



시스템프로그래밍기초 실습 1주차

Orientation



시스템프로그래밍기초 실습

실습조교: **양인식 (3공 318호)**

연구실: 제3공학관 421호 사이버피지컬시스템연구실

연락처: 031-400-4748

이메일: inshik@hanyang.ac.kr

실습조교: **조윤수 (4공 412호)**

연구실: 제3공학관 421호 사이버피지컬시스템연구실

연락처: 031-400-4748

이메일: suejoe@hanyang.ac.kr

Virtual Box 다운로드

다운로드 링크: <https://www.virtualbox.org/wiki/Downloads>



VirtualBox

search...
Login Preferences

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 5.1 packages, see [VirtualBox 5.1 builds](#). Consider upgrading.

VirtualBox 5.2.18 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

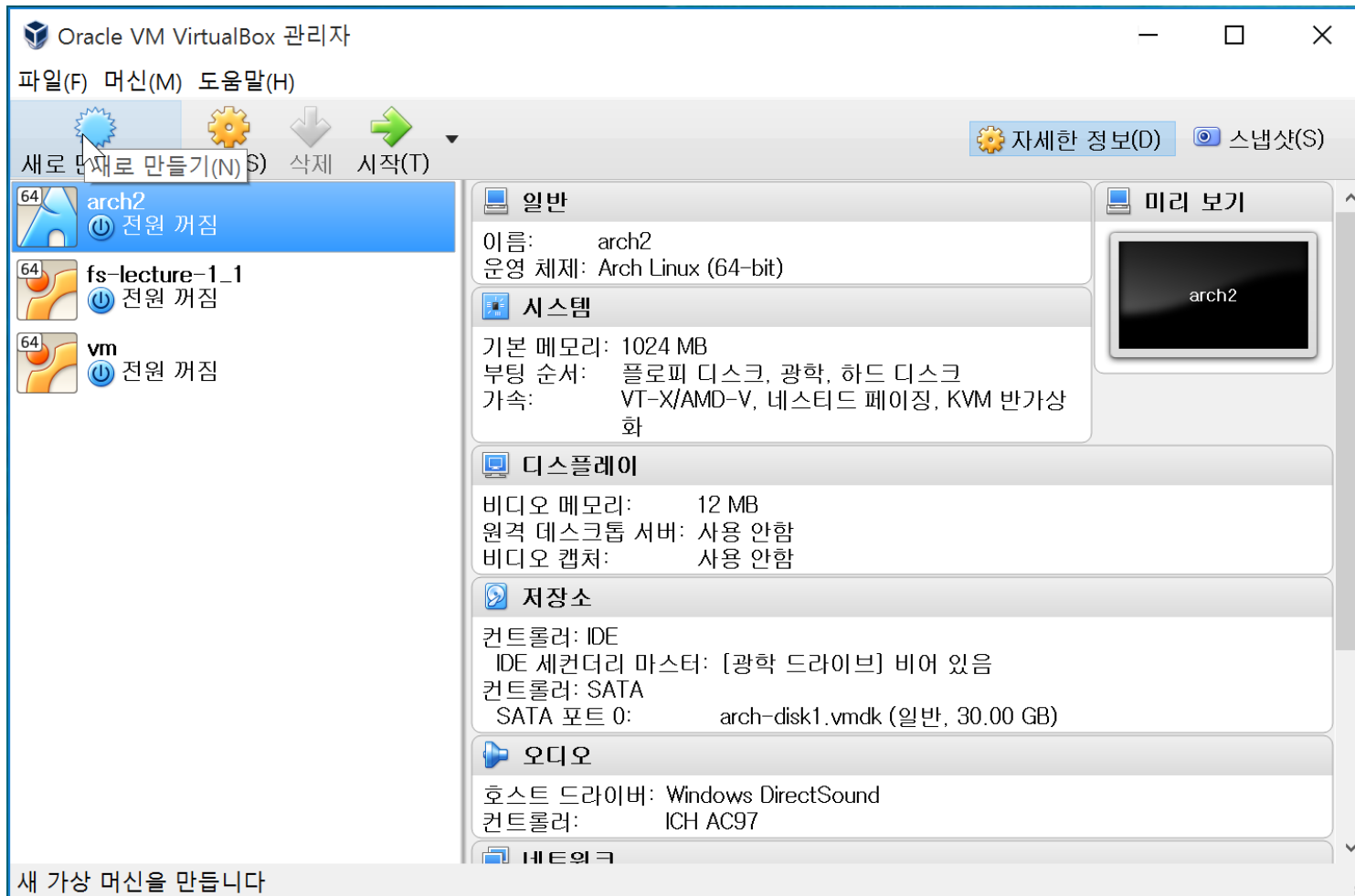
The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

[About](#)
[Screenshots](#)
[Downloads](#)
[Documentation](#)
 [End-user docs](#)
 [Technical docs](#)
[Contribute](#)
[Community](#)

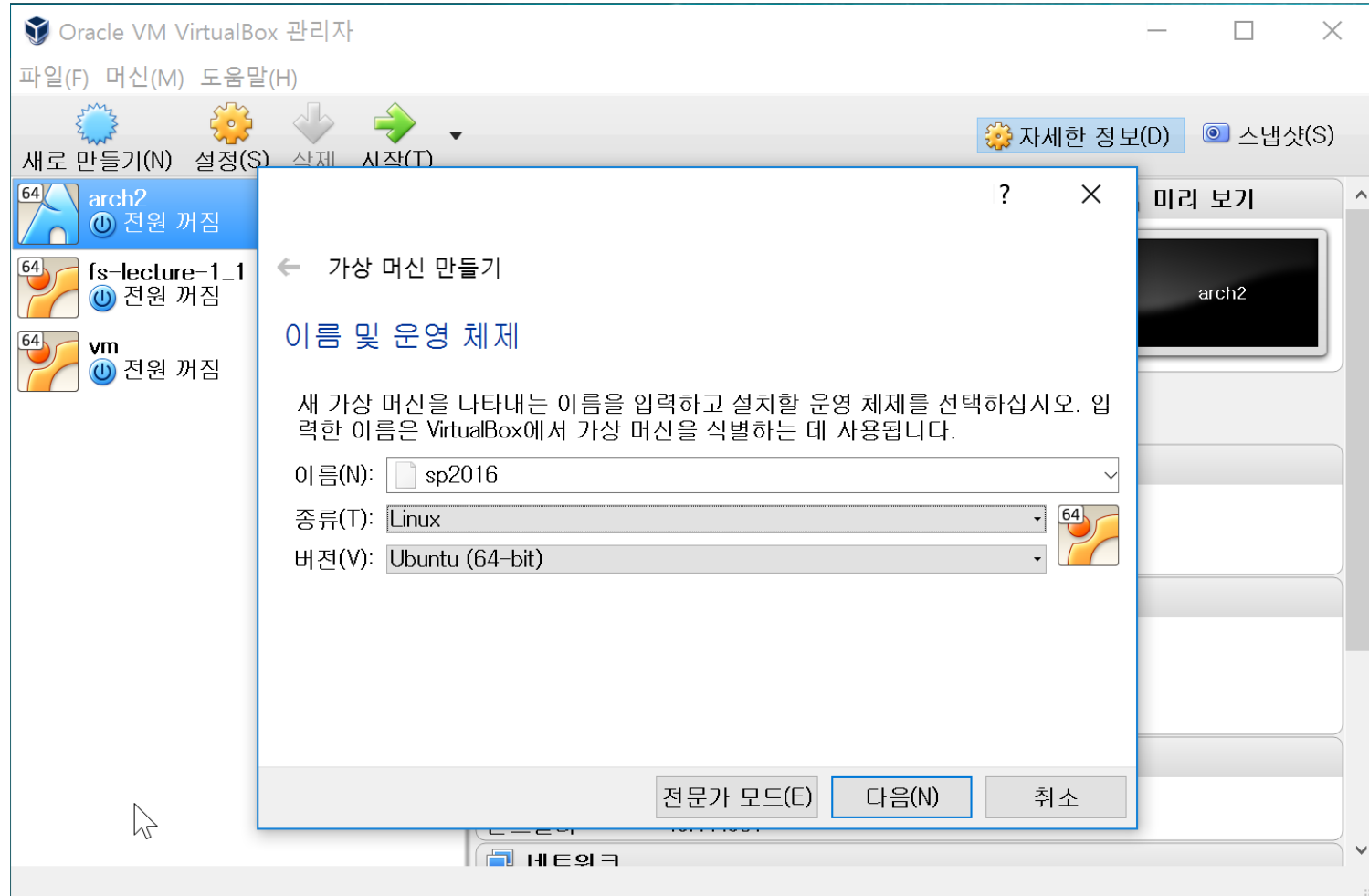
Linux 설치 (1)

