Operating Systems Lab

Part 0: Installing PintOS

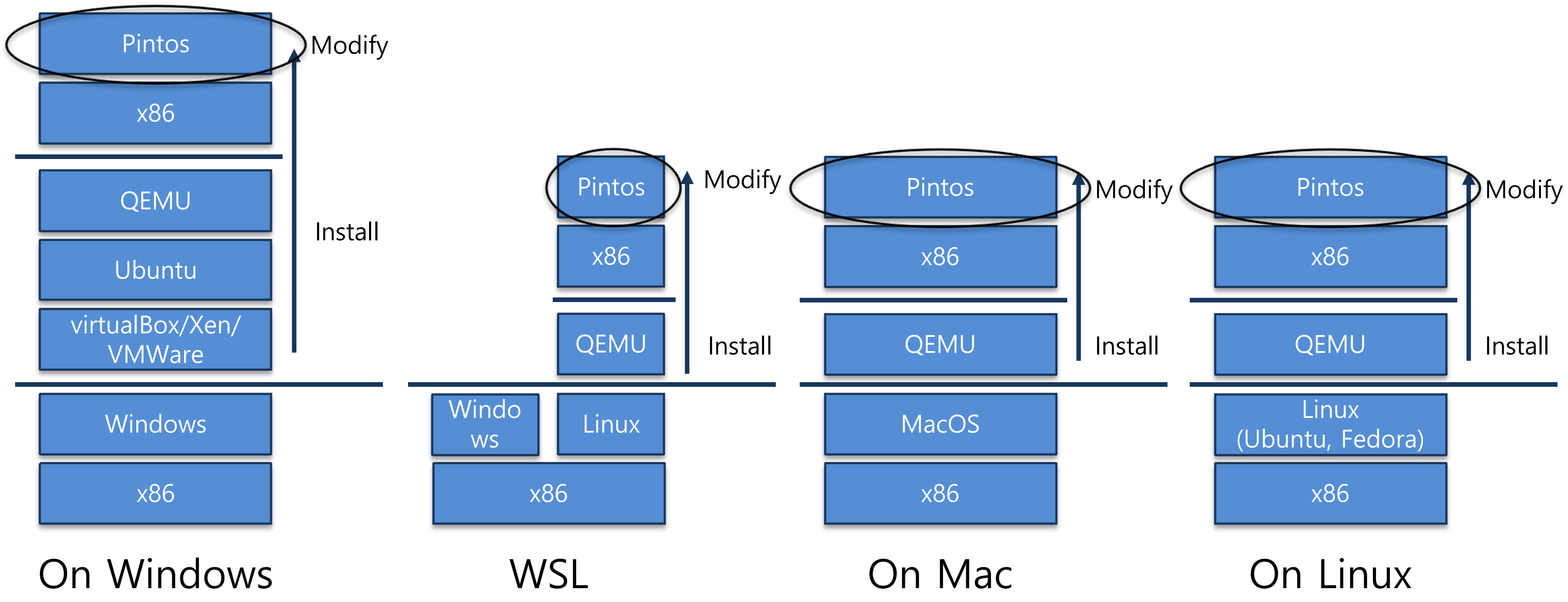
KAIST

Youjip Won

PintOS

- Pintos?
  - Educational operating system for x86 architecture
  - Developed by Ben Pfaff in Stanford Univ, 2004
  - Support kernel threads, loading and running user programs, file system, etc.
  - Uses x86 simulators, such as Bochs or QEMU
- Why we use Pintos?
  - It is important to implement a variety of concepts (threads, processes, memory man agement, and file systems) in the operating system manually
  - Commercial operating systems, such as Linux are very large(1 million lines). More th an 80% of 1 million lines are device driver codes to support hardware.
  - Linux compile: takes at least an hour.
  - PintoS : Simple, easy to understand, easy to compile

