Introduction of Linux

杜威

> oslab2021@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
- 1991 - LINUX

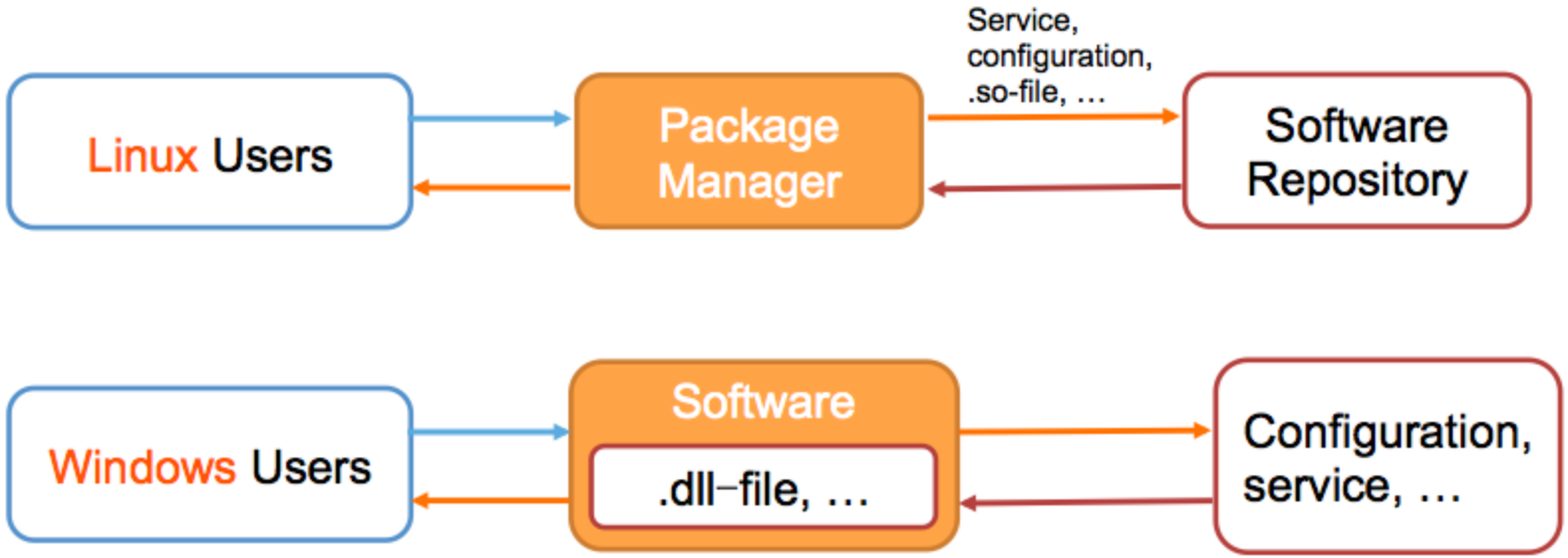
Distribution

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

Features

- Protable
- Open source
- Security
- ...

