

Introduction of **Linux**

韦阳

| oslab2018_class1@163.com

黄子寅

| oslab2018_class2@163.com

PART I

- Brief Introduction
- Basic Conceptions & Environment
- Install & Configure a Virtual Machine
- Basic Commands

PART II

- Shell Script
- Compile & Debug (for C)
- Text Editor (Vim, Sublime text, Atom)

PART I

- Brief Introduction
- Basic Conceptions & Environment
- Install & Configure a Virtual Machine
- Basic Commands

History

- 1969 - UNIX
- 1984 - GNU
- 1987 - MINIX
- 1995 - POSIX
- Internet

Distribution

- Ubuntu
- Debian
- CentOS
- Arch Linux
- Fedora
- ...

Features

- Protable
- Open source
- Security
- Shell
- ...

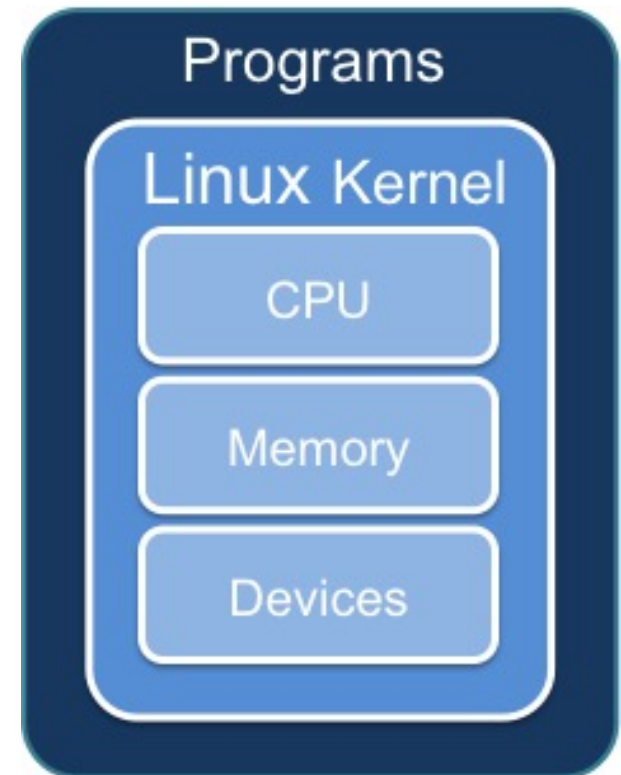
PART I

- Brief Introduction
- **Basic Conceptions & Environment**
- Install & Configure a Virtual Machine
- Basic Commands

Linux Kernel

The most important component of Linux OS, containing all the operating system's **core functions** and the **device drivers**.

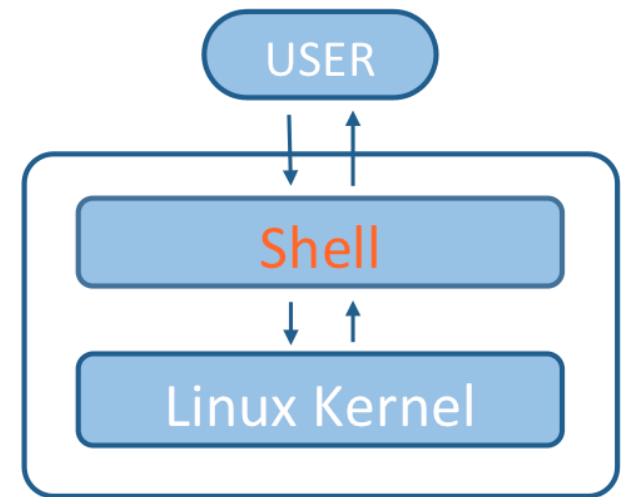
- memory management
- process scheduling
- file system
- ...



Shell (CLI shell)

Command Line Interface

A **program** which accepts commands as text input and **converts commands** to appropriate operating system functions.