Execution of Pintos



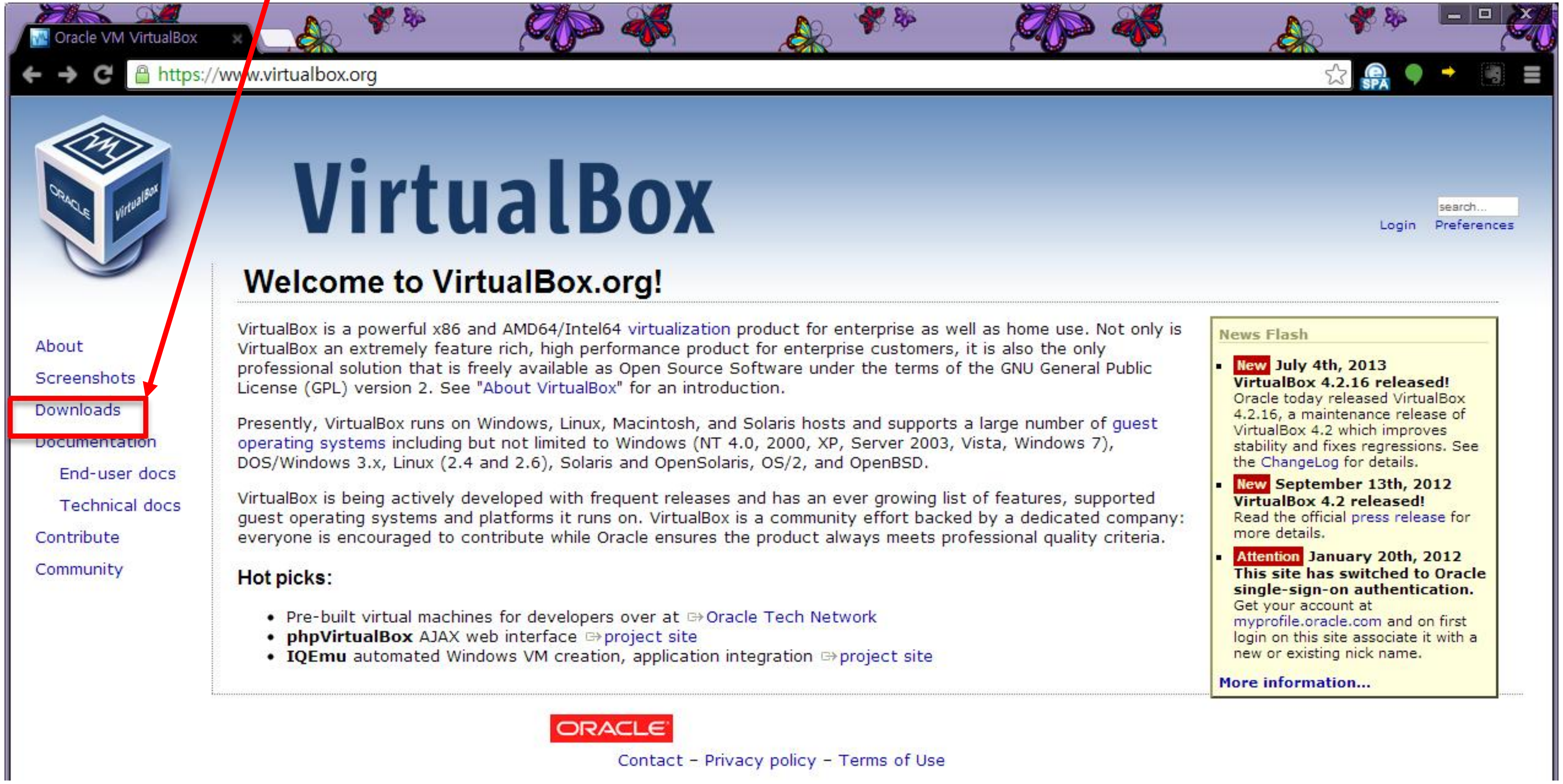
Install pintos

- Install pintos on Windows
  - VM + Linux + QEMU + PintOS
  - WSL + QEMU + PintOS
- Install pintos on Linux
  - QEMU + PintOS
- Install PintOS on MacOS
  - QEMU + PintOS

For Windows user: Install Virtual Box

- Download Virtual Box at http://www.virtualbox.org and install it.

Here, Download



Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7), DOS/Windows 3.x, Linux (2.4 and 2.6), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

Hot picks:

- Pre-built virtual machines for developers over at Oracle Tech Network
- **phpVirtualBox** AJAX web interface project site
- **IQEmu** automated Windows VM creation, application integration project site