Linux vs Windows Software



Linux install software

Package Manager： `apt-get` (Advanced Package Tool)

```
zheng@kernel:~$ sudo apt-get autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
zheng@kernel:~$ sudo apt-get install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  manpages-dev libc-dev-bin linux-libc-dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  binutils gcc-4.4 libc-dev-bin libgomp1 linux-libc-dev manpages-dev
Suggested packages:
  binutils-doc gcc-multilib autoconf automake1.9 libtool flex bison gdb
  gcc-doc gcc-4.4-multilib libmudflap0-4.4-dev gcc-4.4-doc gcc-4.4-locales
  libgcc1-dbg libgomp1-dbg libmudflap0-dbg libcloog-ppl0 libppl-c2 libppl7
Recommended packages:
  libc6-dev libc-dev
The following NEW packages will be installed:
  binutils gcc gcc-4.4 libc-dev-bin libgomp1 linux-libc-dev manpages-dev
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,147kB of archives.
After this operation, 22.8MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

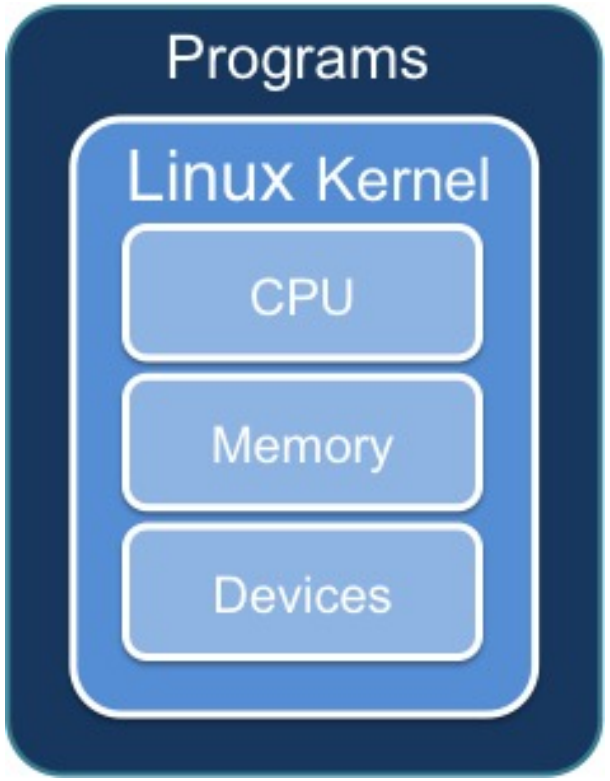
PART I

- 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.

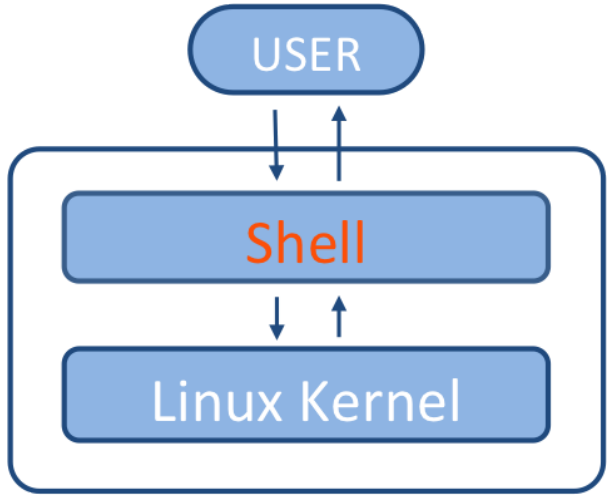
- memory management
- process scheduling
- file system
- ...



Shell (CLI shell)

Command Line Interface

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

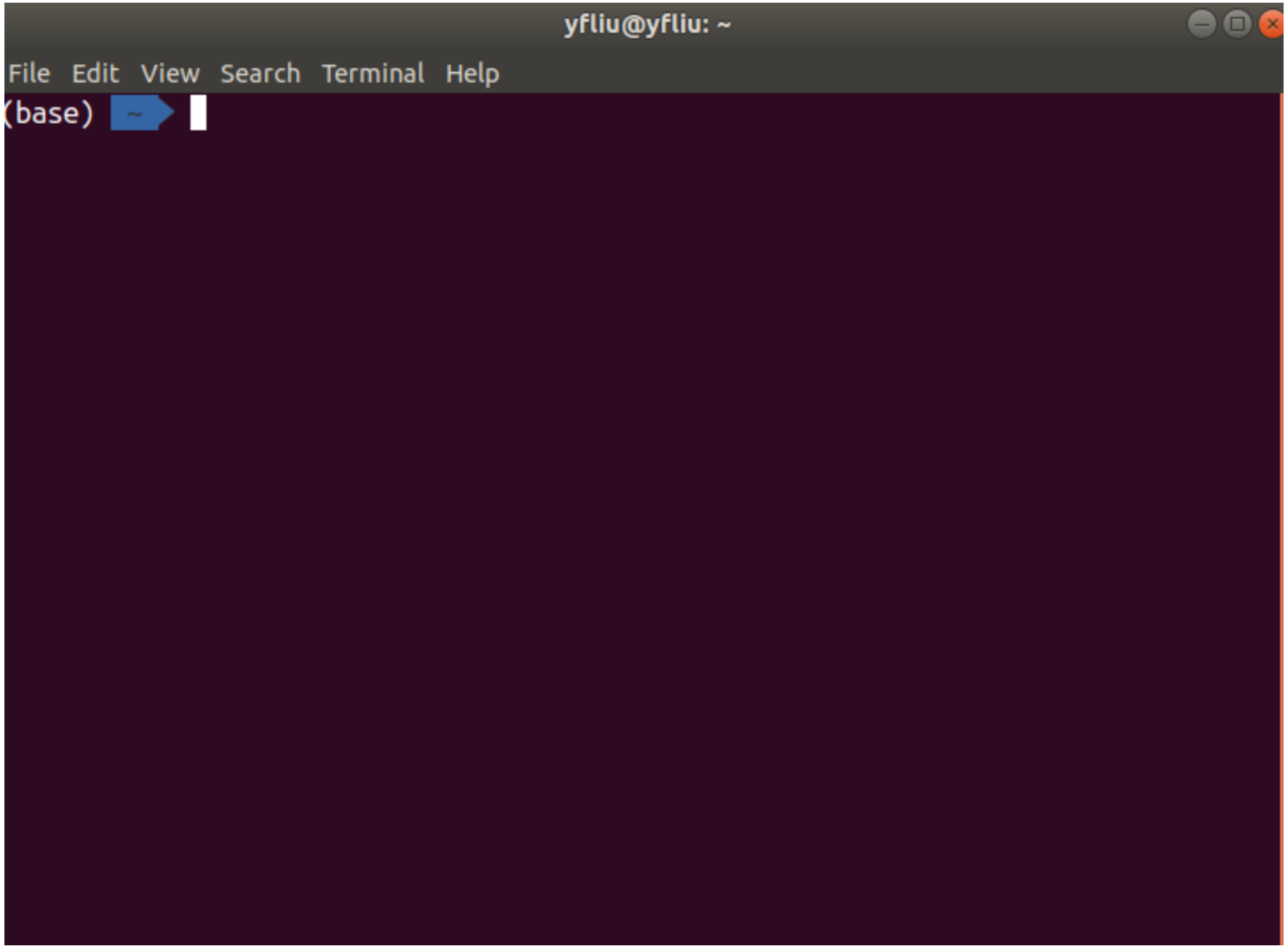


Terminal↔Shell

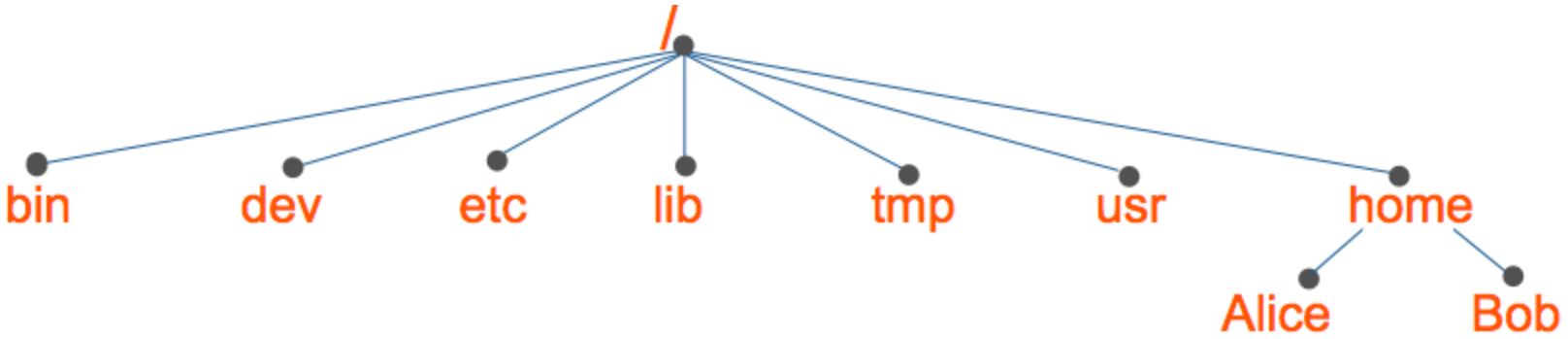
The terminal send information to the shell, receive and display the information from the shell.

Open Terminal

keyboard accelerators: `CTRL+ALT+T`



File System



Tree structure, with the root directory " / "

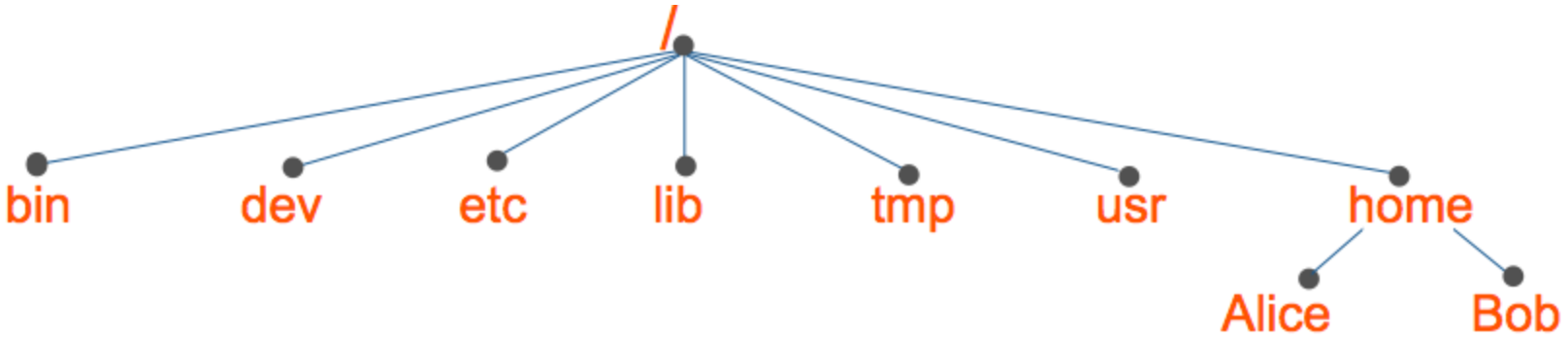
`/home/oslab/...`

`~` = `/home/oslab`

`.`

`..`

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

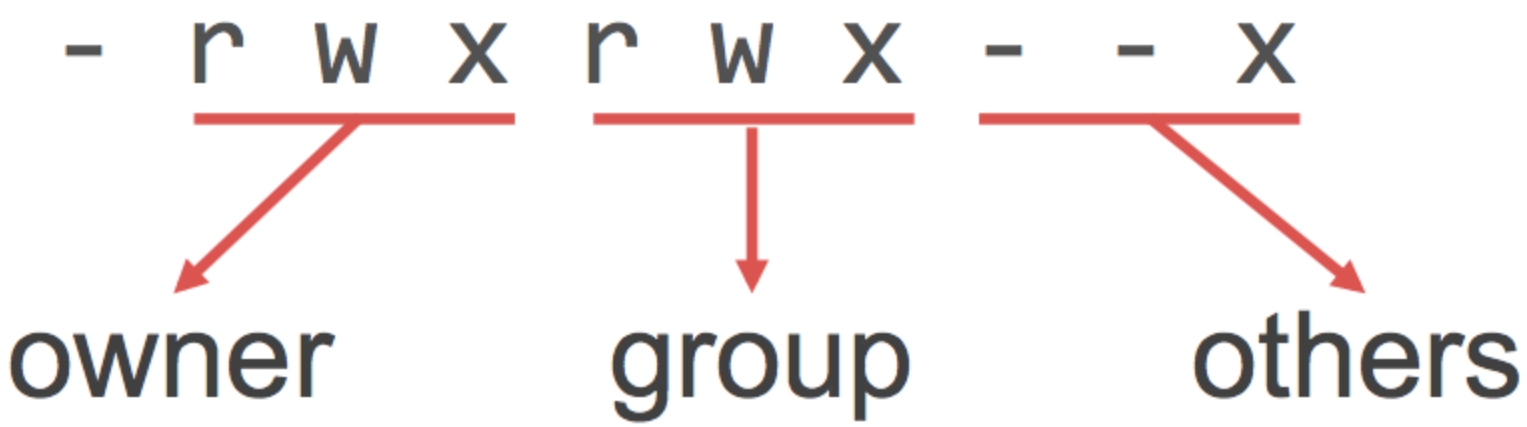
`/usr` : files related to user tools and applications

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
```

