cmd	/ find			

motion

moves the cursor, or defines the range for an operator

command

direct action command, if red, it enters insert mode

operator

requires a motion afterwards, operates between cursor & destination

extra

special functions, requires extra input

q

commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foO, baR, baZ)`

WORDS: `quux(foO, baR, baZ)`

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)

:e f (open file f), :%s/s/y/g (replace 's' by 'y' filewide), :h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim), CTRL-F/B: page up/down, CTRL-E/Y: scroll line up/down, CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,") (e.g.: "ay\$ to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top, zb: bottom, zz: center

(6) gg: top of file (vim only), gF: open file under cursor (vim only)

[그림] VI에디터 명령어

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (2/6)

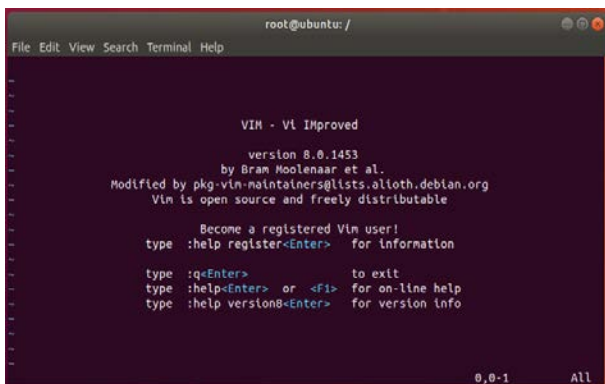
- Vim은 vi와 호환되면서도 독자적으로 다양한 기능을 추가하여 사용자의 편의를 돕고 있음
- 특히 Vim 스크립트 등을 사용해서 자유롭게 편집 환경을 변경하거나, 확장된 정규 표현식 문법, 강력한 문법 강조 기능, 다중 되돌리기, 유니코드를 비롯한 다국어 지원, 문법 검사 등을 쓸 수 있다는 점이 강점
- 한편으로는 vi와 마찬가지로 처음에 배우기 어렵다는 점이 단점으로 지적, 이를 극복하기 위해 쉬운 Vim 모드를 지원

리눅스 버전에 따라 vim이 기본설치 되어 있으며 추가 설치가 필요하면 다음과 같이 입력

```
root@Ubuntu:/# apt-get install vim
```

파일 생성은 vim명령과 함께 파일 이름을 입력

```
root@Ubuntu:/# vim [filename]
```



[그림] vim 시작화면

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (3/6)

- 에디터를 실행하면 기본 상태는 명령모드
- 키 앞에 숫자를 입력하면 그 명령이 숫자만큼 반복되며 입력 모드에서 입력 작업 완료 후, 커서 이동이나 삭제, 복사, 붙여 넣기 등의 편집 작업 등을 할 때는 Esc키를 눌러 vi 명령 모드로 전환시킨 후에 작업을 진행

① 커서이동

- h : 한 문자 왼쪽으로 커서 이동
- j : 한 줄 아래쪽으로 커서 이동
- k : 한 줄 위쪽으로 커서 이동
- l : 한 문자 오른쪽으로 커서 이동
- w : 다음 단어 첫 문자로 커서 이동
- b : 이전 단어 첫 문자로 커서 이동
- Ctrl+f : 한 페이지 아래로 커서 이동
- Ctrl+b : 한 페이지 위로 커서 이동
- G : 마지막 줄로 커서 이동, 숫자G로 눌렀을 때, 숫자에 해당되는 줄로 커서 이동
- ^ : 공백 문자가 아닌, 현재 줄의 첫 문자로 커서 이동
- 0 : 공백 문자를 포함한, 현재 줄 첫 문자로 커서 이동
- \$: 현재 줄 마지막 문자로 커서 이동

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (4/6)