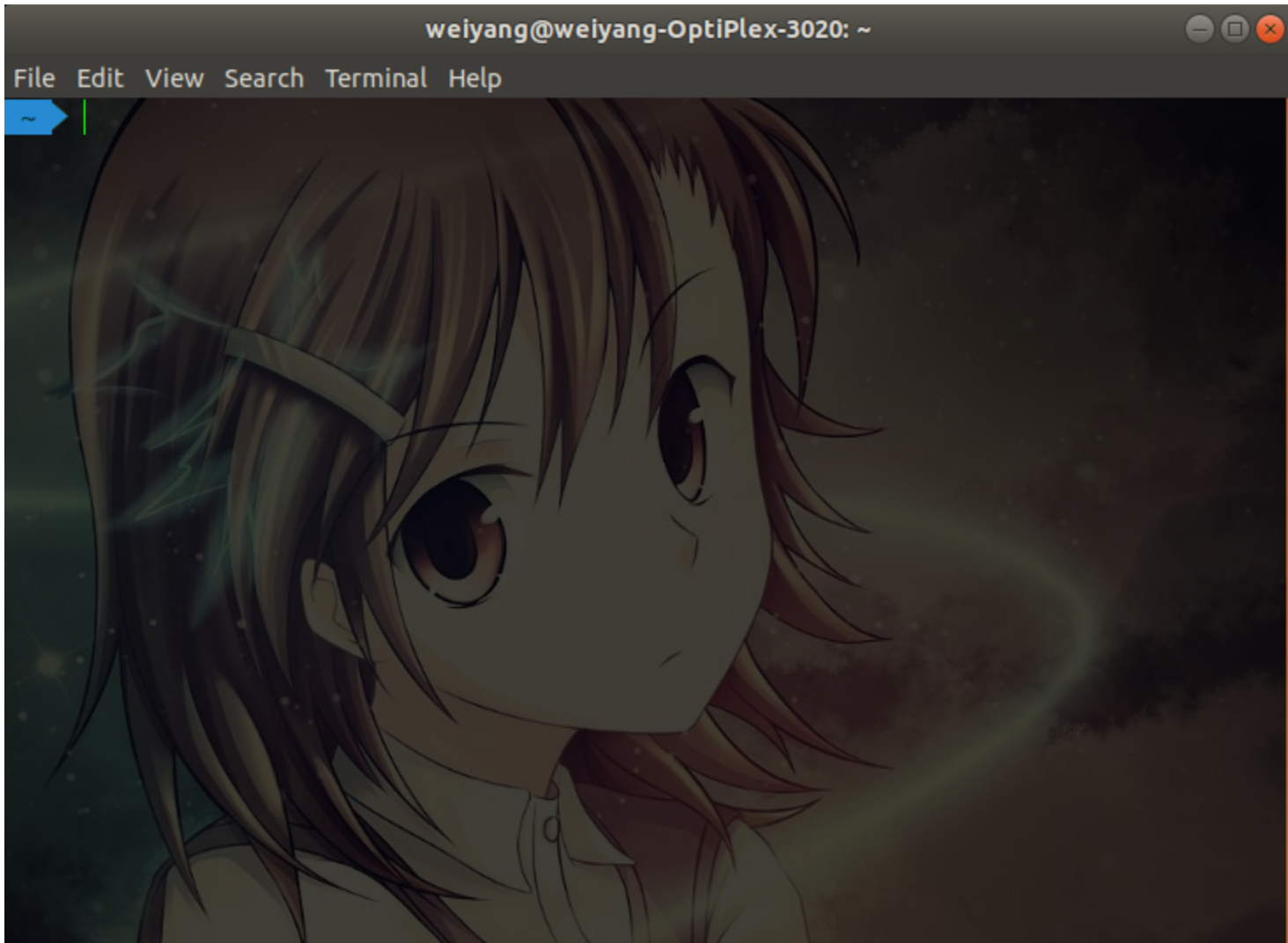


Terminal ↔ Shell

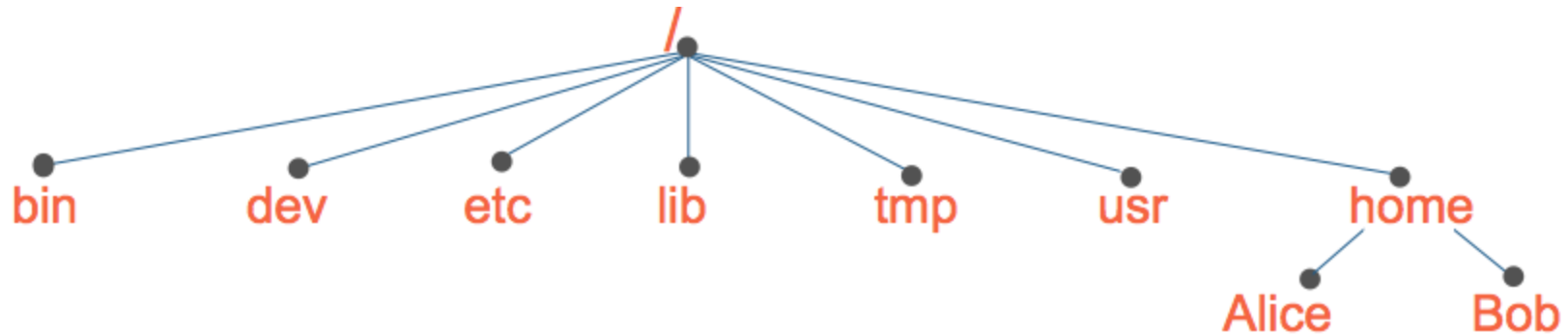
The terminal send information to the shell, receive and display the information from the shell.

Open Terminal

keyboard accelerators: `CTRL+ALT+T`



File System



Tree structure, with the root directory “ / ”

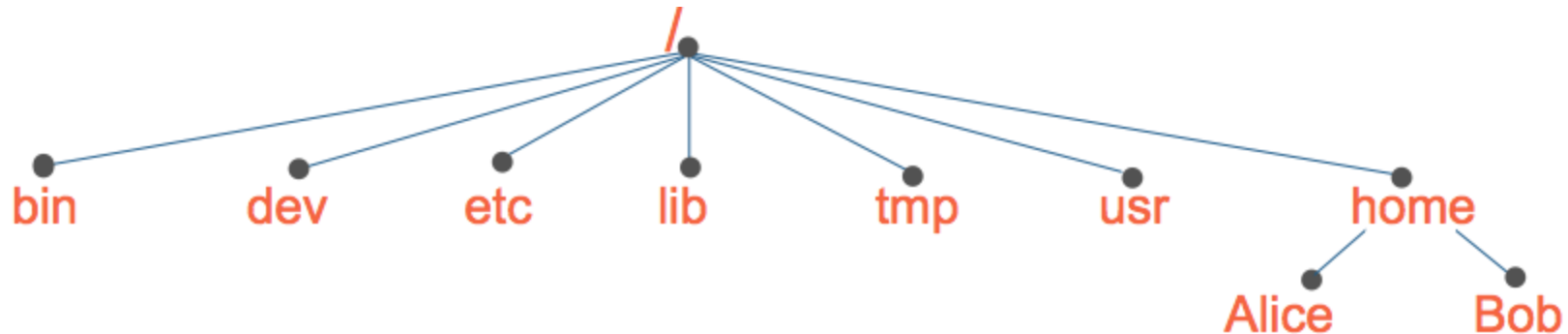
```
/home/godweiyang/...
```

```
~ = /home/godweiyang
```

```
.
```

```
..
```

File System



`/bin` : essential tools and other programs

`/dev` : files representing the system's hardware devices

`/etc` : system configuration files

`/home` : the home directory for all system's users

`/lib` : essential system library files

`/proc` : files that give information about current system

`/usr` : files related to user tools and applications

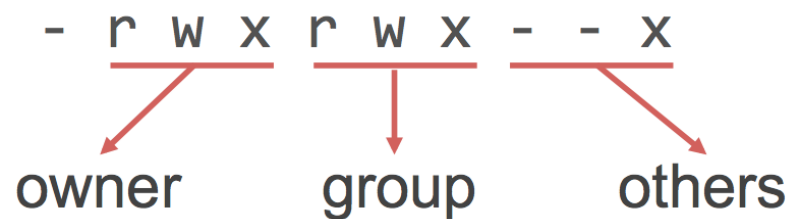
User & Group

The system determines whether or not a **user** or **group** can access a file or directory.

There is a special user called **Super User** or the **root** which has permission to access any file and directory.

Three **Permissions**:

- **r** = read
- **w** = write
- **x** = execute



Environment Variables

Environment variables are a **set of values** that can affect the way running processes will behave on a computer.

- `PATH` -- Contains a colon-separated list of directories that the shell searches for commands that do not contain a slash in their name.
- `HOME` -- Contains the location of the user's home directory.
- ...

Set The Environment Variables:

```
export VARIABLE = value      # temporary
/etc/profile                 # permanent, all users

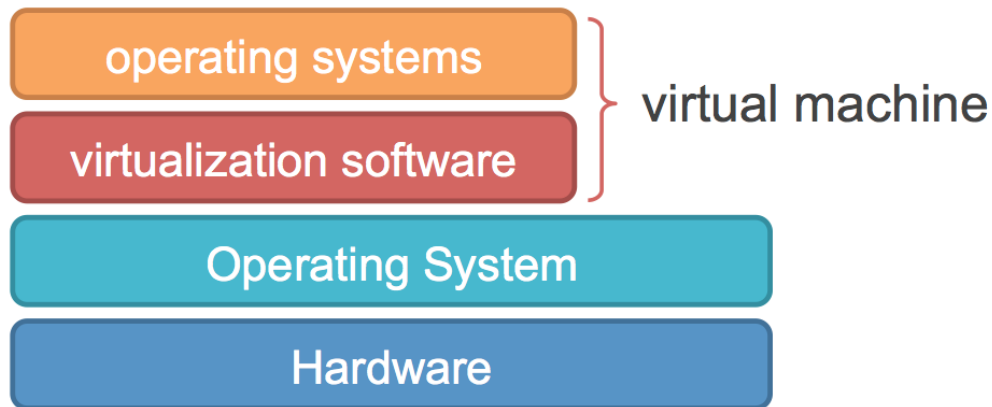
~/.profile                   # permanent, one user
~/.bashrc
```

PART I

- Brief Introduction
- Basic Conceptions & Environment
- **Install & Configure a Virtual Machine**
- Basic Commands

Virtual Machine

A virtual machine is an emulation of a particular computer system.



Virtualization Software provide (hardware) resources virtually to the new OS.