- VirtualBox 실행 – 새로 만들기



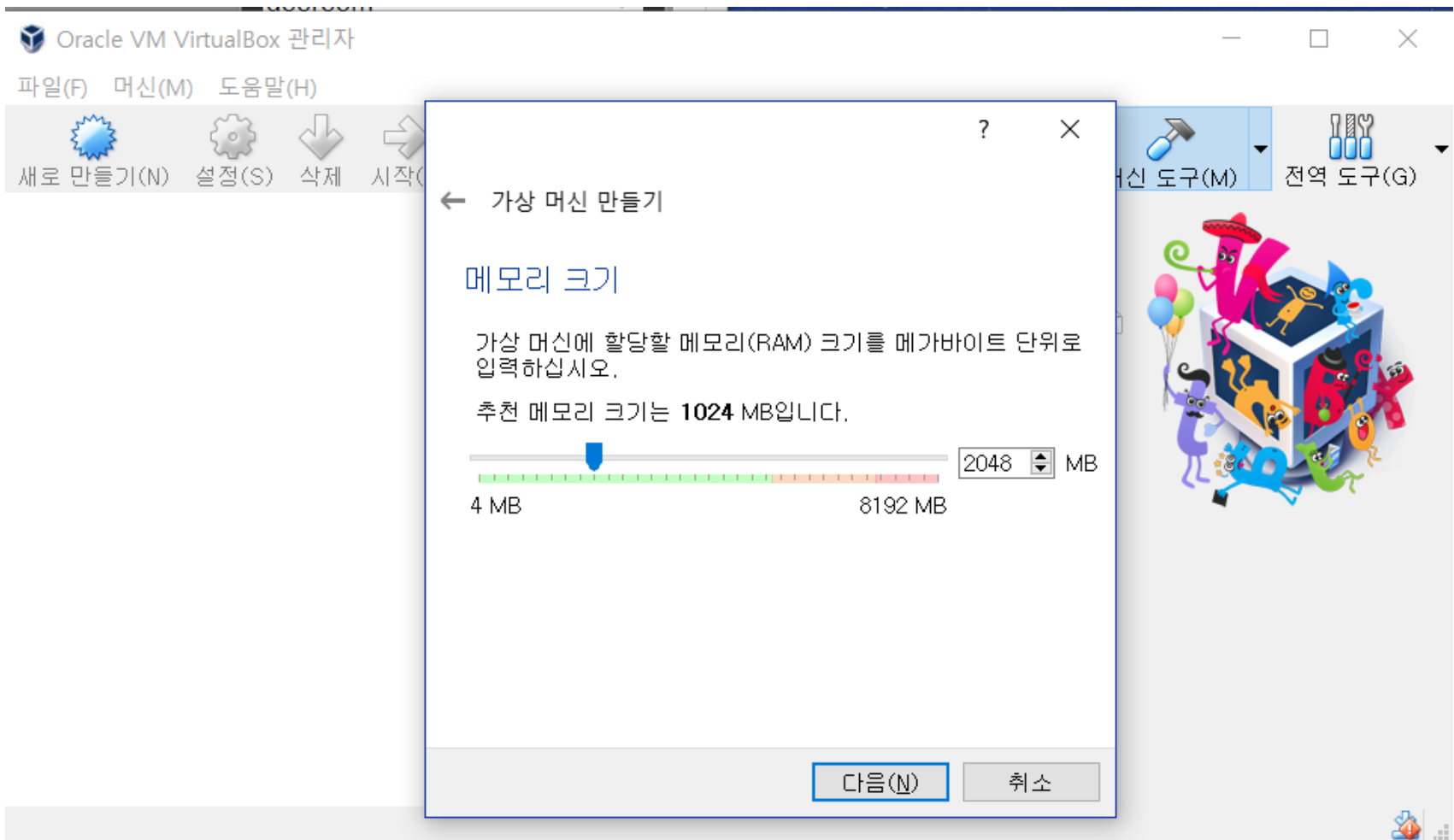
Linux 설치 (2)

- 가상머신 이름 Ubuntu – 종류 Linux – 버전 Ubuntu(64-bit)