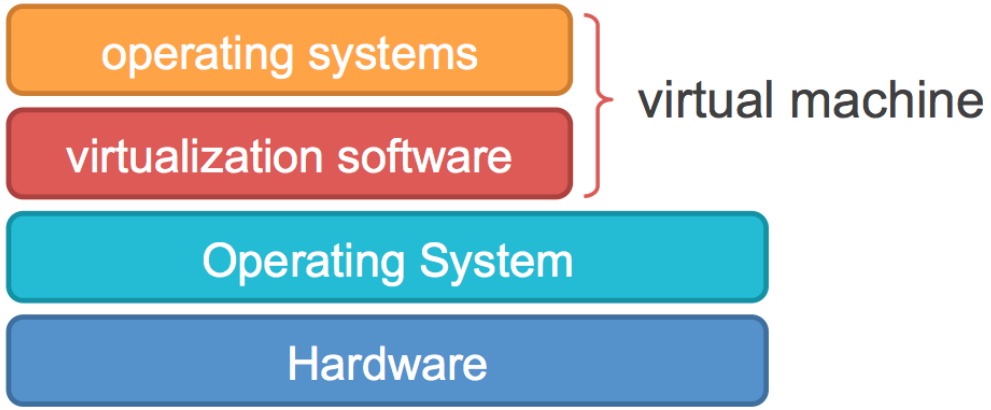
PART I

- 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.