- VMware
- Virtual Box
- Virtual PC

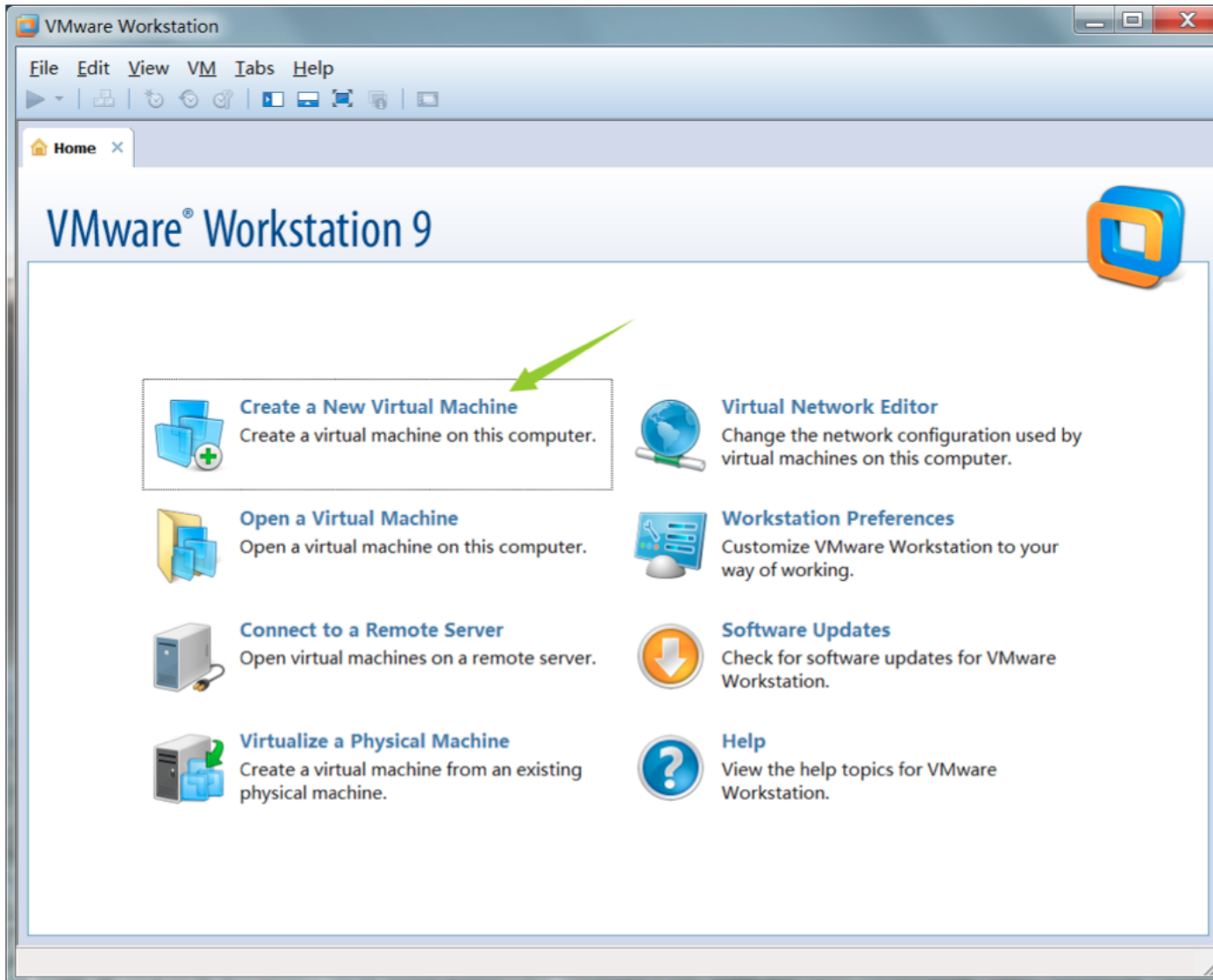
Install the Virtual Machine

VMware Workstation 9.0 + Ubuntu 14.04 LTS (kernel 3.19)

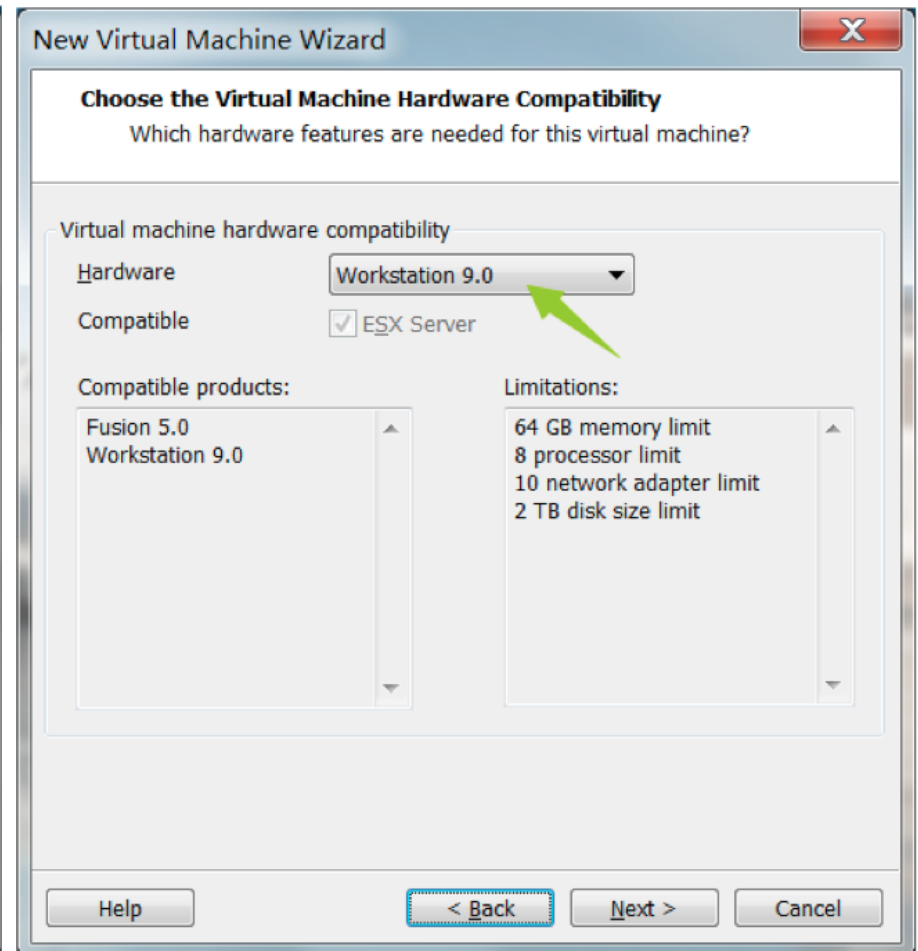
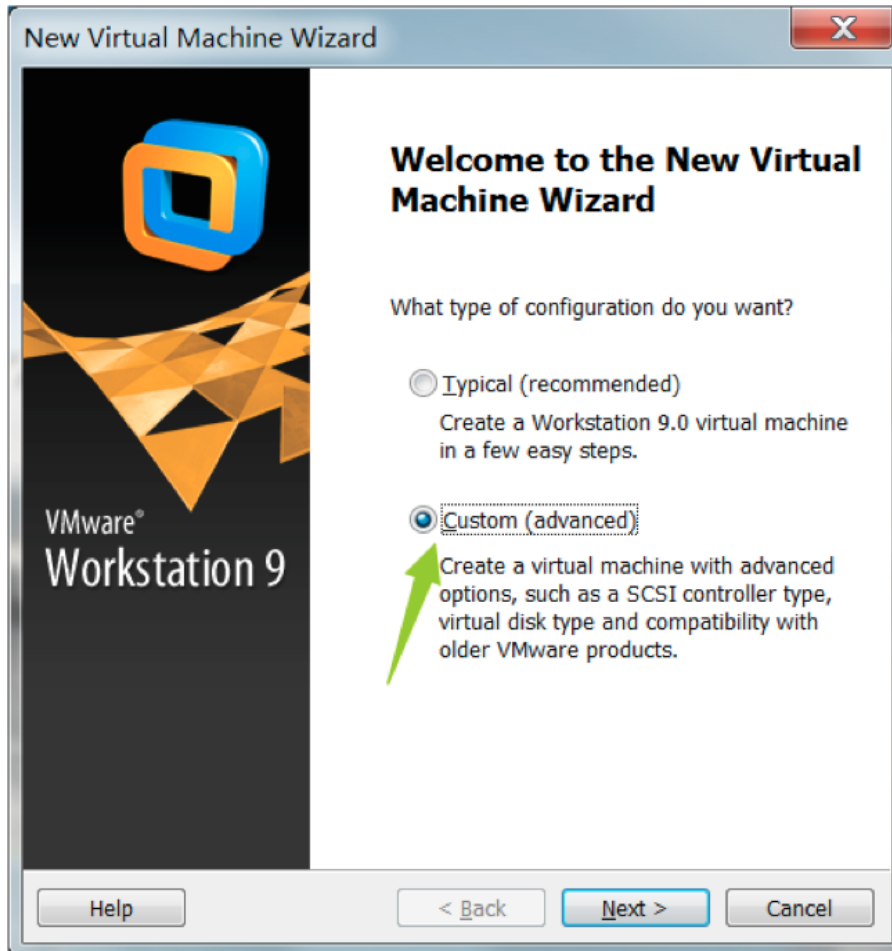