② 삭제

- x : 현재 커서 위치 문자 삭제
- X : 현재 커서 위치 이전 문자 삭제
- dw : 현재 커서 위치 단어 삭제, 숫자dw로 쓰면 숫자만큼의 단어가 삭제됨, 다른 삭제 명령도 마찬가지로 숫자 입력 방식이 적용됨
- db : 현재 커서 위치 이전 단어 삭제
- dd : 현재 커서 위치 줄 삭제, 숫자dd로 쓰면 숫자만큼의 줄이 삭제됨
- d^ : 현재 줄에서 현재 커서 위치 이전 문자열 삭제
- d\$: 현재 줄에서 현재 커서 위치 이후 문자열 삭제
- D : d\$와 같음

③ 복사, 붙여 넣기

- yw : 현재 커서 이후 단어 복사, 숫자yw로 쓰면 숫자만큼의 단어가 복사됨, 다른 복사 명령에도 이와 같은 숫자 입력 방식이 적용됨
- yb : 현재 커서 이전 단어 복사
- yy : 현재 줄 복사, 숫자yy로 쓰면 숫자만큼의 줄이 복사됨
- p : 복사된 항목을 현재 커서 위치 이후에 붙여 넣기, 삭제된 항목도 붙여 넣기 가능
- P : 복사된 항목을 현재 커서 위치 이전에 붙여 넣기, 삭제된 항목도 붙여 넣기 가능

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (5/6)

④ 검색

- /검색어 : 전진 검색
- ?검색어 : 후진 검색
- n : 검색 반복
- N : 반대 방향 검색 반복

⑤ 되돌리기(undo)

- u : 되돌리기(undo). u키를 계속 치게되면, undo 버퍼가 빌 때까지 계속 되돌리기할 수 있음. 유용함
- U : 라인 되돌리기
- Ctrl+r : Redo(undo undo)

1.2 리눅스 시작하기

2) 코드 편집기(VI Editor) (6/6)

- 에디터에 글자를 입력하기 위해서는 명령 모드에서 입력 모드로 모드 전환이 이루어져야 한다.
- i : 현재 커서 위치에 글자 삽입
- I : 현재 줄 처음에 삽입
- a : 현재 커서 다음 위치에 추가
- A : 현재 줄 마지막에 추가
- o : 아랫 줄에 추가
- O : 윗 줄에 추가
- s : 현재 커서 글자 지우고 입력 모드로 전환
- r : 현재 커서 글자 지우고 한 글자 입력 받아 명령 모드로 돌아감
- Esc : 입력 모드 또는 ex 명령 모드에서 vi 명령 모드로 전환

1.3 우분투 환경설정

1) GCC-4.4 / U-boot-tools 패키지 설치

□ GCC-4.4 설치

```
# apt-get install gcc-4.4 g++-4.4 g++-4.4-multilib gcc-4.4-multilib --yes
```

```
root@ubuntu: /
root@ubuntu:/#
root@ubuntu:/# apt-get install gcc-4.4 g++-4.4 g++-4.4-multilib gcc-4.4-multilib --yes
```

□ GCC-4.4 환경 변수 추가 & 적용

```
# echo "export CC=gcc-4.4 CXX=g++-4.4" >> ~/.bashrc
# source ~/.bashrc
```

```
root@ubuntu: /
Reading state information... Done
g++-4.4 is already the newest version.
gcc-4.4 is already the newest version.
g++-4.4-multilib is already the newest version.
gcc-4.4-multilib is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@ubuntu:/# echo "export CC=gcc-4.4 CXX=g++-4.4" >> ~/.bashrc
root@ubuntu:/# source ~/.bashrc
root@ubuntu:/#
```

□ U-boot tools 설치

```
# apt-get install u-boot-tools
```

```
root@ubuntu: /
root@ubuntu:/#
root@ubuntu:/# apt-get install u-boot-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  u-boot-tools
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 60.4 kB of archives.
After this operation, 180 kB of additional disk space will be used.
0% [working]
```