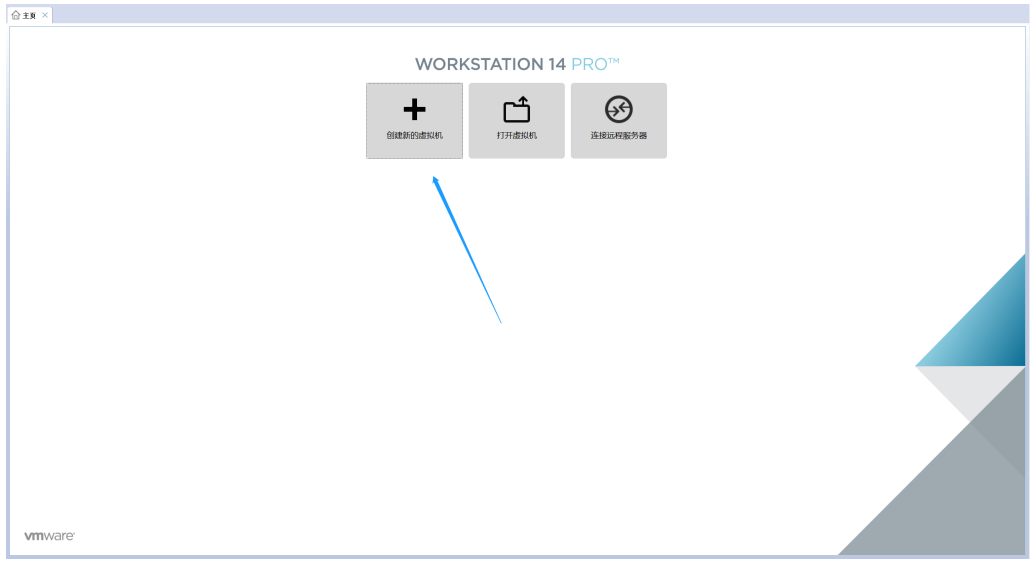
- VMware
- Virtual Box
- Virtual PC

Install the Virtual Machine

VMware Workstation 14.0 + Ubuntu 20.04 LTS

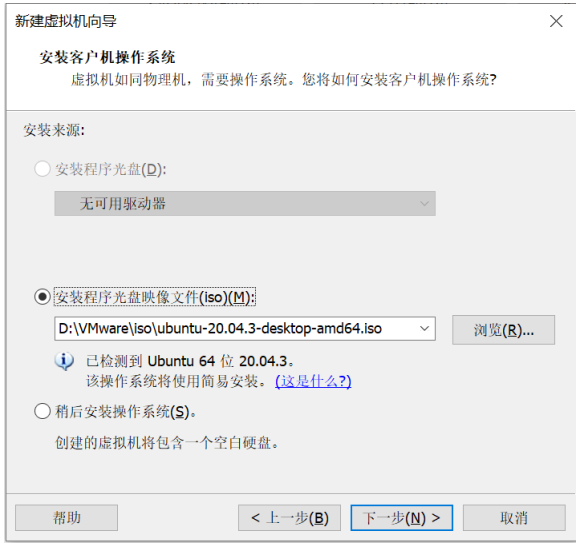
1. Download the Setup File of Vmware14.0
   http://download3.vmware.com/software/wkst/file/VMware-workstation-full-14.0.0-6661328.exe
2. Download the Ubuntu Ubuntu 20.04 LTS from the official website
   www.ubuntu.com/download/desktop
3. Install VMware 14.0
4. Create a Virtual Machine in the VMware