News Flash

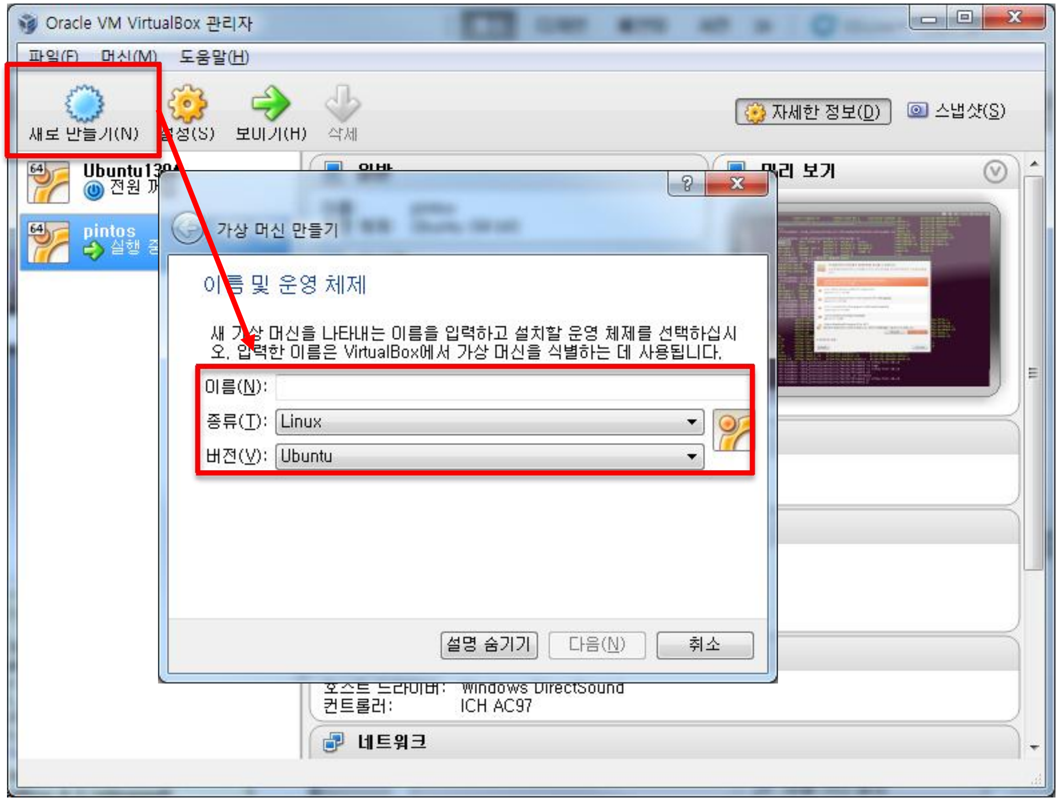
- New **July 4th, 2013**
  VirtualBox 4.2.16 released!
  Oracle today released VirtualBox 4.2.16, a maintenance release of VirtualBox 4.2 which improves stability and fixes regressions. See the ChangeLog for details.
- New **September 13th, 2012**
  VirtualBox 4.2 released!
  Read the official press release for more details.
- Attention **January 20th, 2012**
  This site has switched to Oracle single-sign-on authentication.
  Get your account at myprofile.oracle.com and on first login on this site associate it with a new or existing nick name.

More information...

ORACLE
Contact – Privacy policy – Terms of Use

For Windows user: Install Ubuntu

- Create a Linux Virtual Machine on VirtualBox, and then install Linux (Ubuntu 18.04 LTS)
  - Download ubuntu-18.04.6-desktop-amd64.iso : https://releases.ubuntu.com/18.04.6/?_ga=2.173865891.1436103949.1646090779-879543907.1646090779
  - Create the Virtual Machine



이름 및 운영 체제

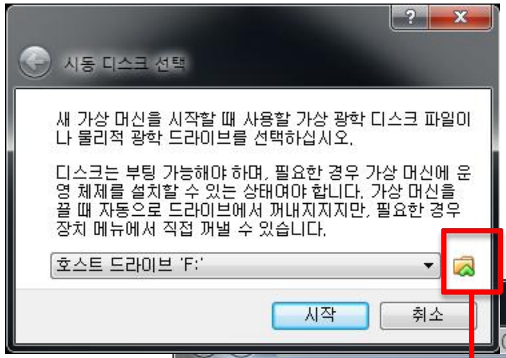
새 가상 머신을 나타내는 이름을 입력하고 설치할 운영 체제를 선택하십시오. 입력한 이름은 VirtualBox에서 가상 머신을 식별하는 데 사용됩니다.

