

HCIA-openEuler Training Courses

HCIA-openEuler

openEuler System Engineer

Lab Guide

(PC Edition)

Issue: 1.0



Huawei Technologies Co., Ltd.

Copyright © Huawei Technologies Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base Bantian, Longgang, Shenzhen People's Republic of China

Website: <http://e.huawei.com>

Introduction to Huawei Certification System

The Huawei certification system is a platform for shared growth, part of a thriving partner ecosystem. There are two types of certification: one for ICT architectures and applications, and one for cloud services and platforms. There are three levels of certification available:

Huawei Certified ICT Associate (HCIA)

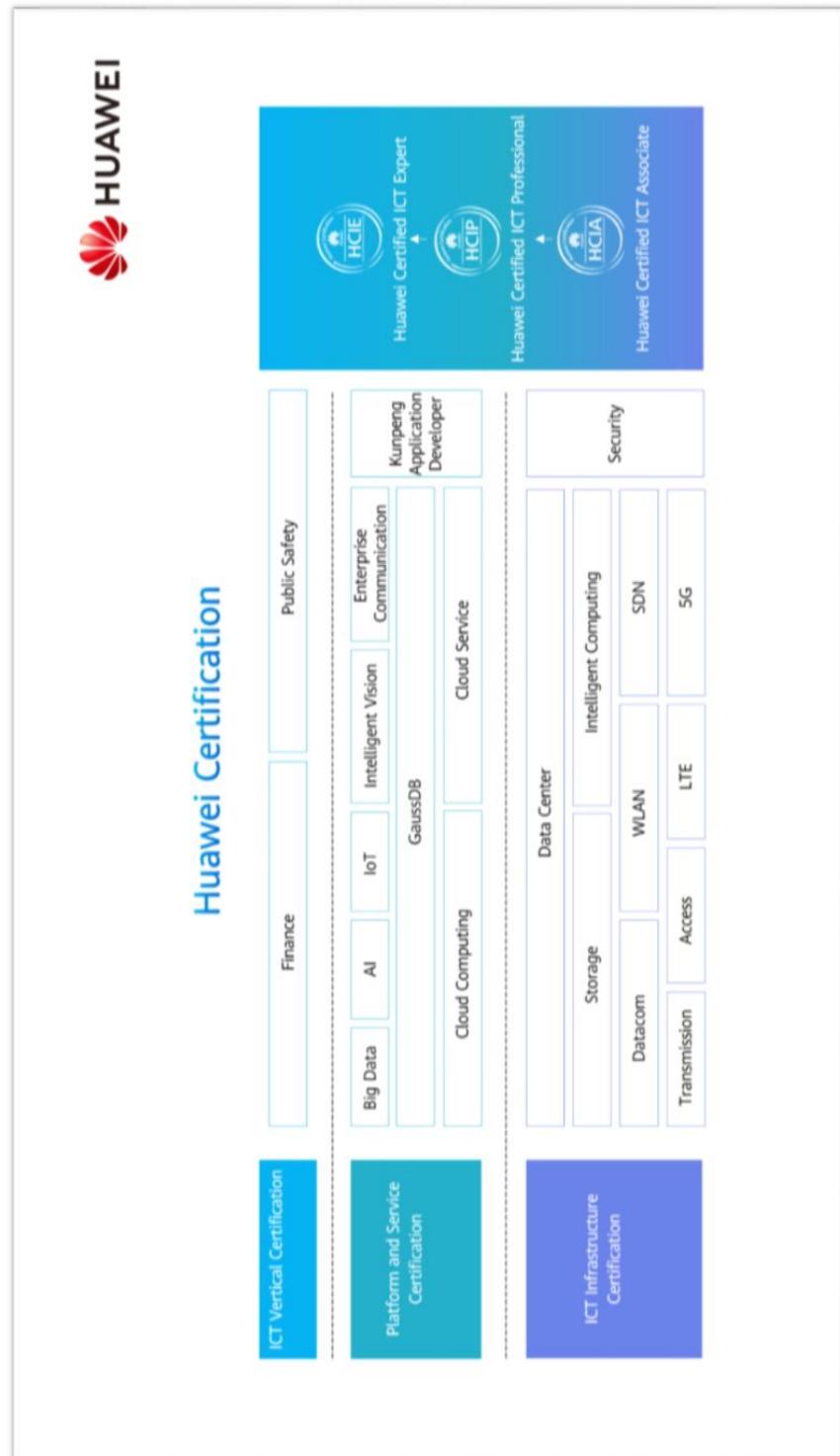
Huawei Certified ICT Professional (HCIP)

Huawei Certified ICT Expert (HCIE)

Huawei certification courses cover the entire ICT domain, with a focus on how today's architecture generates cloud-pipe-device synergy. The courses present the latest developments of all essential ICT aspects to foster a thriving ICT talent ecosystem for the digital age.

HCIA-openEuler is intended for frontline engineers at Huawei regional or representative offices, as well as other personnel who want to expand their knowledge of Linux technologies. The HCIA-openEuler certification includes the introduction to the openEuler OS, software installation, and basic operations, file processing and text editing, user and permission management, logical volume management, and other techniques and best practices aimed at getting the most out of openEuler.

Those who pass this certification will be able to install and use openEuler, as well as apply cutting-edge Linux knowledge.



About This Document

Overview

This document is developed for the training course for HCIA-openEuler certification. It is intended for trainees who are preparing for the HCIA-openEuler exam or readers who want to learn about the openEuler knowledge.

Description

This document consists of nine exercises. It describes how to install and configure the openEuler OS.

- Exercise 1: installing the openEuler OS
- Exercise 2: basic CLI operations on the openEuler OS
- Exercise 3: openEuler text editor
- Exercise 4: openEuler user and permission management
- Exercise 5: openEuler software management
- Exercise 6: openEuler storage and file system management
- Exercise 7: openEuler system and process management
- Exercise 8: shell script language basics
- Exercise 9: openEuler comprehensive practices

Readers' Knowledge Background

To better understand this course, the readers should be able to:

- Have basic computer operation knowledge and be familiar with computer hardware.

Lab Environment

Networking

This lab environment is intended for Linux system engineers who are preparing for the HCIA-openEuler exam. In the lab environment, PCs and virtualization software are used. Devices and VMs must have access to the Internet.

Device Configuration Requirements

To meet the HCIA-openEuler exercise requirements, it is recommended that the PC specifications be greater than or equal to the following:

Component	Specifications
CPU	4 cores
Memory	16 GB
Hard disks	The free space is greater than 100 GB.

Contents

About This Document	1
Overview	1
Description	1
Readers' Knowledge Background	1
Lab Environment.....	1
1 Installing the openEuler OS.....	1
1.1 Overview	1
1.1.1 About This Exercise.....	1
1.1.2 Objectives	1
1.1.3 Software Required for the Exercise.....	1
1.2 Configuring the Virtualization Environment.....	1
1.3 Installing the openEuler OS	2
1.3.1 Creating a VM	2
1.3.2 Installing the openEuler OS.....	7
1.3.3 Verification	12
1.4 Closing the Lab Environment	13
1.5 Quiz	14
2 Basic CLI Operations on the openEuler OS.....	15
2.1 Overview	15
2.1.1 About the Exercises.....	15
2.1.2 Objectives	15
2.2 Basic Operations of the bash Command	15
2.3 Viewing a File	20
2.4 Packaging and Compression Commands	24
2.5 Help Command	26
2.6 Quiz	28
3 openEuler Text Editor.....	29
3.1 Overview	29
3.1.1 About the Exercises.....	29
3.1.2 Objectives	29
3.2 Logging In to the Lab Environment	29
3.2.1 Logging In to the Lab Environment.....	29
3.3 Basic Vim Operations	30
3.3.1 Moving the Cursor by Pressing h , j , k , and l	30

3.3.2 Exiting Vim	31
3.3.3 Text Editing - Deletion	32
3.3.4 Text Editing - Insertion.....	33
3.3.5 Text Editing - Appending	34
3.3.6 Editing a File.....	35
3.4 Vim Operators and Actions	37
3.4.1 Deletion Commands.....	37
3.4.2 Deletion Commands de and d\$	38
3.4.3 Operators and Actions	38
3.4.4 Adding a Number Before an Action.....	39
3.4.5 Adding a Number Before a Deletion Command.....	40
3.4.6 Line Operators.....	40
3.4.7 Undo Operations.....	41
3.5 Vim Modification	42
3.5.1 Paste	42
3.5.2 Replace.....	43
3.5.3 Change	44
3.5.4 Other Changes	45
3.6 Vim Search and Replacement Operations	45
3.6.1 Cursor Position and File Status.....	45
3.6.2 Search	47
3.6.3 Matching Parentheses Search	48
3.6.4 Substitute.....	48
3.7 Vim File Operations.....	49
3.7.1 Running External Commands	49
3.7.2 More Operations on Writing Files.....	50
3.7.3 Selecting Text to Write.....	50
3.7.4 Retrieving File Content.....	52
3.8 Other Vim Operations.....	53
3.8.1 Open.....	53
3.8.2 Append	54
3.8.3 Another Replace Command.....	55
3.8.4 Copy and Paste Text.....	55
3.8.5 Setting Options	56
3.9 Using the Vim Help Document.....	58
3.9.1 Using the Vim Help Document	58
4 openEuler User and Permission Management.....	66
4.1 Overview	66
4.1.1 About This Exercise	66

4.1.2 Objectives	66
4.2 Configuring the Exercise	66
4.2.1 Procedure	66
4.3 Quiz	74

5 openEuler Software Management 75

5.1 Overview	75
5.1.1 About the Exercises.....	75
5.1.2 Objectives	75
5.2 Configuring the Yum Repository.....	75
5.3 Managing Software Through RPM	76
5.3.1 RPM Query Commands	76
5.3.2 RPM Installation Commands.....	76
5.3.3 RPM Upgrade Commands	77
5.3.4 Common RPM Options	77
5.3.5 RPM Uninstallation Command	77
5.4 Managing Software Packages Through DNF	77
5.4.1 Operations.....	77
5.5 Installing Personal Cloud Drive Application.....	78
5.5.1 Overview	78
5.5.2 Procedure	79
5.5.3 Installing Dependency Components.....	79
5.5.4 Installing the Apache Server	79
5.5.5 Installing PHP	79
5.5.6 Installing the Nextcloud Application	81
5.5.7 Result Verification	81

6 openEuler Storage and File System Management..... 83

6.1 Overview	83
6.1.1 About This Exercise	83
6.1.2 Objectives	83
6.2 Adding a Disk	83
6.3 Disk Partition Management in MBR Partition Table Mode	87
6.3.1 Creating a Primary Partition	87
6.3.2 Creating Extended and Logical Partitions	88
6.3.3 Changing the Type of a Partition	89
6.3.4 Deleting a Partition.....	91
6.4 Managing Disk Partitions in GPT Partition Table Mode	91
6.4.1 Using parted to Create a Partition in Interactive Mode	91
6.4.2 Creating a Partition in Non-interactive Mode.....	92
6.4.3 Deleting a Partition.....	93

6.5 Formatting and Mounting	93
6.5.1 Formatting a File System.....	93
6.5.2 Mounting a File System.....	94
6.5.3 Mounting an ISO File.....	94
6.5.4 Setting Automatic Disk Mounting Upon Startup.....	95
6.6 Logical Volume Management	95
6.6.1 Creating and Formatting a Logical Volume	95
6.6.2 Expanding and Reducing the Capacity of an LV	97
7 openEuler System and Process Management	100
7.1 Overview	100
7.1.1 About This Exercise.....	100
7.1.2 Objectives	100
7.2 Task Management	100
7.2.1 Temporary Task Management	100
7.2.2 Periodic Task Management.....	102
7.3 Network Management.....	102
7.3.1 Host Name Management	102
7.3.2 Network Management.....	103
7.4 Firewall Management	106
7.5 Service Management	108
7.5.1 Managing System Services.....	108
8 Shell Script Language Basics	110
8.1 Overview	110
8.1.1 About This Exercise.....	110
8.1.2 Objectives	110
8.2 Shell Variables.....	110
8.2.1 Definition of User Variables	110
8.2.2 Location Parameters	112
8.3 Special Characters in Shell	112
8.4 Condition Judgment and Loop Structure.....	113
8.4.1 if Statement.....	113
8.4.2 Test Statement.....	115
8.4.3 while Statement	116
8.4.4 for Statement.....	116
8.4.5 case Statement.....	117
8.5 Quiz	118
9 openEuler Comprehensive Practices	119
9.1 Overview	119

9.1.1 About This Exercise.....	119
9.1.2 Objectives	119
9.2 Configuring the Exercise	119
9.2.1 Configuration Roadmap	119
9.2.2 Procedure.....	119
9.3 Setting File Sharing Users and Permissions	122
9.3.1 Overview	122
9.3.2 Configuring the Exercise.....	122
9.4 O&M File Sharing Server	125
9.4.1 Overview	125
9.4.2 Configuring the Exercise.....	126

1 Installing the openEuler OS

1.1 Overview

1.1.1 About This Exercise

This document describes how to install the virtualization environment and the openEuler OS.

1.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Master how to set up the lab environment.
- Master the installation and deployment of the openEuler OS.

1.1.3 Software Required for the Exercise

Table 1-1 Software required for the openEuler exercise

Software Version	Download URL
openEuler-20.03-LTS-x86_64-dvd.iso	https://repo.openeuler.org/openEuler-20.03-LTS/ISO/x86_64/openEuler-20.03-LTS-x86_64-dvd.iso
VirtualBox-6.1.14-140239-Win.exe	https://download.virtualbox.org/virtualbox/6.1.14/VirtualBox-6.1.14-140239-Win.exe
putty.exe	https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe

1.2 Configuring the Virtualization Environment

Step 1 Enter the BIOS and enable the CPU virtualization technology. The method for enabling the technology varies according to the computer.

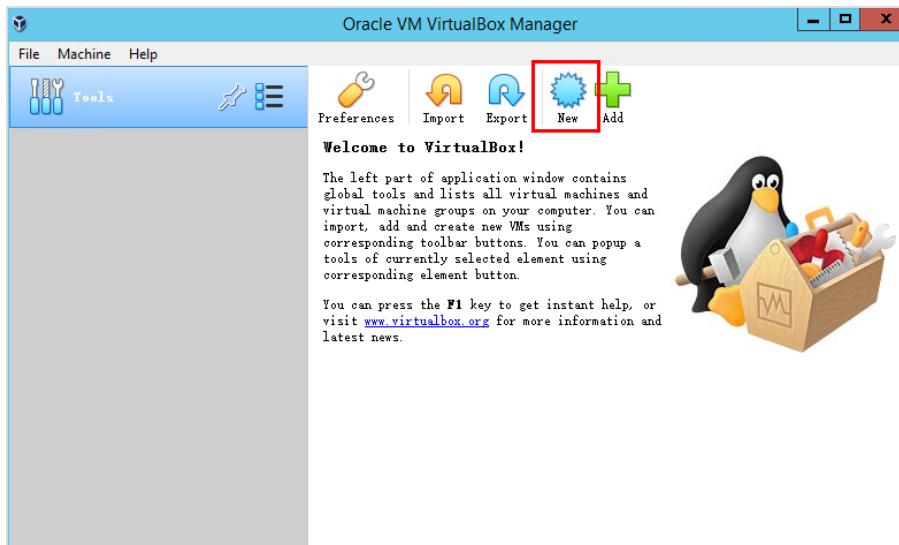
Step 2 Download and install VirtualBox.

----End

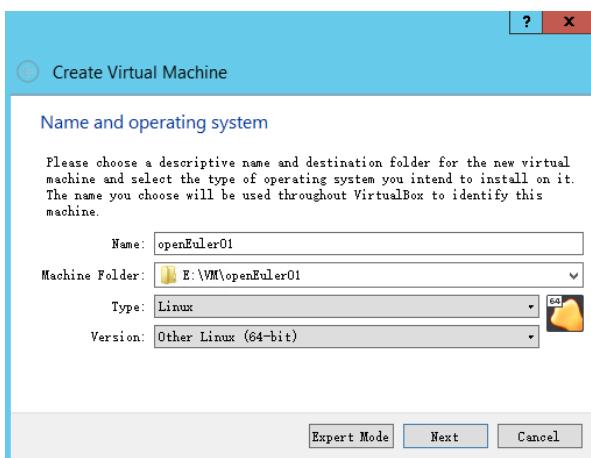
1.3 Installing the openEuler OS

1.3.1 Creating a VM

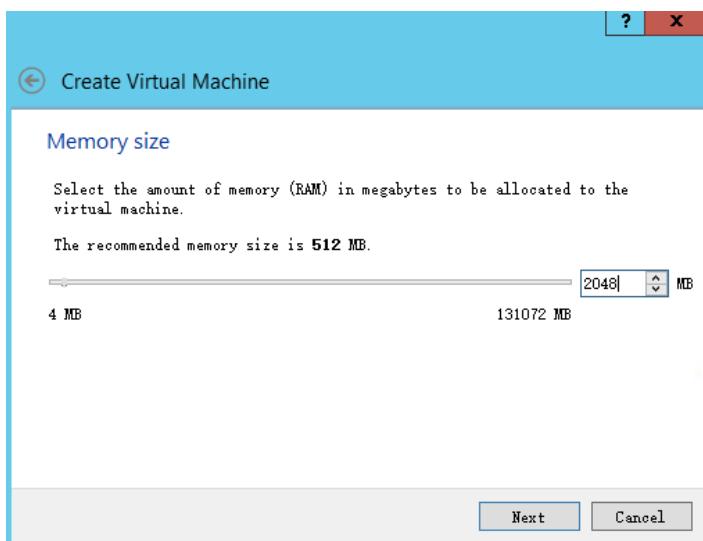
Step 1 Open VirtualBox and click **New** to open the wizard for creating a VM.



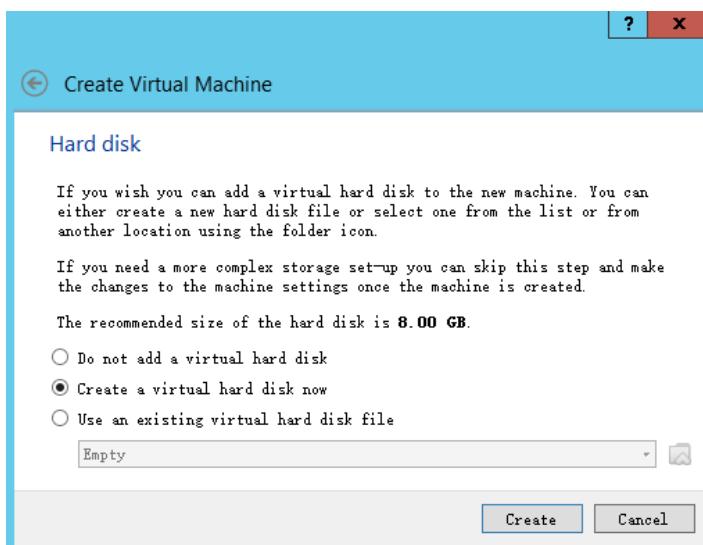
Step 2 Set the VM name, storage location of the VM configuration file, VM type, and version, and click **Next**.



Step 3 Set the memory size of the VM and click **Next**.

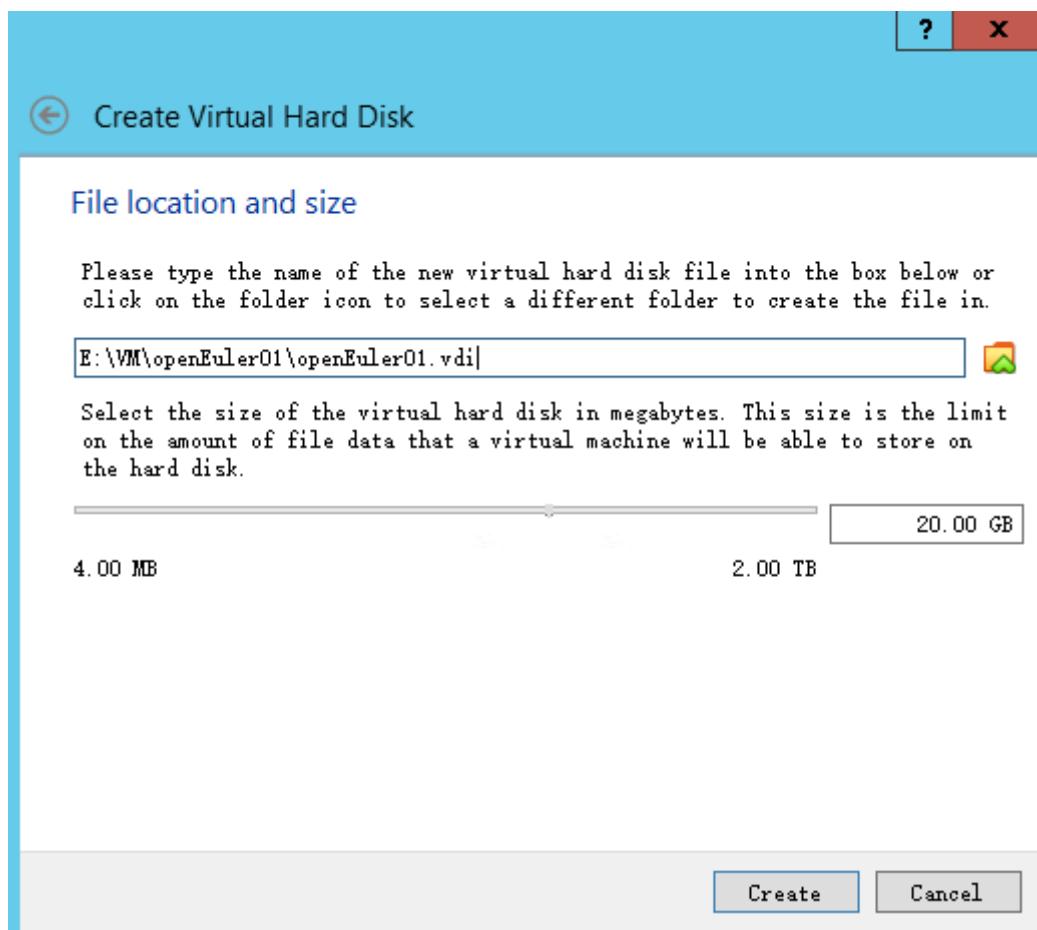


Step 4 Select **Create a virtual hard disk now** and click **Create**.



Step 5 In the **Create Virtual Hard Disk** dialog box, retain the default settings and click **Next** twice.

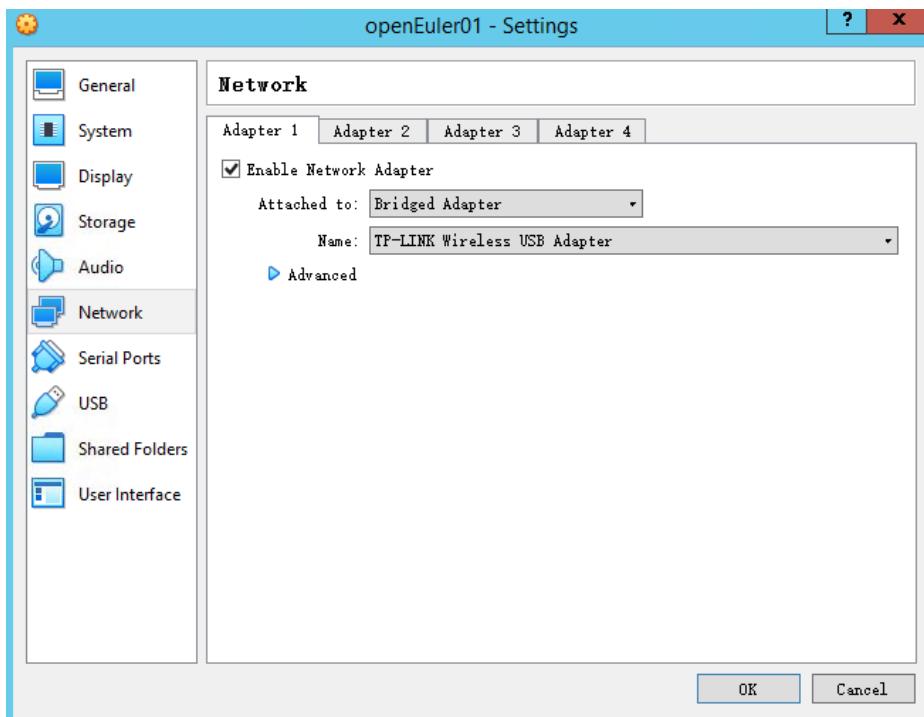
Step 6 In the **File location and size** area, set the storage location and size of the disk file, and click **Create**.



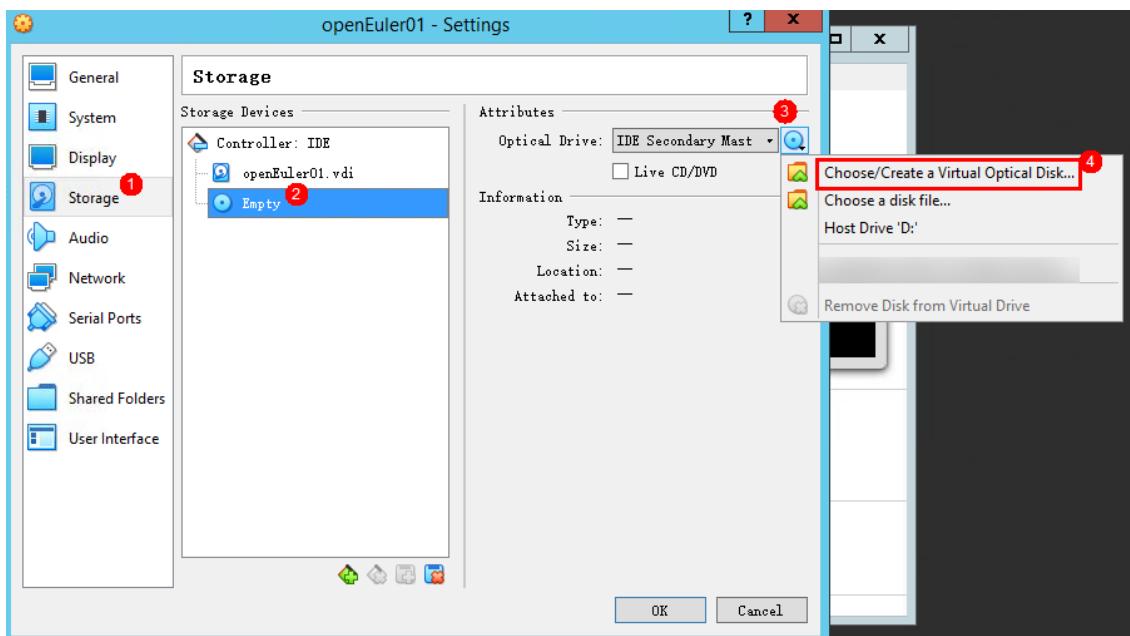
- Step 7 After the creation is complete, the VirtualBox automatically switches to the management page of the **openEuler01** VM. Click **Settings** to open the configuration page of the **openEuler01** VM.



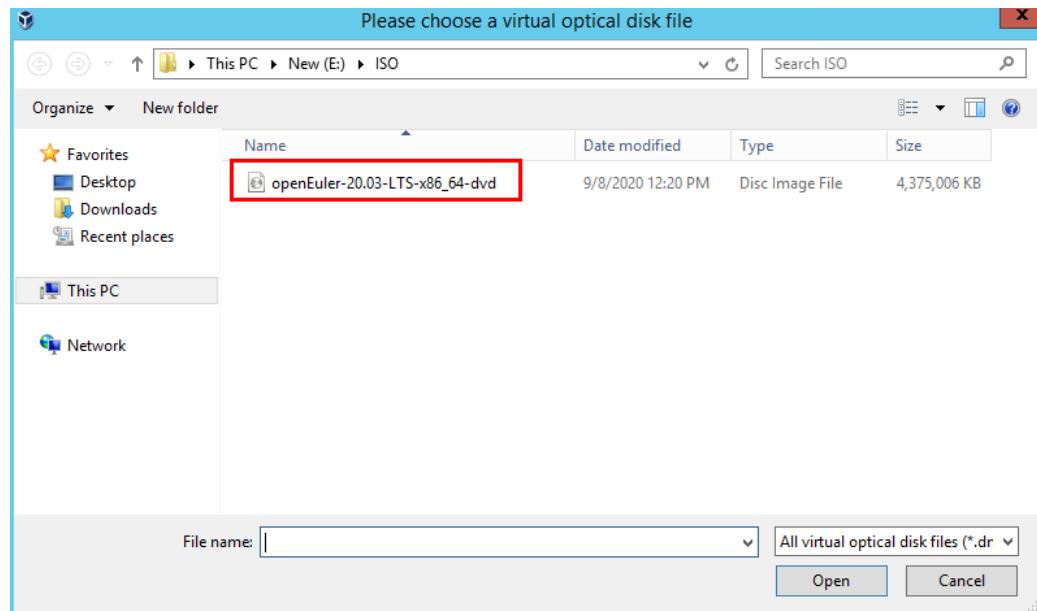
- Step 8 On the **Settings** page, click **Network**, set the connection mode of network adapter 1 to **Bridged Adapter**, and set the network adapter name to one that can access the Internet on the PC.