1. Download the Setup File of Vmware 9.0
2. Download the Ubuntu Ubuntu 14.04 LTS from the official website www.ubuntu.com/download/desktop
3. Install VMware 9.0
4. Create a Virtual Machine in the VMware

Create a Virtual Machine

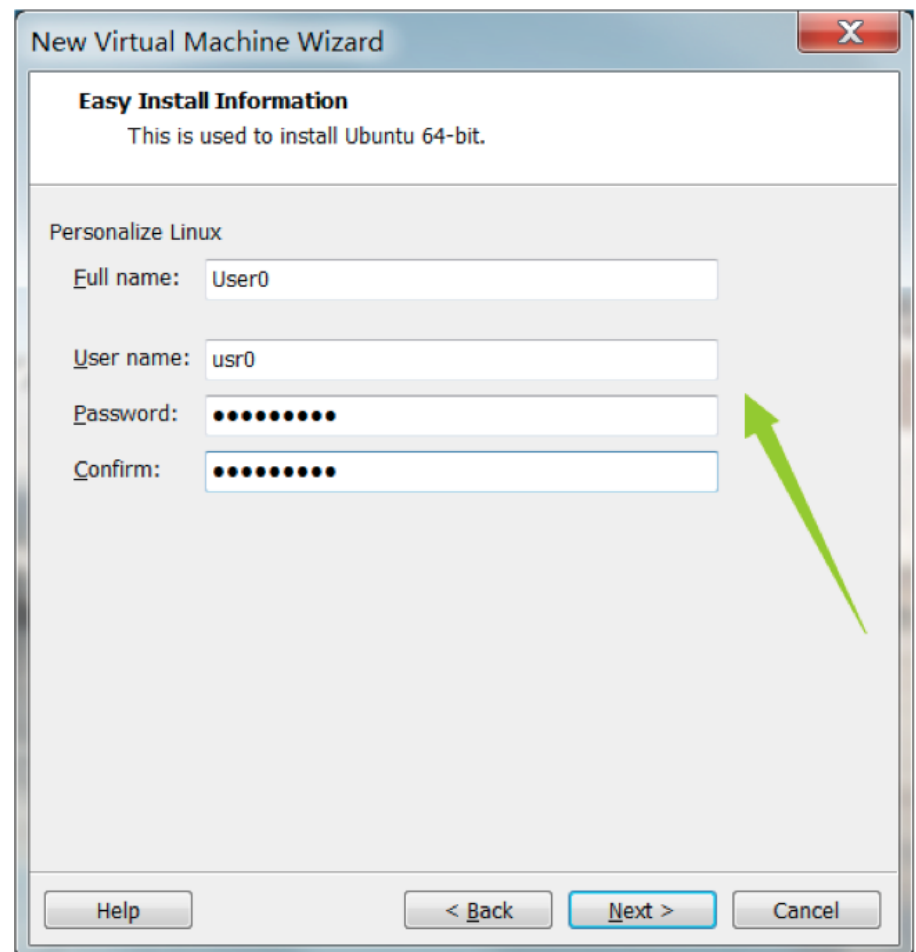
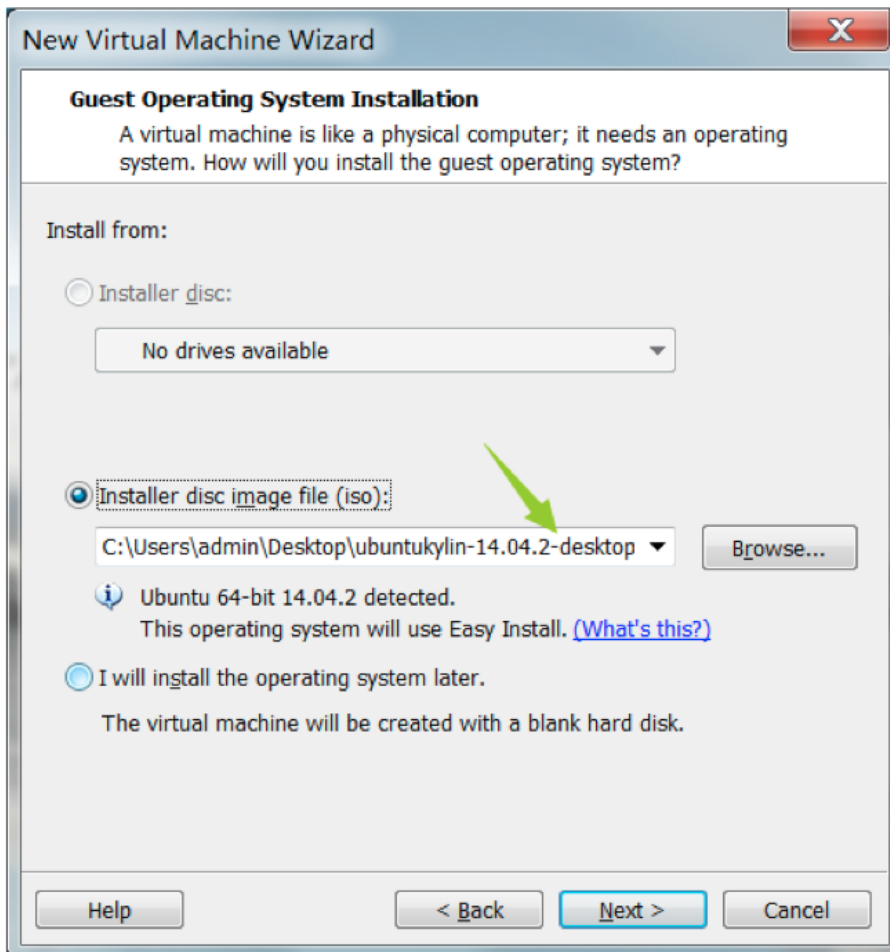


Create a Virtual Machine



Create a Virtual Machine

In the computers in our computer room, our .iso-file lies on path E:/ of the system.



Create a Virtual Machine

New Virtual Machine Wizard

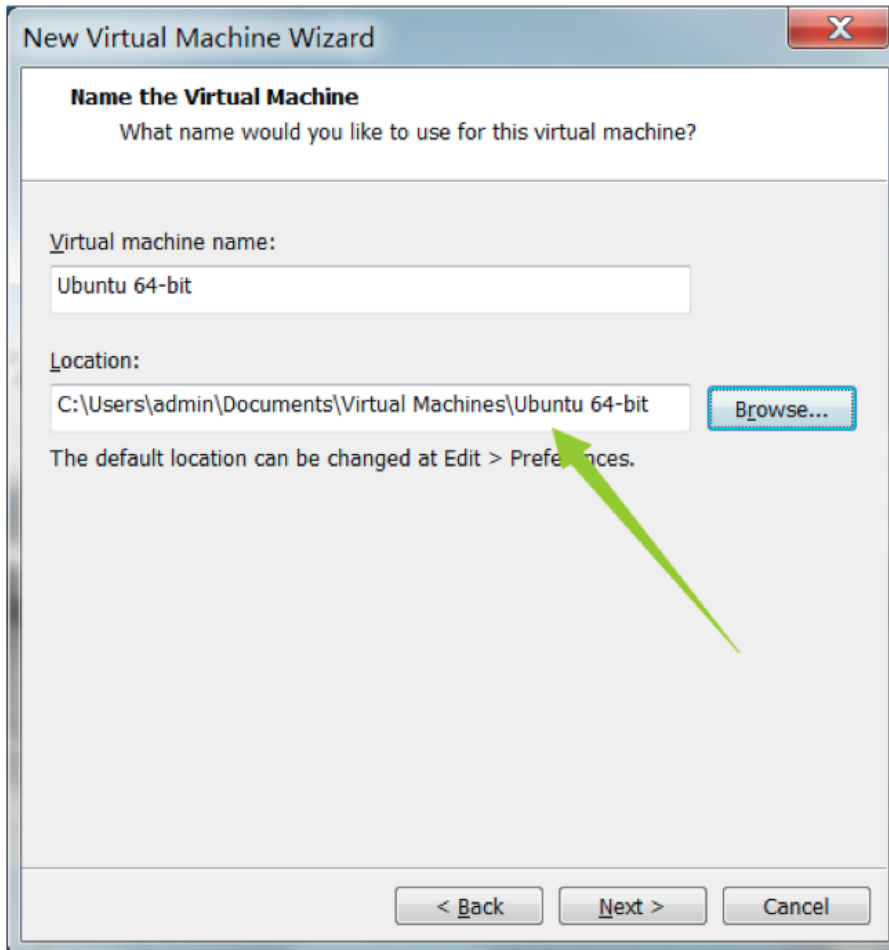
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
Ubuntu 64-bit