이름(N):
종류(T): Linux
버전(V): Ubuntu

호스트 드라이버: Windows DirectSound
컨트롤러: ICH AC97

For Windows user: Install Ubuntu

- Mount the Ubuntu image file, and install



시동 디스크 선택

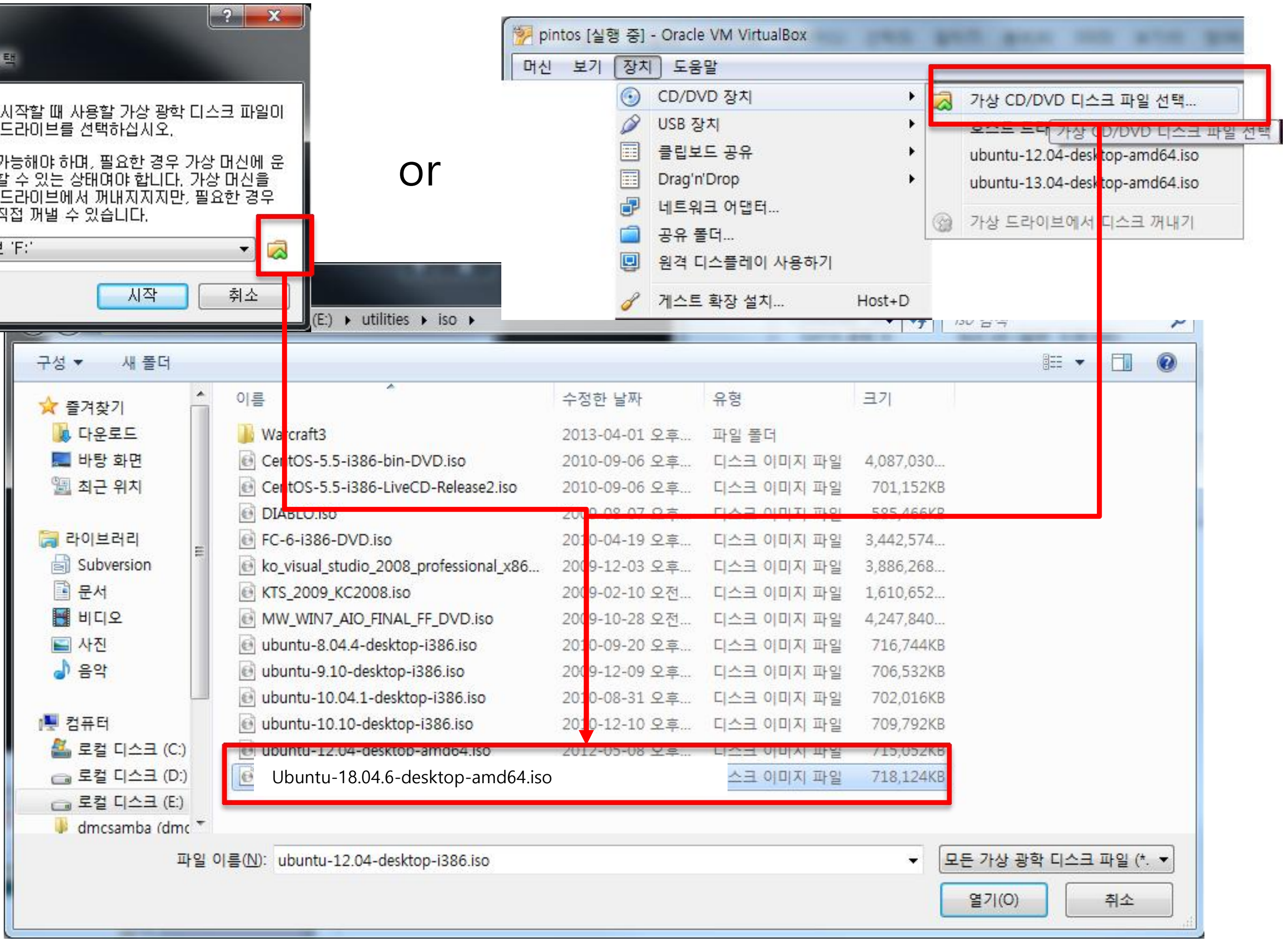
새 가상 머신을 시작할 때 사용할 가상 광학 디스크 파일이나 물리적 광학 드라이브를 선택하십시오.

디스크는 부팅 가능해야 하며, 필요한 경우 가상 머신에 운영 체제를 설치할 수 있는 상태여야 합니다. 가상 머신을 끌 때 자동으로 드라이브에서 꺼내지지만, 필요한 경우 장치 메뉴에서 직접 꺼낼 수 있습니다.