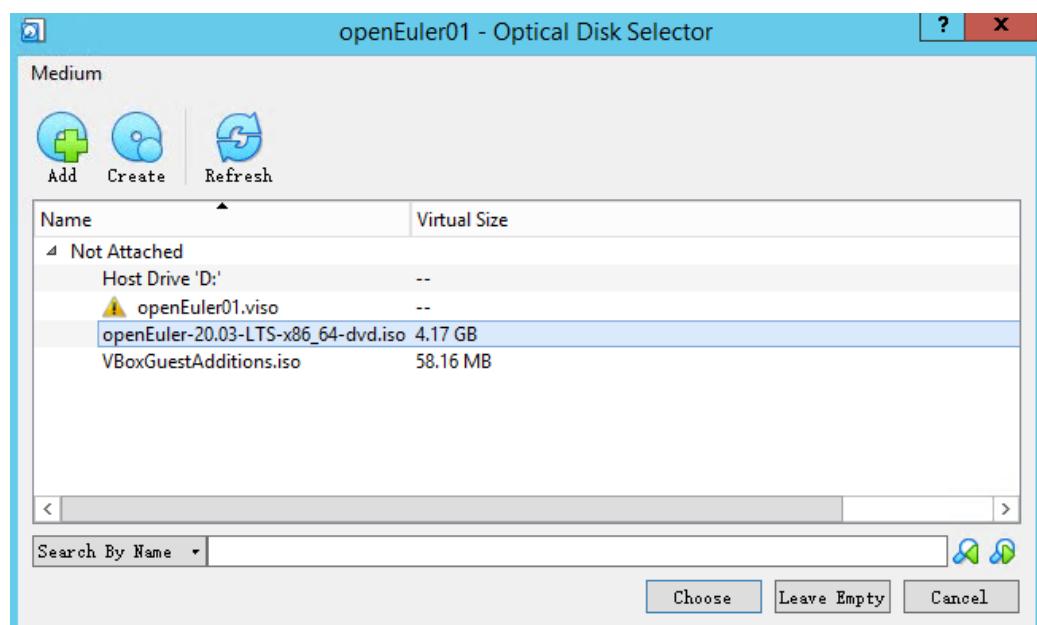
- Step 9 On the **Settings** page, click **Storage**. The storage setting page is displayed. Click the CD-ROM icon in the right pane, and then choose **Choose/Create a Virtual Optical Disk**.



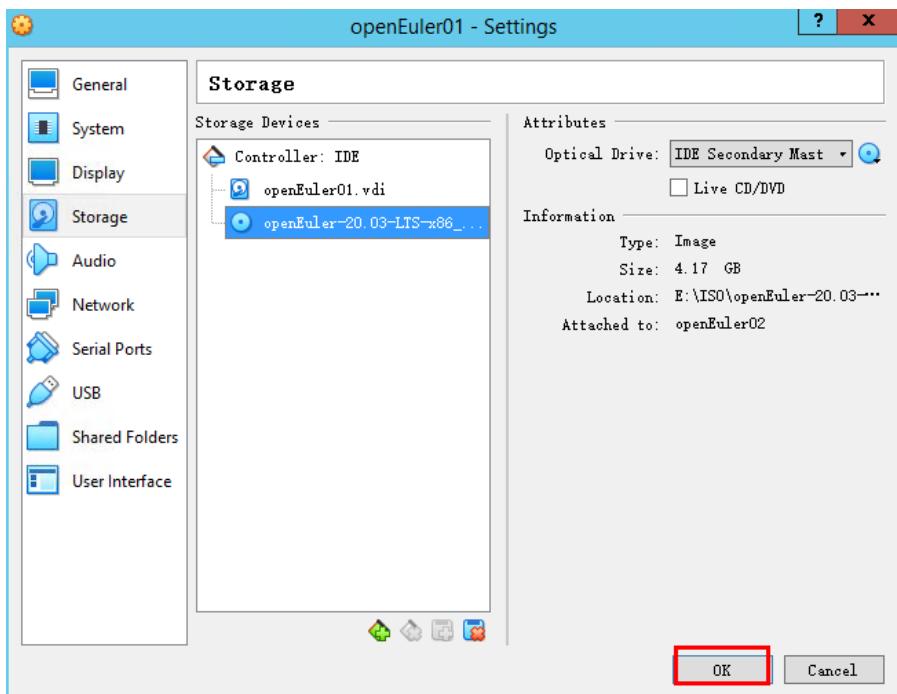
- Step 10 In the displayed window, click **Register**. In the displayed window, select the downloaded **openEuler-20.03-LTS-x86_64-dvd.iso** and click **Open**.



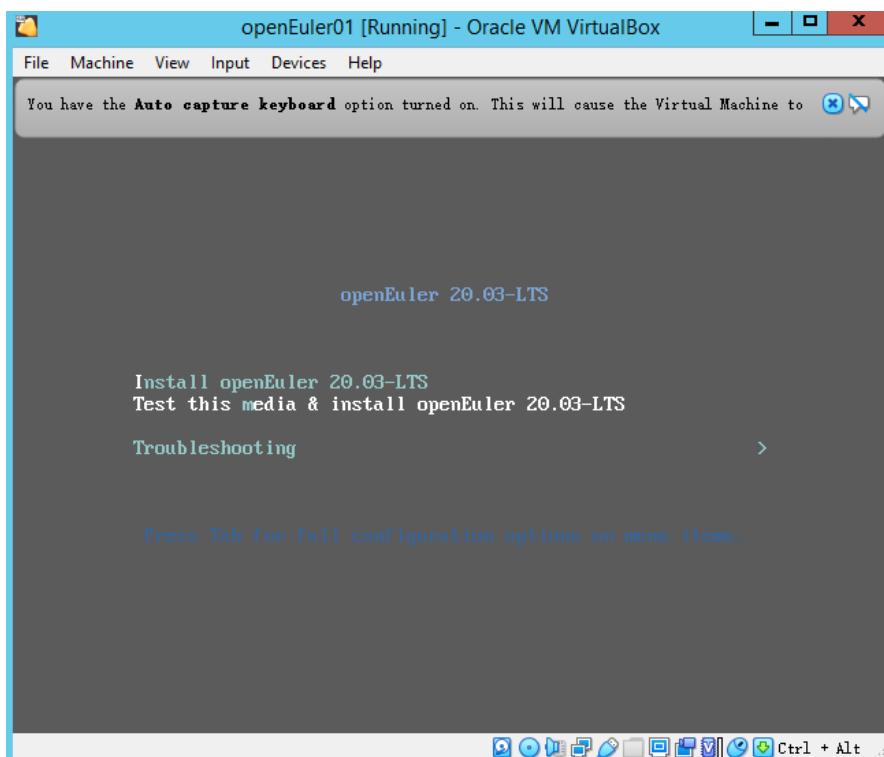
Step 11 On the **Optical Disk Selector** page, select the added openEuler ISO file and click **Choose**.



Step 12 On the storage setting page of the **openEuler01** VM, click **OK** to mount the CD-ROM.



Step 13 Click **Start** to start the **openEuler01** VM. The VM control window is displayed.



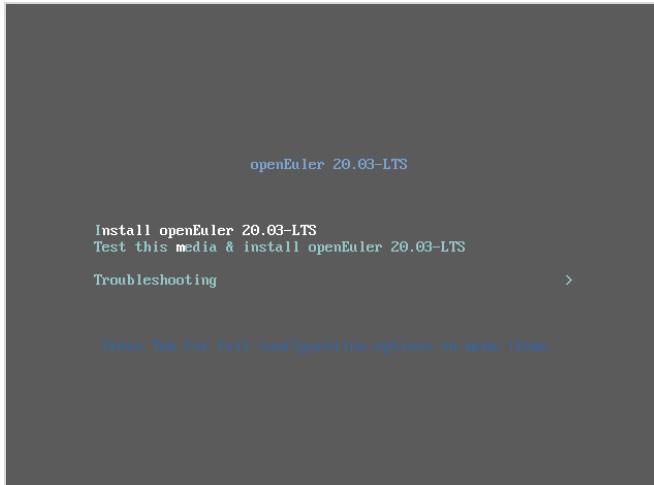
----End

1.3.2 Installing the openEuler OS

Step 1 Move the cursor to the console of the **openEuler01** VM and left-click to enable VirtualBox to capture the cursor. In the displayed dialog box, select **Do not display this message**

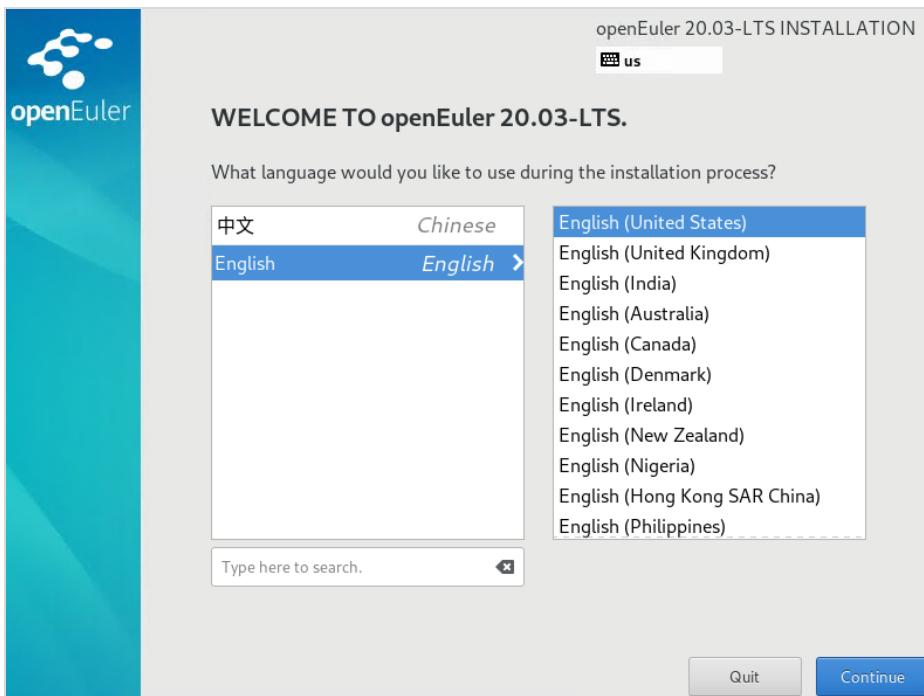
again and click **Capture**. Remember to cancel the combination keys that are exclusively used by the mouse and keyboard. The default combination key is **Right Ctrl**.

- Step 2 Press the up or down arrow key to select **Install openEuler 20.03-LTS** and press **Enter** to install openEuler.

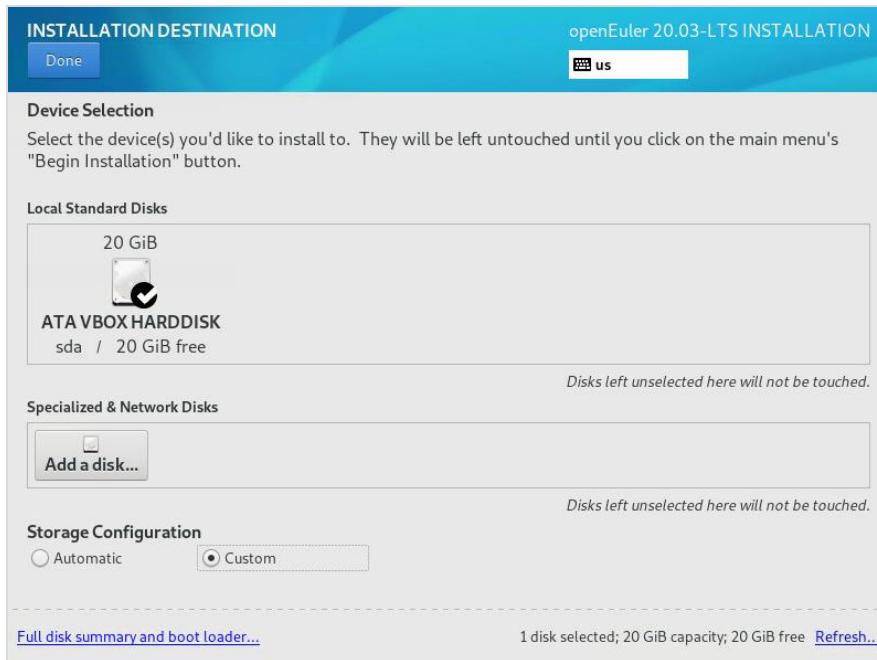


- Step 3 Wait until the installation page is displayed.

- Step 4 Select **English > English (United States)** as the installation language and click **Continue**.

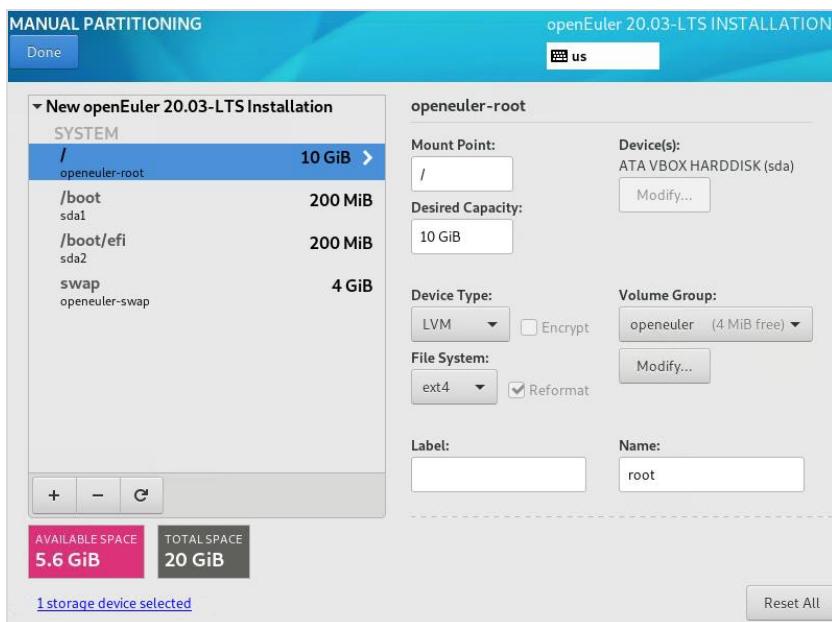


- Step 5 On the **INSTALLATION SUMMARY** page, click **Installation Destination** and set the installation disk and partition of the OS. You are advised to select **sda** as the local standard disk, set **Storage Configuration** to **Custom**, and click **Done**. The storage space configuration page is displayed.



Step 6 Click + in the lower left corner to add a partition, configure the partition according to the following table, and click **Add mount point**.

Mount Point	Desired Capacity
/boot/efi	200 MiB
/boot	200 MiB
/swap	4 GiB
/	10 GiB

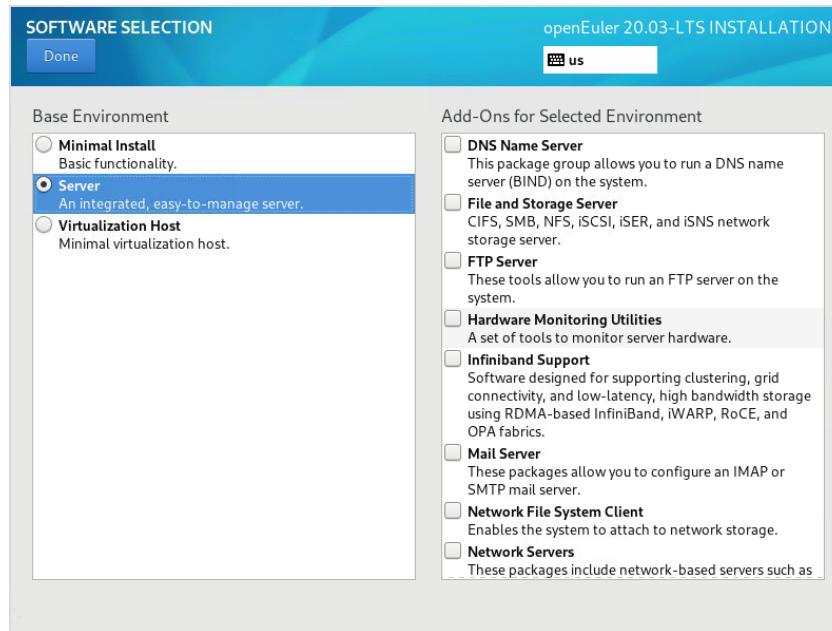


After the configuration is complete, click **Done** in the upper left corner. In the displayed dialog box, click **Accept Changes**.

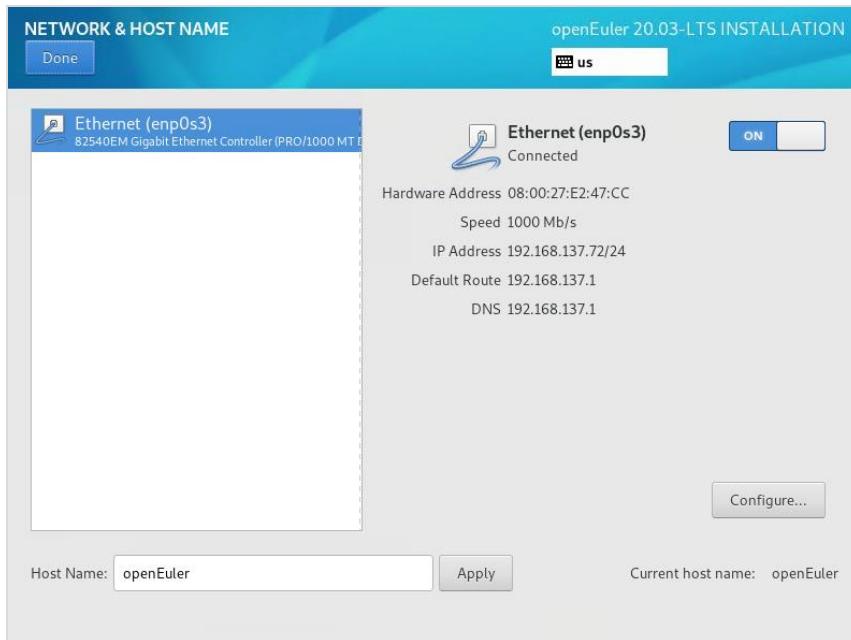
SUMMARY OF CHANGES				
Your customizations will result in the following changes taking effect after you return to the main menu and begin installation:				
Order	Action	Type	Device	Mount point
1	Destroy Format	Unknown	ATA VBOX HARDDISK (sda)	
2	Create Format	partition table (MSDOS)	ATA VBOX HARDDISK (sda)	
3	Create Device	partition	sda1 on ATA VBOX HARDDISK	
4	Create Device	partition	sda2 on ATA VBOX HARDDISK	
5	Create Device	partition	sda3 on ATA VBOX HARDDISK	
6	Create Format	physical volume (LVM)	sda3 on ATA VBOX HARDDISK	
7	Create Device	lvmyg	openeuler	
8	Create Device	lvmlv	openeuler-root	/
9	Create Format	ext4	openeuler-root	/
10	Create Device	lvmlv	openeuler-swap	
11	Create Format	swap	openeuler-swap	
12	Create Format	EFI System Partition	sda2 on ATA VBOX HARDDISK	/boot/efi

[Cancel & Return to Custom Partitioning](#) [Accept Changes](#)

- Step 7 On the **INSTALLATION SUMMARY** page, click **Software Selection** and specify the software package to be installed. Select **Server**. After confirming that the information is correct, click **Done**.



- Step 8 Click **Network & Host Name**. The **Network & Host Name** page is displayed. Change the host name in the lower left corner to **openEuler** and click **Apply**. Enable the network adapter. (Note: The network adapter must be set to the bridge mode in the network settings of the **openEuler01** VM on VirtualBox. Connect the network adapter to a network adapter that has access to the Internet, for example, a wireless network adapter. For details, see Step 8 in section 1.3.1.)

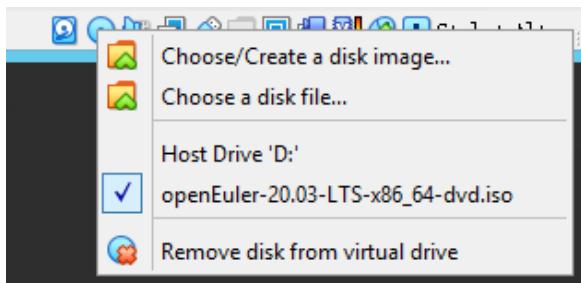


Step 9 Click **Time & Date**. The time configuration page is displayed. Verify that the current time zone corresponds to **Shanghai** and **Network Time** is **OFF**, and click **Done**.



- Step 10 After the preceding configuration is complete, click **Begin Installation** in the lower right corner to start installing the OS.
- Step 11 During the OS installation, click **Root Password** and set a password for the **root** user. The password must contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters. Then, click **Done** in the upper left corner.
- Step 12 During the OS installation, click **User Creation** to add a common user for the system. Set the username and password (the password cannot be the same as the username and must meet the complexity requirements), and click **Done** in the upper left corner.

- Step 13 After the installation is complete, click **Reboot** in the lower right corner to restart the system. Right-click the CD-ROM icon under VirtualBox and choose **Remove disk from virtual drive** from the shortcut menu. (If the installation page is displayed, click **Control > Restart** in the upper left corner.)



----End

1.3.3 Verification

- Step 1 After the system has restarted, log in to the system as the **root** user. (The system does not display any message when you enter the password. Ensure that the entered password is correct.)

```
openEuler 20.03 (LTS)
Kernel 4.19.90-2003.4.0.0036.oe1.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

openEuler login: root
Password:

Authorized users only. All activities may be monitored and reported.

Welcome to 4.19.90-2003.4.0.0036.oe1.x86_64

System information as of time: Tue Sep 8 14:38:29 CST 2020

System load: 0.18
Processes: 97
Memory used: 12.1%
Swap used: 0.0%
Usage On: 28%
IP address: 192.168.137.52
Users online: 1

[root@openEuler ~]# _
```

- Step 2 Run the following command to switch the user and check whether the user is successfully created:

```
[root@openEuler ~]# su - username # username is a common user created during system installation.
Welcome to 4.19.90-2003.4.0.0036.oe1.x86_64

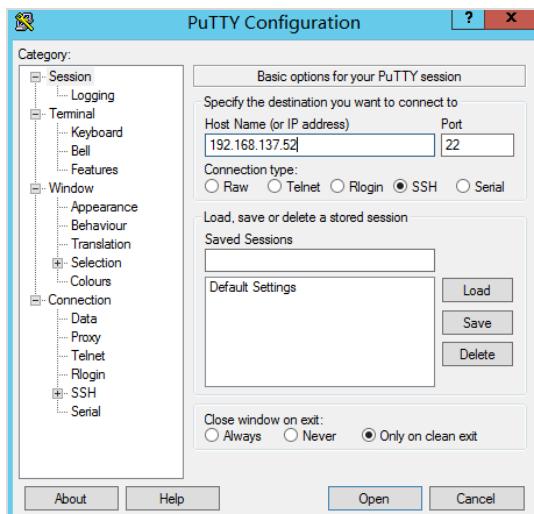
System information as of time: Tue Sep 8 14:47:14 CST 2020

System load: 0.06
Processes: 97
Memory used: 12.3%
Swap used: 0.0%
Usage On: 28%
IP address: 192.168.137.52
```

Users online: 1

```
[username@openEuler ~]$      # If information similar to this is displayed, the login is successful.  
[username@openEuler ~]$ exit  
logout  
[root@openEuler ~]#
```

- Step 3 Open PuTTY, enter the IP address displayed during login in the **Host Name (or IP address)** text box, for example, **192.168.137.52**, and click **Open**. If a warning dialog box is displayed, click **Yes**. Enter the username and password to log in to the system. For details, see Step 1.

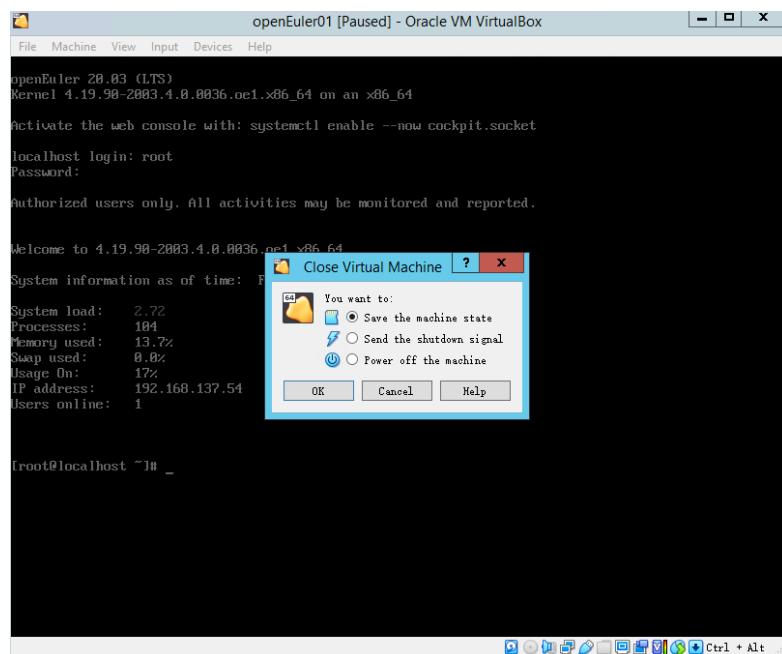


- Step 4 Enter the username **root** and the password to log in to the system.

----End

1.4 Closing the Lab Environment

- Step 1 Close the VirtualBox window of the **openEuler01** VM. In the dialog box that is displayed, select **Save the machine state** and click **OK**.



----End

1.5 Quiz

Which partitions must be configured during openEuler installation?

2

Basic CLI Operations on the openEuler OS

2.1 Overview

2.1.1 About the Exercises

The exercises introduce bash commands and file management commands on the openEuler OS.

2.1.2 Objectives

Upon completion of the exercises, you will be able to:

- Master the basic operations of the bash command.
- Master the basic operations of file management commands.

2.2 Basic Operations of the bash Command

Step 1 Open VirtualBox, start the **openEuler** VM, and log in to the VM as the **root** user.

Step 2 Execute the basic bash commands.

Run the **reboot** command to restart the openEuler OS.

```
[root@localhost ~]# reboot
```

After the restart, log in to the openEuler OS as the **root** user.

Run the **logout** or **exit** command to log out of the system.

```
[root@localhost ~]# logout  
# Log in to the openEuler OS as the root user again.  
[root@localhost ~]# su - openeuler      # Switch to another user.  
[openeuler@localhost ~]# exit          # Exit the current user and switch back to the root user.  
[root@localhost ~]#
```

You can also run the **exit** command to log out of the system. However, if you switch to another user frequently, you are advised to run the **exit** command to log out of the system each time you switch to another user.

----End

2.2.1.2 Basic Directory and File Operations

Step 1 Run the **pwd** command to view the current directory.

```
[root@localhost ~]# pwd  
/root  
[root@localhost ~]#  
# The command output indicates that the current directory is the /root directory.
```

Step 2 Run the **ls** command.

The **ls** command is used to view the files and folders in the current directory.

```
[root@localhost ~]# ls  
anaconda-ks.cfg  
[root@localhost ~]# ls .  
anaconda-ks.cfg  
# The command output indicates that the anaconda-ks.cfg file exists in the current directory.
```

Run the following command to display the files and folders in the upper-level directory:

```
[root@localhost ~]# ls ..  
bin dev home lib64 media opt root sbin sys usr  
boot etc lib lost+found mnt proc run srv tmp var
```

Run the following command to view the files and folders in the **/tmp** directory:

```
[root@localhost ~]# ls /tmp  
systemd-private-92622a8f3c5b45d6b45c4cc9012916e6-chronyd.service-6X7mn1  
systemd-private-92622a8f3c5b45d6b45c4cc9012916e6-systemd-logind.service-GKj4CO
```

Run the following command to display all files and folders in the current directory:

```
[root@localhost ~]# ls -a  
. anaconda-ks.cfg .bash_logout .bashrc .tcshrc  
.. .bash_history .bash_profile .cshrc  
# The command output indicates that the current directory contains hidden files and directories.
```

Run the following command to display the detailed information about the files and folders that are not hidden in the current directory:

```
[root@localhost ~]#ls -l  
total 4  
-rw----- 1 root root 1986 Jul 8 11:07 anaconda-ks.cfg
```

Run the following command to display details about all files and folders in the current directory:

```
[root@localhost ~]#ls -al  
total 36  
dr-xr-x--. 2 root root 4096 Jul 8 11:38 .  
dr-xr-xr-x. 18 root root 4096 Jul 8 11:00 ..  
-rw----- 1 root root 1986 Jul 8 11:07 anaconda-ks.cfg  

```

```
-rw-r--r--. 1 root root 18 Oct 29 2019 .bash_logout
-rw-r--r--. 1 root root 176 Oct 29 2019 .bash_profile
-rw-r--r--. 1 root root 176 Oct 29 2019 .bashrc
-rw-r--r--. 1 root root 100 Oct 29 2019 .cshrc
-rw-r--r--. 1 root root 129 Oct 29 2019 .tcshrc
```

Step 3 Run the **cd** command to switch to another directory.