Location:
C:\Users\admin\Documents\Virtual Machines\Ubuntu 64-bit Browse...

The default location can be changed at Edit > Preferences.

< Back Next > Cancel



New Virtual Machine Wizard

Processor Configuration
Specify the number of processors for this virtual machine.

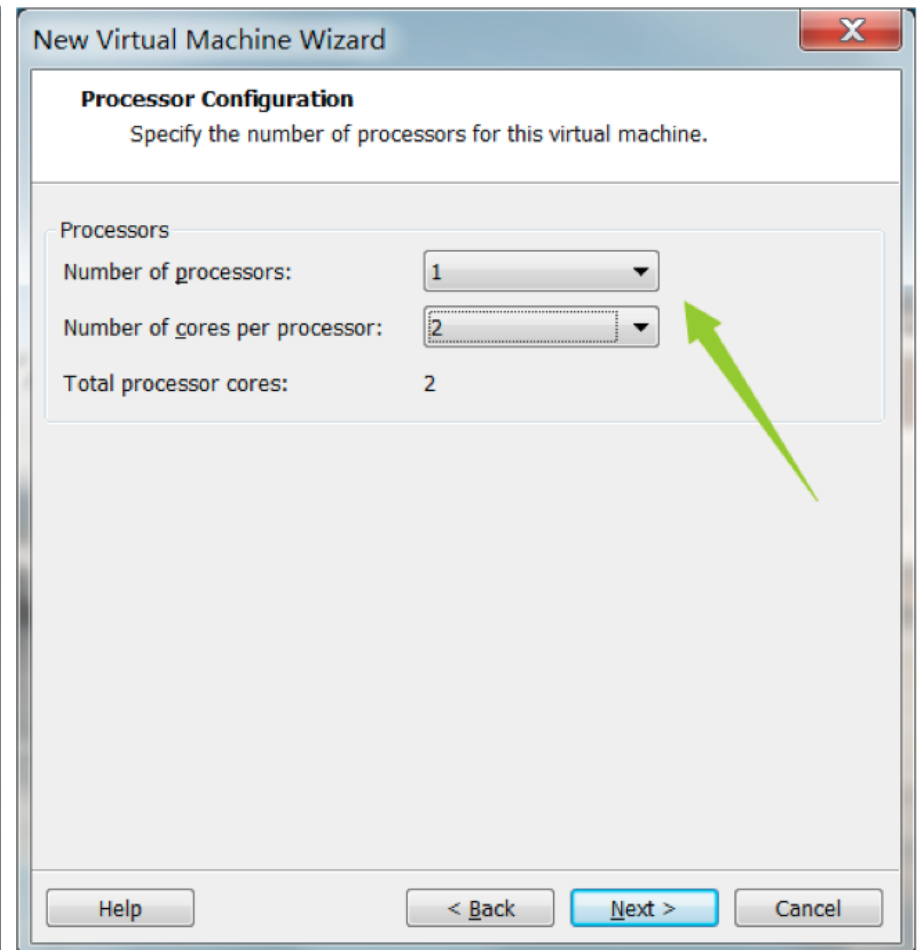
Processors

Number of processors: 1

Number of cores per processor: 2

Total processor cores: 2

Help < Back Next > Cancel



Create a Virtual Machine

New Virtual Machine Wizard

Memory for the Virtual Machine
How much memory would you like to use for this virtual machine?

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

64 GB
32 GB
16 GB
8 GB
4 GB
2 GB
1 GB
512 MB
256 MB
128 MB
64 MB
32 MB
16 MB
8 MB
4 MB

Memory for this virtual machine: 4096 MB

Maximum recommended memory: 13644 MB

Recommended memory: 1024 MB

Guest OS recommended minimum: 512 MB

Help < Back Next > Cancel

New Virtual Machine Wizard

Network Type
What type of network do you want to add?

Network connection

☐ Use bridged networking
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.

☒ Use network address translation (NAT)
Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.

☐ Use host-only networking
Connect the guest operating system to a private virtual network on the host computer.

☐ Do not use a network connection

Help < Back Next > Cancel

Create a Virtual Machine

New Virtual Machine Wizard

Select I/O Controller Types
Which SCSI controller type would you like to use?

I/O controller types

IDE Controller: ATAPI

SCSI Controller: ☐ BusLogic (Not available for 64-bit guests)
☒ LSI Logic (Recommended)
☐ LSI Logic SAS

Help < Back Next > Cancel

New Virtual Machine Wizard

Select a Disk
Which disk do you want to use?