Create a Virtual Machine

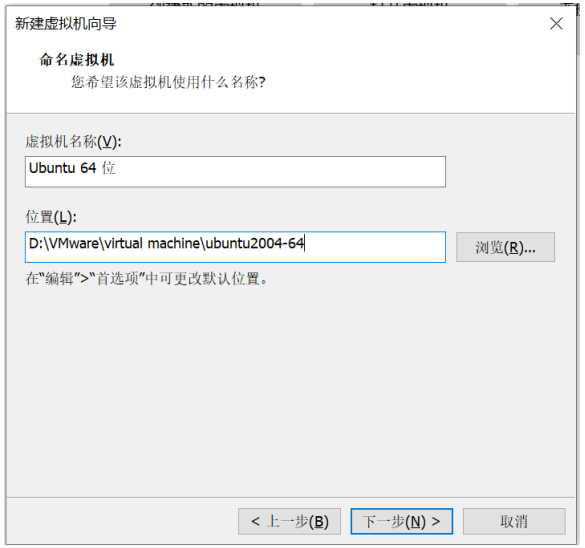


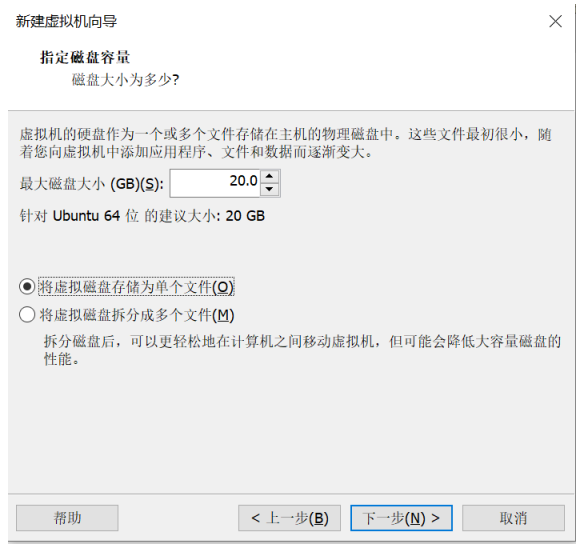
Create a Virtual Machine

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

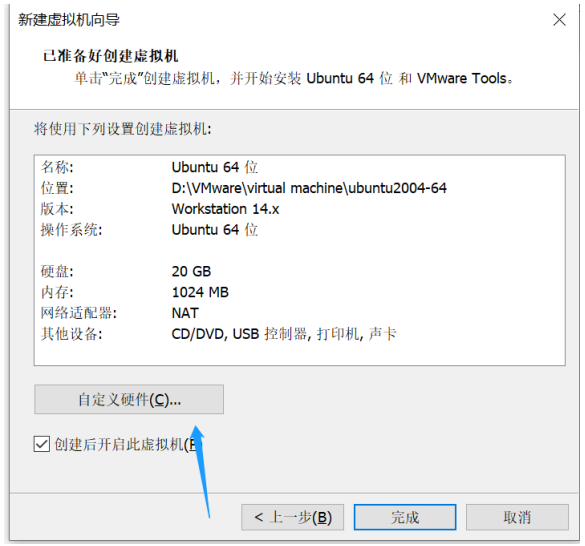


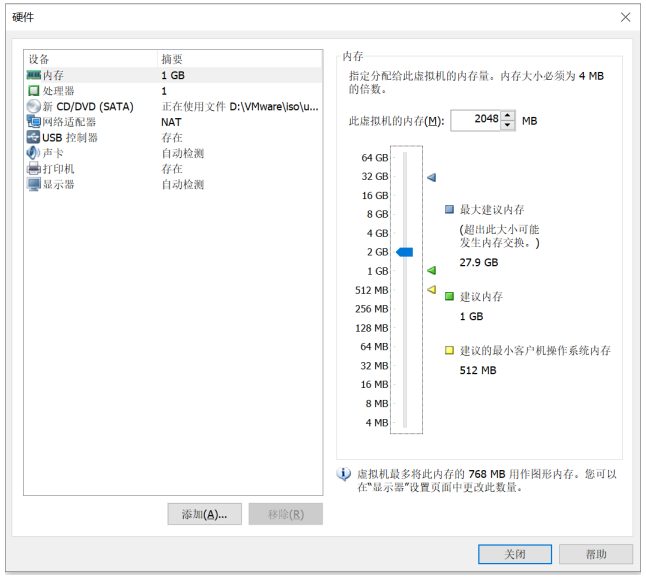
Create a Virtual Machine





Create a Virtual Machine

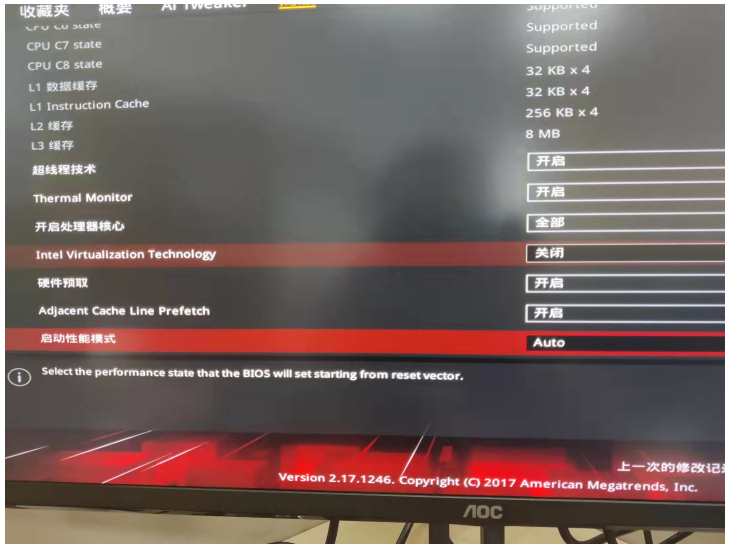




所分配的内存过少会导致虚拟机运行时卡顿。

Create a Virtual Machine

--- 需要在BIOS界面打开虚拟环境的设置



Create a Virtual Machine (Mac)

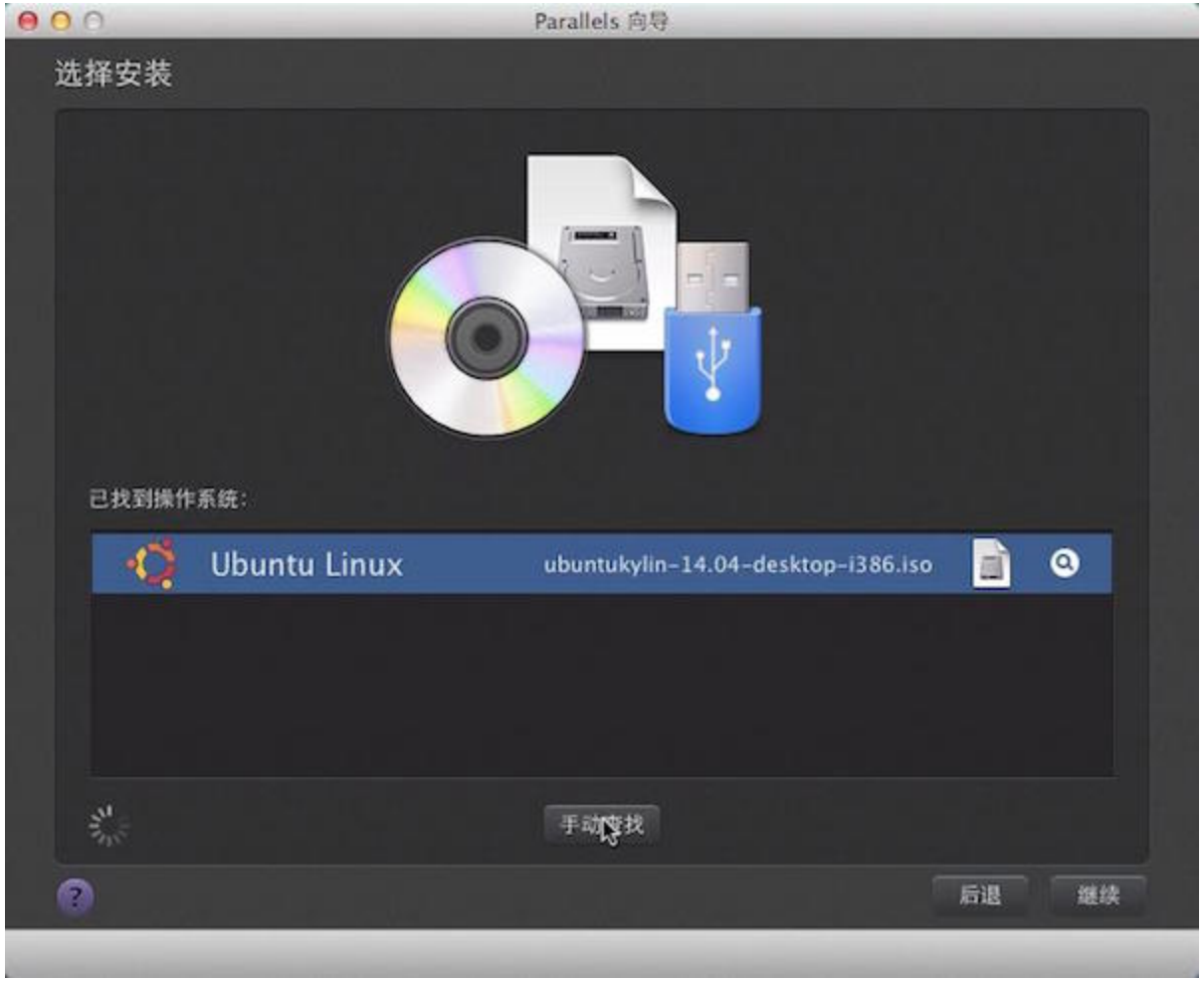
Mac Virtual Machine -- Parallels desktop