Switch to the system root directory.

```
[root@localhost ~]#cd /
[root@localhost /]#
# Note that ~ changes to /.
```

Run the following command to switch to the **/etc/** directory:

```
[root@localhost /]#cd /etc
[root@localhost etc]#
```

Use the relative path to switch to the **/etc/sysconfig/** directory.

```
[root@localhost etc]#cd sysconfig
[root@localhost sysconfig]#
```

Use the absolute path method to switch to the **/etc/sysconfig/** directory.

```
[root@localhost etc]#cd /etc/sysconfig
[root@localhost sysconfig]#
```

Run the **cd ..** command to switch to the upper-level directory:

```
[root@localhost sysconfig]# cd ..
[root@localhost etc]#
```

Run the **cd** command to switch to the home directory of the user.

```
[root@localhost sysconfig]#cd
[root@localhost ~]#
```

Run the **cd -** command to return to the previous directory.

```
[root@localhost sysconfig]# cd -
/etc
[root@localhost etc]#
```

Run the **cd ~** command to switch to the home directory of the user.

```
[root@localhost etc]#cd /etc/sysconfig
[root@localhost sysconfig]#cd ~
[root@localhost ~]#
```

Step 4 Run the **mkdir** command to create a directory.

Create a **test1** directory in the current folder.

```
[root@localhost ~]#mkdir /root/test1  
[root@localhost ~]#ls  
anaconda-ks.cfg  test1
```

Create a directory using a relative path.

```
[root@localhost ~]#mkdir ./test2  
[root@localhost ~]#ls  
anaconda-ks.cfg  test1 test2
```

Create a directory using an absolute path.

```
[root@localhost ~]#mkdir test3  
[root@localhost ~]#ls  
anaconda-ks.cfg  test1 test2 test3
```

Step 5 Run the **touch** command to create a file.

Create a **huawei.txt** file.

```
[root@localhost ~]#cd test1  
[root@localhost test1]#touch /root/huawei.txt  
[root@localhost test1]#touch huawei1.txt  
[root@localhost test1]#ls  
huawei.txt  huawei1.txt
```

Step 6 Run the **cp** command.

Copy **huawei.txt** to the **/root/test2** directory and name it **huawei.txt.bak**.

```
[root@localhost test1]#cp huawei.txt /root/test2/huawei.txt.bak  
[root@localhost test1]#ls /root/test2  
huawei.txt.bak
```

Copy the **text1** directory to the **/root/test2** directory.

```
[root@localhost test1]#cp -r /root/test1 /root/test2/  
[root@localhost test1]#ls /root/test2/  
huawei.txt.bak test1
```

Step 7 Run the **rm** command.

Delete the **huawei.txt** file from the **/root/test1** directory.

```
[root@localhost test1]#rm huawei.txt  
rm: Are you sure you want to delete the common empty file huawei.txt? y      // Enter y to confirm the deletion.  
[root@localhost test1]#ls  
[root@localhost test1]#  
[root@localhost ~]# touch /root/huawei1.txt  
[root@localhost ~]# rm -f /root/huawei1.txt
```

Delete the **test1** folder in the **/root** directory.

```
[root@localhost test1]#cd
```

```
[root@localhost ~]#ls
anaconda-ks.cfg  test1  test2  test3
[root@localhost ~]#rmdir /root/test1
[root@localhost ~]#ls
anaconda-ks.cfg  test2  test3
[root@localhost ~]#mkdir /root/test1
[root@localhost ~]# rm -r /root/test1
rm: remove directory '/root/test1'? y
```

Step 8 Run the **mv** command.

Move the **huawei.txt.bak** file from the **/root/test2** directory to the **/root** directory and rename the file to **huawei.txt**.

```
[root@localhost ~]#mv /root/test2/huawei.txt.bak  ~/huawei.txt
[root@localhost ~]#ls
anaconda-ks.cfg  huawei.txt  test2  test3
```

Step 9 Run the **ln** command.

Create a hard link from **huawei.txt** to **/test3** and name it **huawei1.txt**.

```
[root@localhost ~]#ln huawei.txt /root/test3/huawei1.txt
```

Create a soft link from **huawei.txt** to **/test3** and name it **huawei2.txt**.

```
[root@localhost ~]#ln -s huawei.txt /root/test3/huawei2.txt
```

Check the **inode** information in the file. The node information in the **huawei.txt** file is the same as that in the **huawei1.txt** file. The node information in the **huawei.txt** file is the same as that in the **huawei2.txt** file.

```
[root@localhost ~]#ls -li
798457 -rw----- 1 root root 1631 June 9 16:40 anaconda-ks.cfg
798572 -rw----- 2 root root 0 June 10 10:20 huawei.txt
[root@localhost ~]#cd test3/
[root@localhost test3]#ls -li
798572 -rw----- 2 root root 0 June 10 10:20 huawei1.txt
798551 lrwxrwxrwx. 1 root root 10 June 10 11:37 huawei2.txt -> huawei.txt
```

Delete the **huawei.txt** file and view the file content again.

```
[root@localhost test3]# rm /root/huawei.txt      # Delete the source file huawei.txt.
rm: Are you sure you want to delete the common empty file /root/huawei.txt? y
[root@localhost test3]# ls
huawei1.txt  huawei2.txt
[root@localhost test3]# cat huawei1.txt      # The hard link file huawei1.txt can be opened
properly.
[root@localhost test3]# cat huawei2.txt      # Failed to open the soft link file huawei2.txt.
cat: huawei2.txt: The file or directory does not exist.
```

----End

2.3 Viewing a File

Step 1 Copy the **/etc/passwd** file to the **/root** directory.

```
[root@localhost test3]# cd  
[root@localhost ~]# cp /etc/passwd ~
```

Step 2 Run the **cat** command.

The **cat** command is used to view the content of the **passwd** file.

```
[root@localhost ~]# cat passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin  
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/sbin/nologin  
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin  
systemd-timesync:x:998:995:systemd Time Synchronization:/sbin/nologin  
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin  
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin  
polkitd:x:996:993:User for polkitd:/sbin/nologin  
saslauthd:x:995:76:Saslauthd user:/run/saslauthd:/sbin/nologin  
rpcbind:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin  
libstoragemgmt:x:994:991:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin  
pcp:x:993:990:PCP:/var/lib/pcp:/sbin/nologin  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin  
dnsmasq:x:988:988:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin  
radvd:x:75:75:radvd user:/sbin/nologin  
sanlock:x:179:179:sanlock:/var/run/sanlock:/sbin/nologin  
qemu:x:107:107:qemu user:/sbin/nologin  
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin  
dhcpd:x:177:177:DHCP server:/sbin/nologin  
named:x:25:25:Named:/var/named:/bin/false  
gluster:x:987:985:GlusterFS daemons:/run/gluster:/sbin/nologin  
setroubleshoot:x:986:984::/var/lib/setroubleshoot:/sbin/nologin  
geoclue:x:985:983:User for geoclue:/var/lib/geoclue:/sbin/nologin  
cockpit-ws:x:984:982:User for cockpit-ws:/sbin/nologin  
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM services:/var/lib/Pegasus:/sbin/nologin  
chrony:x:983:981::/var/lib/chrony:/sbin/nologin  
pcpqqa:x:982:980:PCP Quality Assurance:/var/lib/pcp/testsuite:/bin/bash  
pesign:x:981:979:Group for the pesign signing daemon:/var/run/pesign:/sbin/nologin  
postfix:x:89:89::/var/spool/postfix:/sbin/nologin  
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin
```

```
tcpdump:x:72:72:::/sbin/nologin  
dbus:x:978:978:System Message Bus:/usr/sbin/nologin  
openeuler:x:1000:1000:openEuler:/home/openeuler:/bin/bash
```

Step 3 Run the **head** command.

Run the following **head** command to view the first 10 lines of the file:

```
[root@localhost ~]# head passwd # If no option is added to the head command, the  
first 10 lines of the file are displayed by default.  
root:x:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin.sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin
```

Run the following **head** command to view the first 5 lines of the file:

```
[root@localhost ~]# head -n 5 passwd  
root:x:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

Run the following **head** command to view all content except the last 40 lines of the file:

```
[root@localhost ~]# head -n -40 passwd  
root:x:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin.sync
```

Run the following **head** command to view the first 10 bytes of the file:

```
[root@localhost ~]# head -c 10 passwd  
root:x:0:root@localhost ~]#
```

Question: How do I view all contents except the last 100 bytes in a file?

Step 4 Run the **tail** command.

Run the following **tail** command to view the last 10 lines of the file:

```
[root@localhost ~]# tail passwd # Similar to the head command, by default,  
the last 10 lines are displayed.  
cockpit-ws:x:984:982:User for cockpit-ws:/sbin/nologin  
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM services:/var/lib/Pegasus:/sbin/nologin
```

```
chrony:x:983:981::/var/lib/chrony:/sbin/nologin
pcpqqa:x:982:980:PCP Quality Assurance:/var/lib/pcp/testsuite:/bin/bash
pesign:x:981:979:Group for the pesign signing daemon:/var/run/pesign:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
dbus:x:978:978:System Message Bus:/usr/sbin/nologin
openeuler:x:1000:1000:openEuler:/home/openeuler:/bin/bash
```

Run the following **tail** command to view the last 5 lines of the file:

```
[root@localhost ~]# tail -n 5 passwd
```

Run the following **tail** command to view all the content except the first 20 lines of the file:

```
[root@localhost ~]# tail -n -20 passwd
```

Step 5 Run the **less** command.

Run the **less** command to view the file, press the up or down arrow key to go to a line, press the space bar to go to the next page, and press **Q** to exit.

```
[root@localhost ~]# less passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
...
```

Step 6 Run the **more** command.

Run the **more** command to view the file. Press the space bar to go to the next page until you exit. You can also press **Q** to exit.

```
[root@localhost ~]# more passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
...
```

----End

2.3.1.2 Search Commands

Step 1 Run the **find** command.

Search for the file named **passwd** in the **/etc** directory.

```
[root@localhost ~]# find /etc -name passwd
```

```
/etc/raddb/mods-enabled/passwd  
/etc/raddb/mods-available/passwd  
/etc/passwd  
/etc/pam.d/passwd
```

```
[root@localhost ~]# find /root -mtime -2  
/root  
/root/test2  
/root/test2/test1  
/root/test2/test1/huawei.txt  
/root/.bash_history  
/root/passwd  
/root/test3  
/root/test3/huawei1.txt  
/root/test3/huawei2.txt  
/root/anaconda-ks.cfg
```

Search the **/root/** directory for the files that belong to the **root** user.

```
[root@localhost ~]# find /root -user root  
/root  
/root/.bashrc  
/root/.bash_profile  
/root/test2  
/root/test2/test1  
/root/test2/test1/huawei.txt  
/root/.bash_logout  
/root/.bash_history  
/root/.tcshrc  
/root/passwd  
/root/.cshrc  
/root/test3  
/root/test3/huawei1.txt  
/root/test3/huawei2.txt  
/root/anaconda-ks.cfg
```

Search for files larger than 512 KB in the **/etc/** directory.

```
[root@localhost ~]# find /etc -size +512k  
/etc/services  
/etc/ssh/moduli  
/etc/brltty/Contraction/zh-tw.ctb  
/etc/selinux/targeted/policy/policy.31  
/etc/udev/hwdb.bin
```

Step 2 Run the **which** command.

View the absolute path of the **pwd** command. The **which** command is used to search for executable files based on the directory in the *PATH* variable configured by the user. Therefore, the commands found vary according to the *PATH* configuration.

```
[root@localhost ~]# which pwd  
/usr/bin/pwd
```

Step 3 Run the **whereis** command.

This command can only be used to find binary files, source code files, and manual pages.

Run the **whereis** command to view the location of the **bash** command.

```
[root@localhost ~]# whereis bash  
bash: /usr/bin/bash
```

----End

2.4 Packaging and Compression Commands

Step 1 Run the **zip** command.

The **zip** command is used to create a .zip package.

In the first line of the command, the **-r** option indicates that all contents in subdirectories are packaged recursively, the **-q** option indicates that no information is output to the screen, and the **-o** option indicates that a file is output and the file name must be followed. The augment indicating the object to be packaged can be a file or a directory.

```
[root@localhost ~]# zip -r -q -o passwd.zip passwd  
[root@localhost ~]# ls  
anaconda-ks.cfg  passwd  passwd.zip
```

Compress files by level.

If the compression level is 9 or 1 (9 indicates the highest compression level and 1 indicates the lowest compression level), re-packaging is required.

```
[root@localhost ~]# zip -r -9 -q -o passwd1.zip passwd  
[root@localhost ~]# zip -r -1 -q -o passwd2.zip passwd  
[root@localhost ~]# ls -lh  
-rw----- 1 root root 1.6K June 9 16:40 anaconda-ks.cfg  
-rw----- 1 root root 2.5K June 10 16:35 passwd  
-rw----- 1 root root 1.2K June 10 16:35 passwd1.zip  
-rw----- 1 root root 1.3K June 10 16:35 passwd2.zip  
-rw----- 1 root root 1.2K June 10 16:35 passwd.zip
```

Step 2 Run the **unzip** command.

The **unzip** command is used to decompress the package to the current directory.

```
[root@localhost ~]# unzip passwd.zip  
Archive:  passwd.zip  
replace passwd? [y]es, [n]o, [A]ll, [N]one, [r]ename: y  
    inflating: passwd  
[root@localhost ~]# ls  
anaconda-ks.cfg  passwd  passwd1.zip  passwd2.zip  passwd.zip  test2  test3  
[root@localhost ~]#
```

Run the **unzip** command to decompress the package to the specified directory.

Decompress the **password1.zip** file to the **/root/test3** directory. If the file already exists, do not overwrite the original file.

```
[root@localhost ~]# unzip -n passwd1.zip -d /root/test3
Archive: passwd1.zip
  inflating: /root/test3/passwd
[root@localhost ~]# ls /root/test3
huawei1.txt  huawei2.txt  passwd
[root@localhost ~]#
```

Decompress **passwd2.zip** in the **/root/test3** directory. If the file already exists, overwrite the original file.

```
[root@localhost ~]# unzip -o passwd2.zip -d /root/test3
Archive: passwd2.zip
  inflating: /root/test3/passwd
[root@localhost ~]# ls /root/test3
huawei1.txt  huawei2.txt  passwd
[root@localhost ~]#
```

Step 3 Run the **tar** command.

Package all the files in the **/root/test3** directory.

In the **tar** command, **-c** indicates to create a .tar package file and **-f** indicates the name of the file to be created. Note that the file name must follow **-f**. You can also add the **-v** option to output the packaged file in a visual manner. The slash (/) that indicates the absolute path is automatically removed. You can also use **-P** to retain the absolute path character.

```
[root@localhost ~]# cd test3
[root@localhost test3]# ls
huawei1.txt  huawei2.txt  passwd
[root@localhost test3]# tar -cf tartest.tar *
[root@localhost test3]# ls
huawei1.txt  huawei2.txt  passwd  tartest.tar
```

Decompress a package file.

Decompress a file (with the **-x** option) to an existing directory (with the **-C** option) in the specified path.

```
[root@localhost test3]# tar -xvf /root/test3/tartest.tar -C /root/test2
huawei1.txt
huawei2.txt
passwd
[root@localhost test3]# cd /root/test2
[root@localhost test2]# ls
huawei1.txt  huawei2.txt  passwd  test1
```

Step 4 Run the **gzip** command.

Use the **gzip** tool to create a ***.tar.gz** file.

```
[root@localhost ~]# tar -czvf gziptest.tar.gz /root/test2/
```

```
tar: Delete the slash (/) at the beginning of a member name.  
/root/test2/  
/root/test2/passwd  
/root/test2/huawei1.txt  
/root/test2/test1/  
/root/test2/test1/huawei.txt  
/root/test2/huawei2.txt  
[root@localhost ~]# ls  
anaconda-ks.cfg  passwd      passwd2.zip  test2  
gziptest.tar.gz  passwd1.zip  passwd.zip   test3
```

Decompress the ***.tar.gz** file.

```
[root@localhost ~]# ls  
anaconda-ks.cfg  passwd      passwd2.zip  test2  
gziptest.tar.gz  passwd1.zip  passwd.zip   test3  
[root@localhost ~]# ls /root/test2  
huawei1.txt  huawei2.txt  passwd  test1  
[root@localhost ~]# rm -rf /root/test2/*  
[root@localhost ~]# ls /root/test2  
[root@localhost ~]# tar -zxf gziptest.tar.gz -C ~/test2/  
root/test2/  
root/test2/passwd  
root/test2/huawei1.txt  
root/test2/test1/  
root/test2/test1/huawei.txt  
root/test2/huawei2.txt  
[root@localhost ~]# ls /root/test2  
root
```

The commands for decompressing packages in other formats are as follows:

1. The **tar -xvf** command: decompress a *.tar file.
2. The **gzip -d** or **gunzip** command: decompress a *.gz file.
3. The **tar -xzf** command: decompress a .tar.gz or .tgz file.
4. The **bzip2 -d** or **bunzip2** command: decompress a *.bz2 file.
5. The **tar -xjf** command: decompress a *.tar.bz2 file.
6. The **uncompress** command: decompress a *.Z file.
7. The **tar -xZf** command: decompress a *.tar.Z file.
8. The **unrar e** command: decompress a *.rar file.
9. The **unzip** command: decompress a *.zip file.

----End

2.5 Help Command

Step 1 Run the **help** command.

```
[root@localhost ~]# help pwd  
pwd: pwd [-LP]
```

Print the name of the current working directory.

Options:

- L print the value of \$PWD if it names the current working directory
- P print the physical directory, without any symbolic links

By default, `pwd` behaves as if `-L` were specified.

Exit Status:

Returns 0 unless an invalid option is given or the current directory cannot be read.

```
[root@localhost ~]# help -d pwd
pwd - Print the name of the current working directory.
[root@localhost ~]# help -s pwd
pwd: pwd [-LP]
```

----End

2.5.1.2 Other Common Commands

Step 1 Run the **last** command to display the latest login information about a user.

```
[root@localhost ~]# last
root      pts/0        172.19.130.137    Wed Jul  8 14:06    still logged in
root      pts/0        172.19.130.137    Wed Jul  8 11:45 - 13:43  (01:58)
root      tty1          Wed Jul  8 11:23    still logged in
reboot   system boot  4.19.90-2003.4.0 Wed Jul  8 11:19    still running
```

Step 2 Run the **history** command to view historical commands.

```
[root@localhost ~]# history
1  ls -l
```

Step 3 Run the **tab** command.

When a command is output, you can press **Tab** to automatically supplement the command and file path. For example, after you enter **wh** and press **Tab**, the following information is displayed:

```
[root@localhost ~]# wh
whatis  whereis  which  while  whiptail  who  whoami
```

Step 4 Run the **uptime** command to view the system load.

```
[root@localhost ~]# uptime
14:21:42 up  3:02,  2 users,  load average: 0.00, 0.00, 0.00
```

Step 5 Run the **date** command to display or set the date and time of the system.

```
[root@localhost ~]# date
Wed Jul  8 14:23:34 CST 2020
[root@localhost ~]# date '+%c'
Wed 08 Jul 2020 02:24:02 PM CST
```

```
[root@localhost ~]# date '+%D'  
07/08/20  
[root@localhost ~]# date '+%x'  
07/08/2020  
[root@localhost ~]#
```

Step 6 Run the **wget** command to download a file from a specified URL.

Note that the host on which you run the **wget** command must have access to the Internet.

```
[root@localhost ~]# wget https://wordpress.org/latest.zip
```

----End

2.6 Quiz

1. Create an **/iamthebest** directory.
2. Create **/cat** and **/dog** directories in the **/iamthebest** directory.
3. Copy an **/etc/passwd** file to the **/iamthebest** directory and view the operation permission of the copied file.
4. Run the **cp -i /etc/passwd** command. What do you find? Why does this happen?
5. Rename **passwd** to **fun**.
6. Move the **fun** file to the **cat** directory, move the file from the **cat** directory to the **dog** directory, and then move the file to the **/iamthebest** directory.
7. Hard link the **fun** directory to the **/cat** directory.
8. Soft link the **fun** file to the **/dog** directory.
9. Delete the **fun** file from the **/iamthebest** file.
10. View the node information of all **fun** files.
11. Find the **fun** file, package the **fun** file to the **/iamthebest** directory in the *.tar.gz format, and name the package **iamstillfun.tar.gz**.
12. Decompress the **iamstillfun.tar.gz** package to the **cat** directory and name the package **fun**.
13. Find and display the location of all **fun** files.

3 openEuler Text Editor

3.1 Overview

3.1.1 About the Exercises

Vim is a powerful editor with a wide variety of instructions and operations. The exercises mainly involve the use of common vim commands to help trainees proficiently use the vim editor. It introduces how to use the vim editor on the Huawei openEuler OS based on the vimtutor tutorial.

This chapter consists of 8 exercises and describes basic vim operations and vim file operations in the Huawei openEuler lab environment.

- Exercise 1: using the local openEuler lab environment
- Exercise 2: basic vim operations
- Exercise 3: vim operators and actions
- Exercise 4: vim modification
- Exercise 5: vim search and replacement
- Exercise 6: vim file operations
- Exercise 7: other vim operations
- Exercise 8: using the vim help document

3.1.2 Objectives

Upon completion of the exercises, you will be able to:

- Be familiar with the vi and vim editors.
- Master the vim editor installation.
- Master the vim editor mode switching.
- Master common vim editor operations.

3.2 Logging In to the Lab Environment

3.2.1 Logging In to the Lab Environment

Open VirtualBox, start the openEuler VM, and log in to the VM as the **root** user.

3.3 Basic Vim Operations

3.3.1 Moving the Cursor by Pressing **h**, **j**, **k**, and **l**

Step 1 Download vimtutor.

```
[root@openEuler ~]# wget -c http://www.silecs.info/formations/Linux-TP-export/vimtutor-en.txt -O vimtutor
```

Step 2 Input the following code in the command line and press **Enter**. Then, use the **vim** command to open vimtutor.

```
[root@openEuler ~]# vim vimtutor # Use the vim command to access the vimtutor main interface.  
=====  
= Welcome to the VIM Tutor - Version 1.7 =  
=====  
Vim is a very powerful editor that has many commands, too many to  
explain in a tutor such as this. This tutor is designed to describe  
enough of the commands that you will be able to easily use Vim as  
an all-purpose editor.  
The approximate time required to complete the tutor is 25-30 minutes,  
depending upon how much time is spent with experimentation.  
ATTENTION:  
The commands in the lessons will modify the text. Make a copy of this  
file to practise on (if you started "vimtutor" this is already a copy).  
  
It is important to remember that this tutor is set up to teach by  
use. That means that you need to execute the commands to learn them  
properly. If you only read the text, you will forget the commands!  
Now, make sure that your Shift-Lock key is NOT depressed and press  
the j key enough times to move the cursor so that Lesson 1.1  
completely fills the screen.  
=====
```

Step 3 Press **j** to move the cursor down to view all the contents of lesson 1.1. Press **h** to move the cursor left, **j** to move down, **k** to move up, and **l** to move right.

```
Lesson 1.1: MOVING THE CURSOR

** To move the cursor, press the h,j,k,l keys as indicated. **
^
      k          Hint: The h key is at the left and moves left.
< h      l >          The l key is at the right and moves right.
      j          The j key looks like a down arrow.
      v

1. Move the cursor around the screen until you are comfortable.

2. Hold down the down key (j) until it repeats.
   Now you know how to move to the next lesson.

3. Using the down key, move to lesson 1.2.

NOTE: If you are ever unsure about something you typed, press <ESC> to place
you in Normal mode. Then retype the command you wanted.

NOTE: The cursor keys should also work. But using hjkl you will be able to
move around much faster, once you get used to it. Really!
~~~~~~
```

Step 4 Press **Ctrl+D** to go to the next page, press **Ctrl+U** to go to the previous page, and press **h**, **j**, **k**, and **l** to quickly move the cursor.

----End

3.3.2 Exiting Vim

Step 1 Move the cursor downwards to view all the contents of lesson 1.2.

```
Lesson 1.2: EXITING VIM

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Press the <ESC> key (to make sure you are in Normal mode).

2. Type: :q! <ENTER>.
   This exits the editor, DISCARDING any changes you have made.

3. Get back here by executing the command that got you into this tutor. That
   might be: vimtutor <ENTER>

4. If you have these steps memorized and are confident, execute steps
   1 through 3 to exit and re-enter the editor.

NOTE: :q! <ENTER> discards any changes you made. In a few lessons you
      will learn how to save the changes to a file.

5. Move the cursor down to lesson 1.3.
~~~~~~
```

Step 2 Input **:q!** and press **Enter** to forcibly exit the vimtutor without saving any changes.

```
:q!
```

```
Lesson 1.2: EXITING VIM

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Press the <ESC> key (to make sure you are in Normal mode).

2. Type:      :q! <ENTER>.
   This exits the editor, DISCARDING any changes you have made.

3. Get back here by executing the command that got you into this tutor. That
   might be:  vimtutor <ENTER>

4. If you have these steps memorized and are confident, execute steps
   1 through 3 to exit and re-enter the editor.

NOTE:  :q! <ENTER> discards any changes you made. In a few lessons you
      will learn how to save the changes to a file.

5. Move the cursor down to lesson 1.3.

:q!
```

Step 3 Input the following code in the command line and press **Enter** to reopen the vimtutor.

```
[root@openEuler ~]# vimtutor
```

----End

3.3.3 Text Editing - Deletion

Step 1 Move the cursor downwards to view all the contents of lesson 1.3.

```
Lesson 1.3: TEXT EDITING - DELETION

** Press x to delete the character under the cursor. **

1. Move the cursor to the line below marked --->.

2. To fix the errors, move the cursor until it is on top of the
   character to be deleted.

3. Press the x key to delete the unwanted character.

4. Repeat steps 2 through 4 until the sentence is correct.

---> The ccow jumpedd ovverr thhe mooon.

5. Now that the line is correct, go on to lesson 1.4.

NOTE: As you go through this tutor, do not try to memorize, learn by usage.

:
```

Step 2 Move the cursor to the ---> line, and then press x to delete the redundant letter c.

```
Lesson 1.3: TEXT EDITING - DELETION

** Press x to delete the character under the cursor. **

1. Move the cursor to the line below marked --->.

2. To fix the errors, move the cursor until it is on top of the
character to be deleted.

3. Press the x key to delete the unwanted character.

4. Repeat steps 2 through 4 until the sentence is correct.

---> The cow jumpedd ovverr thhe mooon.

5. Now that the line is correct, go on to lesson 1.4.

NOTE: As you go through this tutor, do not try to memorize, learn by usage.

~~~~~
```

Step 3 Repeat the preceding steps to move the cursor and press x to delete redundant letters from the sentence.

```
---> The cow jumped over the moon.

----End
```

3.3.4 Text Editing - Insertion

Step 1 Move the cursor downwards to view all the contents of lesson 1.4.

```
Lesson 1.4: TEXT EDITING - INSERTION

** Press i to insert text. **

1. Move the cursor to the first line below marked --->.

2. To make the first line the same as the second, move the cursor on top
of the first character AFTER where the text is to be inserted.

3. Press i and type in the necessary additions.

4. As each error is fixed press <ESC> to return to Normal mode.
Repeat steps 2 through 4 to correct the sentence.

---> There is text misng this .
---> There is some text missing from this line.

5. When you are comfortable inserting text move to lesson 1.5.

~~~~~
```

Step 2 Move the cursor to the word **text** in the first ---> line and press **i** to enter the insert mode.

```
Lesson 1.4: TEXT EDITING - INSERTION

** Press i to insert text. **

1. Move the cursor to the first line below marked --->.

2. To make the first line the same as the second, move the cursor on top
   of the first character AFTER where the text is to be inserted.

3. Press i and type in the necessary additions.

4. As each error is fixed press <ESC> to return to Normal mode.
   Repeat steps 2 through 4 to correct the sentence.

---> There is text misng this .
---> There is some text missing from this line.

5. When you are comfortable inserting text move to lesson 1.5.

~~~~~
-- INSERT --
```

Step 3 Input the missing word **some** by referring to the line below.

```
---> There is some text misng this .
---> There is some text missing from this line.
```

Step 4 Repeat the preceding steps to move the cursor and enter other missing words and letters to ensure that the two sentences are the same. Press **Esc** to exit the insert mode and return to the normal mode.

```
---> There is some text missing from this line.
---> There is some text missing from this line.
```

----End

3.3.5 Text Editing - Appending

Step 1 Move the cursor downwards to view all the contents of lesson 1.5.

```
Lesson 1.5: TEXT EDITING - APPENDING

** Press A to append text. **

1. Move the cursor to the first line below marked --->.
   It does not matter on what character the cursor is in that line.

2. Press A and type in the necessary additions.

3. As the text has been appended press <ESC> to return to Normal mode.

4. Move the cursor to the second line marked ---> and repeat
   steps 2 and 3 to correct this sentence.

---> There is some text missing from th
      There is some text missing from this line.
---> There is also some text miss
      There is also some text missing here.

5. When you are comfortable appending text move to lesson 1.6.
```

Step 2 Move the cursor to the first ---> line.

```
---> There is some text missing from th
      There is some text missing from this line.
---> There is also some text miss
      There is also some text missing here.
```

Step 3 Press **Shift+A** to enter the insert mode and append the missing words and letters. Press **Esc** to exit the insert mode.

```
---> There is some text missing from this line.
      There is some text missing from this line.
---> There is also some text miss
      There is also some text missing here.
```

Step 4 Repeat the preceding steps to press **Shift+A** to append the missing words and letters in the second ---> line, and then press **Esc** to exit the insert mode.

```
---> There is some text missing from this line.
      There is some text missing from this line.
---> There is also some text missing here.
      There is also some text missing here.
```

----End

3.3.6 Editing a File

Step 1 Move the cursor downwards to view all the contents of lesson 1.6.

```
Lesson 1.6: EDITING A FILE

** Use :wq to save a file and exit. **

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Exit this tutor as you did in lesson 1.2: :q!
   Or, if you have access to another terminal, do the following there.

2. At the shell prompt type this command: vim tutor <ENTER>
   'vim' is the command to start the Vim editor, 'tutor' is the name of the
   file you wish to edit. Use a file that may be changed.

3. Insert and delete text as you learned in the previous lessons.

4. Save the file with changes and exit Vim with: :wq <ENTER>

5. If you have quit vimtutor in step 1 restart the vimtutor and move down to
   the following summary.

6. After reading the above steps and understanding them: do it.