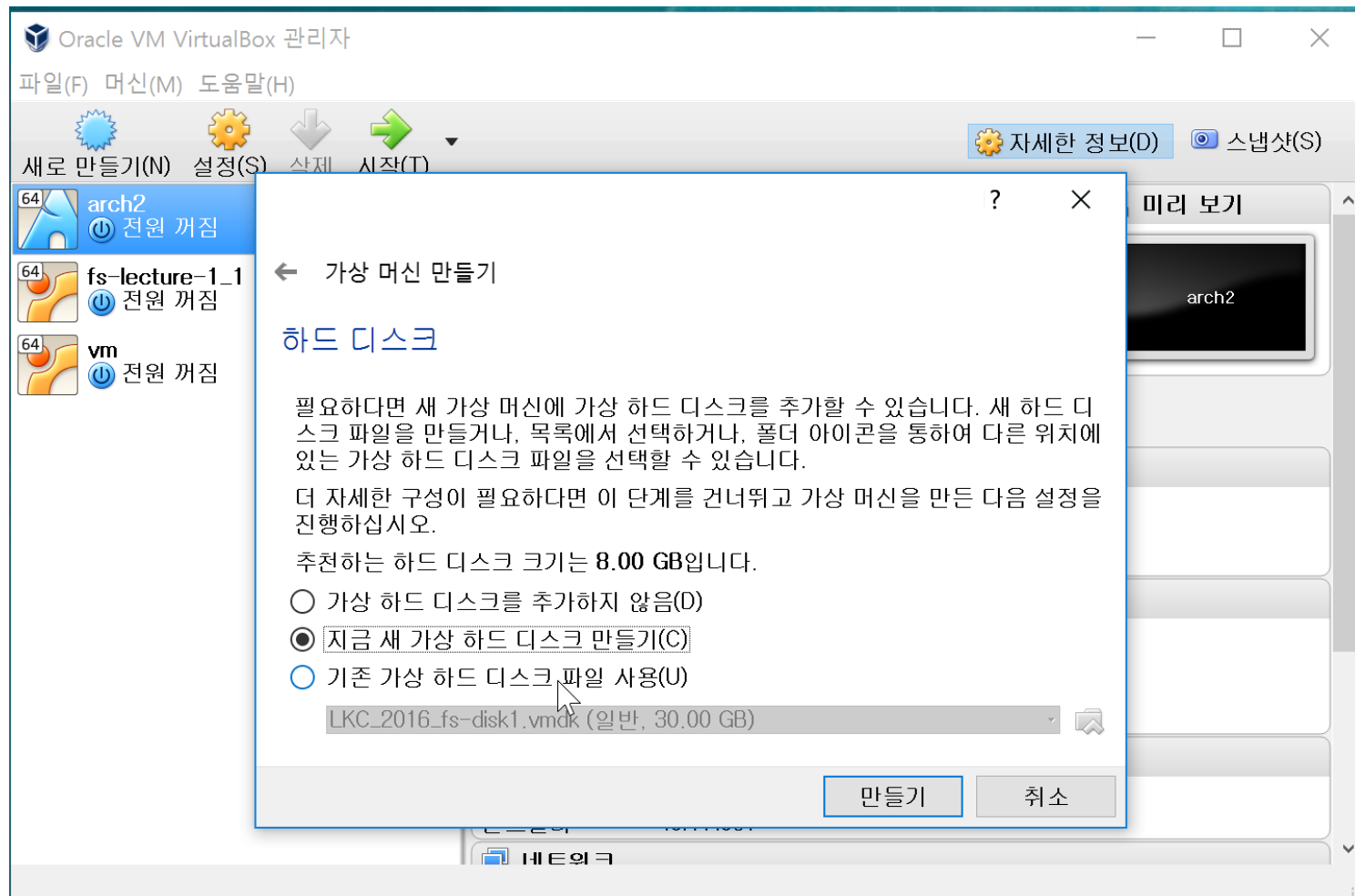
Linux 설치 (3)

- 메모리(RAM) 최소 **2048 MB**로 잡는 것이 좋음



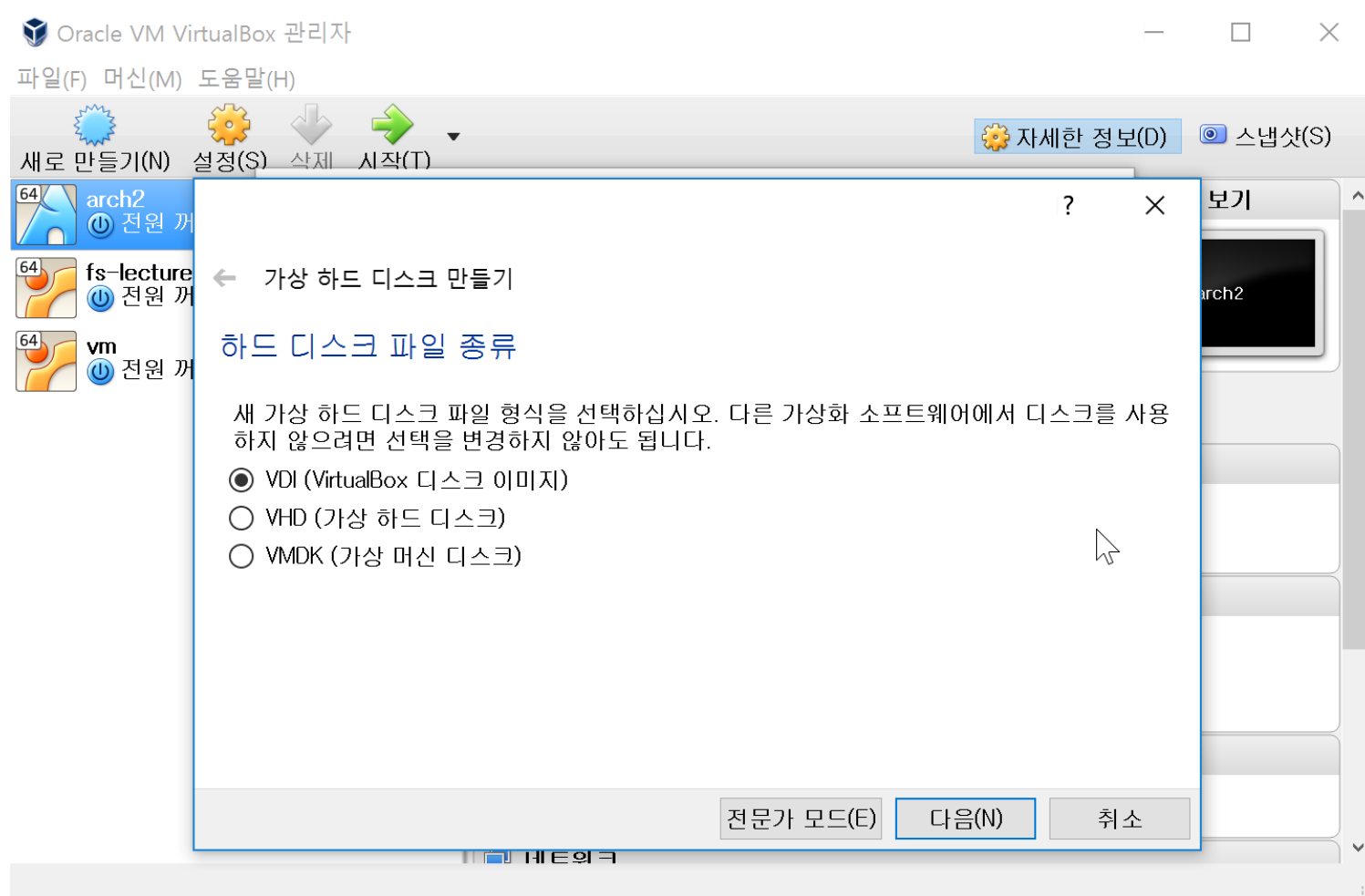
Linux 설치 (4)

- 하드 디스크: 지금 새 가상 하드 디스크 만들기(C)