`Ubuntu14.iso->` `magnet:?xt=urn:btih:5EE7E1DC3E01F362B0E53BFEE9E4D6DCDEDAD61B`

`Parallels desktop->` `http://xclient.info/s/parallels-desktop.html?t=2c5f238779ee02ff6e1b5cda873deeacaeabc304`

Create a Virtual Machine (Mac)

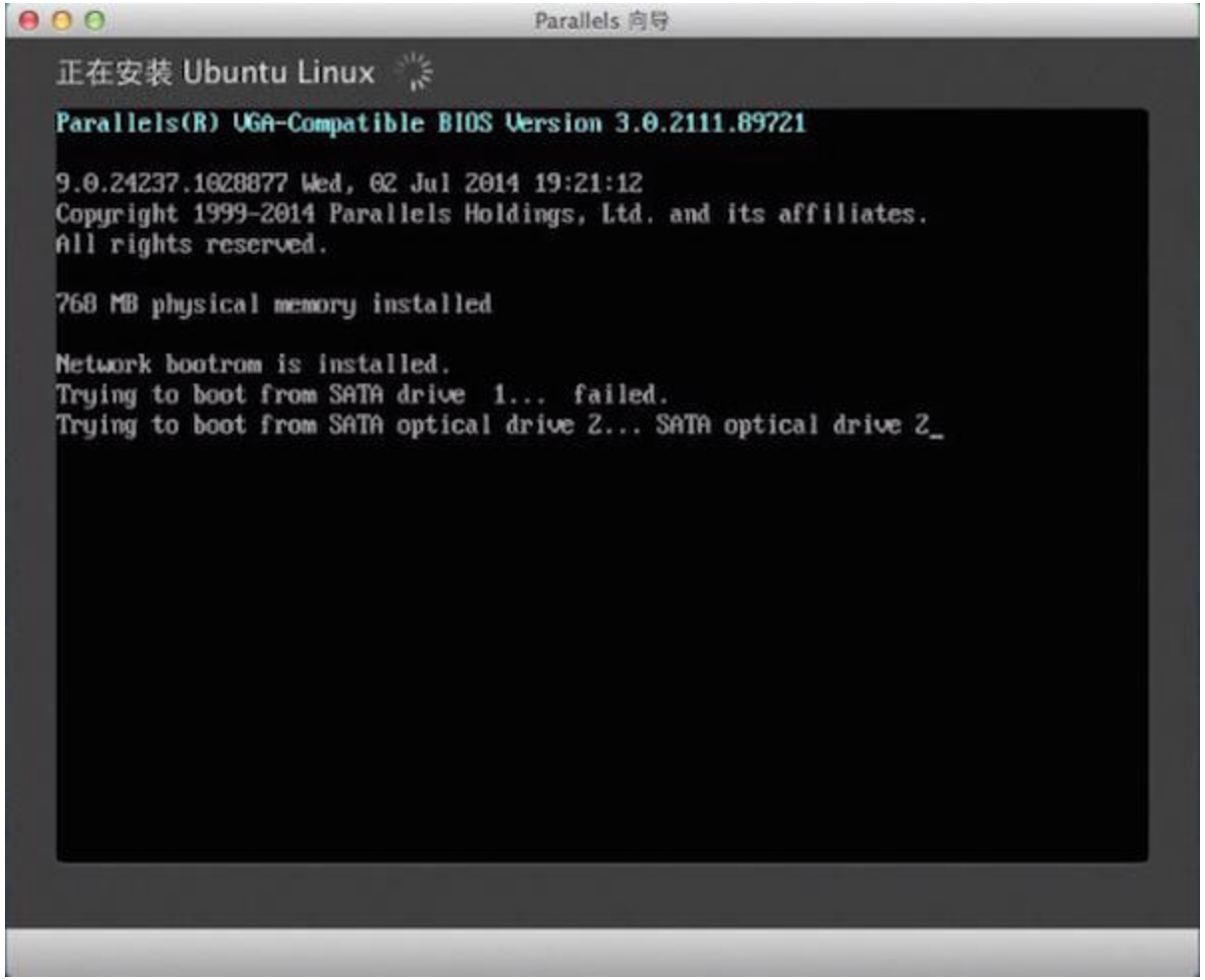
Create a Virtual Machine (Mac)