Disk

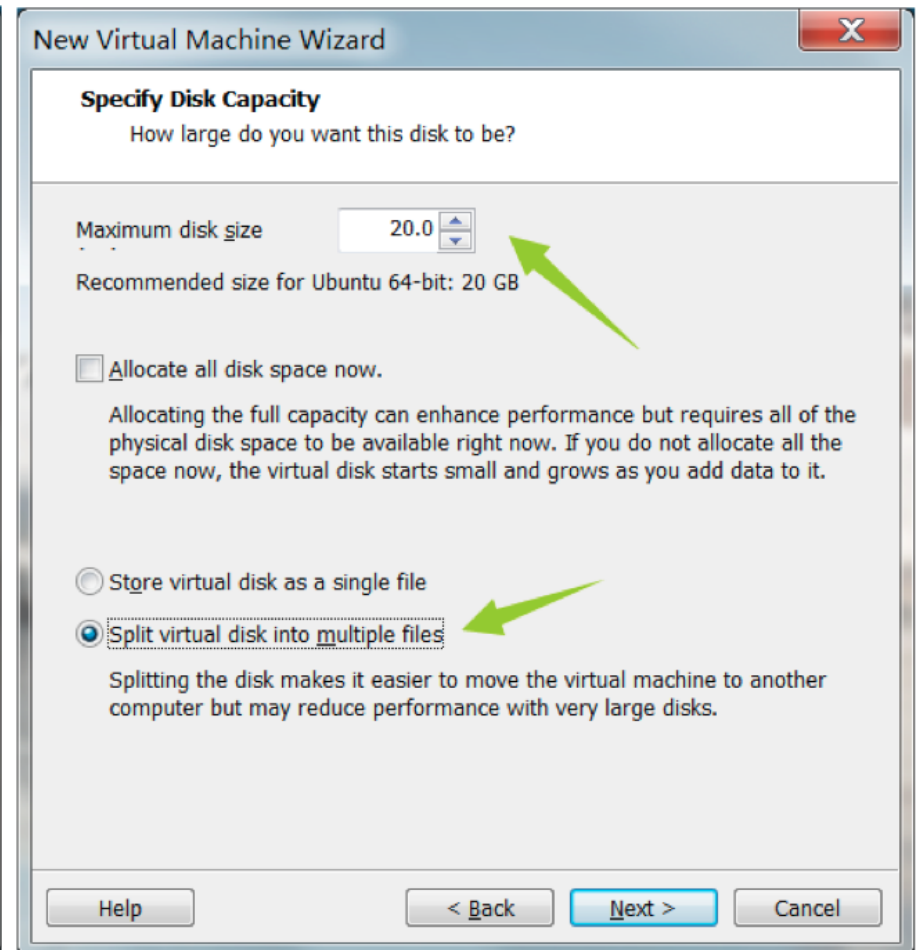
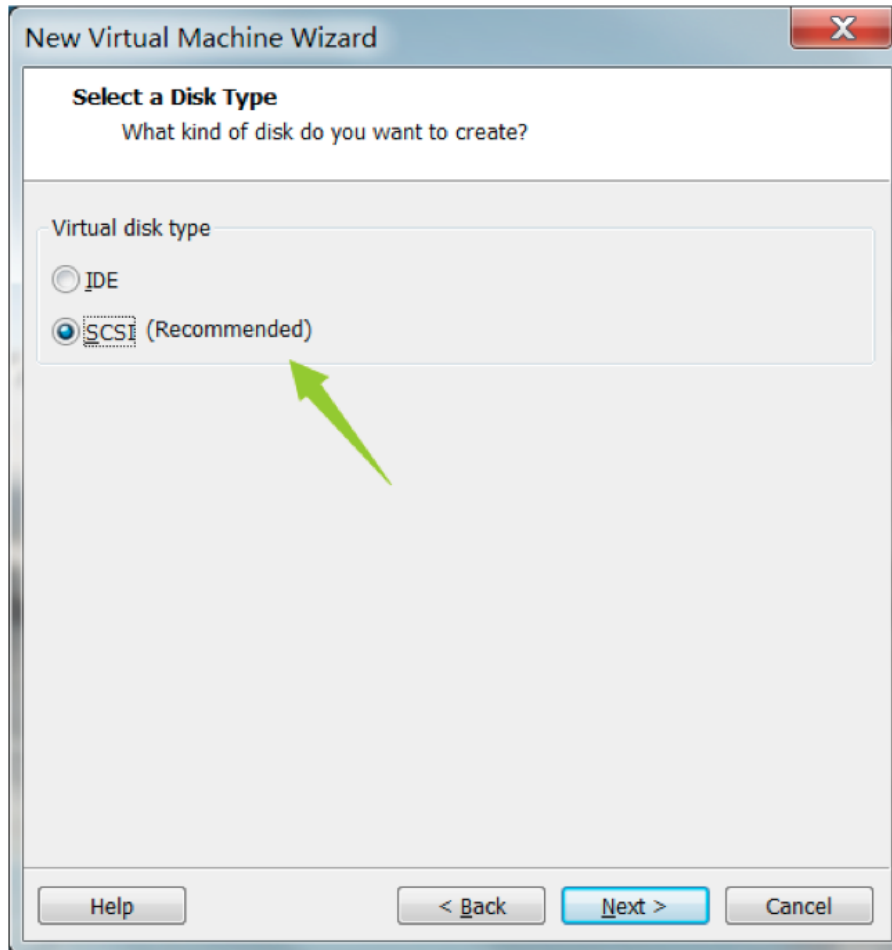
☒ Create a new virtual disk
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

☐ Use an existing virtual disk
Choose this option to reuse a previously configured disk.

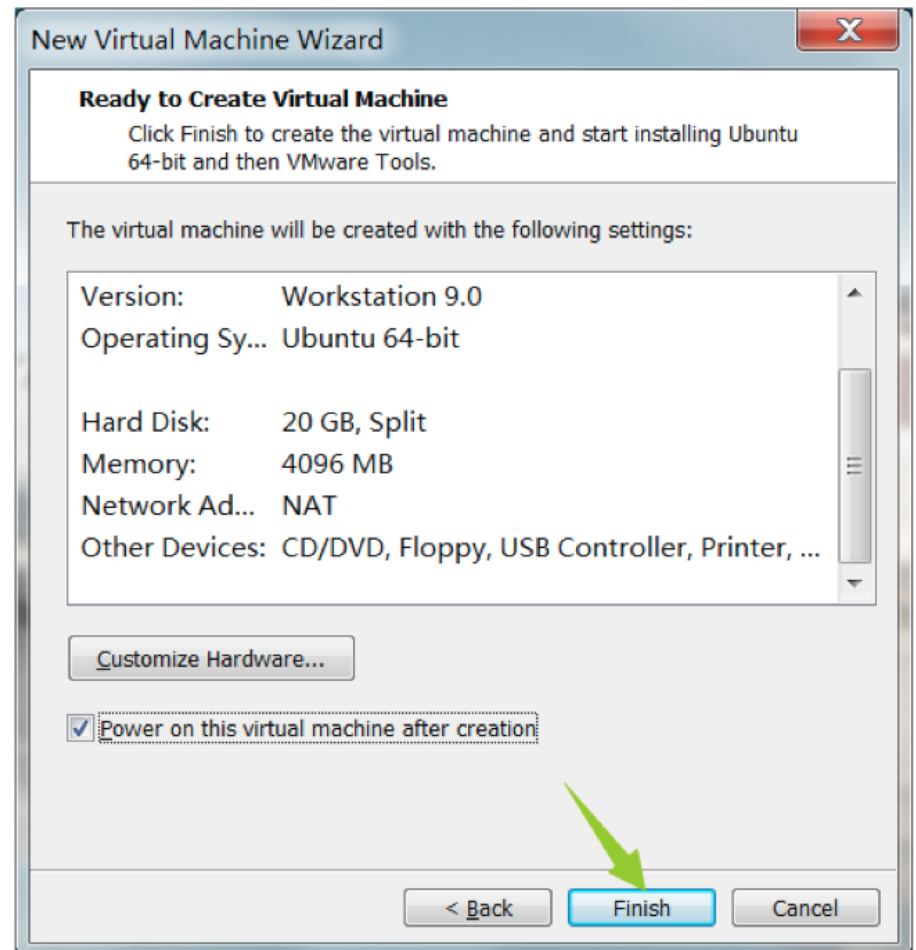
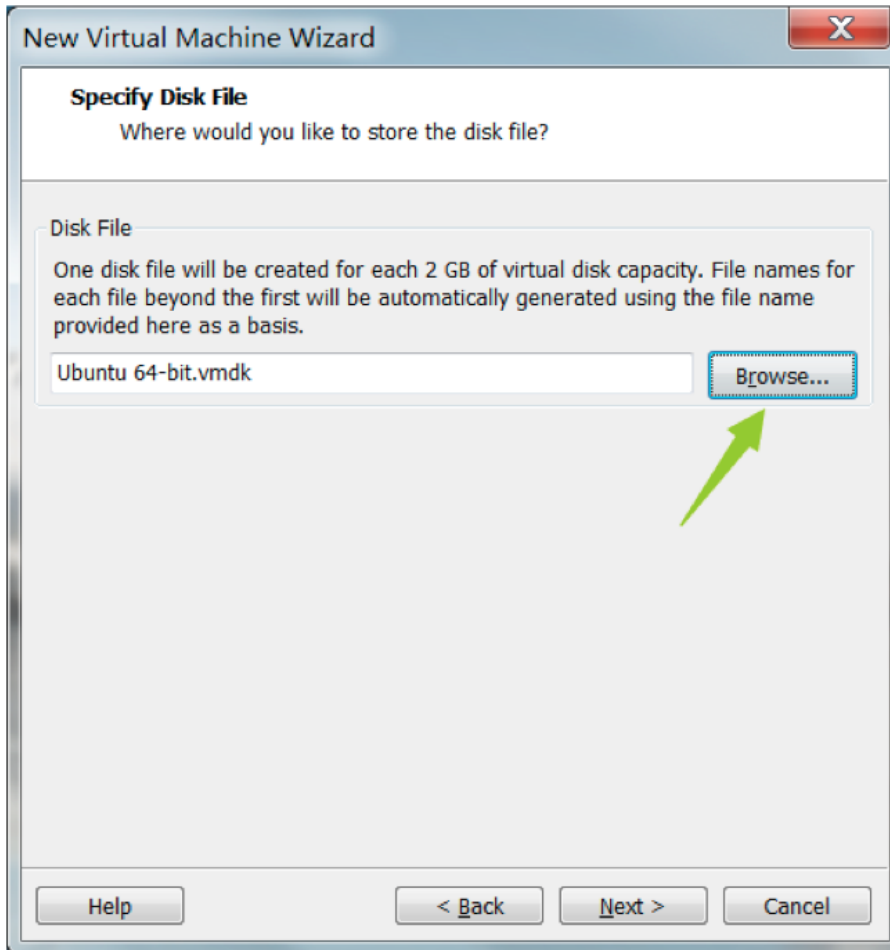
☐ Use a physical disk (for advanced users)
Choose this option to give the virtual machine direct access to a local hard disk.