~~~~~
```

Step 2 Input **:wq** and press **Enter** to save and exit the tutorial.

```
:wq
Lesson 1.6: EDITING A FILE

** Use :wq to save a file and exit. **

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Exit this tutor as you did in lesson 1.2: :q!
   Or, if you have access to another terminal, do the following there.

2. At the shell prompt type this command: vim tutor <ENTER>
   'vim' is the command to start the Vim editor, 'tutor' is the name of the
   file you wish to edit. Use a file that may be changed.

3. Insert and delete text as you learned in the previous lessons.

4. Save the file with changes and exit Vim with: :wq <ENTER>

5. If you have quit vimtutor in step 1 restart the vimtutor and move down to
   the following summary.

6. After reading the above steps and understanding them: do it.

~~~~~
```

Step 3 Input the following code in the command line and press **Enter** to reopen the vimtutor. Move the cursor and verify that all changes are saved.

```
[root@openEuler ~]# vim vimtutor
```

----End

3.4 Vim Operators and Actions

3.4.1 Deletion Commands

Step 1 Press **Esc** to enter the normal mode in vimtutor, input :208, and press **Enter** to go to the 208th line.

```
Lesson 2.1: DELETION COMMANDS

** Type dw to delete a word. **

1. Press <ESC> to make sure you are in Normal mode.
2. Move the cursor to the line below marked --->.
3. Move the cursor to the beginning of a word that needs to be deleted.
4. Type dw to make the word disappear.

NOTE: The letter d will appear on the last line of the screen as you type it. Vim is waiting for you to type w. If you see another character than d you typed something wrong; press <ESC> and start over.

---> There are a some words fun that don't belong paper in this sentence.

5. Repeat steps 3 and 4 until the sentence is correct and go to Lesson 2.2.

~~~~~
:208
```

Step 2 Move the cursor to the ---> line where the word to be deleted is located, and input **dw** to delete the word. Use **dw** to delete **a**, **fun**, and **paper** until the sentence is clear.

```
Lesson 2.1: DELETION COMMANDS

** Type dw to delete a word. **

1. Press <ESC> to make sure you are in Normal mode.
2. Move the cursor to the line below marked --->.
3. Move the cursor to the beginning of a word that needs to be deleted.
4. Type dw to make the word disappear.

NOTE: The letter d will appear on the last line of the screen as you type it. Vim is waiting for you to type w. If you see another character than d you typed something wrong; press <ESC> and start over.

---> There are a some words fun that don't belong paper in this sentence.

5. Repeat steps 3 and 4 until the sentence is correct and go to Lesson 2.2.

~~~~~
203,29 19%
```

----End

3.4.2 Deletion Commands **de** and **d\$**

- Step 1 Move the cursor downwards to view all the contents of lesson 2.2. Move the cursor to the position after the first period (.) in the ---> line.

```
Lesson 2.2: MORE DELETION COMMANDS

    ** Type d$ to delete to the end of the line. **

1. Press <ESC> to make sure you are in Normal mode.

2. Move the cursor to the line below marked --->.

3. Move the cursor to the end of the correct line (AFTER the first .).

4. Type d$ to delete to the end of the line.

---> Somebody typed the end of this line twice. End of this line twice.

5. Move on to Lesson 2.3 to understand what is happening.
```

- Step 2 Input **d\$** to delete the content after the cursor.

```
---> Somebody typed the end of this line twice.

----End
```

3.4.3 Operators and Actions

- Step 1 Move the cursor downwards to view all the contents of lesson 2.3. Use the **d motion** command as an example, where **d** indicates a deletion operator, and *motion* indicates an action. **de** deletes the content from the cursor to the end of the current word. **dw** deletes the content from the cursor to the beginning of the next word. **d\$** deletes the content from the cursor to the end of the current line.

Lesson 2.3: ON OPERATORS AND MOTIONS

Many commands that change text are made from an operator and a motion.
The format for a delete command with the `d` delete operator is as follows:

`d motion`

Where:

`d` - is the delete operator.
`motion` - is what the operator will operate on (listed below).

A short list of motions:

`w` - until the start of the next word, EXCLUDING its first character.
`e` - to the end of the current word, INCLUDING the last character.
`$` - to the end of the line, INCLUDING the last character.

Thus typing `de` will delete from the cursor to the end of the word.

NOTE: Pressing just the motion while in Normal mode without an operator will move the cursor as specified.

254, 47

24%

---End

3.4.4 Adding a Number Before an Action

Step 1 Move the cursor downwards to view all the contents of lesson 2.4. Move the cursor to the beginning of the `--->` line.

Lesson 2.4: USING A COUNT FOR A MOTION

** Typing a number before a motion repeats it that many times. **

1. Move the cursor to the start of the line marked `--->` below.
2. Type `2w` to move the cursor two words forward.
3. Type `3e` to move the cursor to the end of the third word forward.
4. Type `0` (zero) to move to the start of the line.
5. Repeat steps 2 and 3 with different numbers.

`--->` This is just a line with words you can move around in.

6. Move on to Lesson 2.5.

270, 1

26%

Step 2 Input **2w** to move the cursor rightwards by two words.

`---> This [is just a line with words you can move around in.`

Step 3 Input **3e** to move the cursor rightwards to the end of the third word.

```
--> This is just a line with words you can move around in.
```

Step 4 Input **0** to move the cursor to the beginning of the line.

```
--> This is just a line with words you can move around in.
```

----End

3.4.5 Adding a Number Before a Deletion Command

Step 1 Move the cursor downwards to view all the contents of lesson 2.5. Move the cursor to the first uppercase letter **A** in the ---> line.

```
Lesson 2.5: USING A COUNT TO DELETE MORE
```

```
** Typing a number with an operator repeats it that many times. **
```

```
In the combination of the delete operator and a motion mentioned above you  
insert a count before the motion to delete more:  
d number motion
```

1. Move the cursor to the first UPPER CASE word in the line marked --->.
2. Type **d2w** to delete the two UPPER CASE words
3. Repeat steps 1 and 2 with a different count to delete the consecutive
UPPER CASE words with one command

```
--> this ABC DE line FGHI JK LMN OP of words is Q RS TUV cleaned up.
```

Step 2 Input **d2w** to delete **ABC** and **DE**.

```
--> this line FGHI JK LMN OP of words is Q RS TUV cleaned up.
```

Step 3 Move the cursor and run the **d number motion** command, for example, **d4w**, to delete **FGHI**, **JK**, **LMN**, **OP**, **Q**, **RS**, and **TUV**.

```
--> this line of words is cleaned up.
```

----End

3.4.6 Line Operators

Step 1 Move the cursor downwards to view all the contents of lesson 2.6. Move the cursor to the second ---> line.

```
Lesson 2.6: OPERATING ON LINES

** Type dd to delete a whole line. **

Due to the frequency of whole line deletion, the designers of Vi decided it would be easier to simply type two d's to delete a line.

1. Move the cursor to the second line in the phrase below.
2. Type dd to delete the line.
3. Now move to the fourth line.
4. Type 2dd to delete two lines.

---> 1) Roses are red,
---> 2) Mud is fun,
---> 3) Violets are blue,
---> 4) I have a car,
---> 5) Clocks tell time,
---> 6) Sugar is sweet
---> 7) And so are you.
```

Step 2 Enter **dd** to delete the second line.

```
---> 1) Roses are red,
---> 3) Violets are blue,
---> 4) I have a car,
---> 5) Clocks tell time,
---> 6) Sugar is sweet
---> 7) And so are you.
```

Step 3 Move the cursor to the fourth line and enter **2dd** to delete the fourth and fifth lines.

```
---> 1) Roses are red,
---> 3) Violets are blue,
---> 6) Sugar is sweet
---> 7) And so are you.
```

----End

3.4.7 Undo Operations

Step 1 Move the cursor downwards to view all the contents of lesson 2.7. Move the cursor to the first letter **i** in the ---> line.

```
Lesson 2.7: THE UNDO COMMAND

** Press u to undo the last commands, U to fix a whole line. **

1. Move the cursor to the line below marked ---> and place it on the first error.
2. Type x to delete the first unwanted character.
3. Now type u to undo the last command executed.
4. This time fix all the errors on the line using the x command.
5. Now type a capital U to return the line to its original state.
6. Now type u a few times to undo the U and preceding commands.
7. Now type CTRL-R (keeping CTRL key pressed while hitting R) a few times to redo the commands (undo the undo's).

---> Fix the errors on this line and replace them with undo.

8. These are very useful commands. Now move on to the Lesson 2 Summary.
```

- Step 2 Press **x** to delete the letter **i**, and then press **u** to undo the operation, that is, restore the letter **i**.

```
----> Fix the errors oon thhis line and reeplace them with undo.  
8. These are very useful commands. Now move on to the Lesson 2 Summary.
```

~~~~~  
1 change; before #1 2 seconds ago 339,7 34%

- Step 3 Press **x** to delete redundant letters.

```
----> Fix the errors on this line and replace them with undo.
```

- Step 4 Press **Shift+U** to undo all operations in the line and return to original state.

```
----> Fix the errors oon thhis line and reeplace them with undo.
```

- Step 5 Press **u** for multiple times to undo step 4.

```
----> Fix the errors on thhis line and reeplace them with undo.
```

```
----> Fix the errors oon thhis line and reeplace them with undo.
```

- Step 6 Hold down **Ctrl** and press **R** for multiple times to redo step 5 until the sentence is restored to the state in step 3.

```
----> Fix the errors on this line and replace them with undo.
```

----End

## 3.5 Vim Modification

### 3.5.1 Paste

- Step 1 Move the cursor downwards to view all the contents of lesson 3.1. Move the cursor to the first ---> line.

## Lesson 3.1: THE PUT COMMAND

```
** Type p to put previously deleted text after the cursor. **  
1. Move the cursor to the first ---> line below.  
2. Type dd to delete the line and store it in a Vim register.  
3. Move the cursor to the c) line, ABOVE where the deleted line should go.  
4. Type p to put the line below the cursor.  
5. Repeat steps 2 through 4 to put all the lines in correct order.  
---> d) Can you learn too?  
---> b) Violets are blue,  
---> c) Intelligence is learned,  
---> a) Roses are red,
```

Step 2 Input **dd** to delete the line. Vim automatically saves the data to a register.

```
---> b) Violets are blue,  
---> c) Intelligence is learned,  
---> a) Roses are red,
```

Step 3 Move the cursor to the **c** line, and press **p** to paste the **d** line content saved in the register to the line below **c**.

```
---> b) Violets are blue,  
---> c) Intelligence is learned,  
---> d) Can you learn too?  
---> a) Roses are red,
```

Step 4 Repeat the preceding steps to delete and save the data in the **a** line, move the cursor to the **b** line, and press **p** to paste the data.

```
---> a) Roses are red,  
---> b) Violets are blue,  
---> c) Intelligence is learned,  
---> d) Can you learn too?
```

Alternatively, use **nyy** to copy **n** lines of data and press **p** to paste the data to the line below the current line.

----End

### 3.5.2 Replace

Step 1 Move the cursor downwards to view all the contents of lesson 3.2. Move the cursor to the letter **a** of **Whan** in the first ---> line.

## Lesson 3.2: THE REPLACE COMMAND

```
** Type rx to replace the character at the cursor with x . **  
1. Move the cursor to the first line below marked --->.  
2. Move the cursor so that it is on top of the first error.  
3. Type r and then the character which should be there.  
4. Repeat steps 2 and 3 until the first line is equal to the second one.  
---> When this lime was tuoed in, someone presswd some wrojg keys!  
---> When this line was typed in, someone pressed some wrong keys!  
5. Now move on to Lesson 3.3.
```

NOTE: Remember that you should be learning by doing, not memorization.

- Step 2 Input **re** to replace the letter **a** with **e**. In the preceding command, **r** indicates the replacement command, and **e** indicates the new letter.

```
---> When this lime was tuoed in, someone presswd some wrojg keys!  
---> When this line was typed in, someone pressed some wrong keys!
```

- Step 3 Use the **r** command to replace wrong letters in the first line by referring to the second line.

```
---> When this line was typed in, someone pressed some wrong keys!  
---> When this line was typed in, someone pressed some wrong keys!
```

----End

### 3.5.3 Change

- Step 1 Move the cursor downwards to view all the contents of lesson 3.3. Move the cursor to the letter **u** of **lubw** in the first ---> line.

## Lesson 3.3: THE CHANGE OPERATOR

```
** To change until the end of a word, type ce . **  
1. Move the cursor to the first line below marked --->.  
2. Place the cursor on the u in lubw.  
3. Type ce and the correct word (in this case, type ine ).  
4. Press <ESC> and move to the next character that needs to be changed.  
5. Repeat steps 3 and 4 until the first sentence is the same as the second.  
---> This lubw has a few wptfd that mrrf changing usf the change operator.  
---> This line has a few words that need changing using the change operator.  
Notice that ce deletes the word and places you in Insert mode.
```

- Step 2 Input **ce** and then **ine** to change the word to **line** by referring to the second line. Press **Esc** to exit the insert mode.

```
---> This line has a few wptfd that mrrf changing usf the change operator.  
---> This line has a few words that need changing using the change operator.
```

- Step 3 Repeat the preceding steps to change the wrong words such as **wptfd** and **meef** to those in the second line.

```
---> This line has a few words that need changing using the change operator.  
---> This line has a few words that need changing using the change operator.
```

**----End**

### 3.5.4 Other Changes

- Step 1 Move the cursor downwards to view all the contents of lesson 3.4. Move the cursor to the beginning of the first wrong word **some** in the first ---> line.

#### Lesson 3.4: MORE CHANGES USING c

```
** The change operator is used with the same motions as delete. **  
1. The change operator works in the same way as delete. The format is:  
    c [number] motion  
2. The motions are the same, such as w (word) and $ (end of line).  
3. Move to the first line below marked --->.  
4. Move the cursor to the first error.  
5. Type c$ and type the rest of the line like the second and press <ESC>.  
---> The end of this line needs some help to make it like the second.  
---> The end of this line needs to be corrected using the c$ command.  
NOTE: You can use the Backspace key to correct mistakes while typing.
```

- Step 2 Enter **c\$**, enter the sentence content by referring to the second line, and press **Esc** to exit the insert mode. If the input is incorrect, press **Backspace** to delete the wrong content.

```
---> The end of this line needs to be corrected using the c$ command.  
---> The end of this line needs to be corrected using the c$ command.
```

**----End**

## 3.6 Vim Search and Replacement Operations

### 3.6.1 Cursor Position and File Status

- Step 1 Move the cursor downwards to view all the contents of lesson 4.1.

```
Lesson 4.1: CURSOR LOCATION AND FILE STATUS

** Type CTRL-G to show your location in the file and the file status.
Type G to move to a line in the file. **

NOTE: Read this entire lesson before executing any of the steps!!

1. Hold down the Ctrl key and press g . We call this CTRL-G.
A message will appear at the bottom of the page with the filename and the
position in the file. Remember the line number for Step 3.

NOTE: You may see the cursor position in the lower right corner of the screen
This happens when the 'ruler' option is set (see :help 'ruler' )

2. Press G to move you to the bottom of the file.
Type gg to move you to the start of the file.

3. Type the number of the line you were on and then G . This will
return you to the line you were on when you first pressed CTRL-G.

4. If you feel confident to do this, execute steps 1 through 3.
```

Step 2 Press **Ctrl+G** to display the file name and location information. Remember the line number 492.

```
1. Hold down the Ctrl key and press g . We call this CTRL-G.
A message will appear at the bottom of the page with the filename and the
position in the file. Remember the line number for Step 3.

NOTE: You may see the cursor position in the lower right corner of the screen
This happens when the 'ruler' option is set (see :help 'ruler' )

2. Press G to move you to the bottom of the file.
Type gg to move you to the start of the file.

3. Type the number of the line you were on and then G . This will
return you to the line you were on when you first pressed CTRL-G.

4. If you feel confident to do this, execute steps 1 through 3.

~~~~~ "vimtutor" [Modified] 970 lines --50%-- 492,1 51%
```

Step 3 Press **Shift+G** to go to the end of the file.

```
This tutorial was written by Michael C. Pierce and Robert K. Ware,
Colorado School of Mines using ideas supplied by Charles Smith,
Colorado State University. E-mail: bware@mines.colorado.edu.

Modified for Vim by Bram Moolenaar.

~~~~~ 970,1 Bot
```

Step 4 Input **gg** to go to the beginning of the file.

```
===== W e l c o m e t o t h e V I M T u t o r - Version 1.7 =====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.
```

Step 5 Input 492 and press Shift+G to go back to line 492.

1. Hold down the Ctrl key and press g . We call this CTRL-G.  
A message will appear at the bottom of the page with the filename and the position in the file. Remember the line number for Step 3.

NOTE: You may see the cursor position in the lower right corner of the screen  
This happens when the 'ruler' option is set (see :help 'ruler' )

2. Press G to move you to the bottom of the file.  
Type gg to move you to the start of the file.

3. Type the number of the line you were on and then G . This will return you to the line you were on when you first pressed CTRL-G.

492,3                    50%

----End

### 3.6.2 Search

Step 1 Move the cursor downwards to view all the contents of lesson 4.2.

Lesson 4.2: THE SEARCH COMMAND

\*\* Type / followed by a phrase to search for the phrase. \*\*

1. In Normal mode type the / character. Notice that it and the cursor appear at the bottom of the screen as with the : command.

2. Now type 'errroor' <ENTER>. This is the word you want to search for.

3. To search for the same phrase again, simply type n .  
To search for the same phrase in the opposite direction, type N .

4. To search for a phrase in the backward direction, use ? instead of / .

5. To go back to where you came from press CTRL-O (Keep Ctrl down while pressing the letter o). Repeat to go back further. CTRL-I goes forward.

--> "errroor" is not the way to spell error; errroor is an error.

NOTE: When the search reaches the end of the file it will continue at the start, unless the 'wrapscan' option has been reset.

Step 2 Press / to go to the search box, input **errroor**, and press Enter.

2. Now type 'errroor' <ENTER>. This is the word you want to search for.

3. To search for the same phrase again, simply type n .  
To search for the same phrase in the opposite direction, type N .

4. To search for a phrase in the backward direction, use ? instead of / .

5. To go back to where you came from press CTRL-O (Keep Ctrl down while pressing the letter o). Repeat to go back further. CTRL-I goes forward.

--> "errroor" is not the way to spell error; errroor is an error.

NOTE: When the search reaches the end of the file it will continue at the start, unless the 'wrapscan' option has been reset.

~~~~~  
search hit BOTTOM, continuing at TOP 516,16 53%

- Step 3 Press **n** to search for **erroor** again. Press **Shift+N** to search for **erroor** again in the opposite direction. Press **Ctrl+O** to jump to the previous location. Press **Ctrl+I** to jump forward to the next location.

```
---> "errroor" is not the way to spell error; Errroor is an error.  
NOTE: When the search reaches the end of the file it will continue at the  
      start, unless the 'wrapscan' option has been reset.  
~~~~~  
/errroor                                         526, 49          53%
```

----End

3.6.3 Matching Parentheses Search

- Step 1 Move the cursor downwards to view all the contents of lesson 4.4. Move the cursor to the left parenthesis **(** in the first ---> line.

```
Lesson 4.3: MATCHING PARENTHESES SEARCH  
  
** Type % to find a matching ),], or } . **  
  
1. Place the cursor on any (, [, or { in the line below marked --->.  
2. Now type the % character.  
3. The cursor will move to the matching parenthesis or bracket.  
4. Type % to move the cursor to the other matching bracket.  
5. Move the cursor to another (, ), [ , ] , { or } and see what % does.  
---> This ( is a test line with ('s, ['s ] and {'s } in it. )  
  
NOTE: This is very useful in debugging a program with unmatched parentheses!
```

- Step 2 Input **%** to jump to the matching right parenthesis **)**.

```
---> This ( is a test line with ('s, ['s ] and {'s } in it. )
```

- Step 3 Input **%** again to go back to the left parenthesis **(**.

```
---> This ( is a test line with ('s, ['s ] and {'s } in it. )
```

- Step 4 Move the cursor to other brackets such as **[** and use **%** to search for matching brackets.

```
---> This ( is a test line with ('s, ['s ] and {'s } in it. ))
```

----End

3.6.4 Substitute

- Step 1 Move the cursor downwards to view all the contents of lesson 4.4. Move the cursor to the beginning of the ---> line.

Lesson 4.4: THE SUBSTITUTE COMMAND

```
** Type :s/old/new/g to substitute 'new' for 'old'. **
```

1. Move the cursor to the line below marked --->.
2. Type :s/thee/the <ENTER>. Note that this command only changes the first occurrence of "thee" in the line.
3. Now type :s/thee/the/g . Adding the g flag means to substitute globally in the line, change all occurrences of "thee" in the line.
---> thee best time to see thee flowers is in thee spring.
4. To change every occurrence of a character string between two lines, type :#, #s/old/new/g where #, # are the line numbers of the range of lines where the substitution is to be done.
Type :%s/old/new/g to change every occurrence in the whole file.
Type :%s/old/new/gc to find every occurrence in the whole file, with a prompt whether to substitute or not.

Step 2 Input :s/**thee**/the and press **Enter** to replace the first **thee** with **the**.

```
---> the best time to see thee flowers is in thee spring.
```

Step 3 Input :s/**thee**/the/g and press **Enter** to replace all **thee** in the line with **the**.

```
---> the best time to see the flowers is in the spring.
```

----End

3.7 Vim File Operations

3.7.1 Running External Commands

Step 1 Move the cursor downwards to view all the contents of lesson 5.1.

Lesson 5.1: HOW TO EXECUTE AN EXTERNAL COMMAND

```
** Type :! followed by an external command to execute that command. **
```

1. Type the familiar command : to set the cursor at the bottom of the screen. This allows you to enter a command-line command.
2. Now type the ! (exclamation point) character. This allows you to execute any external shell command.
3. As an example type ls following the ! and then hit <ENTER>. This will show you a listing of your directory, just as if you were at the shell prompt. Or use :!dir if ls doesn't work.

NOTE: It is possible to execute any external command this way, also with arguments.

NOTE: All : commands must be finished by hitting <ENTER>
From here on we will not always mention it.

- Step 2 Input **:!ls** to use the exclamation mark (!) to run external shell commands, for example, run the **ls** command to list directories. Press **Enter** to return to the vimtutor.

```
[root@ecs-openeuler ~]# vim vimtutor  
vimtutor  
Press ENTER or type command to continue  
----End
```

3.7.2 More Operations on Writing Files

- Step 1 Move the cursor downwards to view all the contents of lesson 5.2.

```
Lesson 5.2: MORE ON WRITING FILES  
  
** To save the changes made to the text, type :w FILENAME. **  
  
1. Type :!dir or :!ls to get a listing of your directory.  
You already know you must hit <ENTER> after this.  
  
2. Choose a filename that does not exist yet, such as TEST.  
  
3. Now type: :w TEST (where TEST is the filename you chose.)  
  
4. This saves the whole file (the Vim Tutor) under the name TEST.  
To verify this, type :!dir or :!ls again to see your directory.  
  
NOTE: If you were to exit Vim and start it again with vim TEST , the file  
would be an exact copy of the tutor when you saved it.  
  
5. Now remove the file by typing (MS-DOS): :!del TEST  
or (Unix): :!rm TEST
```

- Step 2 Input **:w TEST** to create a file named **TEST** in the current directory.

```
~~~~~  
"TEST" [New] 970L, 33266C written 645,1 65%
```

- Step 3 Input **:!ls** again to view the directory where a file named **TEST** has been created.

```
TEST vimtutor  
Press ENTER or type command to continue  
----End
```

3.7.3 Selecting Text to Write

- Step 1 Move the cursor downwards to view all the contents of lesson 5.3. Move the cursor to line 1.

Lesson 5.3: SELECTING TEXT TO WRITE

** To save part of the file, type `v` motion `:w FILENAME` **

1. Move the cursor to this line.
2. Press `v` and move the cursor to the fifth item below. Notice that the text is highlighted.
3. Press the `:` character. At the bottom of the screen `:<,>` will appear.
4. Type `w TEST`, where `TEST` is a filename that does not exist yet. Verify that you see `:<,>w TEST` before you press <ENTER>.
5. Vim will write the selected lines to the file `TEST`. Use `:!dir` or `:!ls` to see it. Do not remove it yet! We will use it in the next lesson.

NOTE: Pressing `v` starts Visual selection. You can move the cursor around to make the selection bigger or smaller. Then you can use an operator to do something with the text. For example, `d` deletes the text.

Step 2 Press **v** to enter the visual mode and move the cursor to the end of line 5.

1. Move the cursor to this line.
2. Press `v` and move the cursor to the fifth item below. Notice that the text is highlighted.
3. Press the `:` character. At the bottom of the screen `:<,>` will appear.
4. Type `w TEST`, where `TEST` is a filename that does not exist yet. Verify that you see `:<,>w TEST` before you press <ENTER>.
5. Vim will write the selected lines to the file `TEST`. Use `:!dir` or `:!ls` to see it. Do not remove it yet! We will use it in the next lesson.

NOTE: Pressing `v` starts Visual selection. You can move the cursor around to make the selection bigger or smaller. Then you can use an operator to do something with the text. For example, `d` deletes the text.

-- VISUAL -- 663,0-1 68%

Step 3 Input `:w! TEST` and press **Enter** to forcibly write the highlighted content into the **TEST** file.

1. Move the cursor to this line.
2. Press `v` and move the cursor to the fifth item below. Notice that the text is highlighted.
3. Press the `:` character. At the bottom of the screen `:<,>` will appear.
4. Type `w TEST`, where `TEST` is a filename that does not exist yet. Verify that you see `:<,>w TEST` before you press <ENTER>.
5. Vim will write the selected lines to the file `TEST`. Use `:!dir` or `:!ls` to see it. Do not remove it yet! We will use it in the next lesson.

NOTE: Pressing `v` starts Visual selection. You can move the cursor around to make the selection bigger or smaller. Then you can use an operator to do something with the text. For example, `d` deletes the text.

`:<,>w! TEST`

Step 4 Input **:!cat TEST** to view the **TEST** file content, and then press **Enter** to return to vimtutor.

1. Move the cursor to this line.
2. Press **v** and move the cursor to the fifth item below. Notice that the text is highlighted.
3. Press the **:** character. At the bottom of the screen **:'<,'>** will appear.
4. Type **w TEST**, where **TEST** is a filename that does not exist yet. Verify that you see **:'<,'>w TEST** before you press **<ENTER>**.
5. Vim will write the selected lines to the file **TEST**. Use **:!dir** or **:!ls** to see it. Do not remove it yet! We will use it in the next lesson.

Press ENTER or type command to continue █

----End

3.7.4 Retrieving File Content

Step 1 Move the cursor downwards to view all the contents of lesson 5.4. Move the cursor to the line above line 1.

```
Lesson 5.4: RETRIEVING AND MERGING FILES

** To insert the contents of a file, type :r FILENAME **

1. Place the cursor just above this line.

NOTE: After executing Step 2 you will see text from Lesson 5.3. Then move DOWN to see this lesson again.

2. Now retrieve your TEST file using the command :r TEST where TEST is the name of the file you used.
The file you retrieve is placed below the cursor line.

3. To verify that a file was retrieved, cursor back and notice that there are now two copies of Lesson 5.3, the original and the file version.

NOTE: You can also read the output of an external command. For example,
:r !ls reads the output of the ls command and puts it below the cursor.
```

Step 2 Input **:r TEST** and press **Enter** to retrieve text from the **TEST** file.

1. Move the cursor to this line.
2. Press `v` and move the cursor to the fifth item below. Notice that the text is highlighted.
3. Press the `:` character. At the bottom of the screen `:<,>` will appear.
4. Type `w TEST`, where `TEST` is a filename that does not exist yet. Verify that you see `:<,>w TEST` before you press `<ENTER>`.
5. Vim will write the selected lines to the file `TEST`. Use `:!dir` or `:!ls` to see it. Do not remove it yet! We will use it in the next lesson.

1. Place the cursor just above this line.

NOTE: After executing Step 2 you will see text from Lesson 5.3. Then move DOWN to see this lesson again.

----End

3.8 Other Vim Operations

3.8.1 Open

Step 1 Move the cursor downwards to view all the contents of lesson 6.1. Move the cursor to the beginning of the first ---> line.

Lesson 6.1: THE OPEN COMMAND

** Type `o` to open a line below the cursor and place you in Insert mode. **

1. Move the cursor to the line below marked --->.
2. Type the lowercase letter `o` to open up a line BELOW the cursor and place you in Insert mode.
3. Now type some text and press `<ESC>` to exit Insert mode.

---> After typing `o` the cursor is placed on the open line in Insert mode.

4. To open up a line ABOVE the cursor, simply type a capital `O`, rather than a lowercase `o`. Try this on the line below.

---> Open up a line above this by typing `O` while the cursor is on this line.