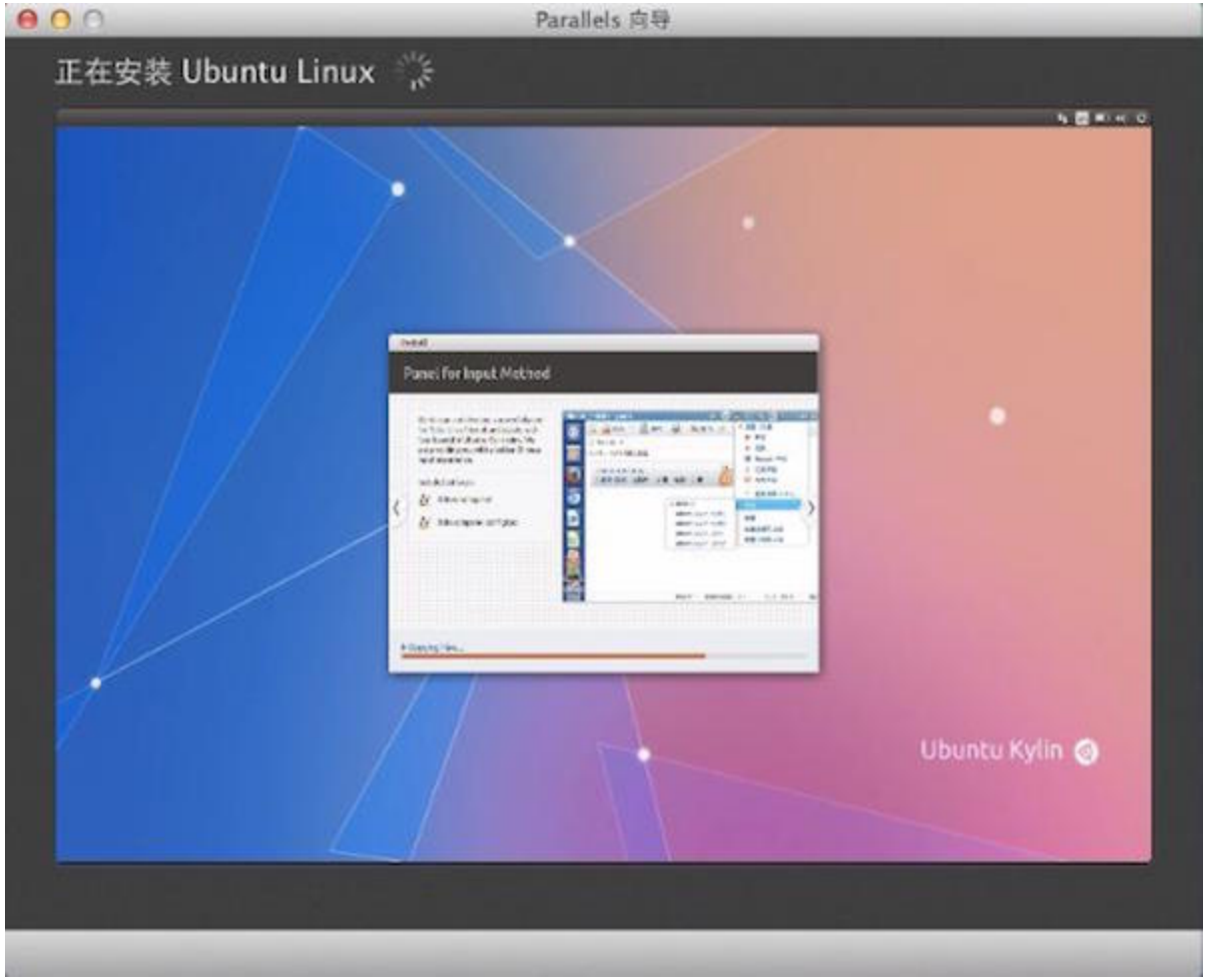
Create a Virtual Machine (Mac)

Create a Virtual Machine (Mac)

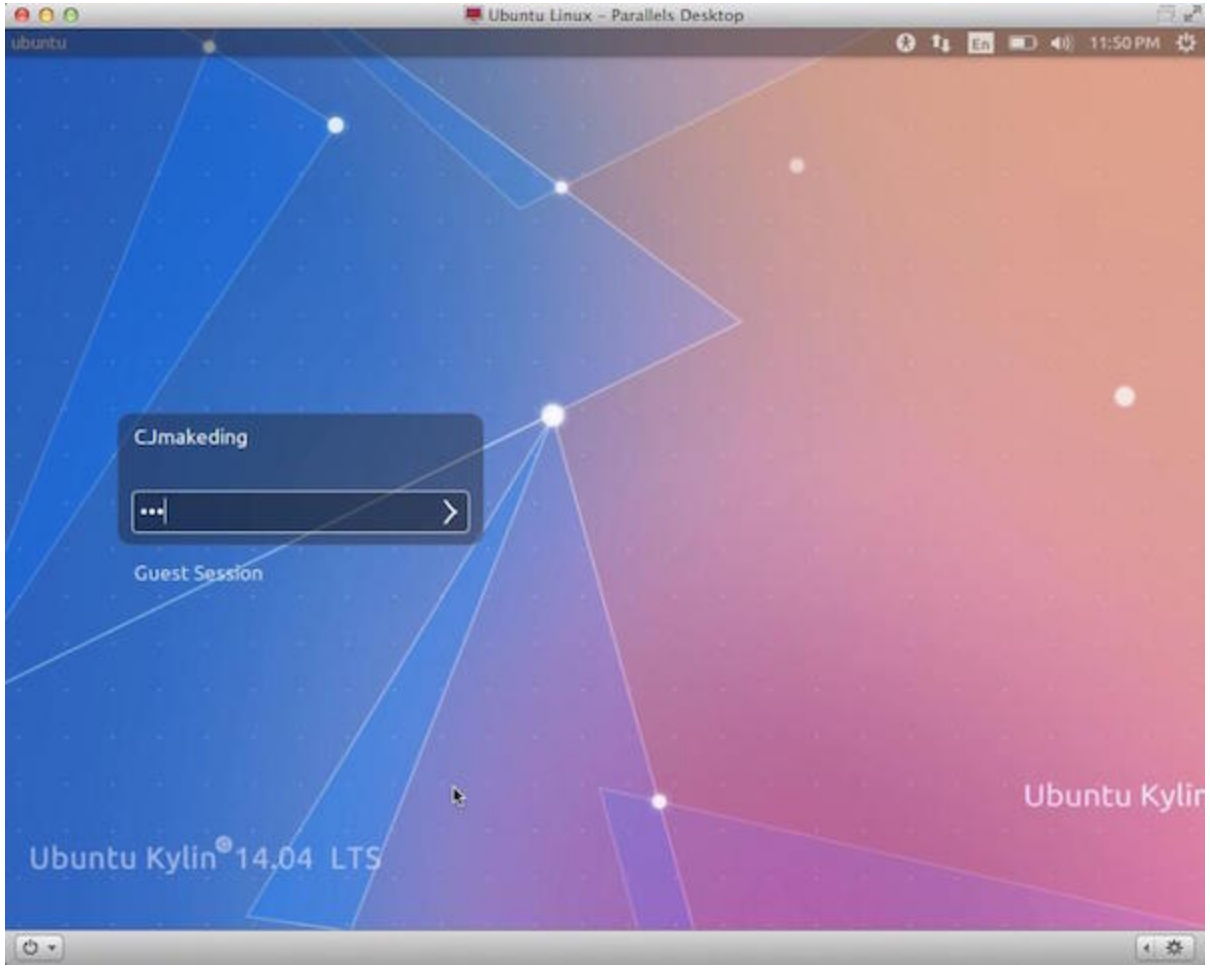
Create a Virtual Machine (Mac)