호스트 드라이브 'F:'

시작 | 취소

or



pintos [실행 중] - Oracle VM VirtualBox

머신 보기 장치 도움말

이름	수정한 날짜	유형	크기
Warcraft3	2013-04-01 오후...	파일 폴더	
CentOS-5.5-i386-bin-DVD.iso	2010-09-06 오후...	디스크 이미지 파일	4,087,030...
CentOS-5.5-i386-LiveCD-Release2.iso	2010-09-06 오후...	디스크 이미지 파일	701,152KB
DIABLO.iso	2010-08-07 오후...	디스크 이미지 파일	585,466KB
FC-6-i386-DVD.iso	2010-04-19 오후...	디스크 이미지 파일	3,442,574...
ko_visual_studio_2008_professional_x86...	2009-12-03 오후...	디스크 이미지 파일	3,886,268...
KTS_2009_KC2008.iso	2009-02-10 오전...	디스크 이미지 파일	1,610,652...
MW_WIN7_AIO_FINAL_FF_DVD.iso	2009-10-28 오전...	디스크 이미지 파일	4,247,840...
ubuntu-8.04.4-desktop-i386.iso	2010-09-20 오후...	디스크 이미지 파일	716,744KB
ubuntu-9.10-desktop-i386.iso	2009-12-09 오후...	디스크 이미지 파일	706,532KB
ubuntu-10.04.1-desktop-i386.iso	2010-08-31 오후...	디스크 이미지 파일	702,016KB
ubuntu-10.10-desktop-i386.iso	2010-12-10 오후...	디스크 이미지 파일	709,792KB
ubuntu-12.04-desktop-amd64.iso	2012-05-08 오후...	디스크 이미지 파일	715,052KB
Ubuntu-18.04.6-desktop-amd64.iso		스크 이미지 파일	718,124KB

파일 이름(N): ubuntu-12.04-desktop-i386.iso

모든 가상 광학 디스크 파일 (*.

열기(O) | 취소

For Windows user: Ubuntu Installation Complete

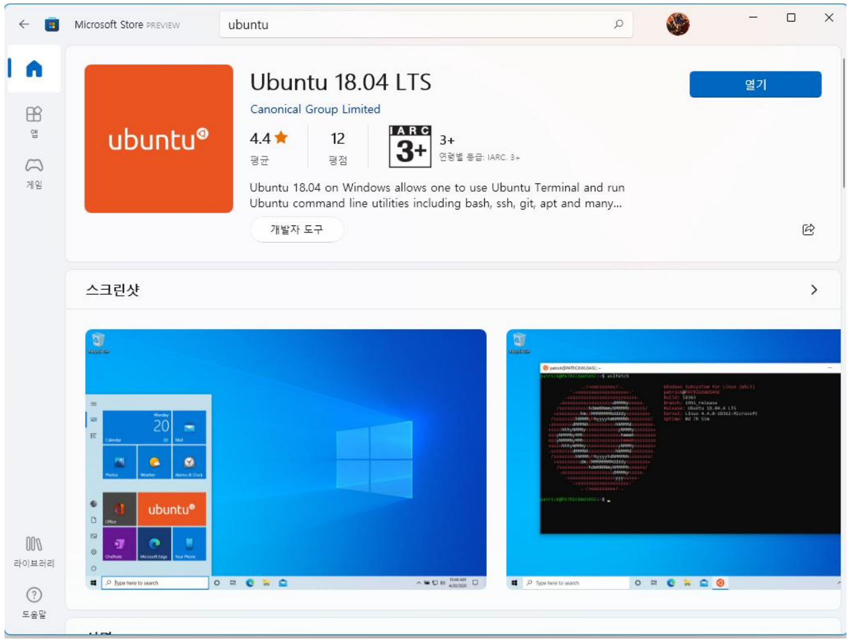
- Ubuntu Installation Complete and booting

WSL: Install WSL (Recommended)

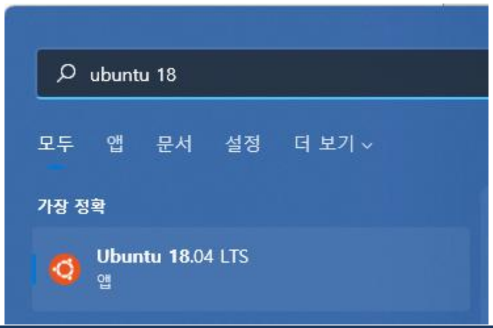
- https://docs.microsoft.com/en-us/windows/wsl/install
- Ensure using Windows 10 Version>=2004 or Windows 11
- Open Powershell or Windows Terminal with administrator
- Run `wsl --install`

WSL: Install Ubuntu 18.04

- Open Microsoft Store (https://aka.ms/wslstore)
- Search 'Ubuntu' and find Ubuntu 18.04 LTS