Step 2 Press `o` to open a new line under the cursor and enter the insert mode. Input any text and press `Esc` to exit the insert mode.

---> After typing `o` the cursor is placed on the open line in Insert mode.
open

Step 3 Move the cursor to the beginning of the second ---> line.

```
--> After typing o the cursor is placed on the open line in Insert mode.  
open
```

```
4. To open up a line ABOVE the cursor, simply type a capital O , rather  
than a lowercase o. Try this on the line below.
```

```
--> Open up a line above this by typing O while the cursor is on this line.
```

- Step 4 Press **Shift+O** to open a new line above the cursor and enter the insert mode, input any text, and press **Esc** to exit the insert mode.

```
open
```

```
--> Open up a line above this by typing O while the cursor is on this line.
```

----End

3.8.2 Append

- Step 1 Move the cursor downwards to view all the contents of lesson 6.2. Then, press **e** to move the cursor to the position after **li** in the first ---> line.

Lesson 6.2: THE APPEND COMMAND

```
** Type a to insert text AFTER the cursor. **
```

1. Move the cursor to the start of the line below marked --->.
2. Press e until the cursor is on the end of li .
3. Type an a (lowercase) to append text AFTER the cursor.
4. Complete the word like the line below it. Press <ESC> to exit Insert mode.
5. Use e to move to the next incomplete word and repeat steps 3 and 4.

```
--> This li will allow you to pract appendi text to a line.
```

```
--> This line will allow you to practice appending text to a line.
```

NOTE: a, i and A all go to the same Insert mode, the only difference is where the characters are inserted.

- Step 2 Press **a** to append content after the cursor, input **ne**, and press **Esc** to exit the insert mode.

```
--> This line will allow you to pract appendi text to a line.
```

```
--> This line will allow you to practice appending text to a line.
```

- Step 3 Repeat the preceding steps to ensure that the content in the first line is the same as that in the second line.

```
--> This line will allow you to practice appending text to a line.
```

```
--> This line will allow you to practice appending text to a line.
```

----End

3.8.3 Another Replace Command

Step 1 Move the cursor downwards to view all the contents of lesson 6.3. Move the cursor to the beginning of **xxx** in the first ---> line.

```
Lesson 6.3: ANOTHER WAY TO REPLACE

** Type a capital R to replace more than one character. **

1. Move the cursor to the first line below marked --->. Move the cursor to the beginning of the first xxx .

2. Now press R and type the number below it in the second line, so that it replaces the xxx .

3. Press <ESC> to leave Replace mode. Notice that the rest of the line remains unmodified.

4. Repeat the steps to replace the remaining xxx.

---> Adding 123 to xxx gives you xxx.
---> Adding 123 to 456 gives you 579.

NOTE: Replace mode is like Insert mode, but every typed character deletes an existing character.
```

Step 2 Press **Shift+R**, input **456** to replace **xxx**, and press **Esc** to exit the replace mode.

```
---> Adding 123 to 456 gives you xxx.
---> Adding 123 to 456 gives you 579.

NOTE: Replace mode is like Insert mode, but every typed character deletes an existing character.

~~~~~
-- REPLACE --                                777,23      80%
```

Step 3 Move the cursor to the beginning of the last **xxx**, and press **Shift+R** to replace the content by referring to the second line.

```
---> Adding 123 to 456 gives you 579.
---> Adding 123 to 456 gives you 579.
```

----End

3.8.4 Copy and Paste Text

Step 1 Move the cursor downwards to view all the contents of lesson 6.4. Move the cursor to the position after **a)** in the ---> line.

Lesson 6.4: COPY AND PASTE TEXT

** Use the **y** operator to copy text and **p** to paste it **

1. Go to the line marked with ---> below and place the cursor after "a)".
2. Start Visual mode with **v** and move the cursor to just before "first".
3. Type **y** to yank (copy) the highlighted text.
4. Move the cursor to the end of the next line: **j\$**
5. Type **p** to put (paste) the text. Then type: **a** second <ESC> .
6. Use Visual mode to select " item.", yank it with **y**, move to the end of the next line with **j\$** and put the text there with **p**.

---> a) this is the first item.
b)

NOTE: you can also use **y** as an operator; **yw** yanks one word.

- Step 2 Press **v** to enter the visual mode, move the cursor to the position before the word **first**, and press **y** to copy the selected highlighted content.

---> a) this is the first item.
b)

NOTE: you can also use **y** as an operator; **yw** yanks one word.

-- VISUAL -- 802,21 82%

- Step 3 Input **j\$** to move the cursor to the end of the next line and press **p** to paste the copied content.

---> a) this is the first item.
b) this is the

- Step 4 Press **a** to append content after the cursor, input **second**, and press **Esc** to exit the insert mode.

---> a) this is the first item.
b) this is the second

- Step 5 In visual mode, select **item.** in the **a)** line, press **y** to copy the content, press **j\$** to move the cursor to the end of the next line, and press **p** to paste the copied content.

---> a) this is the first item.
b) this is the second item.

----End

3.8.5 Setting Options

- Step 1 Move the cursor downwards to view all the contents of lesson 6.5.

Lesson 6.5: SET OPTION

```
** Set an option so a search or substitute ignores case **
```

1. Search for 'ignore' by entering: /ignore <ENTER>
Repeat several times by pressing n .
2. Set the 'ic' (Ignore case) option by entering: :set ic
3. Now search for 'ignore' again by pressing n
Notice that Ignore and IGNORE are now also found.
4. Set the 'hlsearch' and 'incsearch' options: :set hls is
5. Now type the search command again and see what happens: /ignore <ENTER>
6. To disable ignoring case enter: :set noic

NOTE: To remove the highlighting of matches enter: :nohlsearch

NOTE: If you want to ignore case for just one search command, use \c
in the phrase: /ignore\c <ENTER>

Step 2 Input **/ignore** and press **Enter**. Then, press **n** to perform multiple searches, and press **Shift+n** to perform the backward search.

Lesson 6 SUMMARY

1. Type o to open a line BELOW the cursor and start Insert mode.
Type O to open a line ABOVE the cursor.
2. Type a to insert text AFTER the cursor.
Type A to insert text after the end of the line.
3. The e command moves to the end of a word.
4. The y operator yanks (copies) text, p puts (pastes) it.
5. Typing a capital R enters Replace mode until <ESC> is pressed.
6. Typing ":set xxx" sets the option "xxx". Some options are:
'ic' 'ignorecase' ignore upper/lower case when searching
'is' 'incsearch' show partial matches for a search phrase
'hls' 'hlsearch' highlight all matching phrases
You can either use the long or the short option name.

Step 3 Input **:set ic** and press **Enter**. Then, press **n** to perform multiple searches, and press **Shift+n** to perform the backward search. The search is case insensitive. Both **Ignore** and **IGNORE** can be found.

3. Now search for 'ignore' again by pressing n
Notice that Ignore and IGNORE are now also found.
4. Set the 'hlsearch' and 'incsearch' options: :set hls is

```
/ignore 818,29 84%
```

Step 4 Input **:set hls is** and press **Enter** to highlight the search results. Press **n** to perform multiple searches, and press **Shift+n** to perform the backward search.

3. Now search for 'ignore' again by pressing n
Notice that Ignore and IGNORE are now also found.

- Step 5 To cancel case-insensitive search, input :set noic. To unhighlight search results, input :nohlsearch.

----End

3.9 Using the Vim Help Document

3.9.1 Using the Vim Help Document

- Step 1 Move the cursor downwards to view all the contents of lesson 7.1.

```
Lesson 7.1: GETTING HELP

** Use the on-line help system **

Vim has a comprehensive on-line help system. To get started, try one of
these three:
- press the <HELP> key (if you have one)
- press the <F1> key (if you have one)
- type :help <ENTER>

Read the text in the help window to find out how the help works.
Type CTRL-W CTRL-W to jump from one window to another.
Type :q <ENTER> to close the help window.

You can find help on just about any subject, by giving an argument to the
":help" command. Try these (don't forget pressing <ENTER>):

:help w
:help c_CTRL-D
:help insert-index
:help user-manual
```

- Step 2 Input :wq and press **Enter** to save and exit vimtutor.

- Step 3 Run the following command to open vim.

```
[root@ecs-openeuler ~]# vim
```

- Step 4 Input :help and press **Enter**.

```
VIM - Vi IMproved
version 8.1.450
by Bram Moolenaar et al.
Vim is open source and freely distributable

Sponsor Vim development!
type :help sponsor<Enter> for information

type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info

:help
```

Step 5 Enter the main interface of the **help.txt** file.

```
help.txt      For Vim version 8.1.  Last change: 2017 Oct 28

          VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           k
              "j" to go down, "k" to go up, "l" to go right.   h l
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!). j

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-[.
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back:       Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

          WHAT          PREPEND      EXAMPLE
Normal mode command :help x
help.txt [Help] [RO]   1,1          Top

[No Name]           0,0-1        All
"help.txt" [readonly] 228L, 8583C
```

Step 6 Move the cursor to the **bars** position in green in line 10 and column 57.

```
help.txt      For Vim version 8.1.  Last change: 2017 Oct 28

          VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           k
              "j" to go down, "k" to go up, "l" to go right.   h l
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!). j

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-[.
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back:       Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

          WHAT          PREPEND      EXAMPLE
Normal mode command :help x
help.txt [Help] [RO]   10,57        Top

[No Name]           0,0-1        All
"help.txt" [readonly] 228L, 8583C
```

Step 7 Press **Ctrl+]** to go to the **bars** tag.

```

pi_zip.txt      Zip archive explorer

LOCAL ADDITIONS:                                local-additions

-----[bars]----- Bars example

Now that you've jumped here with CTRL-] or a double mouse click, you can use
CTRL-T, CTRL-O, g<RightMouse>, or <C-RightMouse> to go back to where you were.

Note that tags are within | characters, but when highlighting is enabled these
characters are hidden. That makes it easier to read a command.

Anyway, you can use CTRL-] on any word, also when it is not within |, and Vim
will try to find help for it. Especially for options in single quotes, e.g.
'compatible'.

-----[vim:tw=78:fo=tcq2:isk=-~,^*,^\|,^\":ts=8:noet:ft=help:norl:-----[help.txt [Help] [RO]          215,1      Bot
-----[No Name]          0,0-1      All

```

Step 8 Press **Ctrl+T** or **Ctrl+O** to return to the **bars** position.

```

help.txt      For Vim version 8.1.  Last change: 2017 Oct 28

VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           h   l
              "j" to go down, "k" to go up, "l" to go right.       j
Close this window: Use ":q<Enter>".
Get out of Vim:  Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-].
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back:       Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

      WHAT          PREPEND      EXAMPLE
      Normal mode command :help x
-----[help.txt [Help] [RO]          10,57      Top
-----[No Name]          0,0-1      All

```

Step 9 Input **:set mouse=a** and press **Enter**.

```

help.txt      For Vim version 8.1. Last change: 2017 Oct 28

          VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           h   l
              "j" to go down, "k" to go up, "l" to go right.       j
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!). k

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-J.
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
Double-click the left mouse button on a tag, e.g. bars.
Jump back:       Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

          WHAT          PREPEND      EXAMPLE
Normal mode command      :help x
help.txt [Help] [RO]          10,57      Top

[No Name]                  0,0-1      All
:set mouse=a

```

Step 10 Double-click **bars** to go to the **bars** tag.

```

pi_zip.txt      Zip archive explorer

LOCAL ADDITIONS:                                local-additions

-----
bars          Bars example

Now that you've jumped here with CTRL-J or a double mouse click, you can use
CTRL-T, CTRL-O, g<RightMouse>, or <C-RightMouse> to go back to where you were.

Note that tags are within | characters, but when highlighting is enabled these
characters are hidden. That makes it easier to read a command.

Anyway, you can use CTRL-J on any word, also when it is not within |, and Vim
will try to find help for it. Especially for options in single quotes, e.g.
'compatible'.

-----
vim:tw=78:fo=tcq2:isk=-~,^*,^\|,^\":ts=8:noet:ft=help:norl:
~  

help.txt [Help] [RO]          215,1      Bot

[No Name]                  0,0-1      All
:set mouse=a

```

Step 11 Input **g** and right-click to return to the **bars** position.

```
help.txt      For Vim version 8.1. Last change: 2017 Oct 28

          VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           k
              "j" to go down, "k" to go up, "l" to go right.      h l
Close this window: Use ":q<Enter>".                         j
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!). 

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-J.
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag, e.g. bars.
Jump back:       Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

          WHAT          PREPEND      EXAMPLE
Normal mode command      :help x
help.txt [Help] [RO]          10,57      Top

[No Name]                0,0-1      All
:set mouse=a
```

- Step 12 Move the cursor to line 20 and column 60, input **:help x**, and press **Enter**. Move the cursor to view the help document of the **x** command. After viewing the help document, press **Ctrl+O** to return to the previous location.

```
For inserting text see insert.txt.

=====
1. Deleting text                                deleting E470

["x]<Del>      or      <Del> x dl
["x]x          Delete [count] characters under and after the cursor
                  [into register x] (not linewise). Does the same as
                  "dl".
The <Del> key does not take a [count]. Instead, it
deletes the last character of the count.
See :fixdel if the <Del> key does not do what you
want. See 'whichwrap' for deleting a line break
(join lines). {Vi does not support <Del>}

          x dh
["x]X          Delete [count] characters before the cursor [into
                  register x] (not linewise). Does the same as "dh".
                  Also see 'whichwrap'.

change.txt [Help] [RO]          29,26-65      1%
[No Name]                0,0-1      All
"change.txt" [readonly] 1889L, 77212C
```

- Step 13 Input **:help word** and then press **Ctrl+D** to search for the content that matches **word**.

WHAT	PREPEND	EXAMPLE
Normal mode command		
help.txt [Help] [RO]		:help x 10,57 Top
[No Name]		0,0-1 All
:help word		
word	:syn-keyword	
word-count	compl-keyword	
wordcount()	spell-midword	
word-motions	spellbadword()	
spell-wordlist-format	g:ada#Keywords	
internal-wordlist	:syn-iskeyword	
WORD	netrw-password	
ada#Word()	g:ada_omni_with_keywords	
g:ada#WordRegex	b:clojure_syntax_keywords	
spell-WORDCHARS	g:clojure_syntax_keywords	
g:clojure_special_indent_words	b:clojure_syntax_without_core_keywords	
<cword>	<cWORD>	
:<cword>	:<cWORD>	
'cinwords'	g:ada#DotWordRegex	
'lispwords'	spell-FORBIDDENWORD	
'iskeyword'	spell-COMPOUNDWORDMAX	
'keywordprg'		
:help word		

Step 14 Input **-count** and press **Enter** to view the content related to **word-count**.

{count}CTRL-G	Like CTRL-G , but prints the current file name with full path. If the count is higher than 1 the current buffer number is also given. {not in Vi}
g CTRL-G	g CTRL-G word-count byte-count Prints the current position of the cursor in five ways: Column, Line, Word, Character and Byte. If the number of Characters and Bytes is the same then the Character position is omitted. If there are characters in the line that take more than one position on the screen (<Tab> or special character), both the "real" column and the screen column are shown, separated with a dash. Also see the ' ruler ' option and the wordcount() function. {not in Vi}
{Visual}g CTRL-G	v_g_CTRL-G Similar to "g CTRL-G", but Word, Character, Line, and
editing.txt [Help] [RO]	72,17-52 3%
[No Name]	0,0-1 All
"editing.txt" [readonly] 1750L, 73179C	

Step 15 Input **g**, and then press **Ctrl+G** to display the count information.

```

{count}CTRL-G           Like CTRL-G, but prints the current file name with
                        full path. If the count is higher than 1 the current
                        buffer number is also given. {not in Vi}

g CTRL-G              Prints the current position of the cursor in five
                        ways: Column, Line, Word, Character and Byte. If the
                        number of Characters and Bytes is the same then the
                        Character position is omitted.
                        If there are characters in the line that take more
                        than one position on the screen (<Tab> or special
                        character), both the "real" column and the screen
                        column are shown, separated with a dash.
                        Also see the 'ruler' option and the wordcount()
                        function.
                        {not in Vi}

(vi)g CTRL-G          Similar to "g CTRL-G", but Word, Character, Line, and
                        editing.txt [Help] [RO]           72,17-52      3%
[No Name]               0,0-1          All
Col 17-52 of 41-76; Line 72 of 1750; Word 430 of 11843; Byte 2813 of 73179
Press ENTER or type command to continue

```

- Step 16 Press **Enter** to return, move the cursor to line 39, and press **Ctrl+]** and **Ctrl+O** to view **quickref** and **USER MANUAL**.

```

Vim command argument      -      :help -r
Option                   '      :help 'textwidth'
Regular expression        /      :help /[

See help-summary for more contexts and an explanation.

Search for help: Type ":help word", then hit CTRL-D to see matching
                  help entries for "word".
                  Or use ":helpgrep word". :helpgrep

Vim stands for Vi IMproved. Most of Vim was made by Bram Moolenaar, but only
through the help of many others. See credits.
----- doc-file-list Q_ct
BASIC:
quickref      Overview of the most common commands you will use
tutor         30 minutes training course for beginners
copying       About copyrights
iccf         Helping poor children in Uganda
sponsor      Sponsor Vim development, become a registered Vim user
www          Vim on the World Wide Web
help.txt [Help] [RO]           39,1      11%
[No Name]               0,0-1          All

```

- Step 17 Input `:q` and press **Enter** to exit the help document. Input `:q` and press **Enter** again to exit vim.

```
BASIC:
quickref          Overview of the most common commands you will use
tutor             30 minutes training course for beginners
copying           About copyrights
iccf              Helping poor children in Uganda
sponsor            Sponsor Vim development, become a registered Vim user
www               Vim on the World Wide Web
bugs               Where to send bug reports

USER MANUAL: These files explain how to accomplish an editing task.

usr_toc.txt       Table Of Contents

Getting Started
usr_01.txt        About the manuals
usr_02.txt        The first steps in Vim
usr_03.txt        Moving around
usr_04.txt        Making small changes
usr_05.txt        Set your settings
usr_06.txt        Using syntax highlighting
help.txt [Help] [RO]                                52,1      17%
[No_Name]                                              0,0-1      All
:q
```

----End

4 openEuler User and Permission Management

4.1 Overview

4.1.1 About This Exercise

This exercise introduces the file and permission management on the openEuler OS.

4.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Master the user and user group management.
- Master the file permission management.
- Master the file access control.

4.2 Configuring the Exercise

4.2.1 Procedure

4.2.1.1 User and User Group Management

Step 1 Run the **who** command to display information about the current login user.

```
[root@localhost ~]# who
root      tty1          2020-07-08 11:23
root      pts/0          2020-07-08 14:06 (172.19.130.137)
```

Step 2 Run the **id** command to display the user ID and related group ID.

```
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Step 3 Log in to the system as the **root** user, create users **tom**, **bob**, and **jack**, and set the UID to **1024** for user **jack**.

```
[root@localhost ~]# useradd tom
[root@localhost ~]# useradd bob
[root@localhost ~]# useradd -u 1024 jack
[root@localhost ~]# tail -3 /etc/passwd
```

```
tom:x:1001:1001::/home/tom:/bin/bash
bob:x:1002:1002::/home/bob:/bin/bash
jack:x:1024:1024::/home/jack:/bin/bash
[root@localhost ~]# useradd -d /home/myd bob1 # Specify the home directory for the new user.
Creating mailbox file: File exists
[root@localhost ~]# useradd -d /usr/local/apache -g apache -s /bin/false bob2
# Add a user that cannot log in to the system. Add a user bob2 whose login directory is /usr/local/apache, user group is apache, and shell is /bin/false. Set the user shell to /usr/sbin/nologin or /bin/false, indicating that the system user is not allowed to log in.
```

Step 4 Change the user name **tom** to **tony** and the home directory to **/home/tony**.

```
[root@localhost ~]# usermod -l tony tom
[root@localhost ~]# cp -r /home/tom/ /home/tony/
[root@localhost ~]# cd /home/tony/
[root@localhost tony]# cd ~
[root@localhost ~]# usermod -d /home/tony/ tony
[root@localhost ~]# tail -3 /etc/passwd
bob:x:1002:1002::/home/bob:/bin/bash
jack:x:1024:1024::/home/jack:/bin/bash
tony:x:1001:1001::/home/tony/:/bin/bash
```

Change the user **tom's** private group name **tom** to **tony**.

```
[root@localhost ~]# groupmod -n tony tom
[root@localhost ~]# tail -1 /etc/group
tony:x:1001:
```

Step 5 Delete the user **bob** and its home directory.

```
[root@localhost ~]# userdel -r bob
[root@localhost ~]# tail -2 /etc/passwd
jack:x:1024:1024::/home/jack:/bin/bash
tony:x:1001:1001::/home/tony/:/bin/bash
```

Note: You can see that the user **bob** does not exist in the last two lines in the preceding user configuration file.

The **bob** directory does not exist in the home directory.

```
[root@localhost ~]# ls /home/
jack  openeuler  tom  tony
```

Step 6 Use **sudo** to switch the user.

Switch from user **root** to user **jack** on the device.

```
[root@localhost ~]# su jack
[jack@localhost root]$ pwd
/root
[jack@localhost root]$ exit
exit
[root@localhost ~]# su - jack
[jack@localhost ~]$ pwd
/home/jack
[jack@localhost ~]$ exit
```

Question: What are the differences between the two **su** commands for switching users?

----End

4.2.1.2 Locking a User Account

Step 1 Set passwords of accounts **tony** and **jack** to **Huawei@123**, which is not displayed.

```
[root@localhost ~]# passwd tony
Changing password for user tony.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# passwd jack

Changing password for user jack.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Step 2 Then, lock the **jack** account, test the effect, and unlock the account again.

Run the following command to check the user **jack**'s account status:

```
[root@localhost ~]# passwd -S jack
jack PS 2020-07-08 0 99999 7 -1 (Password set, SHA512 crypt.)
[root@localhost ~]# passwd -l jack
Locking password for user jack.
passwd: Success
[root@localhost ~]#
[root@localhost ~]# passwd -S jack
jack LK 2020-07-08 0 99999 7 -1 (Password locked.)
[root@localhost ~]# passwd -uf jack          // openEuler hardens the security so the -f option must be
added to forcibly unlock the user.
Unlocking password for user jack.
passwd: Success
[root@localhost ~]# passwd -S jack
jack PS 2020-07-08 0 99999 7 -1 (Password set, SHA512 crypt.)
```

Step 3 Run the **chage** command to check and change the password expiration time.

Check the password expiration time.

```
[root@localhost ~]# chage -l jack
Last password change : Jul 08, 2020
Password expires      : never
Password inactive     : never
Account expires        : never
Minimum number of days between password change : 0
Maximum number of days between password change   : 99999
Number of days of warning before password expires : 7
```

Edit the password expiration time. Options are described as follows:

- **-m** specifies the minimum number of days between password changes. If it is set to **0**, a password can be changed at any time.
- **-M** specifies the maximum number of days between password changes.
- **-W** specifies the number of days to notify users before they need to change their password.
- **-E** specifies the date when an account expires. The account is unavailable after this date.
- **-d** specifies the date when the password was last changed.
- **-I** specifies the grace period after a password expires. If a password still expires after the grace period, the account will be unavailable.
- **-l** lists the current settings. Non-privileged users determine when their passwords or accounts expire.

----End

4.2.1.3 Managing User Groups

Step 1 Create a **hatest** group and add users **tony** and **jack** to the **hatest** group.

```
[root@localhost ~]# groupadd hatest
[root@localhost ~]# gpasswd -M tony,jack hatest
[root@localhost ~]# tail -1 /etc/group      # Check whether the user group is created.
hatest:x:1025:tony,jack
```

Step 2 Delete or modify the user group.

```
[root@localhost ~]# groupadd group1
[root@localhost ~]# groupadd -g 101 group2
[root@localhost ~]# groupdel group1      # Delete the user group.
[root@localhost ~]# groupmod -g 102 group2      # Change the user group ID.
[root@localhost ~]# cat /etc/group      #View the user group.
root:x:0:
bin:x:1:
daemon:x:2:
```

----End

4.2.1.4 Manually Creating Accounts and Creating Accounts in Batches

Step 1 Edit a text user file. Write each column in the format of the **/etc/passwd** password file. Note that the user name, UID, and home directory of each user must be unique. Leave the password column blank or enter **x**.

```
[root@localhost ~]# vim users.txt
user1:x:1200:1200:user001:/home/user1:/bin/bash
user2:x:1201:1201:user002:/home/user2:/bin/bash
user3:x:1202:1202:user003:/home/user3:/bin/bash
```

- Step 2 Run the **newusers** command as the **root** user to import data from the **user.txt** file to create a user.

```
[root@localhost ~]# newusers < users.txt
[root@localhost ~]# tail -3 /etc/passwd
user1:x:1200:1200:user001:/home/user1:/bin/bash
user2:x:1201:1201:user002:/home/user2:/bin/bash
user3:x:1202:1202:user003:/home/user3:/bin/bash
```

----End

4.2.1.5 Viewing Common Files Associated with Users

- Step 1 View the user account information file **/etc/passwd**.

```
[root@localhost ~]# cat /etc/passwd
root:x:0:root:/root:/bin/bash
bin:x:1:bin:/bin:/sbin/nologin
daemon:x:2:daemon:/sbin:/sbin/nologin
...
```

- Step 2 View the user account information encryption file **/etc/shadow**.

```
[root@localhost ~]# cat /etc/shadow
root:$6$4KT4vnGt0.9B/FQS$lcrlSwJmvkyFjrhPrg0Ctg.b2FbTdQx4XWqTBiuRzUN7EoRCgDkkepeLq3KXdesc
uFnHNCf.zPVt6L4..N7Mw.:18451:0:99999:7:::
bin:*:18344:0:99999:7:::
...
```

- Step 3 Check the group information file **/etc/group**.

```
[root@localhost ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
...
```

- Step 4 View the group information encryption file **/etc/gshadow**.

```
[root@localhost ~]# cat /etc/gshadow
root:::
bin:::
daemon:::
...
```

----End

4.2.1.6 Setting Permissions and Ownerships of Files and Directories

- Step 1 Create a **/test** directory as the **root** user, create **file1** and **file2** files in the directory, and check their default permissions and ownership.

```
[root@localhost ~]# mkdir test
```

```
[root@localhost ~]# cd /test
[root@localhost test]# touch file1
[root@localhost test]# touch file2
[root@localhost test]# ls -l
total 0
-rwxr-xr-x. 1 root root 0 Jul  8 15:48 file1
-rwxr-xr-x. 1 root root 0 Jul  8 15:48 file2
[root@localhost test]# ls -l / | grep test
drwxrwxrwt. 2 root root 4096 Jul  8 15:41 test
```

- Step 2 Change the **/test** directory to a public shared directory, that is, set the **t** permission bit for the directory.

```
[root@localhost test]# cd
[root@localhost ~]# chmod 1777 /test/
[root@localhost ~]# ls -l / | grep test
drwxrwxrwt. 2 root root 4096 Jul  8 15:41 test
```

Set the permission on **file1** and **file2** to **755**.

```
[root@localhost ~]# chmod 755 /test/file1 /test/file2
[root@localhost ~]# ls -l /test/
total 0
-rwxr-xr-x. 1 root root 0 Jul  8 15:41 file1
-rwxr-xr-x. 1 root root 0 Jul  8 15:41 file2
```

- Step 3 Set **file1** to be readable by all users.

```
[root@localhost test]# chmod ugo+r file1
```

- Step 4 Alternatively, run the following command to set **file1** to be readable by all users.

```
[root@localhost test]# chmod a+r file1
```

- Step 5 Set **file1** and **file2** to be writable by the file owner and users in the same group as the owner. Other users cannot write to **file1** and **file2**.

```
[root@localhost test]# chmod ug+w,o-w file1 file2
```

- Step 6 Set all files and subdirectories in the current directory to be readable by all users.

```
[root@localhost test]# chmod -R a+r *
```

- Step 7 Change the user and user group of **file1** to **jack** and **hatest** respectively.

```
[root@localhost ~]# chown jack:hatest /test/file1
[root@localhost ~]# ls -l /test/
total 0
-rwxr-xr-x. 1 root root 0 Jul  8 15:41 file1
-rwxr-xr-x. 1 root root 0 Jul  8 15:41 file2
```

- Step 8 Modify the file group attributes.

```
[root@localhost test]# chgrp -v bin file1
changed group of 'file1' from root to bin
[root@localhost test]# ll
```

```
total 4.0K
-rwxrwxr-x+ 1 root bin  0 Jul  8 15:48 file1
-rwxrwxr-x. 1 root root 0 Jul  8 15:48 file2
```

- Step 9 Run the **umask** command to check the umask value before modification.

```
[root@localhost test]# umask
0077
```

- Step 10 Run the **umask** command to modify the permission.

```
[root@localhost test]# umask 022
[root@localhost test]# umask
0022
```

----End

4.2.1.7 ACL Settings

- Step 1 Grant the write permission to the group to which **file1** belongs.