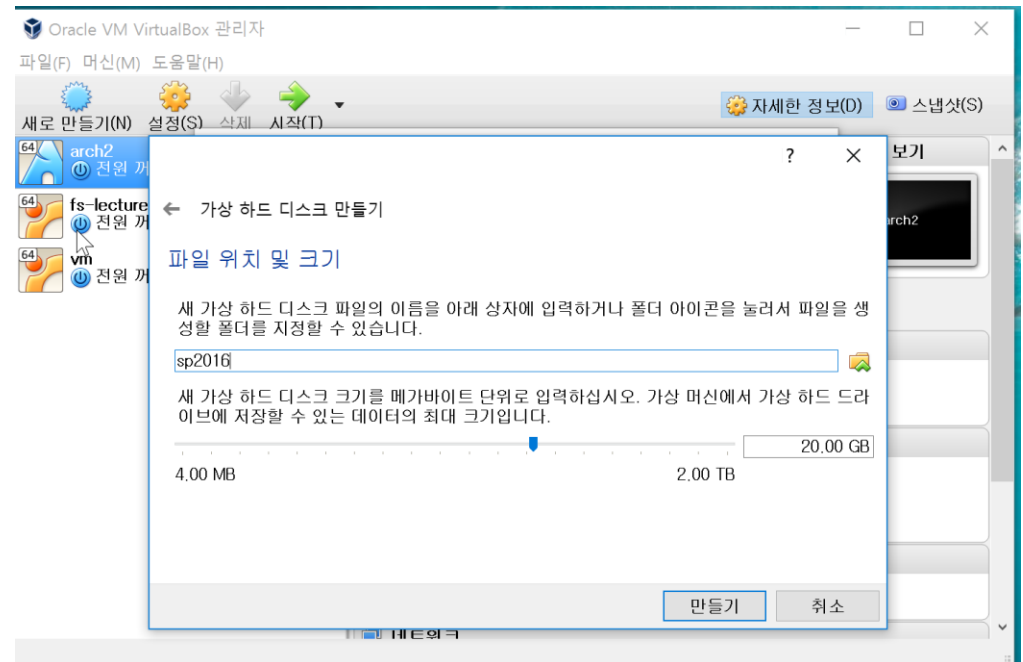
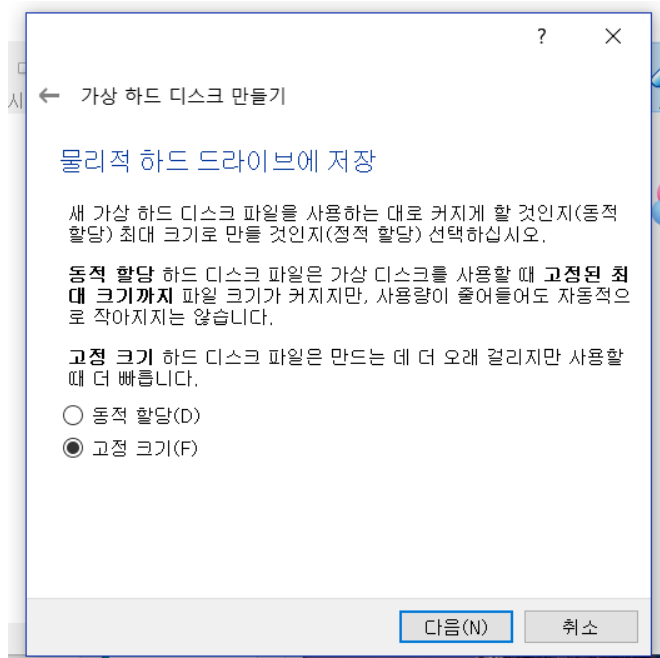
Linux 설치 (5)

- 하드 디스크: VDI (VirtualBox 디스크 이미지)



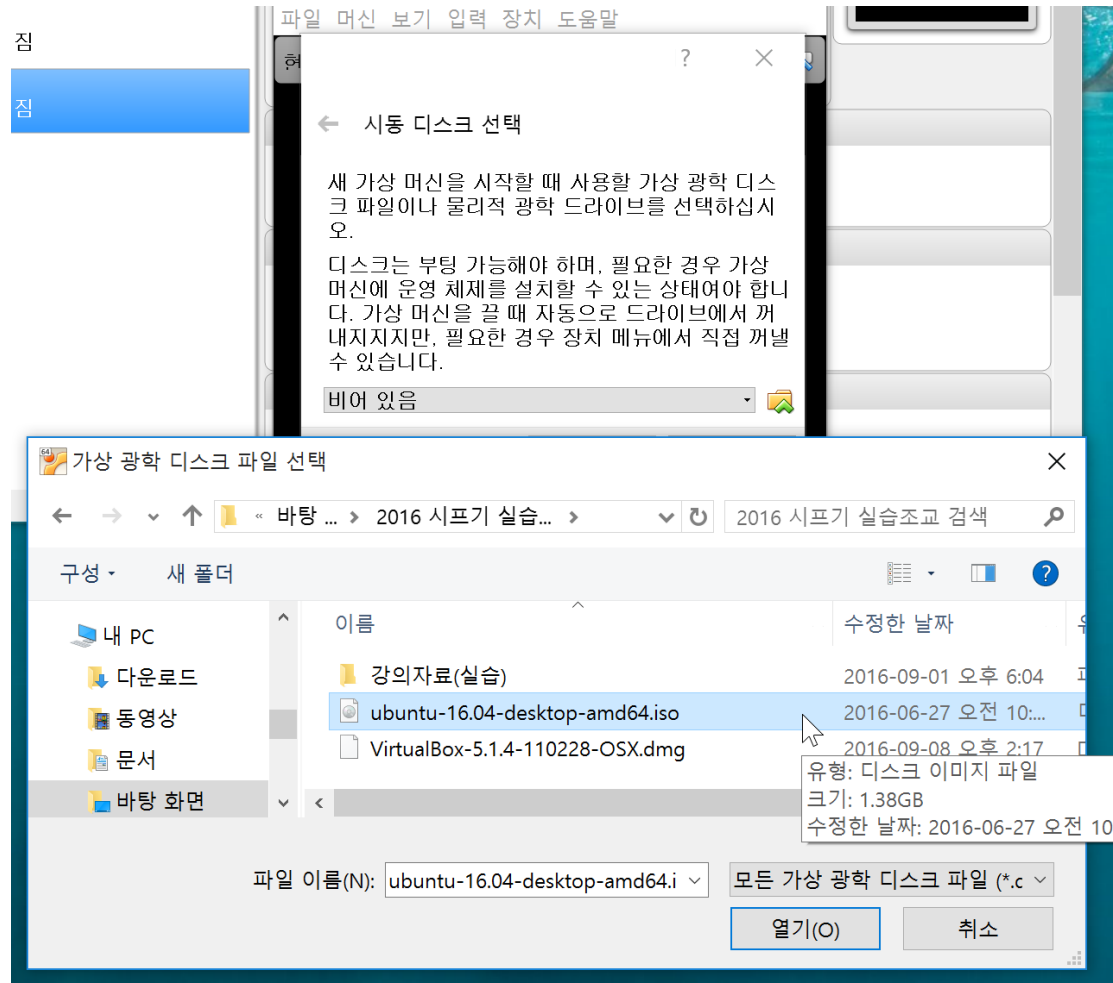
Linux 설치 (6)