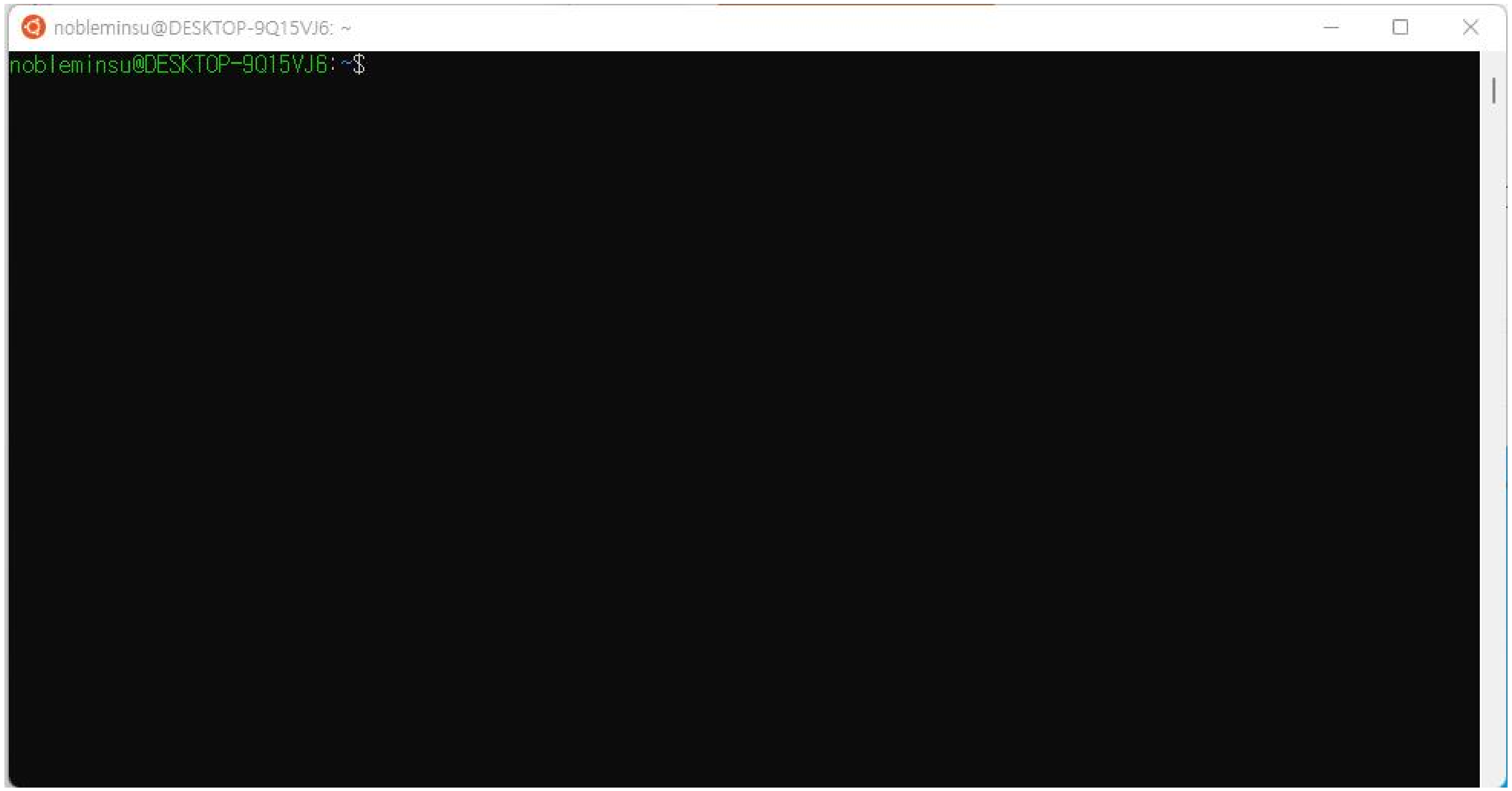

Microsoft Store PREVIEW
ubuntu
Ubuntu 18.04 LTS
Canonical Group Limited
열기
4.4 평균
12 평점
3+
Ubuntu 18.04 on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many...
개발자 도구
스크린샷

- Install it
- Find the installed program and run it



WSL: Ubuntu Installation Complete

- Set up account and you will see shell screen



```
nobleminsu@DESKTOP-9Q15VJ6:~$
```

For Linux: Install QEMU

- Install QEMU on system

  (“qemu-system-i386” command is available after below command)

```
$ sudo apt-get install qemu
```

- Make link “qemu”

  (“qemu” command is available after executing the command below)

```
$ sudo ln -s /usr/bin/qemu-system-i386 /usr/bin/qemu
```