```
[root@localhost ~]# chmod 775 /test/file1
[root@localhost ~]# ls -l /test | grep file1
-rwxrwxr-x. 1 root root 0 Jul  8 15:48 file1
```

- Step 2 Configure the file ACL so that the user **tony** in the **hatest** group has only the read-only permission on the **file1**.

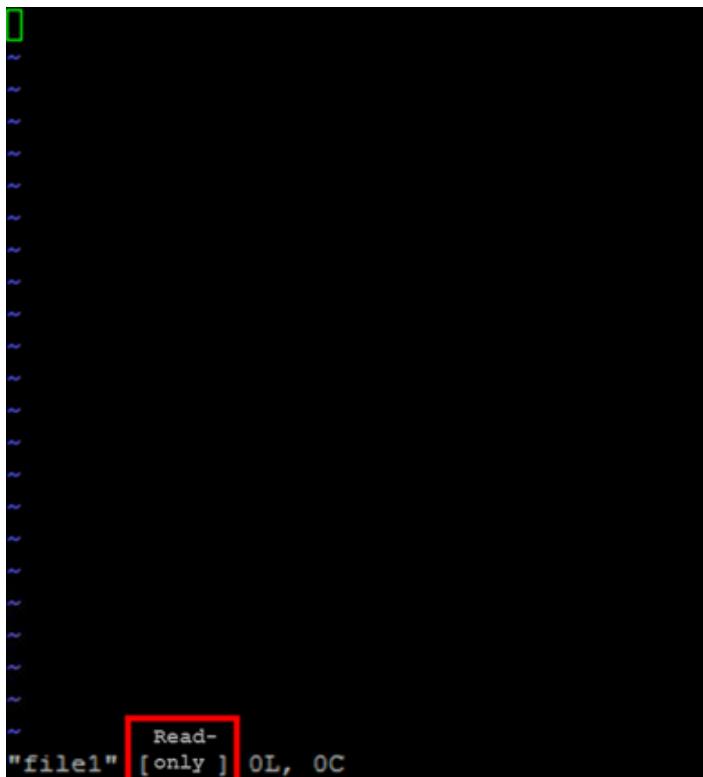
```
[root@localhost ~]# getfacl /test/file1
getfacl: Removing leading '/' from absolute path names
# file: test/file1
# owner: jack
# group: hatest
user::rwx
group::rwx
other::r-x

[root@localhost ~]# setfacl -m u:tony:r /test/file1
[root@localhost ~]# getfacl /test/file1
getfacl: Removing leading '/' from absolute path names
# file: test/file1
# owner: jack
# group: hatest
user::rwx
user:tony:r--
group::rwx
mask::rwx
other::r-x
```

- Step 3 Switch to the user **tony** and check whether the **file1** can be written.

Note: To avoid conflict during the test, the **t** bit in the **/test** directory is deleted.

```
[root@localhost ~]# chmod 777 /test/  
[root@localhost ~]# ls -l / | grep test  
drwxrwxrwx. 2 root root 4096 Jul 8 15:41 test  
[root@localhost ~]# su - tony  
[tony@localhost root]$ cd /test/  
[tony@localhost test]$ vim file1
```



The terminal window shows the command 'ls -l / | grep test' being run, listing the directory structure. The 'file1' entry under the 'test' directory is highlighted with a red box. Above the 'file1' entry, the text 'Read-only' is displayed in a small white box with a black border, also highlighted with a red box.

The file is read-only and cannot be written.

Step 4 Clear ACL settings from the **file1**.

```
[root@localhost test]# getfacl -e file1  
# file: file1  
# owner: root  
# group: bin  
user::rwx  
user:tony:r--          #effective:r--  
group::rwx             #effective:rwx  
mask::rwx  
other::r-x  
  
[root@localhost test]# chacl -B file1  
[root@localhost test]# getfacl -e file1  
# file: file1  
# owner: root  
# group: bin  
user::rwx  
group::rwx
```

other::r-x

----End

4.3 Quiz

- Create a **/data** directory.
- Create users **user1**, **user2**, and **user3** based on the following requirements:
user1's home directory is **/data**. The user description is **testuser**.
user2's UID is 2000.
user3 should log in to the shell using **/bin/tcsh**.
- Add the preceding three users to the **it** group whose GID is 3000.
- Create **/it**.
- All members in the **it** group can create and delete files in the **/it** directory.
- Change the **it** group name to **cloud**.
- Set **user1**'s password validity period (days) to **15** and the notification days before password expiration to **3**.
- Configure that **user2** is not allowed to log in to the system.
- Create a directory **/test** and set the directory ACL. Only **user1** has the read and write permissions, and **user3** has the read permission.

5

openEuler Software Management

5.1 Overview

5.1.1 About the Exercises

The exercises help trainees master openEuler software management operation commands, including RPM, software source code, yum, and systemd/systemctl management service. This section consists of five exercises and describes the yum repo configuration, RPM commands, and software management in the Huawei openEuler lab environment.

- Exercise 1: using the local openEuler lab environment
- Exercise 2: configuring the yum repository
- Exercise 3: using RPM to manage software
- Exercise 4: using DNF to manage software packages
- Exercise 5: installing personal cloud drive application

5.1.2 Objectives

Upon completion of the exercises, you will be able to:

- Master the software management methods on the openEuler OS.
- Understand the differences between RPM and DNF.
- Master the DNF configuration.
- Master software installation, uninstallation, and upgrade.

5.2 Configuring the Yum Repository

Step 1 Go to the **yum** repo directory.

```
[root@openEuler ~]# cd /etc/yum.repos.d/
```

Step 2 Create a repository named **openeuler**.

```
[root@openEuler yum.repos.d]# vim openeuler_x86_64.repo
```

Step 3 Enter the following code at the end of the file, save the file, and exit.

```
[openEuler]
name=openeuler
baseurl=https://repo.openeuler.org/openEuler-20.03-LTS/everything/x86_64/
enabled=1
```

```
gpgcheck=1  
gpgkey=https://repo.openeuler.org/openEuler-20.03-LTS/everything/x86_64/
```

- Step 4 Refresh the software list.

```
[root@openEuler yum.repos.d]# yum list all
```

----End

5.3 Managing Software Through RPM

5.3.1 RPM Query Commands

- Step 1 Run the **yum** or **rpm** command to query the OpenJDK package name.

```
[root@openEuler yum.repos.d]# cd  
[root@openEuler ~]# yum list all | grep 1.8.0-openjdk
```

- Step 2 Check whether OpenJDK is installed.

```
[root@openEuler ~]# rpm -q java-1.8.0-openjdk-1.8.0.242.b08-1.h5.oe1.x86_64.rpm
```

----End

5.3.2 RPM Installation Commands

- Step 1 Download the OpenJDK and ZZIPlib installation packages.

```
[root@openEuler ~]# wget  
https://repo.openeuler.org/openEuler-20.03-LTS/everything/x86_64/Packages/java-1.8.0-openjdk-1.8.0.24  
2.b08-1.h5.oe1.x86_64.rpm  
[root@openEuler ~]# wget  
https://repo.openeuler.org/openEuler-20.03-LTS/everything/x86_64/Packages/zziplib-0.13.69-5.oe1.x86_6  
4.rpm
```

- Step 2 Use **rpm** to install OpenJDK 1.8, but the dependency issue between software packages cannot be resolved.

```
[root@openEuler ~]# rpm -ivh java-1.8.0-openjdk-1.8.0.242.b08-1.h5.oe1.x86_64.rpm
```

- Step 3 Use **yum** to install OpenJDK.

```
[root@openEuler ~]# yum -y install java-1.8.0-openjdk-1.8.0.242.b08-1.h5.oe1.x86_64.rpm
```

- Step 4 Verify the OpenJDK installation.

```
[root@openEuler ~]# java -version
```

- Step 5 Run the **rpm** command to install ZZIPlib.

```
[root@openEuler ~]# rpm -ivh zziplib-0.13.69-5.oe1.x86_64.rpm
```

----End

5.3.3 RPM Upgrade Commands

Step 1 Upgrade OpenJDK.

```
[root@openEuler ~]# rpm -Uvh java-1.8.0-openjdk-1.8.0.242.b08-1.h5.oe1.x86_64.rpm
```

Step 2 Upgrade ZZIPlib.

```
[root@openEuler ~]# rpm -Uvh zziplib-0.13.69-5.oe1.x86_64.rpm
```

----End

5.3.4 Common RPM Options

Step 1 Query the files and their full path in an installed software package.

```
[root@openEuler ~]# rpm -ql python3-libxml2-2.9.8-9.oe1.x86_64
```

Step 2 Query details about the software package.

```
[root@openEuler ~]# rpm -qi python3-libxml2-2.9.8-9.oe1.x86_64
```

----End

5.3.5 RPM Uninstallation Command

Step 1 Check whether ZZIPlib has been installed.

```
[root@openEuler ~]# rpm -qa | grep zziplib
```

Step 2 Uninstall ZZIPlib.

```
[root@openEuler ~]# rpm -e zziplib-0.13.69-5.oe1.x86_64
```

Step 3 Enter **rpm -qa | grep zziplib** again. If an error message is displayed indicating that ZZIPlib is not installed, ZZIPlib has been uninstalled.

----End

5.4 Managing Software Packages Through DNF

5.4.1 Operations

Step 1 Use WinSCP to upload **openEuler-20.03-LTS-x86_64-dvd.iso** to the **/root** directory of the **openeuler01** VM and run the following command to mount the ISO file:

```
[root@openEuler ~]# mount -o loop /root/openEuler-20.03-LTS-x86_64-dvd.iso /mnt/
```

Step 2 View the content of the DNF configuration file.

```
[root@openEuler ~]# cat /etc/dnf/dnf.conf
```

Step 3 Add a software repository.

```
[root@openEuler ~]# dnf config-manager --add-repo file:///mnt/
```

- Step 4 Run the **vim /etc/yum.repos.d/mnt_.repo** command to open the file and add the following code to the end of the file.

```
gpgcheck=1  
gpgkey=file:///mnt/RPM-GPG-KEY-openEuler
```

- Step 5 Verify the function of enabling and disabling a software repository.

```
[root@openEuler ~]# dnf repolist  
[root@openEuler ~]# dnf config-manager --set-disable mnt_  
[root@openEuler ~]# dnf repolist  
[root@openEuler ~]# dnf config-manager --set-enable mnt_  
[root@openEuler ~]# dnf repolist
```

- Step 6 Search for the OpenJDK software package.

```
[root@openEuler ~]# dnf search openjdk
```

- Step 7 List the software packages, installed software packages, and available software packages.

```
[root@openEuler ~]# dnf list all  
[root@openEuler ~]# dnf list installed  
[root@openEuler ~]# dnf list available
```

- Step 8 Display RPM package information.

```
[root@openEuler ~]# dnf info java-1.8.0-openjdk.x86_64
```

- Step 9 Display the managed software package groups.

```
[root@openEuler ~]# dnf groups summary  
[root@openEuler ~]# dnf group list
```

- Step 10 Display the installed software package groups.

```
[root@openEuler ~]# dnf group --installed -v
```

- Step 11 Check for updates.

```
[root@openEuler ~]# dnf check-update
```

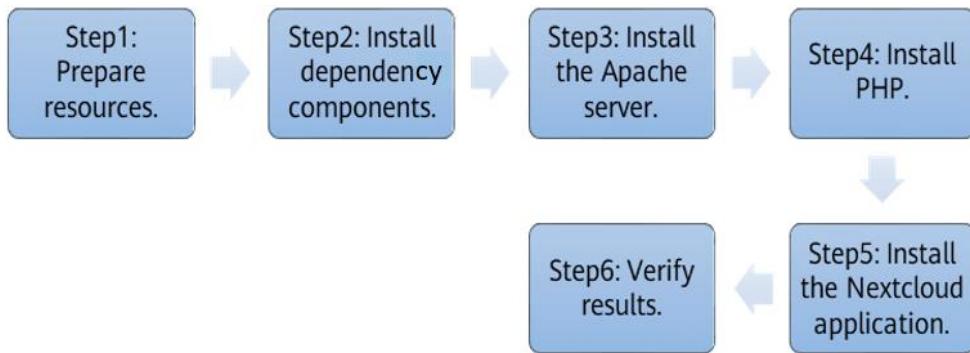
----End

5.5 Installing Personal Cloud Drive Application

5.5.1 Overview

Nextcloud is a popular open-source personal cloud drive application. In this exercise, the Nextcloud application is installed and verified on the openEuler server by compiling and installing dependent components and installing the DNF software package.

5.5.2 Procedure



5.5.3 Installing Dependency Components

Step 1 Install the dependency and basic tools.

```
[root@openEuler ~]# dnf install -y unzip curl wget bash-completion policycoreutils-python-utils mlocate bzip2
```

Note: If an error is reported during the installation, run the installation command again.

----End

5.5.4 Installing the Apache Server

Step 1 Install the Apache web server.

```
[root@openEuler ~]# dnf install -y httpd
```

Step 2 Start the Apache network service.

```
[root@openEuler ~]# systemctl enable httpd.service  
[root@openEuler ~]# systemctl start httpd.service
```

----End

5.5.5 Installing PHP

Step 1 Reset and install PHP.

```
[root@openEuler ~]# dnf install -y php php-devel
```

Step 2 Install modules required by PHP.

```
[root@openEuler ~]# dnf install -y php php-gd php-mbstring php-intl php-mysqlnd php-opcache  
php-json php-pgsql php-fpm php-dom
```

Step 3 Install CMake.

```
[root@openEuler ~]# dnf -y install cmake
```

Step 4 Install libzip using the software package.

```
[root@openEuler ~]# dnf -y install zlib-devel  
[root@openEuler ~]# wget https://nih.at/libzip/libzip-1.2.0.tar.gz
```

```
[root@openEuler ~]# tar -zvxf libzip-1.2.0.tar.gz
[root@openEuler ~]# cd libzip-1.2.0
[root@openEuler libzip-1.2.0]# ./configure
[root@openEuler libzip-1.2.0]# make -j2 && make install
```

Step 5 Install php-zip.

```
[root@openEuler libzip-1.2.0]# cp /usr/local/lib/libzip/include/zipconf.h /usr/local/include/zipconf.h
[root@openEuler libzip-1.2.0]# cd
[root@openEuler ~]# wget http://pecl.php.net/get/zip-1.19.0.tgz
[root@openEuler ~]# tar -zvxf zip-1.19.0.tgz
[root@openEuler ~]# cd zip-1.19.0
[root@openEuler zip-1.19.0]# /usr/bin/phpize
[root@openEuler zip-1.19.0]# ./configure --with-php-config=/usr/bin/php-config
[root@openEuler zip-1.19.0]# make && make install
```

Step 6 Run the **vi** command to open the **/etc/php.ini** file and add the following code under [PHP]:

```
extension=zip.so
```

Step 7 Start PHP-FPM.

```
[root@openEuler zip-1.19.0]# cd
[root@openEuler ~]# systemctl start php-fpm
```

Step 8 Verify the PHP installation version.

```
[root@openEuler ~]# php -v
[root@localhost ~]# php -v
PHP 7.2.10 (cli) (built: Mar 23 2020 21:25:23) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
```

Step 9 Verify the PHP installation modules.

```
[root@openEuler ~]# php -m
```

```
exif
fileinfo
filter
ftp
gettext
hash
iconv
libxml
openssl
pcntl
pcre
Phar
readline
Reflection
session
sockets
SPL
standard
tokenizer
zlib

[Zend Modules]
```

----End

5.5.6 Installing the Nextcloud Application

- Step 1 Go to the `/home` directory and download the Nextcloud software package.

```
[root@openEuler ~]# wget https://download.nextcloud.com/server/releases/nextcloud-18.0.4.tar.bz2
```

- Step 2 Decompress the Nextcloud software package.

```
tar -jxvf nextcloud-18.0.4.tar.bz2
```

- Step 3 Copy the folder to the root directory of the Apache web server.

```
cp -R nextcloud/ /var/www/html/
```

- Step 4 Create a data folder.

```
mkdir /var/www/html/nextcloud/data
```

- Step 5 Change Apache's read and write permissions on the **nextCloud** folder.

```
chown -R apache:apache /var/www/html/nextcloud
```

- Step 6 Restart Apache.

```
systemctl restart httpd.service
```

- Step 7 Disable the firewall service.

```
systemctl stop firewalld.service
```

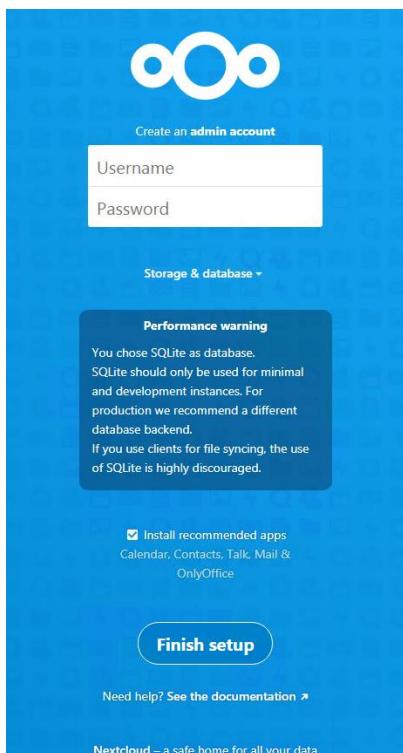
- Step 8 Disable SELinux temporarily.

```
setenforce 0
```

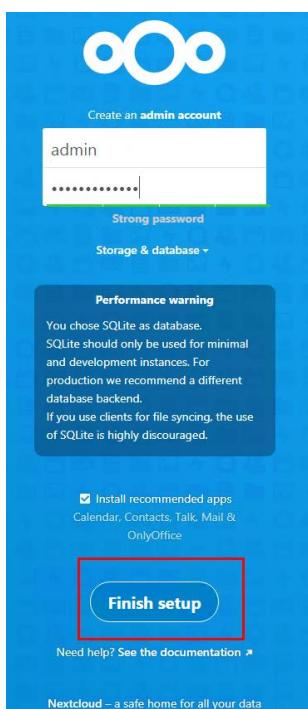
----End

5.5.7 Result Verification

- Step 1 In an address box of a local browser, enter **http://IP address/nextcloud**.



Step 2 Enter the username and password of the administrator and click **Finish setup**.



Note: If the page is being loaded for a long time, stop loading and refresh the page.

Step 3 Wait until the system initialization is complete. Click the close button in the upper right corner to go to the Nextcloud home page.

----End

6 openEuler Storage and File System Management

6.1 Overview

6.1.1 About This Exercise

In a disk OS, the storage is mainly used to store data. For an OS, a grid disk is usually partitioned and then formatted into a file system to store file data of the system. This exercise describes how to format disk partitions and how to configure logical volumes.

6.1.2 Objectives

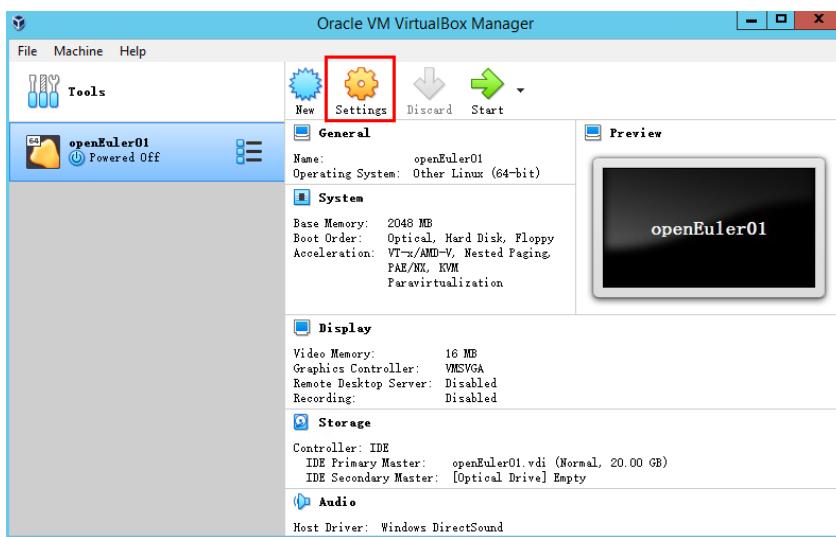
Upon completion of this exercise, you will be able to:

- Master the methods of creating the primary partition in MBR partition table mode.
- Master the methods of creating extended and logical partitions in MBR partition table mode.
- Master the partition configuration method in GPT partition table mode.
- Master the method of formatting a file system.
- Master how to mount and unmount a file system.
- Master how to mount ISO files.
- Master how to configure the **/etc/fstab** file.
- Master the procedure for creating a logical volume.
- Master the methods of expanding and reducing the capacity of logical volumes.
- Master the procedure for deleting a logical volume.

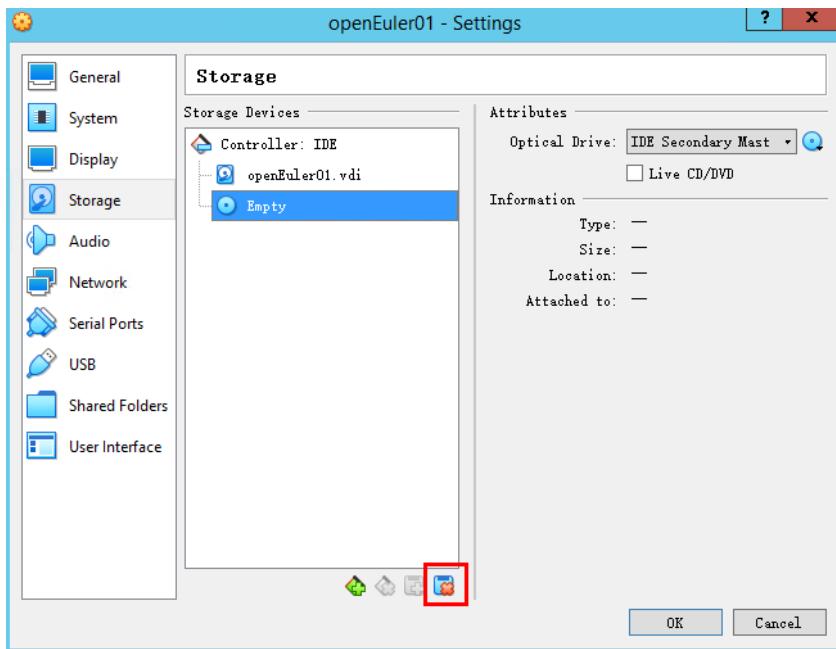
6.2 Adding a Disk

Step 1 Shut down the VM.

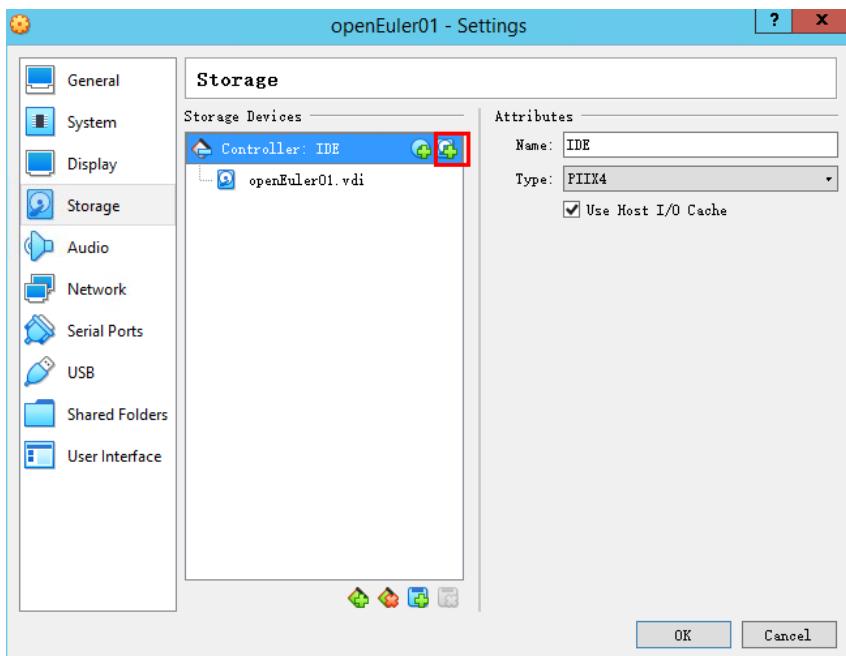
Step 2 Open the **openEuler01** VM console of VirtualBox and click **Settings**.



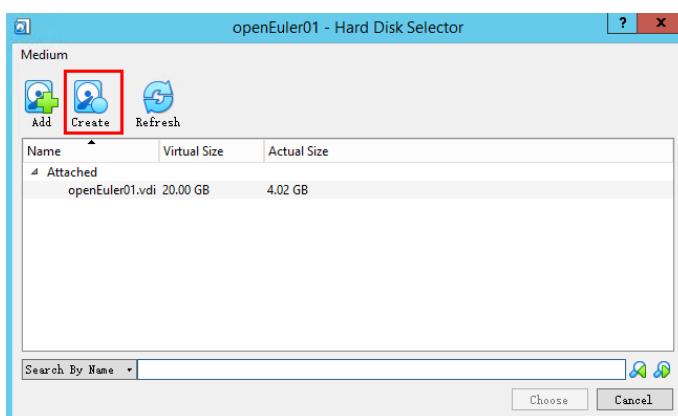
Step 3 Click the CD-ROM icon, and then click  to remove the CD-ROM.



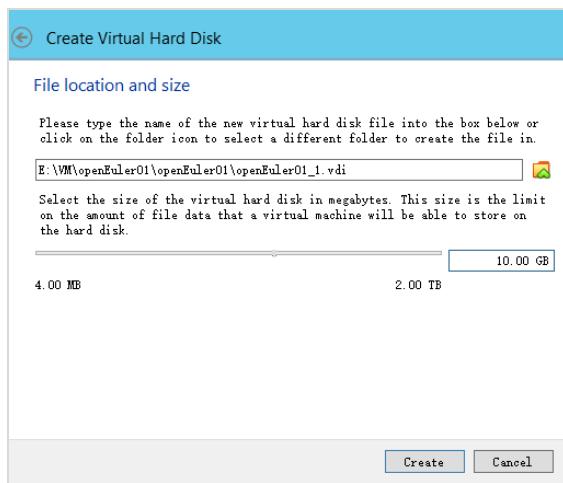
Step 4 Choose **Storage**, and then click  from **Controller: IDE**.



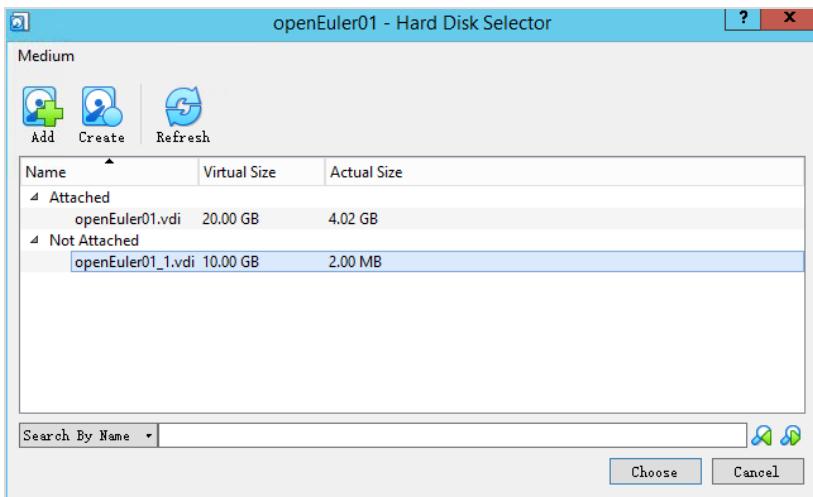
Step 5 Click **Create** in the displayed dialog box.



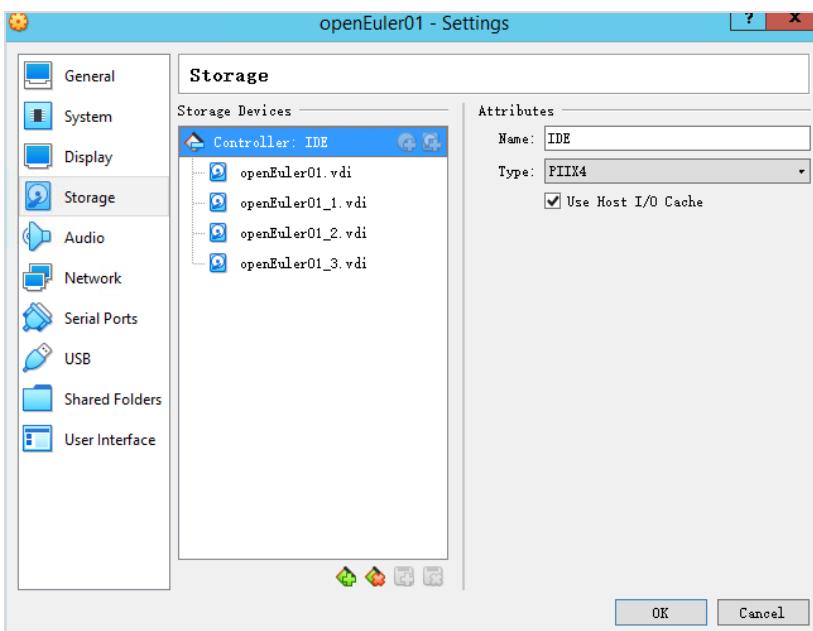
Step 6 Retain the default settings and click **Next** twice. Set the disk size to 10 GB on the **File location and size** page, and click **Create**.



Step 7 Select the disk created in the **Hard Disk Selector** window, and click **Choose**.



Step 8 Create another two disks according to Step 4 to Step 7 and attach them to the VM. The VM now has four disks.



Step 9 Click **OK** to close the **Settings** window, and click **Start** to start the **openEuler01** VM. After the VM is started, use PuTTY to log in to the VM.

Step 10 Run the **fdisk -l** command to view the local disk information. You will see three additional 10 GB disks (**/dev/sdb** to **/dev/sdd**).