GCC-4.4 설치 후 version 확인

```
# gcc --version
# g++ --version
```

```
Terminal
root@ubuntu: /
root@ubuntu:/# gcc --version
gcc (Ubuntu 4.8.4-2ubuntu1~14.04.4) 4.8.4
Copyright (C) 2013 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

root@ubuntu:/# g++ --version
The program 'g++' is currently not installed. You can install it by typing:
apt-get install g++
root@ubuntu:/# sudo dpkg -l | grep gcc | awk '{print $2}'
gcc
gcc-4.4
gcc-4.4-base:amd64
gcc-4.4-multilib
gcc-4.8
gcc-4.8-base:amd64
gcc-4.8-multilib
gcc-4.9-base:amd64
gcc-multilib
lib32gcc-4.8-dev
lib32gcc1
libgcc-4.8-dev:amd64
libgcc1:amd64
libx32gcc-4.8-dev
libx32gcc1
```

GCC-4.4 version을 우선 순위 설정

```
root@ubuntu:/# sudo dpkg -l | grep g++ | awk '{print $2}'
g++-4.4
g++-4.4-multilib
root@ubuntu:/# sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.4 10
root@ubuntu:/# sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 20
root@ubuntu:/# sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.4 10
root@ubuntu:/#
root@ubuntu:/# sudo update-alternatives --install /usr/bin/cc cc /usr/bin/gcc 30
root@ubuntu:/# sudo update-alternatives --set cc /usr/bin/gcc
root@ubuntu:/# sudo update-alternatives --install /usr/bin/c++ c++ /usr/bin/g++ 30
root@ubuntu:/# sudo update-alternatives --set c++ /usr/bin/g++
root@ubuntu:/# sudo update-alternatives --config gcc
```

1.3 우분투 환경설정

2) Toolchain 설치

- Toolchain(arm-2010q1) 복사 및 압축해제

```
# cd /Smart4412Linux/Development/Toolchain/
# cp /arm-2010q1.tgz /usr/local/
# cd /usr/local/
# tar zxvf arm-2010q1.tgz
```

```
root@ubuntu: /usr/local
root@ubuntu: /# cd /Smart4412Linux/Development/Toolchain/
root@ubuntu: /Smart4412Linux/Development/Toolchain# ls
arm-2010q1.tgz  gnueabi.tgz
root@ubuntu: /Smart4412Linux/Development/Toolchain# cp arm-2010q1.tgz /usr/local/
root@ubuntu: /Smart4412Linux/Development/Toolchain# cd /usr/local/
root@ubuntu: /usr/local# ls
arm-2010q1.tgz  bin  etc  games  include  lib  man  sbin  share  src
root@ubuntu: /usr/local# tar zxvf arm-2010q1.tgz
```

Toolchain 설치 경로로 이동하여 개발에 사용되는 컴파일러 정보를 확인한다.

- Toolchain(arm-2010q1) 설치 확인

```
# cd /usr/local/CodeSourcery/Sourcery_G++_Lite/bin/
```

```
root@ubuntu: /usr/local/CodeSourcery/Sourcery_G++_Lite/bin
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Preprocessor.html
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Debugger.pdf
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/C_Library_NewLib.html
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Compiler.html
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Debugger.html
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Profiler.pdf
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Binary Utilities.html
CodeSourcery/Sourcery_G++_Lite_for_ARM_EABI/Getting_Started_Guide.html
root@ubuntu: /usr/local# cd CodeSourcery/Sourcery_G++_Lite/bin/
root@ubuntu: /usr/local/CodeSourcery/Sourcery_G++_Lite/bin# ls
arm-none-eabi-addr2line  arm-none-eabi-g++  arm-none-eabi-gprof  arm-none-eabi-readelf
arm-none-eabi-ar  arm-none-eabi-gcc  arm-none-eabi-ld  arm-none-eabi-run
arm-none-eabi-as  arm-none-eabi-gcc-4.4.1  arm-none-eabi-nm  arm-none-eabi-size
arm-none-eabi-c++  arm-none-eabi-gcov  arm-none-eabi-objcopy  arm-none-eabi-sprite
arm-none-eabi-c++filt  arm-none-eabi-gdb  arm-none-eabi-objdump  arm-none-eabi-strings
arm-none-eabi-cpp  arm-none-eabi-gdbtui  arm-none-eabi-ranlib  arm-none-eabi-strip
root@ubuntu: /usr/local/CodeSourcery/Sourcery_G++_Lite/bin#
```

1.3 우분투 환경설정

2) Toolchain 설치

응용프로그램 컴파일에 필요한 두 번째 Toolchain을 설치해보도록 한다. 불체인을 “/opt” 디렉터리에 복사 후 압축을 해제한다.