For MacOS: Install QEMU

- `Install qemu`

```
$ brew install qemu
```

- Make link "`qemu`"

  ("`qemu`" command is available after below command)

```
 $ sudo ln -s /usr/bin/qemu-system-i386 /usr/bin/qemu
```

Errors

- Error 1: C compiler cannot create executables
  - Install gcc, g++ and library package

```
$ sudo apt-get install libc6-dev g++ gcc
```

- Error 2: X windows libraries were not found
  - Install X windows library

```
$ sudo apt-get install xorg-dev
```

Install PintOS

- Download the source code from the class piazza.
- `Unzip and untar`

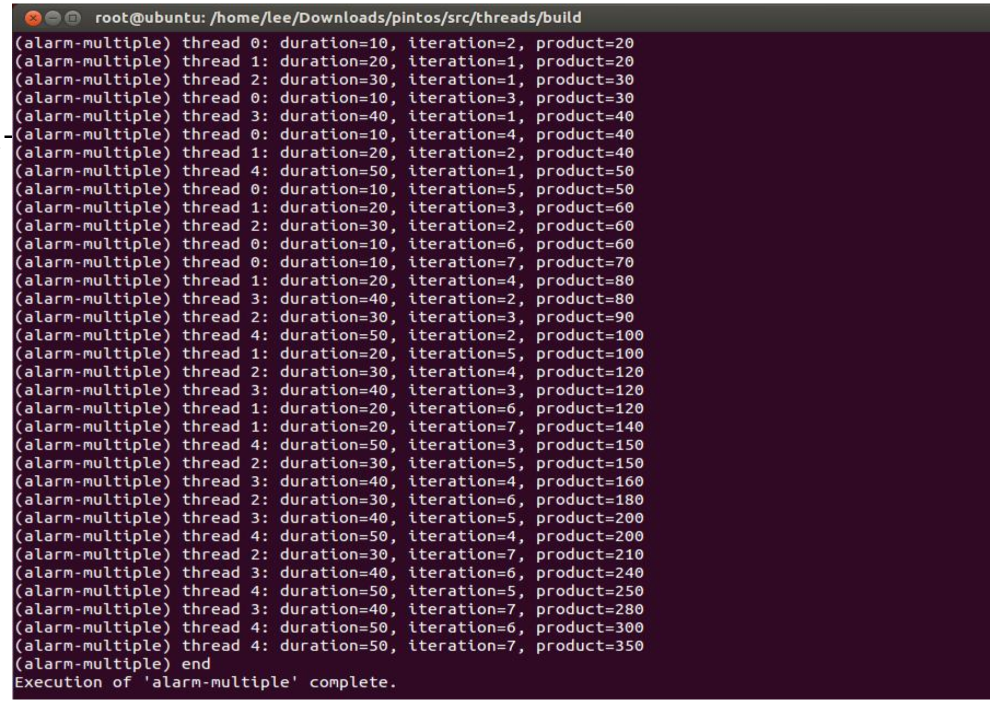
```
$ tar xvf pintos.tar.gz
```

- `cd` to pintos/src/threads/

```
$ make

$ cd build

$ pintos --qemu -- -q run alarm-
```



root@ubuntu: /home/lee/Downloads/pintos/src/threads/build

```
(alarm-multiple) thread 0: duration=10, iteration=2, product=20
(alarm-multiple) thread 1: duration=20, iteration=1, product=20
(alarm-multiple) thread 2: duration=30, iteration=1, product=30
(alarm-multiple) thread 0: duration=10, iteration=3, product=30
(alarm-multiple) thread 3: duration=40, iteration=1, product=40
(alarm-multiple) thread 0: duration=10, iteration=4, product=40
(alarm-multiple) thread 1: duration=20, iteration=2, product=40
(alarm-multiple) thread 4: duration=50, iteration=1, product=50
(alarm-multiple) thread 0: duration=10, iteration=5, product=50
(alarm-multiple) thread 1: duration=20, iteration=3, product=60
(alarm-multiple) thread 2: duration=30, iteration=2, product=60
(alarm-multiple) thread 0: duration=10, iteration=6, product=60
(alarm-multiple) thread 0: duration=10, iteration=7, product=70
(alarm-multiple) thread 1: duration=20, iteration=4, product=80
(alarm-multiple) thread 3: duration=40, iteration=2, product=80
(alarm-multiple) thread 2: duration=30, iteration=3, product=90
(alarm-multiple) thread 4: duration=50, iteration=2, product=100
(alarm-multiple) thread 1: duration=20, iteration=5, product=100
(alarm-multiple) thread 2: duration=30, iteration=4, product=120
(alarm-multiple) thread 3: duration=40, iteration=3, product=120
(alarm-multiple) thread 1: duration=20, iteration=6, product=120
(alarm-multiple) thread 1: duration=20, iteration=7, product=140
(alarm-multiple) thread 4: duration=50, iteration=3, product=150
(alarm-multiple) thread 2: duration=30, iteration=5, product=150
(alarm-multiple) thread 3: duration=40, iteration=4, product=160
(alarm-multiple) thread 2: duration=30, iteration=6, product=180
(alarm-multiple) thread 3: duration=40, iteration=5, product=200
(alarm-multiple) thread 4: duration=50, iteration=4, product=200
(alarm-multiple) thread 2: duration=30, iteration=7, product=210
(alarm-multiple) thread 3: duration=40, iteration=6, product=240
(alarm-multiple) thread 4: duration=50, iteration=5, product=250
(alarm-multiple) thread 3: duration=40, iteration=7, product=280
(alarm-multiple) thread 4: duration=50, iteration=6, product=300
(alarm-multiple) thread 4: duration=50, iteration=7, product=350
(alarm-multiple) end
Execution of 'alarm-multiple' complete.
```