```
[root@localhost ~]# fdisk -l | grep /dev/
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
/dev/sda1      2048    411647   409600  200M EFI System
/dev/sda2    411648  2508799  2097152    1G Linux filesystem
/dev/sda3  2508800 83884031 81375232 38.8G Linux LVM
Disk /dev/mapper/openeuler-root: 34.82 GiB, 37367054336 bytes, 72982528 sectors
Disk /dev/mapper/openeuler-swap: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
```

```
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors  
Disk /dev/sdd: 10 GiB, 10737418240 bytes, 20971520 sectors
```

----End

6.3 Disk Partition Management in MBR Partition Table Mode

6.3.1 Creating a Primary Partition

Step 1 Perform the following operations on the **/dev/sdb** disk:

```
[root@localhost ~]# fdisk /dev/sdb          # fdisk is in the MBR-based partition table mode by default.  
  
Welcome to fdisk (util-linux 2.34).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Device does not contain a recognized partition table.  
Created a new DOS disklabel with disk identifier 0xc5edf2a0.  
  
Command (m for help): m      # If you do not know how to perform this operation, enter m to view the help information.  
Help:  
  
DOS (MBR)  
a    toggle a bootable flag  
b    edit nested BSD disklabel  
c    toggle the dos compatibility flag  
  
Generic  
d    delete a partition  
F    list free unpartitioned space  
l    list known partition types  
n    add a new partition      # You can enter n to add a new partition.  
p    print the partition table  
t    change a partition type  
v    verify the partition table  
i    print information about a partition  
  
Misc  
m    print this menu  
u    change display/entry units  
x    extra functionality (experts only)  
  
Script  
l    load disk layout from sfdisk script file  
O    dump disk layout to sfdisk script file  
  
Save & Exit
```

```
w  write table to disk and exit
q  quit without saving changes
```

Create a new label

```
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table
```

Command (m for help): **n** # Add a new partition.

Partition type

```
p  primary (0 primary, 0 extended, 4 free)
e  extended (container for logical partitions)
```

Select (default p): # The default partition is the primary partition. You can retain it and press **Enter**.

Using default response p.

Partition number (1-4, default 1): # Set this parameter to the partition number. By default, the partition number starts from 1 in ascending order. In MBR mode, there can be only four primary partitions. You can retain the default value 1 and press **Enter**.

First sector (2048-20971519, default 2048): # Enter the start cylinder of the partition or use the default value and press **Enter**.

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): **+2G** # Enter the size of the partition. There are three methods. You can select **+2G**, which means to create a partition with the size of 2 GB.

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): **w** # Enter **w** to save the partition table configuration and exit.

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

```
[root@localhost ~]# fdisk -l /dev/vbd
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc5edf2a0

Device      Boot Start      End Sectors Size Id Type
/dev/vdb1        2048 4196351 4194304 2G 83 Linux      # Create the first partition of 2 GB.
```

----End

6.3.2 Creating Extended and Logical Partitions

Step 1 Perform the following operations to configure an extended partition and a logical partition for **/dev/vbd**:

```
[root@localhost ~]# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.34).
```

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

```
Command (m for help): n # Create a partition.
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p): e # Set the partition type as an extended partition.
Partition number (2-4, default 2): # Retain the default partition number 2.
First sector (4196352-20971519, default 4196352):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4196352-20971519, default 20971519): # Retain the
default value to allocate all space.
```

Created a new partition 2 of type 'Extended' and of size 8 GiB.

```
Command (m for help): n # Create a logical partition.
All space for primary partitions is in use.
Adding logical partition 5 # Create logical partition 5 when the space allocated to the primary
partition is used up.
First sector (4198400-20971519, default 4198400): # Retain the default value.
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4198400-20971519, default 20971519): +3G # Create
a logical partition of 3 GB.
```

Created a new partition 5 of type 'Linux' and of size 3 GiB.

```
Command (m for help): w # Save the partition table configuration and exit.
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@localhost ~]# fdisk -l /dev/sdb # View information about the /dev/sdb disk.
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc5edf2a0

Device      Boot   Start     End   Sectors Size Id Type
/dev/sdb1        2048 4196351 4194304   2G 83 Linux
/dev/sdb2        4196352 20971519 16775168   8 G  5 Extended # An extended partition.
/dev/sdb5        4198400 10489855 6291456   3G 83 Linux # A logical partition allocated in the
extended partition.
```

----End

6.3.3 Changing the Type of a Partition

Step 1 Perform the following operations to change the type of the **/dev/sdb5** partition:

```
[root@localhost ~]# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.34).
```

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

```
Command (m for help): t      # Change the partition type.
Partition number (1,2,5, default 5):      # Retain default mode and change the fifth partition.
Hex code (type L to list all codes): L      # List the partition type and change the code.

0  Empty          24  NEC DOS          81  Minix / old Lin      bf  Solaris
1  FAT12         27  Hidden NTFS Win  82  Linux swap / So     c1  DRDOS/sec (FAT-
2  XENIX root    39  Plan 9          83  Linux             c4  DRDOS/sec (FAT-
3  XENIX usr     3c  PartitionMagic   84  OS/2 hidden or    c6  DRDOS/sec (FAT-
4  FAT16 <32M    40  Venix 80286    85  Linux extended    c7  Syrinx
5  Extended       41  PPC PreP Boot   86  NTFS volume set   da  Non-FS data
6  FAT16          42  SFS             87  NTFS volume set   db  CP/M / CTOS /
7  HPFS/NTFS/exFAT 4d  QNX4.x        88  Linux plaintext   de  Dell Utility
8  AIX            4e  QNX4.x 2nd part  8e  Linux LVM          df  BootIt
9  AIX bootable   4f  QNX4.x 3rd part 93  Amoeba           e1  DOS access
a  OS/2 Boot Manag 50  OnTrack DM   94  Amoeba BBT        e3  DOS R/O
b  W95 FAT32     51  OnTrack DM6 Aux 9f  BSD/OS           e4  SpeedStor
c  W95 FAT32 (LBA) 52  CP/M          a0  IBM Thinkpad hi  ea  Rufus alignment
e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux  a5  FreeBSD          eb  BeOS fs
f  W95 Ext'd (LBA) 54  OnTrackDM6   a6  OpenBSD          ee  GPT
10 OPUS          55  EZ-Drive        a7  NeXTSTEP         ef  EFI (FAT-12/16/
11 Hidden FAT12    56  Golden Bow    a8  Darwin UFS        f0  Linux/PA-RISC b
12 Compaq diagnos 5c  Priam Edisk   a9  NetBSD           f1  SpeedStor
14 Hidden FAT16 <3 61  SpeedStor    ab  Darwin boot      f4  SpeedStor
16 Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      f2  DOS secondary
17 Hidden HPFS/NTF 64  Novell Netware b7  BSDI fs          fb  VMware VMFS
18 AST SmartSleep  65  Novell Netware b8  BSDI swap       fc  VMware VMKCORE
1b Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fd  Linux raid auto
1c Hidden W95 FAT3 75  PC/IX          bc  Acronis FAT32 L fe  LANstep
1e Hidden W95 FAT1 80  Old Minix    be  Solaris boot     ff  BBT
```

The Hex code (type L to list all codes): **8e** # Set the partition type to Linux LVM, which will be used in subsequent operations.

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): **p** # Display the information about the current partition.

Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xc5edf2a0

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	4196351	4194304	2G	83	Linux
/dev/sdb2		4196352	20971519	16775168	8G	5	Extended
/dev/vdb5		4198400	10489855	6291456	3G	8e	Linux LVM

The partition type has changed to Linux LVM.

Command (m for help): **w**

The partition table has been altered.

Calling ioctl() to re-read partition table.
Syncing disks.

----End

6.3.4 Deleting a Partition

Step 1 Perform the following operations to delete the **/dev/sdb1** partition:

```
[root@localhost ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): d      # Delete a partition.
Partition number (1,2,5, default 5): 1      # Select the number of the partition to be deleted, that is, 1.

Partition 1 has been deleted.

Command (m for help): p  # View information about existing partitions.
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc5edf2a0

Device      Boot   Start     End   Sectors Size Id Type      # No /dev/sdb1 exists now.
/dev/sdb2        4196352 20971519 16775168   8G  5 Extended
/dev/sdb5        4198400 10489855  6291456   3G  8e Linux LVM

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

----End

6.4 Managing Disk Partitions in GPT Partition Table Mode

6.4.1 Using parted to Create a Partition in Interactive Mode

Step 1 Perform the following operations to partition the **/dev/sdc** disk:

```
[root@localhost ~]# parted /dev/sdc # Use the parted partition command.
GNU Parted 3.3
Using /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) help  # If you do not know how to operate, enter help to view the help information.
```

```

align-check TYPE N           check partition N for TYPE(min|opt) alignment
help [COMMAND]              print general help, or help on COMMAND
mklabel,mktable LABEL-TYPE create a new disklabel (partition table)
mkpart PART-TYPE [FS-TYPE] START END   make a partition
name NUMBER NAME            name partition NUMBER as NAME
print [devices|free|list,all|NUMBER]      display the partition table, available devices, free space, all
found partitions, or a particular partition
quit                         exit program
rescue START END            rescue a lost partition near START and END
resizepart NUMBER END       resize partition NUMBER
rm NUMBER                   delete partition NUMBER
select DEVICE                choose the device to edit
disk_set FLAG STATE         change the FLAG on selected device
disk_toggle [FLAG]          toggle the state of FLAG on selected device
set NUMBER FLAG STATE       change the FLAG on partition NUMBER
toggle [NUMBER [FLAG]]      toggle the state of FLAG on partition NUMBER
unit UNIT                   set the default unit to UNIT
version                     display the version number and copyright information of
GNU Parted
(parted) mklabel gpt # Set the format of the disk partition table to GPT.
Warning: The existing disk label on /dev/sdc will be destroyed and all data on this disk will be lost. Do you
want
to continue?
Yes/No? yes # Confirm the operation.
(parted) mkpart # Create a new partition.
Partition name? []? gpt1 # Set the partition name.
File system type? [ext2]? xfs # Set the file system type used during partition formatting.
Start? 0 KB # Set the start position of the partition.
End? 2 GB # Set the end position of the partition.
Warning: You requested a partition from 0.00B to 2000MB (sectors 0..3906250).
The closest location we can manage is 17.4kB to 2000MB (sectors 34..3906250).
Is this still acceptable to you?
Yes/No? yes # Confirm the operation.
Warning: The resulting partition is not properly aligned for best performance: 34s % 2048s != 0s
Ignore/Cancel? Ignore # Ignore the alarm.
(parted) print # Output the partition information.
Model: Virtio Block Device (virtblk)
Disk /dev/sdc: 10.7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      17.4kB  2000 MB  2000MB  xfs        gpt1  # The partition created in the preceding
steps.
(parted) quit
Information: You may need to update /etc/fstab.

```

----End

6.4.2 Creating a Partition in Non-interactive Mode

Step 1 Create a partition.

```
[root@localhost ~]# parted /dev/sdc mkpart gpt2 2001M 5G      # Create a partition and set the  
start and end positions.  
Information: You may need to update /etc/fstab.
```

```
[root@localhost ~]# parted /dev/sdc p    # Output partition information.  
Model: Virtio Block Device (virtblk)  
Disk /dev/sdc: 10.7GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:  
  
Number  Start   End     Size    File system  Name  Flags  
1        17.4kB  2000MB  2000MB          gpt1  
2        2001 MB  5000MB  2999 MB         gpt2  # The partition created in the preceding  
steps.  
  
# The partition table format of the /dev/sdc disk has been set to GPT. Do not need to set the partition  
table format again. For a new disk, run the following command:  
[root@localhost ~]# parted /dev/sdc mklabel gpt
```

----End

6.4.3 Deleting a Partition

Step 1 Delete the **/dev/sdc1** partition.

```
[root@localhost ~]# parted /dev/sdc rm 1  
Information: You may need to update /etc/fstab.  
  
[root@localhost ~]# parted /dev/sdc p  
Model: Virtio Block Device (virtblk)  
Disk /dev/sdc: 10.7GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:  
  
Number  Start   End     Size    File system  Name  Flags  
2        2001MB  5000MB  2999MB          gpt2
```

----End

6.5 Formatting and Mounting

6.5.1 Formatting a File System

Step 1 Format the file system.

```
[root@localhost ~]# mkfs -t xfs /dev/sdc2  
meta-data=/dev/sdc2          isize=512    agcount=4, agsize=183040 blks  
                      =           sectsz=512  attr=2, projid32bit=1  
                      =           crc=1     finobt=1, sparse=1, rmapbt=0
```

```

        =
        reflink=0
data   =
        bsize=4096  blocks=732160, imaxpct=25
        =
        sunit=0    swidth=0 blks
naming =version 2           bsize=4096  ascii-ci=0, ftype=1
log    =internal log        bsize=4096  blocks=2560, version=2
        =
        sectsz=512  sunit=0 blks, lazy-count=1
realtime =none              extsz=4096  blocks=0, rtextents=0
[root@localhost ~]# parted /dev/sdc2 p      # Check partition details.
Model: Virtio Block Device (virtblk)
Disk /dev/sdc2: 2999MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start   End     Size   File system  Flags
1       0.00B  2999 MB  2999MB  xfs          # The file system is XFS.

```

----End

6.5.2 Mounting a File System

Step 1 Mount a file system.

```

[root@localhost ~]# mkdir /mnt/xfs01      # Create a mount point for a file system.
[root@localhost ~]# mount /dev/sdc2 /mnt/xfs01/      # Mount a file system.
[root@localhost ~]# df -h  # Check the system mounting status.
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.2G    0  1.2G  0% /dev
tmpfs          1.5G    0  1.5G  0% /dev/shm
tmpfs          1.5G  19M  1.5G  2% /run
tmpfs          1.5G    0  1.5G  0% /sys/fs/cgroup
/dev/mapper/openeuler-root  35G  2.8G  30G  9% /
tmpfs          1.5G  64K  1.5G  1% /tmp
/dev/sda2       976M 113M  796M 13% /boot
/dev/sda1       200M  5.8M  195M  3% /boot/efi
tmpfs          298M    0  298M  0% /run/user/0
/dev/sdc2       2.8G  36M  2.8G  2% /mnt/xfs01
[root@localhost ~]# mount | grep /dev/sdc2      # Check the file system mounting status.
/dev/sdc2 on /mnt/xfs01 type xfs (rw,relatime,seclabel,attr2,inode64,noquota)

```

----End

6.5.3 Mounting an ISO File

Step 1 Use WinSCP to upload **openEuler-20.03-LTS-x86_64-dvd.iso** to the **/root** directory of the **openEuler01** VM.

Step 2 Mount the ISO file.

```

[root@localhost ~]# mkdir /mnt/cdrom      # Create a mount point.
[root@localhost ~]# mount openEuler-20.03-LTS-x86_64-dvd.iso /mnt/cdrom/
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
# Mount the ISO file. Some systems can be mounted only after the [-o loop] option is added.

```

```
[root@localhost ~]# cd /mnt/cdrom/  
[root@localhost cdrom]# ls      # Check the data in the disk.  
boot.catalog  EFI      Packages  RPM-GPG-KEY-openEuler  
docs          images   repodata  TRANS.TBL
```

----End

6.5.4 Setting Automatic Disk Mounting Upon Startup

Step 1 Check the UUID of the partition (**/dev/sdc2** is used as an example).

```
[root@localhost ~]# blkid /dev/sdc2  
/dev/sdc2: UUID="ab72ab7d-93e6-4857-8a82-54374473d44b" TYPE="xfs" PARTLABEL="gpt2"  
PARTUUID="27d21c27-db60-455e-b455-37e04ff4ab4b"  
# Remember the UUID, which will be used later. The UUID varies depending on the environment.
```

Step 2 Configure automatic mounting upon system startup in the **/etc/fstab** file.

```
[root@localhost ~]# df -h | grep /dev/vdc # Check the mounting status of /dev/sdc2.  
/dev/sdc2              2.8G  36M  2.8G  2% /mnt/xfs01  
[root@localhost ~]# umount -a      # Unmount all additional mount points.  
[root@localhost ~]# df -h | grep /dev/sdc      # /dev/sdc2 is not mounted.  
[root@localhost ~]# vim /etc/fstab      # Edit the /etc/fstab file to configure automatic mounting.  
# Add the following information to the last line of the file:  
UUID=ab72ab7d-93e6-4857-8a82-54374473d44b  /mnt/xfs01  xfs  defaults  0 0  
# Use the UUID obtained in Step 1. After the writing is complete, save the file and exit.  
[root@localhost ~]# mount -a      # Mount all devices.  
[root@localhost ~]# df -h | grep /dev/sdc      # Check the device mounting status.  
/dev/sdc2              2.8G  36M  2.8G  2% /mnt/xfs01
```

----End

6.6 Logical Volume Management

6.6.1 Creating and Formatting a Logical Volume

Step 1 Create a logical volume (LV).

```
[root@openEuler ~]# pvcreate /dev/sdb5  
Physical volume "/dev/sdb5" successfully created.  
[root@openEuler ~]# pvdisplay  
"/dev/sdb5" is a new physical volume of "3.00 GiB"  
--- NEW Physical volume ---  
PV Name          /dev/sdb5  
VG Name  
PV Size          3.00 GiB  
Allocatable      NO  
PE Size          0  
Total PE         0  
Free PE          0  
Allocated PE     0
```

```
PV UUID           wSgzbu-oFkH-rxL7-tt7e-QQA4-RZsX-rC2BSb
[root@openEuler ~]# vgcreate testvg /dev/sdb5
  Volume group "testvg" successfully created
[root@openEuler ~]# vgdiskl testvg
--- Volume group ---
VG Name          testvg
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 1
VG Access        read/write
VG Status         resizable
MAX LV            0
Cur LV            0
Open LV            0
Max PV            0
Cur PV            1
Act PV            1
VG Size           <3.00 GiB
PE Size           4.00 MiB
Total PE          767
Alloc PE / Size  0 / 0
Free  PE / Size  767 / <3.00 GiB
VG UUID          O63DUX-eojV-iUfw-75ib-yeZP-N0xx-JXyOeY [root@openEuler ~]# lvcreate
-L 2G -n testlv testvg
Logical volume "testlv" created.
[root@openEuler ~]# lvdisplay /dev/testvg/testlv
--- Logical volume ---
LV Path           /dev/testvg/testlv
LV Name           testlv
VG Name           testvg
LV UUID           wfSIth-O3cf-9pxa-emMW-BSwl-ZfWQ-uS6vYL
LV Write Access   read/write
LV Creation host, time openEuler, 2020-07-02 09:24:41 +0800
LV Status         available
# open            0
LV Size           2.00 GiB
Current LE        512
Segments          1
Allocation        inherit
Read ahead sectors auto
- currently set to 8192
Block device      253:22
```

Step 2 Format and mount the LV.

```
[root@openEuler ~]# mkfs.ext4 /dev/testvg/testlv
mke2fs 1.45.3 (14-Jul-2019)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 59f52483-742c-41bd-b83b-3edb9c315611
Superblock backups stored on blocks:
            32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
[root@openEuler ~]# mkdir /mnt/testlv
[root@openEuler ~]# mount /dev/testvg/testlv /mnt/testlv/
[root@openEuler ~]# df -h | grep testvg
/dev/mapper/testvg-testlv 2.0G 6.0M 1.8G 1% /mnt/testlv
```

----End

6.6.2 Expanding and Reducing the Capacity of an LV

Step 1 Expand the LV and file system.

```
[root@openEuler ~]# fdisk /dev/sdd      # Create a 3 GB LVM partition.

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc5aa0a0a.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-20971519, default 2048):
Last sector, +/sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): +3G

Created a new partition 1 of type 'Linux' and of size 3 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): p
Disk /dev/sdd: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc5aa0a0a

Device     Boot Start      End Sectors Size Id Type
/dev/sdd1        2048 6293503 6291456   3G 8e Linux LVM
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
[root@openEuler ~]# pvcreate /dev/sdd1
  Physical volume "/dev/sdd1" successfully created.
[root@openEuler ~]# vgextend testvg /dev/sdd1 # Extend only when the PE sizes are the same.
  Volume group "testvg" successfully extended
[root@openEuler ~]# vgdisplay testvg
--- Volume group ---
VG Name          testvg
System ID
Format          lvm2
Metadata Areas   2
Metadata Sequence No  3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV           1
Open LV           1
Max PV           0
Cur PV           2
Act PV           2
VG Size          5.99 GiB
PE Size          4.00 MiB
Total PE         1534
Alloc PE / Size  512 / 2.00 GiB
Free  PE / Size  1022 / 3.99 GiB
VG UUID          O63DUX-eojV-iUfw-75ib-yeZP-N0xx-JXyOeY
[root@openEuler ~]# pvs
PV      VG      Fmt Attr PSize  PFree
/dev/sda3  openeuler lvm2 a--  38.80g    0
/dev/sdb5  testvg   lvm2 a-- <3.00g 1020.00m
/dev/sdd1  testvg   lvm2 a-- <3.00g    <3.00g
[root@openEuler ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
openeuler  1   2   0 wz--n- 38.80g    0
testvg  2  1  0 wz--n- 5.99g  3.99g # The LV has been expanded to 6 GB.
[root@openEuler ~]# lvs /dev/testvg/testlv # View testlv information.
  LV      VG      Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  testlv testvg -wi-ao----  2.00g
[root@openEuler ~]# lvextend -L +3G /dev/testvg/testlv
  Size of logical volume testvg/testlv changed from 2.00 GiB (512 extents) to 5.00 GiB (1280 extents).
  Logical volume testvg/testlv successfully resized.
[root@openEuler ~]# lvs /dev/testvg/testlv
  LV      VG      Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  testlv testvg -wi-ao----  5.00g
[root@openEuler ~]# resize2fs /dev/testvg/testlv
resize2fs 1.45.3 (14-Jul-2019)
Filesystem at /dev/testvg/testlv is mounted on /mnt/testlv; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/testvg/testlv is now 1310720 (4k) blocks long.
[root@openEuler ~]# df -h /dev/testvg/testlv
Filesystem            Size  Used  Avail Use% Mounted on
/dev/mapper/testvg-testlv 4.9G 8.0M 4.7G 1% /mnt/testlv # The file system is expanded to 5 GB.
```

- Step 2 Reduce the capacity of the file system and LV. (High-risk operation warning. Be careful when running the following commands.)

```
[root@openEuler ~]# umount /mnt/testlv      # Unmount a file system.  
[root@openEuler ~]# e2fsck -f /dev/testvg/testlv      # Check the file system usage.  
e2fsck 1.45.3 (14-Jul-2019)  
Pass 1: Checking inodes, blocks, and sizes  
Pass 2: Checking directory structure  
Pass 3: Checking directory connectivity  
Pass 4: Checking reference counts  
Pass 5: Checking group summary information  
/dev/testvg/testlv: 11/327680 files (0.0% non-contiguous), 39006/1310720 blocks  
[root@openEuler ~]# resize2fs /dev/testvg/testlv 2G      # Re-specify the logical size of the file  
system.  
resize2fs 1.45.3 (14-Jul-2019)  
Resizing the filesystem on /dev/testvg/testlv to 524288 (4k) blocks.  
The filesystem on /dev/testvg/testlv is now 524288 (4k) blocks long.  
[root@openEuler ~]# lvs /dev/testvg/testlv  
  LV   VG   Attr   LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert  
The testlv testvg -wi-a---- 5.00g      # The LV is not reduced.  
[root@openEuler ~]# lvchange -a n /dev/testvg/testlv      # Change the status of an LV to inactive.  
[root@openEuler ~]# lvreduce -L 2G /dev/testvg/testlv      # LV is reduced to 2 GB.  
  Size of logical volume testvg/testlv changed from 5.00 GiB (1280 extents) to 2.00 GiB (512 extents).  
  Logical volume testvg/testlv successfully resized.  
[root@openEuler ~]# lvchange -a y /dev/testvg/testlv      # Change the status of an LV to active.  
[root@openEuler ~]# lvs /dev/testvg/testlv      # Check the LV capacity.  
  LV   VG   Attr   LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert  
  testlv testvg -wi-a----  2.00g  
[root@openEuler ~]# e2fsck -f /dev/testvg/testlv      # Check the file system.  
e2fsck 1.45.3 (14-Jul-2019)  
Pass 1: Checking inodes, blocks, and sizes  
Pass 2: Checking directory structure  
Pass 3: Checking directory connectivity  
Pass 4: Checking reference counts  
Pass 5: Checking group summary information  
/dev/testvg/testlv: 11/131072 files (0.0% non-contiguous), 26156/524288 blocks  
[root@openEuler ~]# mount /dev/testvg/testlv /mnt/testlv/      # Remount the directory.  
[root@openEuler ~]# df -h /dev/testvg/testlv      # Check the file size.  
Filesystem           Size   Used  Avail Use% Mounted on  
/dev/mapper/testvg-testlv  2.0G  6.0M  1.8G  1% /mnt/testlv
```

- Step 3 Delete the created LVM configuration.

```
[root@openEuler ~]# umount /mnt/testlv  
[root@openEuler ~]# lvremove -y /dev/testvg/testlv  
  Logical volume "testlv" successfully removed  
[root@openEuler ~]# vgremove testvg      # Remove a VG.  
  Volume group "testvg" successfully removed  
[root@openEuler ~]# pvremove /dev/sdb5 /dev/sdd1      # Removing a PV  
  Labels on physical volume "/dev/sdb5" successfully wiped.  
  Labels on physical volume "/dev/sdd1" successfully wiped.
```

- Task: Create an LV. The required PE size is 8 MB and the LV contains 30 LEs.

----End

7 openEuler System and Process Management

7.1 Overview

7.1.1 About This Exercise

This exercise introduces the scheduled tasks, system management, and service management on the openEuler OS.

7.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Master the system task management methods.
- Master the openEuler network management configuration.
- Master the service management of the openEuler OS.

7.2 Task Management

7.2.1 Temporary Task Management

Step 1 Use PuTTY to log in to the VM.

Step 2 Add a one-time task and press **Ctrl+D**.

```
[root@openEuler ~]# at now+5min
warning: commands will be executed using /bin/sh
at> echo "aaa" >> /tmp/at.log
at> echo "bbb" >> /tmp/at.log
at> date >> /tmp/at.log
at>      # Press Ctrl+D.
job 1 at Thu Jul  2 11:21:00 2020
[root@openEuler ~]# at 22:00
warning: commands will be executed using /bin/sh
at> pwd >> /tmp/check.log
at>      # Press Ctrl+D.
job 2 at Thu Jul  2 22:00:00 2020
```

Step 3 Query the task list.

```
[root@openeuler ~]# atq    # The number on the left indicates the task ID.  
2      Thu Jul  2 22:00:00 2020 a root  
1      Thu Jul  2 11:21:00 2020 a root
```

Step 4 Check the task details.

```
[root@openEuler ~]# at -c 1  
#!/bin/sh  
# atrun uid=0 gid=0  
# mail root 0  
umask 77  
SHELL=/bin/bash; export SHELL  
HISTCONTROL=; export HISTCONTROL  
HISTSIZE=1000; export HISTSIZE  
HOSTNAME=openEuler; export HOSTNAME  
HISTTIMEFORMAT=; export HISTTIMEFORMAT  
GOMP_CPU_AFFINITY=0-1; export GOMP_CPU_AFFINITY  
PWD=/root; export PWD  
LOGNAME=root; export LOGNAME  
XDG_SESSION_TYPE=tty; export XDG_SESSION_TYPE  
HOME=/root; export HOME  
SSH_ASKPASS=/usr/libexec.openssh/gnome-ssh-askpass; export SSH_ASKPASS  
LANG=en_US.UTF-8; export LANG  
PROMPT_COMMAND=openEuler_history; export PROMPT_COMMAND  
SSH_CONNECTION=172.19.130.180\ 51500\ 192.168.110.248\ 22; export SSH_CONNECTION  
XDG_SESSION_CLASS=user; export XDG_SESSION_CLASS  
SELINUX_ROLE_REQUESTED=; export SELINUX_ROLE_REQUESTED  
USER=root; export USER  
SELINUX_USE_CURRENT_RANGE=; export SELINUX_USE_CURRENT_RANGE  
SHLVL=1; export SHLVL  
XDG_SESSION_ID=3; export XDG_SESSION_ID  
XDG_RUNTIME_DIR=/run/user/0; export XDG_RUNTIME_DIR  
SSH_CLIENT=172.19.130.180\ 51500\ 22; export SSH_CLIENT  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin; export PATH  
SELINUX_LEVEL_REQUESTED=; export SELINUX_LEVEL_REQUESTED  
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/0/bus; export DBUS_SESSION_BUS_ADDRESS  
MAIL=/var/spool/mail/root; export MAIL  
SSH_TTY=/dev/pts/0; export SSH_TTY  
OLDPWD=/root; export OLDPWD  
cd /root || {  
    echo 'Execution directory inaccessible' >&2  
    exit 1  
}  
${SHELL:-/bin/sh} << 'marcinDELIMITER42ada33e'  
echo "aaa" >> /tmp/at.log  
echo "bbb" >> /tmp/at.log  
date >> /tmp/at.log  
marcinDELIMITER42ada33e
```

Step 5 Delete the temporary task.

```
[root@openEuler ~]# atrm 2  
[root@openEuler ~]# atq # If the preceding steps are performed slowly, no task may be found.  
The [root@openEuler ~]# cat /tmp/at.log # View the execution result of the first temporary task.  
aaa
```

```
bbb  
Thu Jul 2 11:21:00 CST 2020
```

----End

7.2.2 Periodic Task Management

Step 1 Perform the following operations to manage periodic tasks:

```
[root@openeuler ~]# crontab -l # Query the cron scheduled task of the current user.  
no crontab for root  
[root@openeuler ~]#crontab -e # crontab will open an editor. Enter the following content in the editor,  
save the file, and exit.  
5 * * * * date >> /tmp/croncheck.log  
*/2 * * * * id >> /tmp/cronuser.log  
[root@openeuler ~]# crontab -l # Query the cron scheduled task of the current user.  
5 * * * * date >> /tmp/croncheck.log  
*/2 * * * * id >> /tmp/cronuser.log  
[root@openEuler ~]# crontab -r # Delete all scheduled tasks of the current user.  
[root@openEuler ~]# crontab -l  
no crontab for root
```

- Task: Create a scheduled task to record the current system time in the **/mnt/test.txt** file every five minutes from 01:00 to 02:00 from Monday to Friday.