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
```

`man` (manual)

```
man ls
```

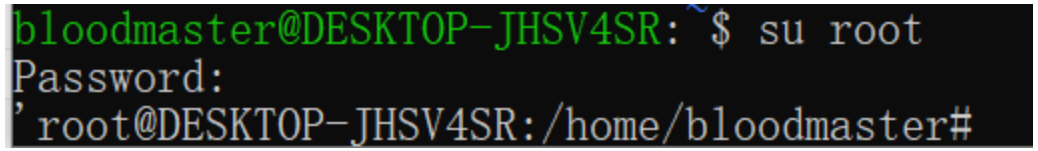
`ls` (list segment)

- `-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

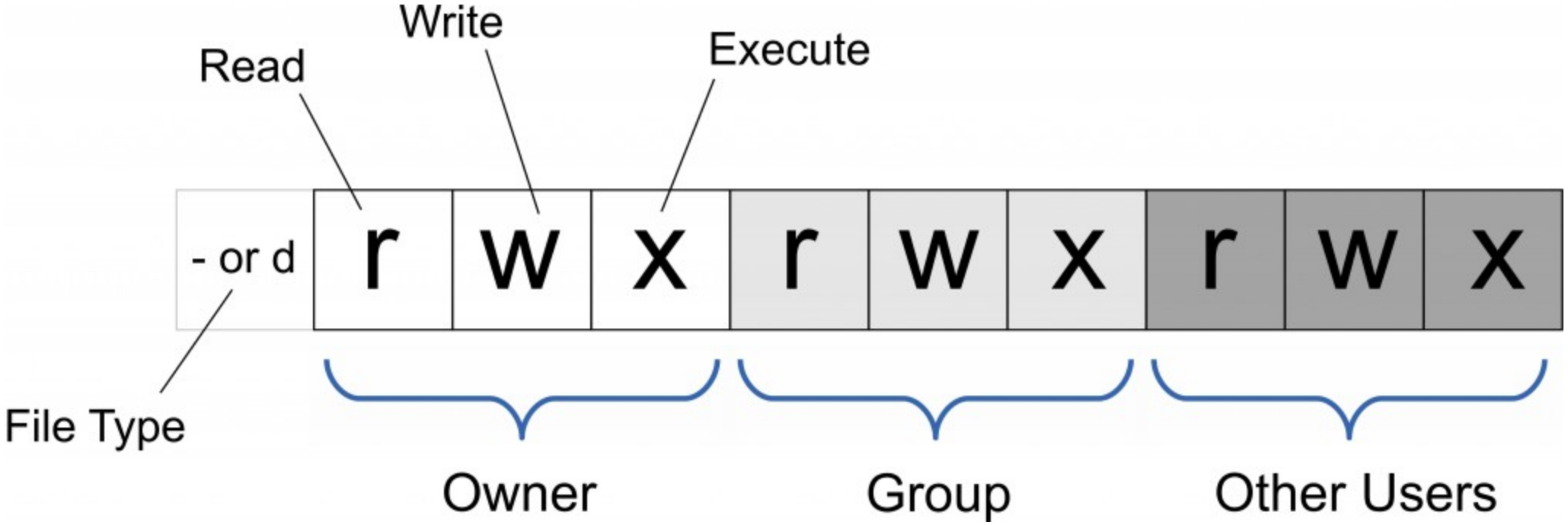
```
ls
ls -l
ls -a
ls -R
ls -d
ls -la
ls -ld
...
```

`su` (switch user)

```
su root
```



`chmod` (change mode)



```
ls -la
output:
---------- 1 bloodmaster bloodmaster    0 Sep  2 21:25 test1.txt

chmod 660 class1.txt
ls -la
output:
-rw-rw---- 1 bloodmaster bloodmaster    0 Sep  2 21:25 test1.txt

chmod u-r test1.txt
ls -la
output:
--w-rw---- 1 bloodmaster bloodmaster    0 Sep  2 21:25 test1.txt
```

`cat` (concatenate)

```
cat test1.txt
```

`touch`

```
touch test1.txt
```

`rename`

```
rename 's/test1/test11/'  test1.txt
```

`mv` (move)

```
mv test.txt test1.txt
mv test01.txt test02.txt /home/bloodmaster/test
```

`cp` (copy)

```
cp test.txt /home/bloodmaster/test
```

`mkdir` (make directory)

```
mkdir Lesson1/rename
```

`rmdir` (remove empty directory)

```
rmdir empty_directory
```

`rm` (remove)

`-r` recursive

`-i` interactive

`-f` force

`-rf`

```
rm -rf ~/Lesson1/*
rm -i test1.txt
```

find

```
find ~ -name "*.txt"
find . -type f
find . -type d
```

grep

> globally search a regular expression and print

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

`>` & `>>` (redirection)

```
覆盖
cat test1.txt test2.txt > test3.txt
追加
cat test1.txt test2.txt >> test3.txt
```

| (pipeline)

- input1 output1 | output2 | output3

```
command1 | command2
cat test3.txt | grep test | grep te
```

xargs

```
test1.txt:test2.txt test3.txt
test2.txt:test2
test3.txt:test3
cat test1.txt | xargs cat -n
```

`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 }

```
log.txt
2 this is a test
3 Are you like awk
This's a test
10 There are orange,apple,mongo
```

```
awk '{[pattern] action}' {filenames}
awk '{print $1,$4}' log.txt
awk -F[,] '{print $1,$4}' log.txt
awk 'BEGIN { print "Hello, world!" }'
awk 'BEGIN { for (i = 1; i <= 5; ++i) print i }'
```

`help`

```
help cd
```

`--help`

```
ls --help
```

Wikipedia

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