- 하드 디스크: 고정크기, 디스크이름변경없음, 20GB



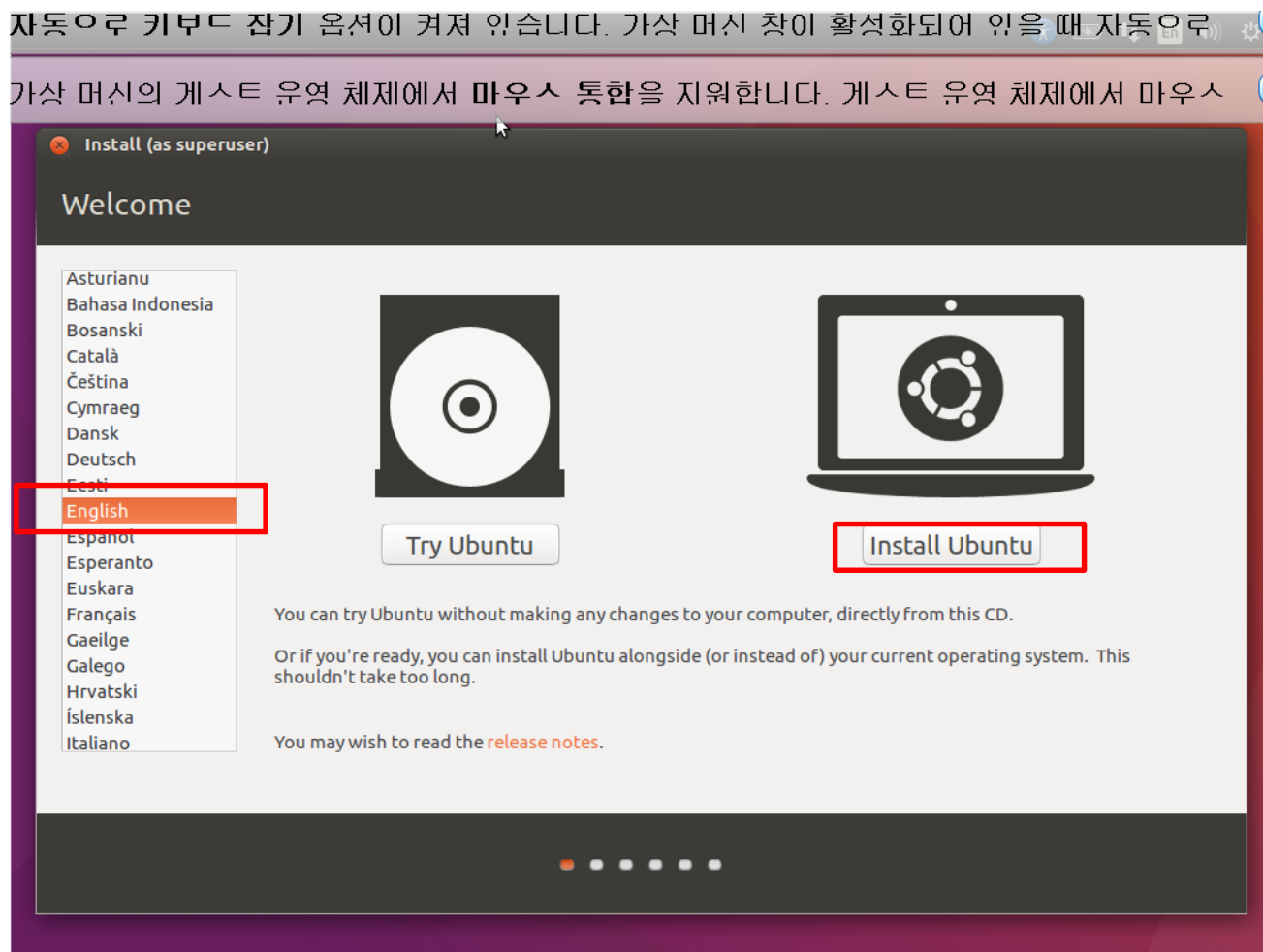
Linux 설치 (7)

- 설정 완료 후 시작 : 우분투 이미지 파일 가져오기



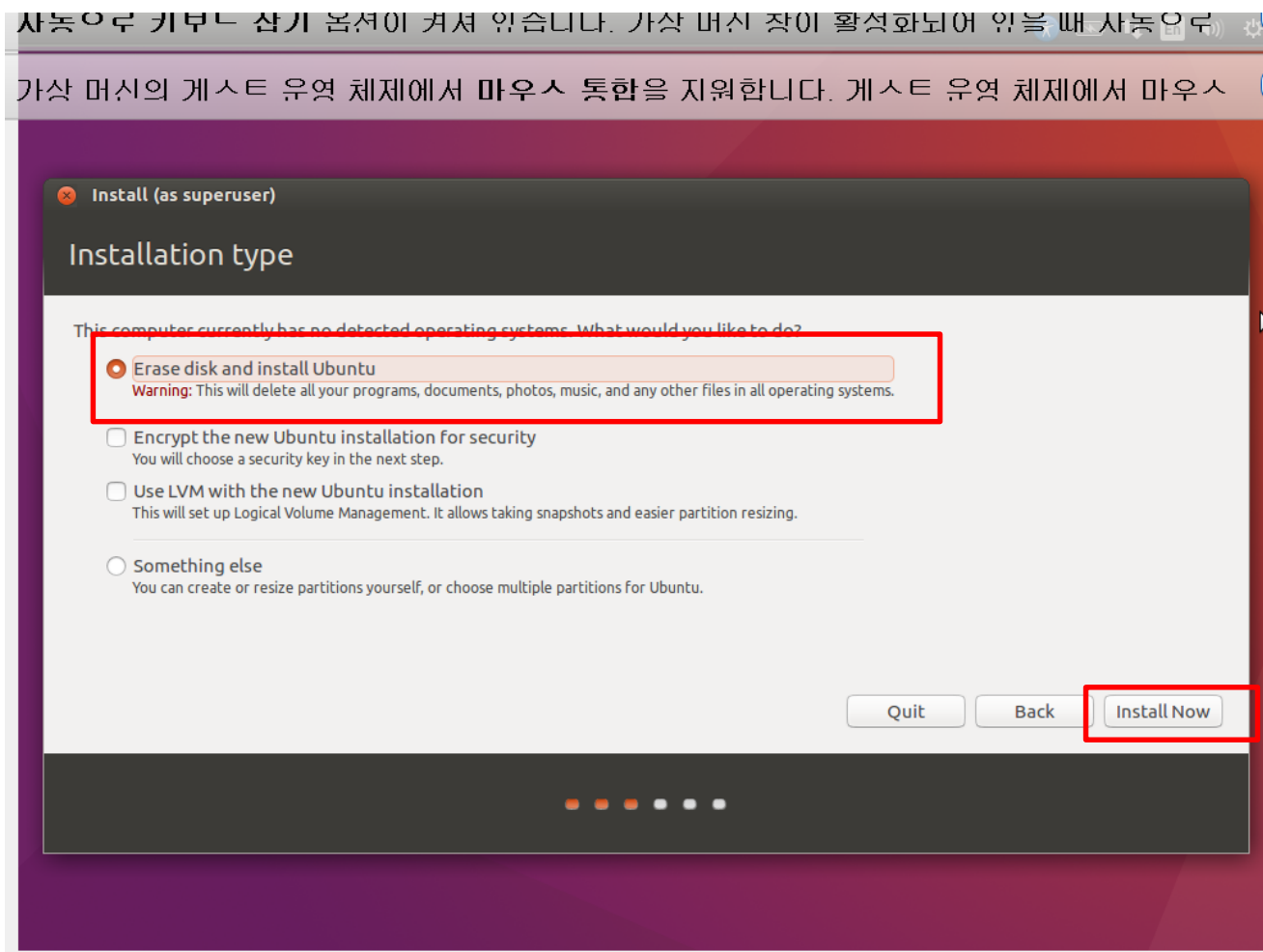
Linux 설치 (8)