Help < Back Next > Cancel

Create a Virtual Machine



Create a Virtual Machine



Create a Virtual Machine (Mac)

Mac Virtual Machine -- Parallels desktop

magnet:?

xt=urn:btih:5EE7E1DC3E01F362B0E53BFEE9E4D6DCDEDAD61B

Create a Virtual Machine (Mac)



Create a Virtual Machine (Mac)



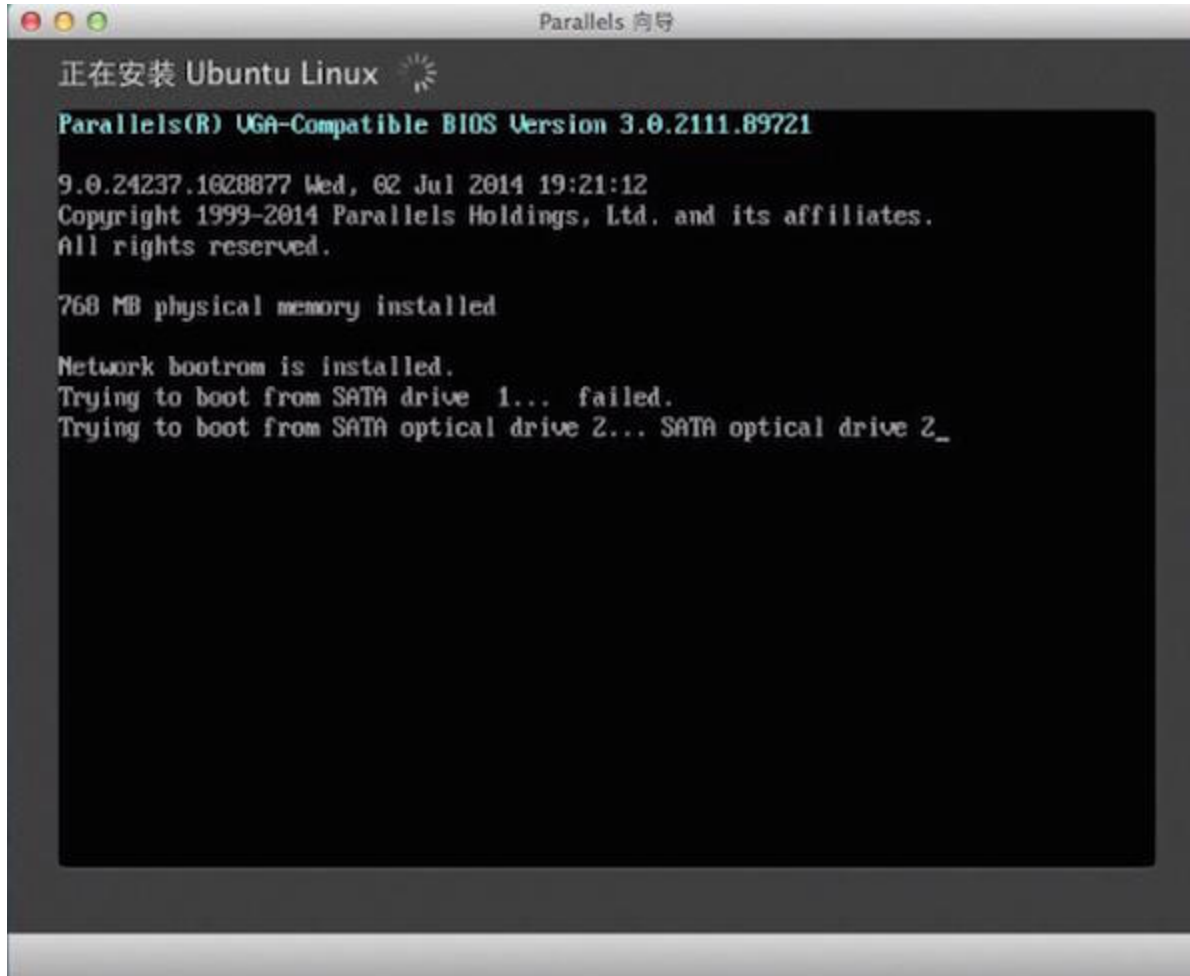
Create a Virtual Machine (Mac)



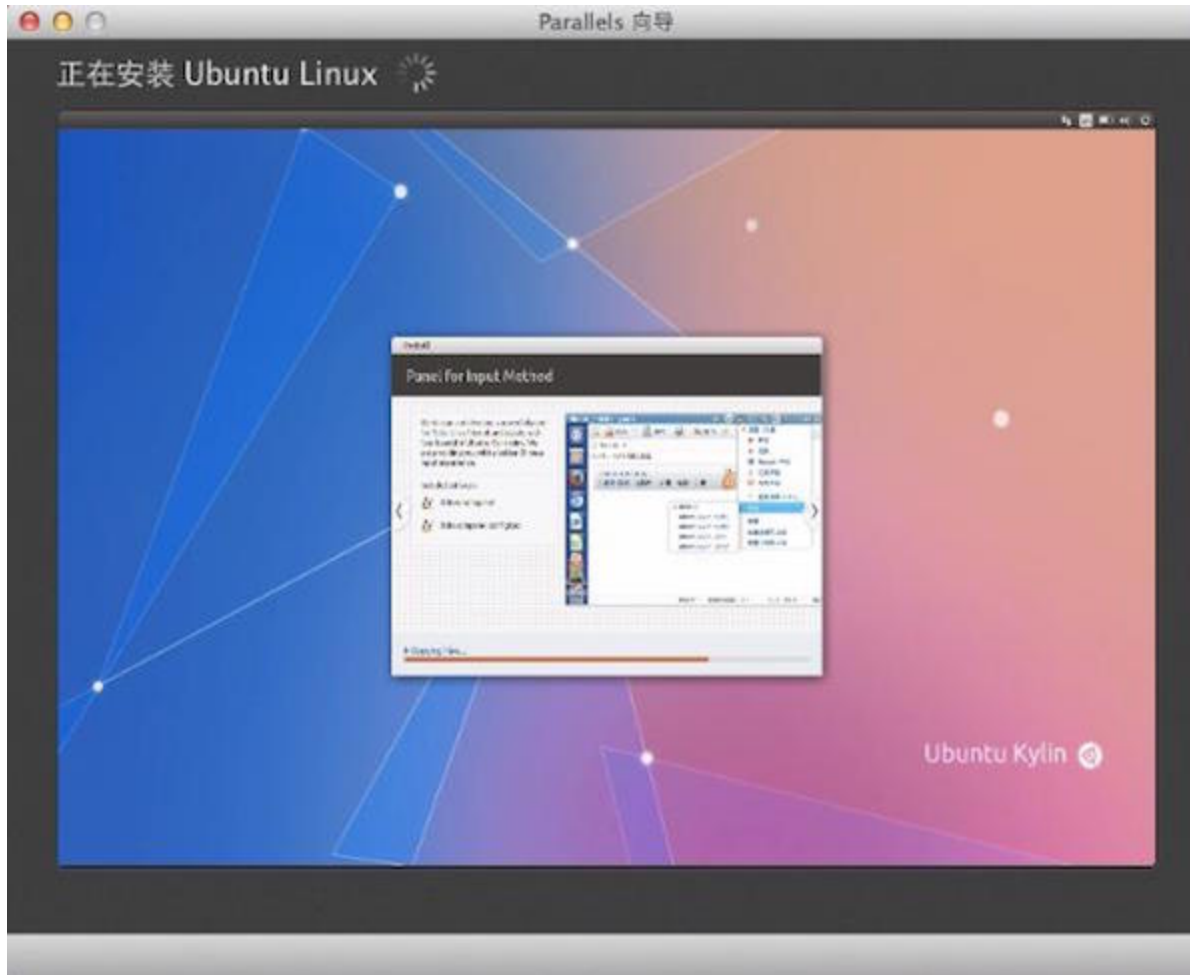
Create a Virtual Machine (Mac)