- Toolchain(gnueabi) 복사 및 압축해제

```
# cd /Smart4412Linux/Development/Toolchain/
# cp gnueabi.tgz /opt/
# cd /opt/
# tar zxvf gnueabi.tgz
```

```
root@ubuntu: /opt
root@ubuntu: /# cd /Smart4412Linux/Development/Toolchain/
root@ubuntu: /Smart4412Linux/Development/Toolchain# ls
arm-2010q1.tgz  gnueabi.tgz
root@ubuntu: /Smart4412Linux/Development/Toolchain# cp gnueabi.tgz /opt/
root@ubuntu: /Smart4412Linux/Development/Toolchain# cd /opt/
root@ubuntu: /opt# ls
gnueabi.tgz
root@ubuntu: /opt# tar zxvf gnueabi.tgz
```

Toolchain 설치 경로로 이동하여 개발에 사용되는 컴파일러 정보를 확인한다.

- Toolchain(gnueabi) 설치 확인

```
# cd /opt/gnueabi/opt/ext-toolchain/bin
```

```
root@ubuntu: /opt/gnueabi/opt/ext-toolchain/bin
gnueabi/opt/ext-toolchain/lib/gcc/arm-linux-gnueabihf/4.7.3/libgcc.a
gnueabi/opt/ext-toolchain/lib/gcc/arm-linux-gnueabihf/4.7.3/libgcc.a
gnueabi/opt/ext-toolchain/lib/gcc/arm-linux-gnueabihf/4.7.3/crtbeginS.o
root@ubuntu: /opt# cd gnueabi/opt/ext-toolchain/bin/
root@ubuntu: /opt/gnueabi/opt/ext-toolchain/bin# ls
arm-linux-gnueabihf-addrd2line  arm-linux-gnueabihf-gcc-ar  arm-linux-gnueabihf-nm
arm-linux-gnueabihf-ar  arm-linux-gnueabihf-gcc-nm  arm-linux-gnueabihf-objcopy
arm-linux-gnueabihf-as  arm-linux-gnueabihf-gcc-ranlib  arm-linux-gnueabihf-objdump
arm-linux-gnueabihf-c++  arm-linux-gnueabihf-gcov  arm-linux-gnueabihf-pkg-config
arm-linux-gnueabihf-c++filt  arm-linux-gnueabihf-gdb  arm-linux-gnueabihf-pkg-config-real
arm-linux-gnueabihf-cpp  arm-linux-gnueabihf-gfortran  arm-linux-gnueabihf-ranlib
arm-linux-gnueabihf-ct-ng.config  arm-linux-gnueabihf-gprof  arm-linux-gnueabihf-readelf
arm-linux-gnueabihf-elfedit  arm-linux-gnueabihf-ld  arm-linux-gnueabihf-size
arm-linux-gnueabihf-g++  arm-linux-gnueabihf-ld.bfd  arm-linux-gnueabihf-strings
arm-linux-gnueabihf-gcc  arm-linux-gnueabihf-ldd  arm-linux-gnueabihf-strip
arm-linux-gnueabihf-gcc-4.7.3  arm-linux-gnueabihf-ld.gold
root@ubuntu: /opt/gnueabi/opt/ext-toolchain/bin#
```

1.4 Homework

- 하기 캡처 사진과 함께 고찰을 포함한 보고서 작성
 - ✓ [23페이지] /smart4412linux/development 디렉토리에서 ls 명령 결과
 - ✓ [43페이지] /usr/local/CodeSourcery/Sourcery_G++_Lite/bin 디렉토리에서 ls 명령 결과
 - Toolchain(arm-2010q1) 설치 확인
 - ✓ [44페이지] /opt/gnueabi/opt/ext-toolchain/bin 디렉토리에서 ls 명령 결과
 - Toolchain(gnueabi) 설치 확인

Any Questions?

Young-woo Lee
E-mail. ylee@jnu.ac.kr