- 우분투 설치 : English, Install Ubuntu



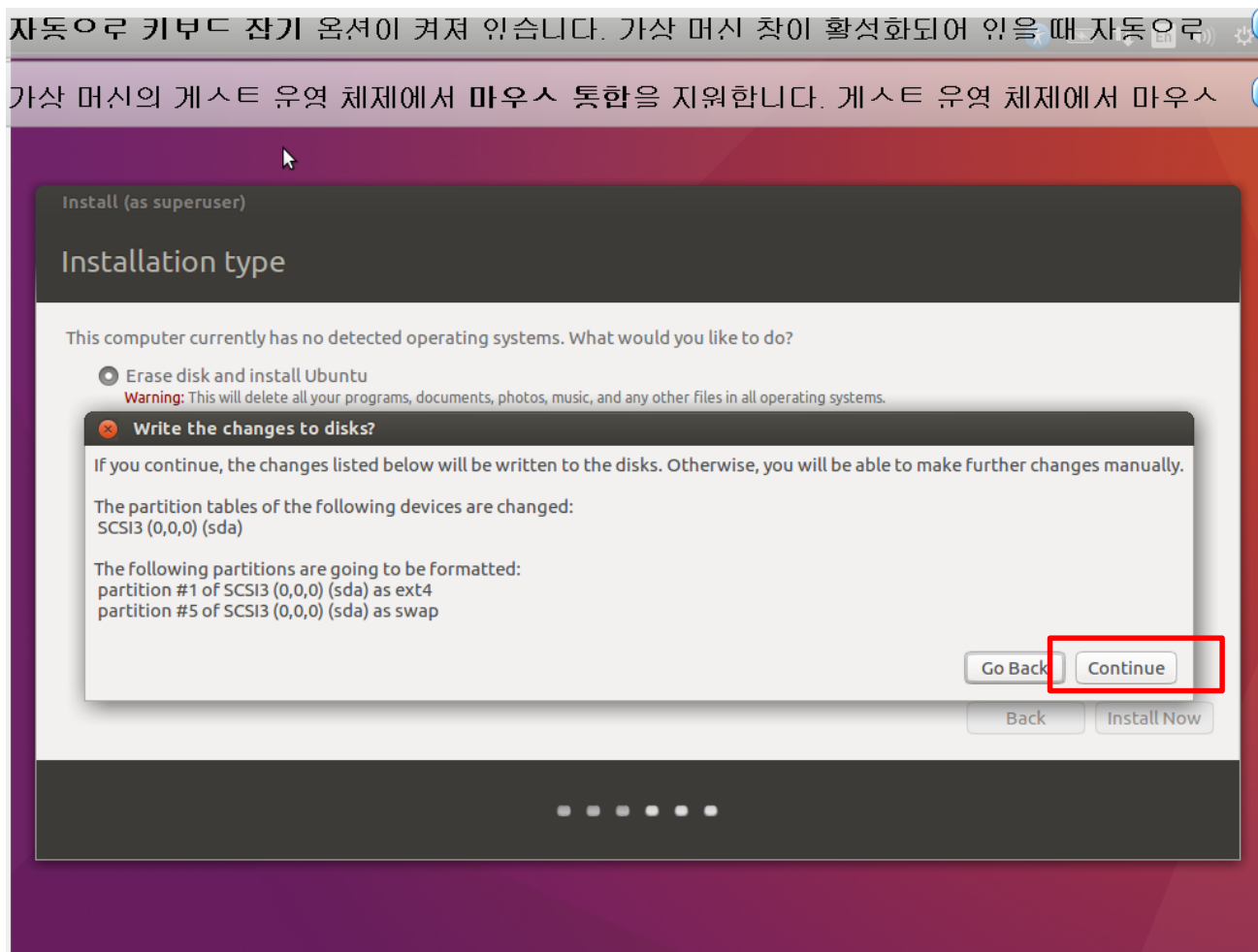
Linux 설치 (9)

- 새로운 디스크 생성 : Erase disk and install Ubuntu



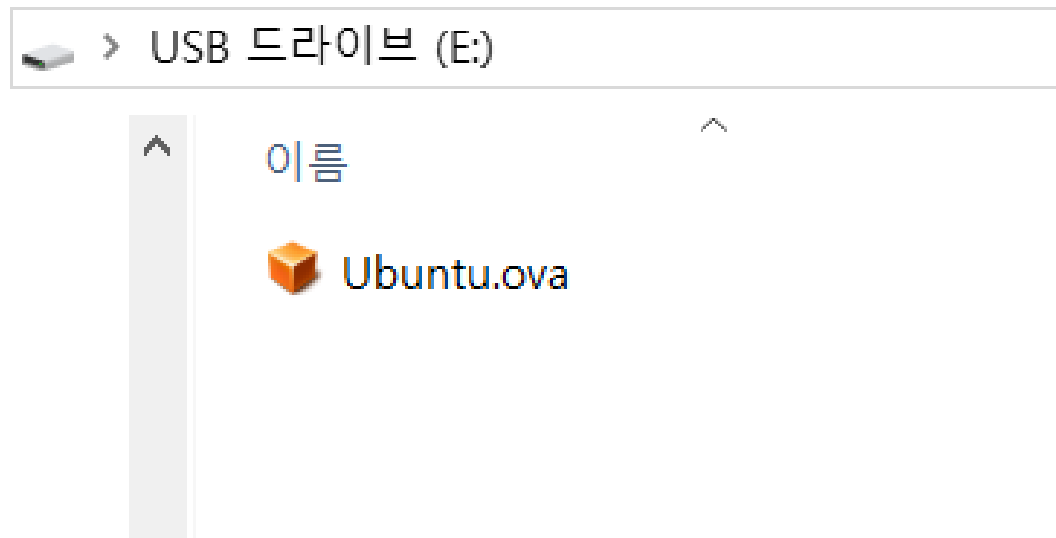
Linux 설치 (10)