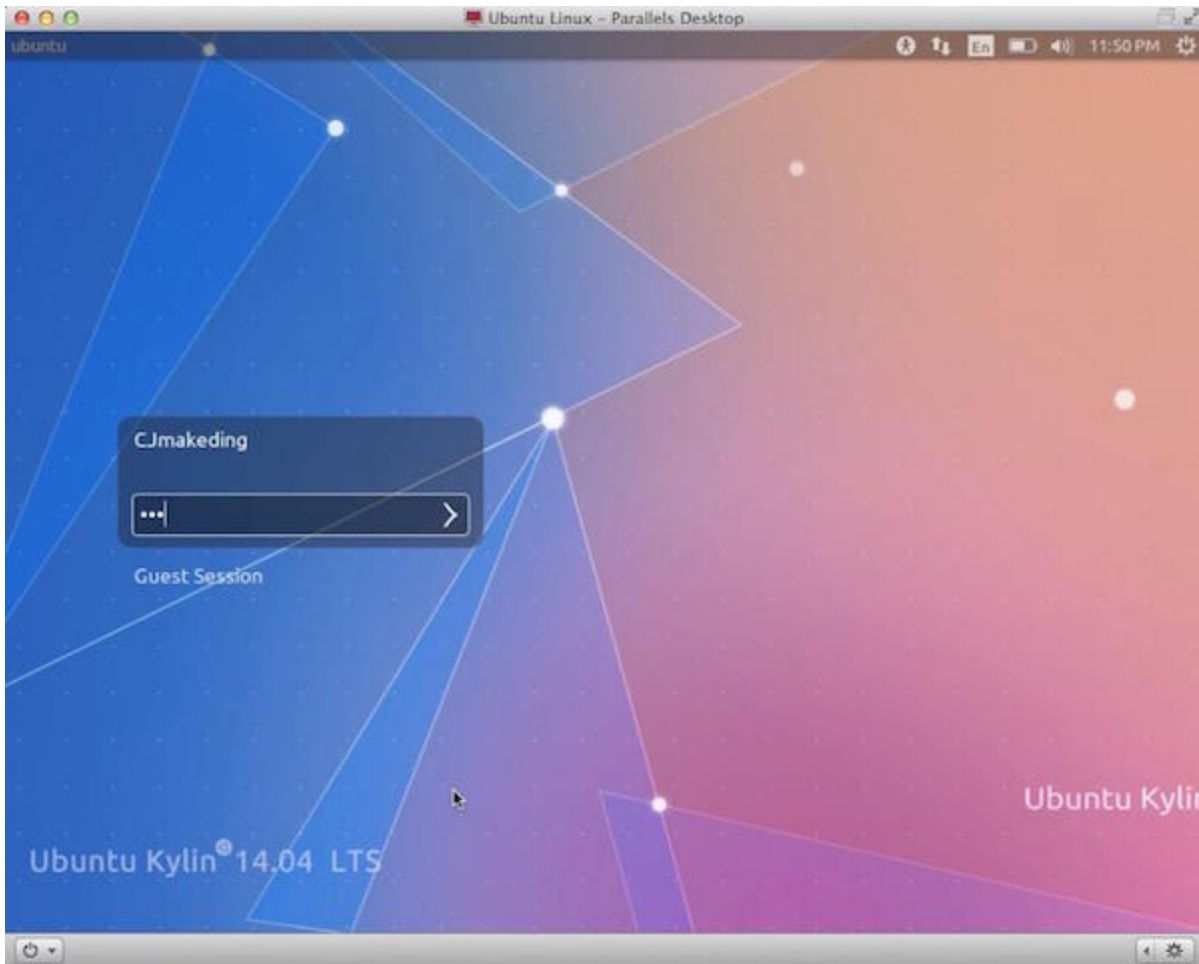
Create a Virtual Machine (Mac)



Create a Virtual Machine (Mac)



Create a Virtual Machine (Mac)



PART I

- Brief Introduction
- Basic Conceptions & Environment
- Install & Configure a Virtual Machine
- Basic Commands

Basic Commands

command [-options] [arguments]

- cd pwd ls
- su chmod cat
- touch rename mv cp
- mkdir rmdir rm
- find grep
- > >> | xargs
- awk
- man help --help

cd (change directory)

```
cd  
cd ~  
cd -  
cd ..
```

pwd (print working directory)

```
pwd
```

ls (list segment)

- l long - Displaying long format
- a all - Lists all files in the given directory
- R recursive - Recursively lists subdirectories.
- d directory - Shows information about a directory

```
ls
ls -l
ls -a
ls -R
ls -d
ls -la
ls -ld
...
```

su (switch user)

```
su weiyang
```

chmod (change mode)

```
chmod 660 class1.txt  
chmod u-r class1.txt
```

cat (concatenate)

```
cat class1.txt  
cat godweiyang.txt
```

touch

```
touch class1.txt
```

rename

```
rename 's/oslab/oslab0/' o*b?.txt
```

mv (move)

```
mv oslab.txt oslab1.txt  
mv oslab01.txt oslab02.txt /home/oslab
```

cp (copy)

```
cp oslab03.txt /home/oslab
```

mkdir (make directory)

```
mkdir Lesson1/rename
```

rmdir (remove empty directory)

```
rmdir empty_directory
```

rm (remove)

-r recursive

-i interactive

-f force

```
rm -rf ~/Lesson1/*  
rm -i oslab04.txt
```


find

```
find ~ -name "*.txt"
```

grep

globally search a regular expression and print

```
grep match_pattern file_name  
grep apple oslab05.txt  
grep -i apple oslab05.txt
```

> & >> (redirection)

```
cat oslab06.txt oslab07.txt > oslab08.txt  
cat oslab06.txt oslab07.txt >> oslab08.txt
```

| (pipeline)

```
command1 | command2  
cat oslab09.txt | grep jt
```

xargs

```
cat oslab09.txt | ls -l  
cat oslab09.txt | xargs ls -l
```

awk (Aho, Weinberg & Kernighan)

AWK is a programming language designed for text processing and typically used as a data extraction and reporting tool.

pattern { action }

BEGIN、 regular expression、 END

{ function calls, variable assignments, calculations }

```
awk 'BEGIN { print "Hello, world!" }'
```

man (manual)

```
man ls
```

help

```
help cd
```

--help

```
ls --help
```

Wikipedia

<https://en.wikipedia.org/wiki/AWK>