Install PintOS (MacOS)

- Change the simulator to qemu in Make.vars in src directory.

```
SIMULATOR = --qemu
```

- Trouble shooting I

  Can't exec "qemu": No such file or directory at /home/arpith/pintos/src/utils/pintos line 923.

  Change the line 623 of perl script (filename: pintos in src/utils)

```
before: my (@cmd) = ('qemu');
After:  my (@cmd) = ('qemu-system-i386');
```

-

Install PintOS (MacOS)

- Trouble shooting II

```
baekdu:threads yjwon$ make
cd build && /Applications/Xcode.app/Contents/Developer/usr/bin/make all
../../Make.config:37: *** Compiler (i386-elf-gcc) not found.  Did you set $PATH
properly? Please refer to the Getting Started section in the documentation for d
etails. ***
/bin/sh: i386-elf-ld: command not found
make[1]: Nothing to be done for `all'.
```

- Add the location of i386-elf-ld to your $PATH.

```
% echo PATH="$PATH:/opt/local/bin" >> ~/.bashrc

% source ~/.bashrc
```

Install PintOS (MacOS)

- Trouble shoot III

```
cd build && /Applications/Xcode.app/Contents/Developer/usr/bin/make all
ld: unknown option: -melf_i386
gcc -m32 -E -nostdinc -I../.. -I../../lib -I../../lib/kernel -P ../../threads/ke
rnel.lds.S > threads/kernel.lds.s
gcc -m32 -c ../../threads/start.S -o threads/start.o -Wa,--gstabs -nostdinc -I..
/.. -I../../lib -I../../lib/kernel  -MMD -MF threads/start.d
clang: error: unsupported argument '--gstabs' to option 'Wa,'
make[1]: *** [threads/start.o] Error 1
make: *** [all] Error 2
```

- It failed to locate the compiler. We need to enforce the compiler selection.

Install PintOS (MacOS)

- Enforce the compiler selection. Change the compiler setting. Comment out the line 24-30 in src/Make.config as follows.

```
#ifneq (0, $(shell expr `uname -m` : '$(X86)'))
#  CC = $(CCPROG)
#  LD = ld
#  OBJCOPY = objcopy
#else
#  ifneq (0, $(shell expr `uname -m` : '$(X86_64)'))
#     CC = $(CCPROG) -m32
#     LD = ld -melf_i386
#     OBJCOPY = objcopy
#   else
    CC = i386-elf-gcc
    LD = i386-elf-ld
    OBJCOPY = i386-elf-objcopy
#   endif
#endif
```

Install PintOS (MacOS)

- Trouble shoot IV

In normal situation, PintOS should quit with '-q' option and shell needs to be ready to acce pt the next command. If the shell hangs, the PintOS failed to shutdown the system properly. There is a problem with the ACPI shutdown module. Please make the following update in src/de vices/shutdown.c.

```
printf ("Powering off...\n");
serial_flush ();

  //add the following line
  ++ outw( 0x604, 0x0 | 0x2000 );

  /* This is a special power-off sequence supported by Bochs and
    QEMU, but not by physical hardware. */
  for (p = s; *p != '\0'; p++)
    outb (0x8900, *p);
```

GCC version issue

- Downgrade the gcc version to 4.5 (recommended for pintos)

```
$ sudo apt-get install gcc-4.5
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/
bin/gcc-4.5 50
```