- 새로운 디스크 생성 : Continue



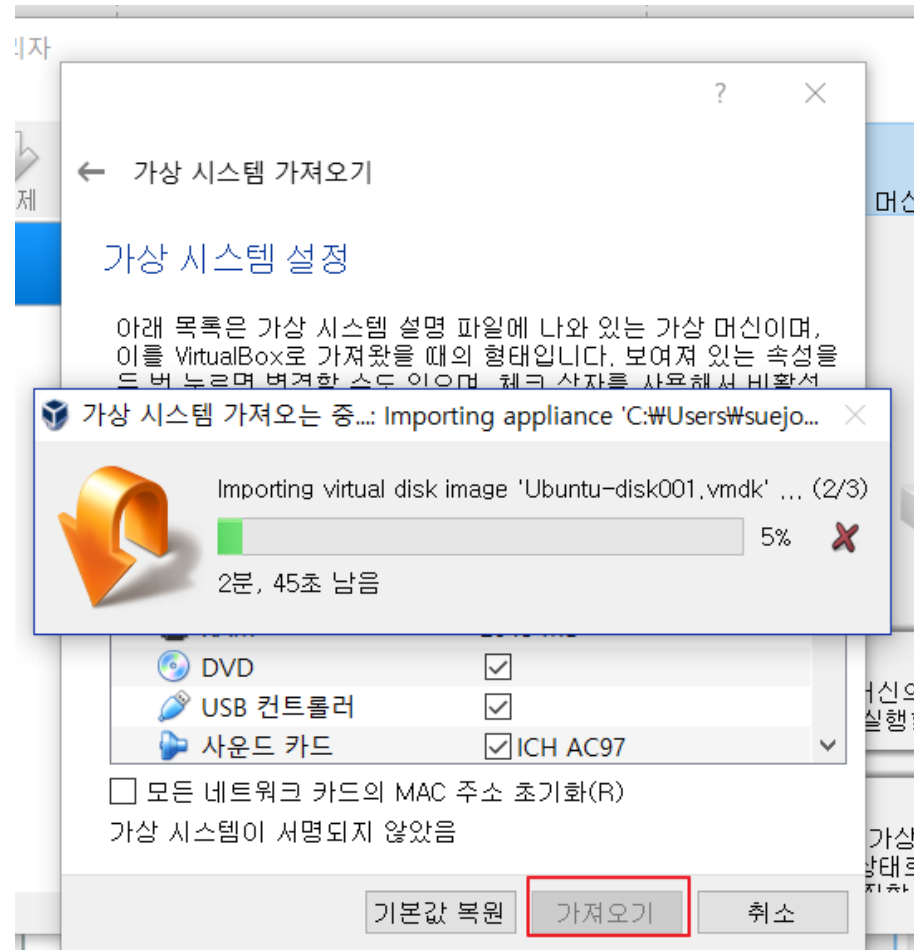
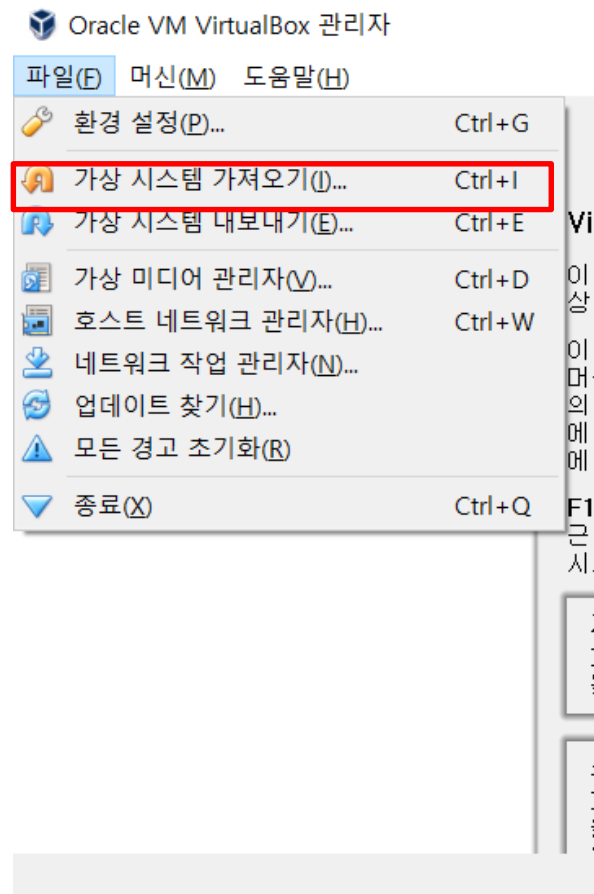
빠른 사용을 위해 설치된 리눅스 가져오기 (1)

- 아래 구글 드라이브 링크에서 Ubuntu.ova 다운로드
- https://drive.google.com/drive/folders/1uex8boYRNZ80fW_SjXzqa_j_HghIxCW9B?usp=sharing



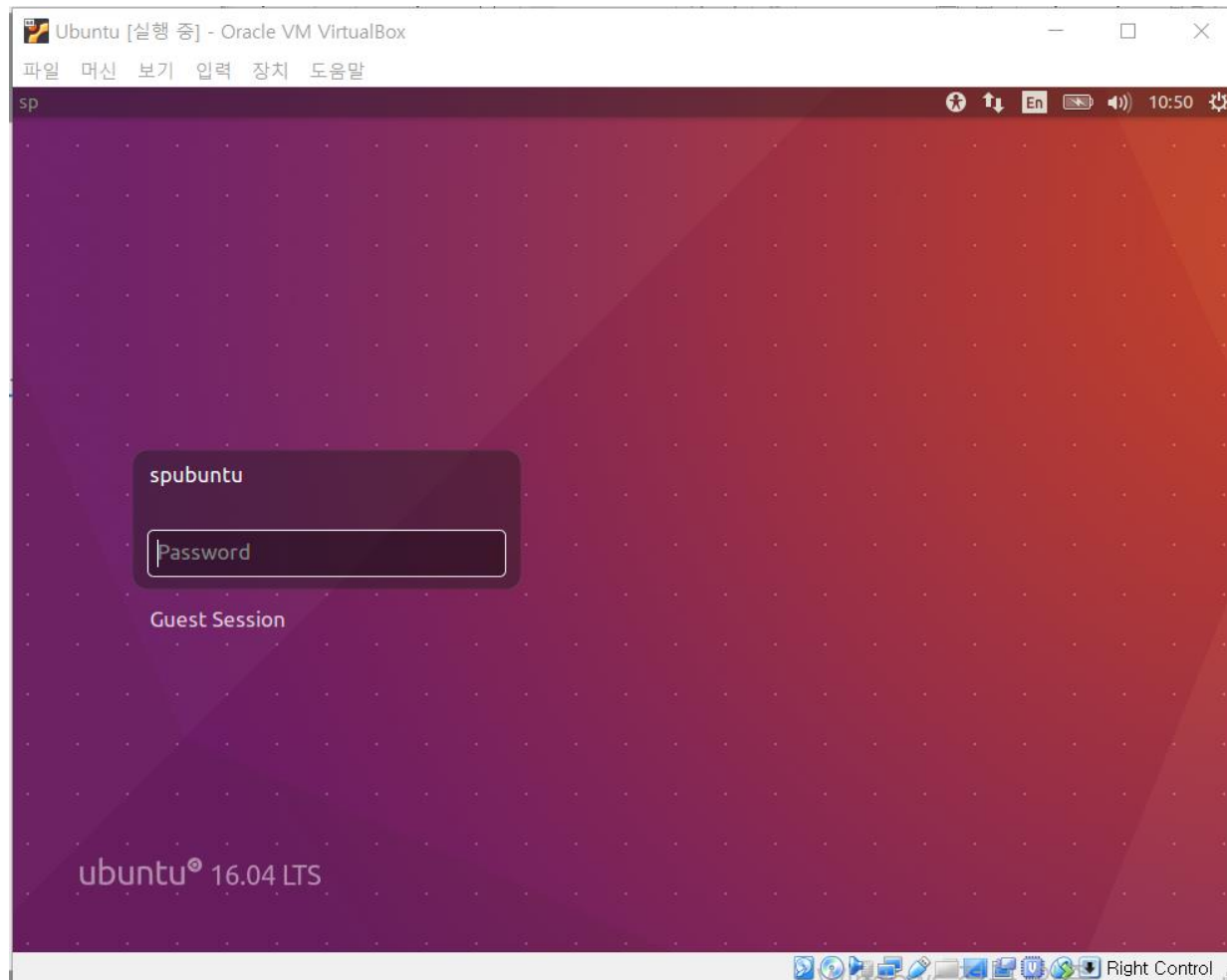
빠른 사용을 위해 설치된 리눅스 가져오기 (2)

- 복사한 가상 시스템(Ubuntu.ova)을 불러온다.
- VirtuaBox 실행 - 파일 - 가상 시스템 가져오기 - 가져오기



우분투 계정 및 비밀번호

- 계정 : spubuntu
- 비밀번호 : 1234





터미널 사용법 (1)

- 터미널이란?
 - 입출력장치 (혹은 사용자)와 컴퓨터 간의 소통을 가능하게 해주는 인터페이스
- 우분투 어플리케이션에서 터미널 찾기
- 단축키 :
 - Ctrl + Alt + T

터미널 사용법 (2)

설명	명령어
현재 디렉토리	\$ pwd
디렉토리 내용 보기	\$ ls
디렉토리 생성	\$ mkdir dir_name
디렉토리로 이동하기	\$ cd directory_name
삭제하기	\$ rm
디렉토리 삭제	\$ rm -r
파일 내용 보기	\$ cat file_name
명령어 정보 보기	\$ man instruction_name

터미널 명령어 실습

```
cprog@cprog-VirtualBox: ~/Documents/cprog
cprog@cprog-VirtualBox:~$ pwd
/home/cprog
cprog@cprog-VirtualBox:~$ ls
Desktop      Downloads      Music          Public          Videos
Documents    examples.desktop Pictures        Templates
cprog@cprog-VirtualBox:~$ cd Documents/
cprog@cprog-VirtualBox:~/Documents$ ls
cprog@cprog-VirtualBox:~/Documents$ mkdir cprog
cprog@cprog-VirtualBox:~/Documents$ mkdir temp
cprog@cprog-VirtualBox:~/Documents$ cd temp
cprog@cprog-VirtualBox:~/Documents/temp$ mkdir temp
cprog@cprog-VirtualBox:~/Documents/temp$ cd ..
cprog@cprog-VirtualBox:~/Documents$ rm temp
rm: cannot remove 'temp': Is a directory
cprog@cprog-VirtualBox:~/Documents$ rm -r temp
cprog@cprog-VirtualBox:~/Documents$ man rm
cprog@cprog-VirtualBox:~/Documents$ cd cprog
```

vi 사용법 (1)

- Vi란?
 - 유닉스 환경에서 많이 쓰이는 텍스트 기반의 편집기
- Command mode (ESC 눌린 후)

설명	명령어
커서 이동	h, j, k, l
블록 지정	v
복사	y
붙여놓기	p
저장	: w
종료	: q
저장 후 종료	: wq
저장하지 않고 종료	: q!

vi 사용법 (2)

- Insert mode

설명	명령어
현재 커서 위치에 글자 추가	i
현재 줄 처음에 글자 추가	I
현재 커서 다음 위치에 글자 추가	a
현재 줄 마지막에 글자 추가	A
아랫줄에 글자 추가	o
윗줄에 글자 추가	O
Command mode로 전환	ESC

Vim 설정

- Vim이란?
 - 우분투에서 사용되는 vi 에디터에 설정을 더하여 보다 편리하게 사용할 수 있도록 해주는 에디터.

- Vim 설치

```
spubuntu@sp:~$ sudo apt-get update
```

```
spubuntu@sp:~$ sudo apt-get install vim
```

- Vim 설정

```
spubuntu@sp:~$ vi .vimrc
```

```
spubuntu@sp: ~
```

```
1 set nu  
2 set autoindent  
3 set ts=4  
4 set cindent  
5 set shiftwidth=4  
6 set smartindent
```

```
~
```

- 설정값 바로 적용

```
spubuntu@sp:~$ source ~/.vimrc
```

실습 예제 1) 터미널에 hello world 출력

```
cprog@cprog-VirtualBox:~/Documents/cprog$ vim hello.c
```

```
#include <stdio.h>
int main(void){
    printf("hello world\n");
    return 0;
}
```

- printf() 함수가 정의되어 있는 stdio.h 라이브러리
 - String \approx char*

C언어에는 string이라는 data type이 없음.

<code>int printf(const char* format, ...);</code>	(1)
<code>int fprintf(std::FILE* stream, const char* format, ...);</code>	(2)
<code>int sprintf(char* buffer, const char* format, ...);</code>	(3)
<code>int snprintf(char* buffer, std::size_t buf_size, const char* format, ...);</code>	(4) (since C++11)

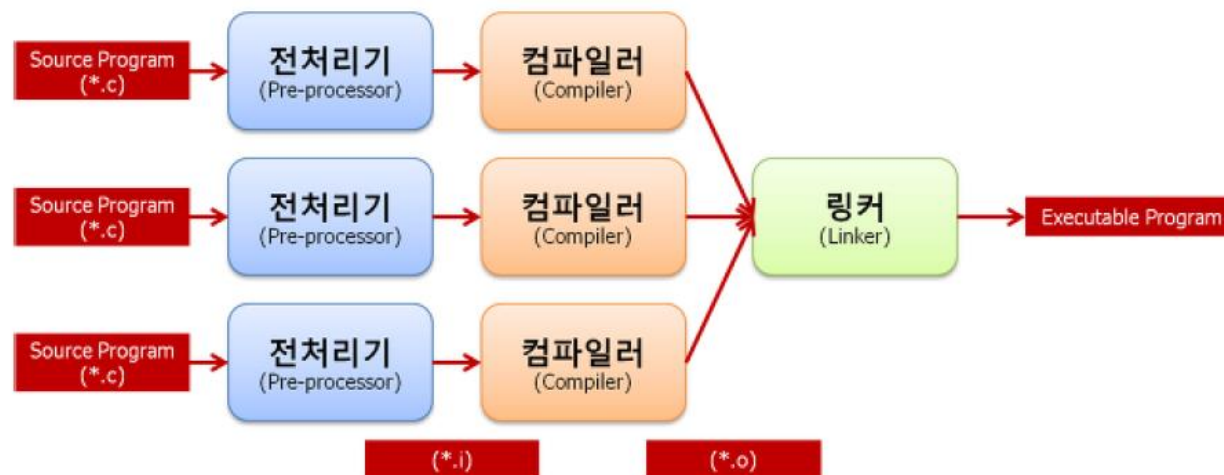


gcc 사용법

```
cprog@cprog-VirtualBox:~/Documents/cprog$ vim hello.c
cprog@cprog-VirtualBox:~/Documents/cprog$ gcc hello.c
cprog@cprog-VirtualBox:~/Documents/cprog$ ls
a.out  hello.c
cprog@cprog-VirtualBox:~/Documents/cprog$ ./a.out
hello world
cprog@cprog-VirtualBox:~/Documents/cprog$ gcc -o hello hello.c
cprog@cprog-VirtualBox:~/Documents/cprog$ ls
a.out  hello  hello.c
cprog@cprog-VirtualBox:~/Documents/cprog$ ./hello
hello world
```


gcc compile 과정

- Pre-processing :
#include, #define, #if 와 같은 지시자들을 처리하는 과정
- Compile :
어셈블리 코드로 변환하는 과정
- Assemble :
바이너리 파일로 변환하는 과정
- Linking



gcc 사용법

- gcc 매뉴얼 확인 `$ man gcc`

- 자주 사용되는 gcc option

GCC options

```
$man gcc
```

- -o : output 으로 나올 파일 이름을 설정
- -W : 경고 메시지 설정, -Wall 을 많이 사용함
- -O : 컴파일러 최적화 -O, -O2, -O3. 숫자가 높을수록 최적화 레벨이 높다
- -g : 디버거(GDB)를 위한 정보를 삽입함
- -E : 전 처리과정 화면에 출력
- -S : C code를 기반으로 Assembly 생성 (.c -> .s)
- -c : object file 생성 (.s -> .o)

GCC example

```
$gcc -o [binary] [anyfile.c]
```

실습 과제 1) sea.c

- 아래 코드를 작성해 compile해 실행하여라.

```
#include <stdio.h>

int main(void) {
    printf("from sea to ");
    printf("shining C");
    printf("\n");
    return 0;
}
```



감사합니다.