----End

7.3 Network Management

7.3.1 Host Name Management

Step 1 Query the host name.

```
[root@openEuler ~]# hostname  
openEuler  
[root@openEuler ~]# cat /etc/hostname # This file is the configuration file of the host name.  
openEuler
```

Step 2 Temporarily change the host name.

```
[root@openEuler ~]# hostname huawei # The host name is temporarily changed and becomes invalid  
after the host is restarted.  
[root@openEuler ~]# hostname  
huawei  
[root@openEuler ~]# bash # Reopen a session.  
Welcome to 4.19.90-2003.4.0.0036.oe1.x86_64  
System information as of time: Thu Jul 2 11:45:09 CST 2020  
System load: 0.00  
Processes: 178  
Memory used: 18.5%  
Swap used: 0.0%
```

```
Usage On:      9%
IP address:   192.168.110.248
Users online: 2
[root@huawei ~]#  # The host name is changed to huawei.
[root@huawei ~]# exit
exit
[root@openEuler ~]#
```

Step 3 Perform the following operations to change the host name permanently:

Method 1:

```
[root@openEuler ~]# hostnamectl set-hostname huawei  # This method does not require a restart. You
only need to log in again.
```

Method 2:

```
[root@openEuler ~]# vim /etc/hostname
# Change the host name in the file, save the change, and exit. The host name is changed only after the
system is restarted.
[root@openEuler ~]# reboot
```

----End

7.3.2 Network Management

Step 1 Change the IP address of the network adapter.

```
[root@openEuler ~]# ip addr show  # Display the IP address of the current host.
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether 28:6e:d4:8a:2b:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.247/24 brd 192.168.110.255 scope global dynamic noprefixroute enp4s0
        valid_lft 847084sec preferred_lft 847084sec
    inet6 fe80::382e:b477:10e8:f0b6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
    default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group
    default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
[root@openEuler ~]# ip addr add 192.168.110.100/24 dev enp4s0
[root@openEuler ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
```

```
valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 28:6e:d4:8a:2b:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.247/24 brd 192.168.110.255 scope global dynamic noprefixroute enp4s0
        valid_lft 846990sec preferred_lft 846990sec
    inet 192.168.110.100/24 scope global secondary enp4s0
        valid_lft forever preferred_lft forever
    inet6 fe80::382e:b477:10e8:f0b6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group
default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
[root@openEuler ~]# ip addr del 192.168.110.100/24 dev enp4s0
[root@openEuler ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 28:6e:d4:8a:2b:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.247/24 brd 192.168.110.255 scope global dynamic noprefixroute enp4s0
        valid_lft 846884sec preferred_lft 846884sec
    inet6 fe80::382e:b477:10e8:f0b6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group
default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group
default qlen 1000
    link/ether 52:54:00:69:ec:cd brd ff:ff:ff:ff:ff:ff
```

Step 2 Configure a static route.

```
[root@openEuler ~]# ip route
default via 192.168.110.254 dev enp4s0 proto dhcp metric 100
192.168.110.0/24 dev enp4s0 proto kernel scope link src 192.168.110.247 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
[root@openEuler ~]# ip route add 192.168.2.1 via 192.168.110.254 dev enp4s0
[root@openEuler ~]# ip route
default via 192.168.110.254 dev enp4s0 proto dhcp metric 100
192.168.2.1 via 192.168.110.254 dev enp4s0
192.168.110.0/24 dev enp4s0 proto kernel scope link src 192.168.110.247 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```

Step 3 Run the **nmcli** command to change the host IP address.

```
[root@openEuler ~]# nmcli general status
STATE      CONNECTIVITY WIFI-HW WIFI      WWAN-HW WWAN
connected   full        enabled    enabled    enabled
[root@openEuler ~]# nmcli connection show
NAME        UUID                                  TYPE      DEVICE
enp4s0      90dbef32-15b2-3400-81c7-f98ece257a55  ethernet  enp4s0
virbr0      d798e4ce-2296-4b6e-9dba-9a4b994d0b82  bridge    virbr0
[root@openEuler ~]# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp4s0      ethernet  connected  enp4s0
virbr0      bridge    connected  virbr0
lo          loopback  unmanaged  --
virbr0-nic  tun      unmanaged  --
[root@openEuler ~]# nmcli con add type ethernet con-name net-static ifname enp4s0 ip4
192.168.110.10/24 gw4 192.168.110.254
[root@openEuler ~]# nmcli con up net-static ifname enp4s0
```

Step 4 Run the **nmcli** command to configure a static route.

```
[root@openEuler ~]# nmcli connection modify enp4s0 +ipv4.routes "192.168.100.0/24
192.168.110.254"
```

Step 5 Perform the following operations to change the host IP address by modifying the **ifcfg** file:

```
[root@openEuler ~]# cat /etc/sysconfig/network-scripts/ifcfg-enp4s0
TYPE=Ethernet      # Configuration file interface type.
PROXY_METHOD=none  # Proxy mode.
BROWSER_ONLY=no   # Browse only.
BOOTPROTO=dhcp    # System startup address protocol.
DEFROUTE=yes      # Default route.
IPV4_FAILURE_FATAL=no  # Whether to perform IPv4 check.
IPV6INIT=yes     # Whether to execute IPv6.
IPV6_AUTOCONF=yes # IPv6 is automatically configured.
IPV6_DEFROUTE=yes # IPv6 default route.
IPV6_FAILURE_FATAL=no  # Whether to perform IPv6 check.
IPV6_ADDR_GEN_MODE=stable-privacy # IPv6 address generation mode.
NAME=enp4s0       # Name of the network connection.
UUID=90dbef32-15b2-3400-81c7-f98ece257a55 # Device UUID.
ONBOOT=yes        # Start with the system.
AUTOCONNECT_PRIORITY=-999      # Automatic connection priority.
DEVICE=enp4s0      # Name of the physical device.
### When configuring the network, set dhcp to static or none and add the following information:
IPADDR=192.168.100.11
NETMASK=255.255.255.0
GATEWAY=192.168.110.254
DNS1=114.114.114.114
### How to make the configuration work.
[root@openEuler ~]# ifdown enp4s0
[root@openEuler ~]# ifup enp4s0
```

Step 6 Change the **/etc/resolv.conf** file to point to the IP address of the DNS server.

```
[root@openEuler ~]# dnf -y install bind-utils
```

```
[root@openEuler ~]# vim /etc/resolv.conf
nameserver 192.168.137.1 # The file format is nameserver IP, and the IP address is the IP address of
the DNS server. # Save the change and exit.
[root@openEuler ~]# nslookup openeuler.org
Server: 192.168.137.1
Address: 192.168.137.1#53

Non-authoritative answer:
Name: openeuler.org
Address: 159.138.7.136
[root@openEuler ~]# ping openeuler.org -c 3
PING openeuler.org (159.138.7.136) 56(84) bytes of data.
64 bytes from ecs-159-138-7-136.compute.hwclouds-dns.com (159.138.7.136): icmp_seq=1 ttl=40
time=52.3 ms
64 bytes from ecs-159-138-7-136.compute.hwclouds-dns.com (159.138.7.136): icmp_seq=2 ttl=40
time=51.8 ms
64 bytes from ecs-159-138-7-136.compute.hwclouds-dns.com (159.138.7.136): icmp_seq=3 ttl=40
time=52.5 ms

--- openeuler.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 51.764/52.157/52.452/0.289 ms
```

Step 7 Modify the **/etc/hosts** file.

```
[root@openEuler ~]# vim /etc/hosts # The hosts file is used to quickly resolve IP addresses and
domain names. Add the following information to the end of the file:
159.138.7.136 server # Add this line, save the configuration, and exit.
[root@openEuler ~]# ping server -c 3
PING server (159.138.7.136) 56(84) bytes of data.
64 bytes from server (159.138.7.136): icmp_seq=1 ttl=40 time=53.3 ms
64 bytes from server (159.138.7.136): icmp_seq=2 ttl=40 time=51.3 ms
64 bytes from server (159.138.7.136): icmp_seq=3 ttl=40 time=52.4 ms

--- server ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 51.288/52.351/53.338/0.838 ms
```

----End

7.4 Firewall Management

Step 1 Check the firewall status.

```
[root@openEuler ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables: [ OK ]
[root@openEuler ~]# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Fri 2020-07-31 10:51:17 CST; 5h 17min ago
     Docs: man:firewalld(1)
 Main PID: 1908 (code=exited, status=0/SUCCESS)
```

```
Jul 08 11:19:39 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 08 11:19:40 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 31 10:51:15 openEuler systemd[1]: Stopping firewalld - dynamic firewall daemon...
Jul 31 10:51:17 openEuler systemd[1]: firewalld.service: Succeeded.
Jul 31 10:51:17 openEuler systemd[1]: Stopped firewalld - dynamic firewall daemon.
[root@openEuler ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination

#####iptables default rule chain#####
INPUT: processes inbound data packets.
OUTPUT: processes outbound data packets.
FORWARD: processes forwarded data packets.
POSTROUTING chain: Data packets are processed after route selection.
PREROUTING chain: Data packets are processed before route selection.
```

Step 2 Start the firewall.

```
[root@openEuler ~]# systemctl start firewalld.service # Enable the firewall service.
[root@openEuler ~] # firewall-cmd --version # Check the firewall version.
0.6.2
[root@openEuler ~]# firewall-cmd --help # Check the help information.
[root@openEuler ~] # firewall-cmd --state # Check the running status.
running
[root@openEuler ~]# firewall-cmd --list-all # Check the firewall configuration.
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp4s0
  sources:
  services: ssh mdns dhcpcv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Step 3 Configure the firewall bypass rules.

```
[root@openEuler ~]# firewall-cmd --panic-on # Reject all packets.
[root@openEuler ~]# firewall-cmd --panic-off # Cancel the rejection of all packets.
[root@openEuler ~]# firewall-cmd --query-panic # Check whether the request is rejected.
[root@openEuler ~]# firewall-cmd --reload # Update a firewall rule without disconnecting the firewall.
[root@openEuler ~]# firewall-cmd --zone=public --add-interface=enp4s0 # Add the network port to the zone. By default, the network port is in the public zone.
[root@openEuler ~]# firewall-cmd --set-default-zone=public # Set the default interface zone.
```

```
[root@openEuler ~]# [root@openEuler ~]# firewall-cmd --zone=public --list-ports # Check all enabled ports.  
[root@openEuler ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent # Permanently enable TCP port 8080.  
The [root@openEuler ~]# [root@openEuler ~]# firewall-cmd --zone=public --add-service=http # Start a service.  
[root@openEuler ~]# systemctl restart firewalld.service  
[root@openEuler ~]# firewall-cmd --list-all  
public (active)  
    target: default  
    icmp-block-inversion: no  
    interfaces: enp4s0  
    sources:  
    services: ssh mdns dhcpcv6-client http  
    ports: 8080/tcp  
    protocols:  
    masquerade: no  
    forward-ports:  
    source-ports:  
    icmp-blocks:  
    rich rules:
```

----End

7.5 Service Management

7.5.1 Managing System Services

Step 1 Display the current services.

```
[root@openEuler ~]# systemctl list-units --type service  
UNIT          LOAD   ACTIVE   SUB      DESCRIPTION  
atd.service    loaded  active  running  Deferred execution scheduler  
auditd.service loaded  active  running  Security Auditing Service  
chronyd.service loaded  active  running  NTP client/server  
crond.service  loaded  active  running  Command Scheduler  
dbus.service    loaded  active  running  D-Bus System Message Bus  
dkms.service   loaded  active  exited   Builds and install new kernel modules through  
dracut-shutdown.service loaded  active  exited   Restore /run/initramfs on shutdown  
firewalld.service loaded  active  running  firewalld - dynamic firewall daemon  
getty@tty1.service loaded  active  running  Getty on tty1  
gssproxy.service loaded  active  running  GSSAPI Proxy Daemon  
hwclock-save.service loaded  active  exited   Update RTC With System Clock
```

Step 2 Display the service status, such as the firewall service.

```
[root@openEuler ~]# systemctl status firewall.service # Check the service status.  
● firewalld.service - firewalld - dynamic firewall daemon  
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)  
  Active: active (running) since Mon 2020-07-06 09:52:14 CST; 5h 33min ago  
    Docs: man:firewalld(1)  
   Main PID: 1863 (firewalld)
```

```
Tasks: 2
```

```
Memory: 61.3M
```

```
CGroup: /system.slice/firewalld.service
```

```
    └─1863 /usr/bin/python3 /usr/sbin/firewalld --nofork --nopid
```

```
Jul 06 09:52:13 openEuler systemd[1]: Starting firewalld - dynamic firewall daemon...
```

```
Jul 06 09:52:14 openEuler systemd[1]: Started firewalld - dynamic firewall daemon.
```

```
[root@openEuler ~]# systemctl is-active firewall.service # Check whether the service is running.  
active
```

```
[root@openEuler ~]# systemctl is-enabled firewall.service # Check whether the service is enabled.  
enabled
```

Step 3 Terminate a service, such as the firewall service.

```
[root@openEuler ~]# systemctl stop firewalld.service  
[root@openEuler ~]# systemctl is-active firewalld.service  
inactive
```

Step 4 Restart a service, such as the firewall service.

```
[root@openEuler ~]# systemctl restart firewalld.service  
[root@openEuler ~]# systemctl is-active firewalld.service  
active
```

Step 5 Disable a service, such as the firewall service.

```
[root@openEuler ~]# systemctl disable firewalld.service  
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.  
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.  
[root@openEuler ~]# systemctl is-enabled firewalld.service  
disabled
```

Step 6 Enable a service, such as the firewall service.

```
[root@openEuler ~]# systemctl enable firewalld.service  
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service →  
/usr/lib/systemd/system/firewalld.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service →  
/usr/lib/systemd/system/firewalld.service.  
[root@openEuler ~]# systemctl is-enabled firewalld.service  
enabled
```

----End

8 Shell Script Language Basics

8.1 Overview

8.1.1 About This Exercise

Shell scripts are used for system O&M, helping significantly reduce workloads. This exercise introduces the concepts and usage of Shell scripts.

8.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Master global and local variables.
- Master the usage of location-based parameters.
- Master special characters in Shell scripts.
- Master common Shell statements.

8.2 Shell Variables

8.2.1 Definition of User Variables

Step 1 Set user-defined local variables.

```
[root@openeuler ~]# dir=/usr/tmp/  
[root@openeuler ~]# echo $dir  
/usr/tmp/  
[root@openeuler ~]# today=Sunday  
[root@openeuler ~]# echo $today  
Sunday  
[root@openeuler ~]# echo $Today  
//No command output is displayed.  
[root@openeuler ~]# str="Happy New Year!"  
[root@openeuler ~]# echo "Wish You $str"  
Wish You Happy New Year!  
### As an interactive input method, you can use the read command to read data from the standard input  
(that is, the keyboard) and assign the data to a specified variable. The general format is as follows: read  
Variable 1 [Variable 2...]  
[root@openeuler ~]# read name  
openeuler //Enter openeuler.  
[root@openeuler ~]# echo $name  
openeuler
```

```
[root@openeuler ~]# read a b c
kunpeng 2020 huawei      # Enter kunpeng 2020 huawei.
[root@openeuler ~]# echo $a
kunpeng
[root@openeuler ~]# echo $b
2020
[root@openeuler ~]# echo $c
huawei
```

Step 2 Perform the following operations to configure user variables:

```
[root@openEuler ~]# vim .bash_profile      # ~/.bash_profile is a configuration file exclusively used by a
user.
# Add a line to the end of the file, save the modification, and exit.
a=1000
When the [root@openEuler ~]# echo $a      # The value a does not change when the file is not updated.
kunpeng
After the [root@openEuler ~]# source .bash_profile  # The value a changes when the variable file is
updated.
[root@openEuler ~]# echo $a
1000
[root@openEuler ~]# bash  #Open a new bash.
[root@openEuler ~]# echo $a      #a variable is not defined.

[root@openEuler ~]# exit  #Exit the newly opened Shell.
[root@openEuler ~]# su - root  # Switch the user by logging in to the system. In this case,
the .bash_profile file is automatically read.
The value of [root@openEuler ~]# echo $a # a takes effect.
1000
[root@openEuler ~]# exit  # Exit the current login Shell.
```

Step 3 Perform the following operations to configure system environment variables:

```
[root@openEuler ~]# vim/etc/profile      # /etc/profile is the configuration file of the system and is
shared by all users.
# Add a line to the end of the file, enter the following information, save the file, and exit:
b=2000
The [root@openEuler ~]# echo $b          # The variable is not updated.
2020
[root@openEuler ~]# source /etc/profile      # Update variables.
[root@openEuler ~]# echo $b
2000          # The variable value is changed.
[root@openEuler ~]# bash      # Open a new bash.
[root@openEuler ~]# echo $b      # b variable is not defined
[root@openEuler ~]# exit      # Exit the newly opened Shell.
[root@openEuler ~]# su - root  # Switch the user by logging in to the system. In this case,
the .bash_profile file is automatically read.
The value of [root@openEuler ~]# echo $b # b takes effect.
2000
[root@openEuler ~]# useradd huawei  # Add a user.
[root@openEuler ~]# su - huawei  # Switch to the user.
[huawei@openEuler ~]$ echo $b
2000  # Variable b takes effect.
[huawei@openEuler ~]$ exit  # Exit the current user.
logout
```

----End

8.2.2 Location Parameters

Step 1 Create three files *****m1.c**, **m2.c**, and **ex1.sh***** in the **/root** directory. The content of the files is as follows:

```
m1.c:  
main()  
{  
printf("Begin \n");  
}  
  
m2.c:  
# include < stdio.h >  
{  
printf("OK! \n");  
}  
  
ex1.sh:  
#!/bin/bash  
# ex1.sh: shell script to combine files and count lines  
cat $1 $2 $3 $4 $5 $6 $7 $8 $9 | wc -l  
# end
```

Step 2 Run the **ex1.sh** script.

```
[root@openeuler ~]# chmod u+x ex1.sh  
[root@openeuler ~]# sh ex1.sh m1.c m2.c  
8
```

----End

8.3 Special Characters in Shell

Step 1 Create an **ex3.sh** script and add the following content to the script:

```
#!/bin/bash  
echo "current directory is `pwd`"  
echo "home directory is $HOME"  
echo "file*.?"  
echo " directory '$HOME'"
```

Step 2 Run the **ex3.sh** script.

```
[root@openeuler ~]# chmod u+x ex3.sh  
[root@openeuler ~]# sh ex3.sh  
current directory is 'pwd'  
home directory is /root  
file*.?
```

```
directory '/root'
```

- Other special variables in Linux

Variable	Description
\$0	Name of the current script file.
\$n	Parameter passed to a script or function. <i>n</i> is a number, indicating the sequence number of the parameter. For example, the first parameter is \$1, and the second parameter is \$2.
\$#	Number of parameters passed to a script or function.
\$*	All parameters passed to a script or function.
\$@	All parameters passed to a script or function. When it is enclosed in double quotation marks (""), it is slightly different from \$*, which will be described in the following.
\$?	Exit status of the previous command or the return value of the function.
\$\$	ID of the current Shell process. For Shell scripts, the value is the ID of the process where the scripts are located.

----End

8.4 Condition Judgment and Loop Structure

8.4.1 if Statement

8.4.1.1 Syntax Example

Step 1 Create an **ex4.sh** script and add the following content to the script:

```
#!/bin/bash
a=3
b=$1
if [ $a == $b ]
then
    echo "You win!"
else
    echo "Please guess again."
fi
```

Step 2 Run the **ex4.sh** script.

```
[root@openeuler ~]# chmod u+x ex4.sh
[root@openeuler ~]# ./ex4.sh 3
You win!
[root@openeuler ~]# ./ex4.sh 4
Please guess again.
```

----End

8.4.1.2 [Appendix] Common Judgment of the if Statement

- File/Directory judgment
 - [-a FILE] If *FILE* exists, **true** is returned.
 - [-b FILE] If *FILE* exists and is a block file, **true** is returned.
 - [-c FILE] If *FILE* exists and is a character file, **true** is returned.
 - [-d FILE] If *FILE* exists and is a directory, **true** is returned.
 - [-e FILE] If the specified file or directory exists, **true** is returned.
 - [-f FILE] If *FILE* exists and is a common file, **true** is returned.
 - [-g FILE] If *FILE* exists and SGID is set, **true** is returned.
 - [-h FILE] If *FILE* exists and is a symbolic link file, **true** is returned. (This option is invalid in some old systems.)
 - [-k FILE] If *FILE* exists and the sticky bit has been set, **true** is returned.
 - [-p FILE] If *FILE* exists and is a command pipeline, **true** is returned.
 - [-r FILE] If *FILE* exists and is readable, **true** is returned.
 - [-s FILE] If *FILE* exists and its size is not 0, **true** is returned.
 - [-u FILE] If *FILE* exists and the SUID bit is set, **true** is returned.
 - [-w FILE] If *FILE* exists and is writable, **true** is returned. (A directory must be executable so that its content can be accessed.)
 - [-x FILE] If *FILE* exists and is executable, **true** is returned.
 - [-O FILE] If *FILE* exists and belongs to a valid user ID, **true** is returned.
 - [-G FILE] If *FILE* exists and the default group is the current group, **true** is returned. (Only the default group is checked.)
 - [-L FILE] If *FILE* exists and is a symbolic link, **true** is returned.
 - [-N FILE] If *FILE* exists and has been modified since it was last read, **true** is returned.
 - [-S FILE] If *FILE* exists and is a socket, **true** is returned.
 - [FILE1 -nt FILE2] If *FILE1* is later than *FILE2* or *FILE1* exists but *FILE2* does not exist, **true** is returned.
 - [FILE1 -ot FILE2] If *FILE1* is older than *FILE2* or *FILE2* exists but *FILE1* does not exist, **true** is returned.
 - [FILE1 -ef FILE2] If *FILE1* and *FILE2* point to the same device and node ID, **true** is returned.
- Character string judgment
 - [-z STRING] If the length of *STRING* is zero, **true** is returned, that is, null indicates **true**.
 - [-n STRING] If the length of *STRING* is not 0, the return value is **true**.
 - [STRING1] If the character string is not null, **true** is returned, which is similar to **-n**.
 - [STRING1 == STRING2] If the two strings are the same, **true** is returned.

[STRING1 != STRING2] If the two strings are different, **true** is returned.

[STRING1 < STRING2] If *STRING1* is listed before *STRING2*, **true** is returned.

[STRING1 > STRING2] If *STRING1* is listed after *STRING2*, **true** is returned.

- Value judgment

[INT1 -eq INT2] If the values of *INT1* and *INT2* are the same, **true** is returned.

If the values of [INT1 -ne INT2] *INT1* and *INT2* are not equal, **true** is returned.

If the value of [INT1 -gt INT2] *INT1* is greater than *INT2*, **true** is returned.

If the value of [INT1 -ge INT2] *INT1* is greater than or equal to *INT2*, **true** is returned.

If the value of [INT1 -lt INT2] *INT1* is less than *INT2*, **true** is returned.

If the value of [INT1 -le INT2] *INT1* is less than or equal to *INT2*, **true** is returned.

- Logic judgment.

[! EXPR] The logic is NOT. If *EXPR* is **false**, **true** is returned.

[EXPR1 -a EXPR2] The logic is AND. If EXPR1 and EXPR2 are **true**, **true** is returned.

[EXPR1 -o EXPR2] The logic is OR. If EXPR1 or EXPR2 is **true**, **true** is returned.

[] || [] Use OR to combine two conditions.

[] && [] Use AND to combine two conditions.

8.4.1.3 Quiz

- Write a script to determine whether the current user is **root**.
- Write a script that can determine whether today is a day off.

8.4.2 Test Statement

Step 1 Create an **ex5.sh** script and add the following content to the script:

```
#!/bin/bash
echo "Please enter your filename:"
read filename
if [ -f "$filename" ]
then cat $filename
elif [ -d "$filename" ]
then cd $filename
    pwd
    ls -l -a
else echo "$filename:bad filename"
fi
```

Step 2 Run the **ex5.sh** script.

```
[root@openeuler ~]# chmod u+x ex5.sh
[root@openeuler ~]# mkdir test
[root@openeuler ~]# sh ex5.sh
Please enter your filename:
test
/root/test
```

```
total 8
drwx----- 2 root root 4096 Feb 18 11:29 .
dr-xr-x--- 3 root root 4096 Feb 18 11:30 ..
[root@openeuler ~]# sh ex5.sh
Please enter your filename:
ex3.sh
echo "current directory is 'pwd'"
echo "home directory is $HOME"
echo "file*.?"
echo "directory '$HOME'"
```

----End

8.4.3 while Statement

Step 1 Create an **ex6.sh** script and add the following content to the script:

```
#!/bin/bash
while [ $1 ]
do
    if [ -f $1 ]
    then echo "display:$1"
        cat $1
    else echo "$1 is not a file name"
    fi
    shift
done
```

Step 2 Run the **ex6.sh** script.

```
[root@openeuler ~]# sh ex6.sh ex3.sh
display:ex3.sh
echo "current directory is 'pwd'"
echo "home directory is $HOME"
echo "file*.?"
echo "directory '$HOME'"
[root@openeuler ~]# sh ex6.sh a
a is not a file name
```

----End

8.4.4 for Statement

Step 1 Create an **ex7.sh** script and add the following content to the script:

```
#!/bin/bash
for Num in {1..10}
do
    echo $Num
done
```

Step 2 Run the **ex7.sh** script.

```
[root@openeuler ~]# sh ex7.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

- Step 3 Create a **namefile** file and write the following content to the file:

```
user1  
user2  
user3  
user4
```

- Step 4 Create an **ex8.sh** script and add the following content to the script:

```
#!/bin/bash  
for Name in $(cat ./namefile)  
do  
    echo $Name  
done
```

- Step 5 Run the **ex8.sh** script.

```
[root@openeuler ~]# sh ex8.sh  
user1  
user2  
user3  
user4
```

----End

8.4.5 case Statement

- Step 1 Create an **ex9.sh** script and add the following content to the script:

```
#!/bin/bash  
echo 'Input a number between 1 to 4'  
printf 'Your number is:\n'  
read aNum  
case $aNum in  
    1) echo 'You select 1'  
    ;;  
    2) echo 'You select 2'  
    ;;  
    3) echo 'You select 3'  
    ;;  
    4) echo 'You select 4'  
    ;;
```

```
*) echo 'You do not select a number between 1 to 4'  
;;  
esac
```

Step 2 Run the **ex9.sh** script.

```
[root@openEuler ~]# sh ex9.sh  
Input a number between 1 to 4  
Your number is:  
1  
You select 1
```

----End

8.5 Quiz

- Obtain an inverse output.
 - If you enter **yes**, **no** is displayed.
 - If you enter **no**, **yes** is displayed.
 - When the user enters other information, the user is prompted to enter **yes** or **no**.
 - Case-insensitive
- Create a user.
 - Create a username list **namedfile**.
 - Create a script to automatically create a user based on the **namefile** file and generate a random password. After the user is created, import the username and password to **/root/loginname.txt**.
 - To generate a random password, run the **openssl rand -base64 6** command.
- Check whether a user exists and whether the user is a superuser.
 - Write a script. If the specified user exists, the script indicates that the user exists, displays the ID and Shell of the user, and checks whether the user is a superuser. Otherwise, the user is created and its ID is displayed.

9

openEuler Comprehensive Practices

9.1 Overview

9.1.1 About This Exercise

In this exercise, a file sharing server is set up on the openEuler OS, and the file sharing server is started using process control commands.

9.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Understand the working principles of a file sharing server.
- Understand the installation method of a file sharing server.
- Master the methods of managing file sharing services.

9.2 Configuring the Exercise

9.2.1 Configuration Roadmap

1. After logging in to the openEuler OS, configure the software installation source on the openEuler OS so that the system can download and install the file sharing server software.
2. After the Samba file sharing service is installed and deployed, start related services and verify that the services can be accessed on the client.

9.2.2 Procedure

Step 1 Mount the system ISO file and configure the local DNF source.

Step 2 Configure the software installation source.

Use PuTTY to log in to the openEuler OS, and enter the system IP address, system account, and password.

After logging in to openEuler, check the system status.

Authorized users only. All activities may be monitored and reported.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Jul 29 09:47:22 2020 from 172.19.130.180

```
Welcome to 4.19.90-2003.4.0.0036.oe1.x86_64
System information as of time: Fri Jul 31 10:14:49 CST 2020
System load: 0.00
Processes: 191
Memory used: 21.2%
Swap used: 0.0%
Usage On: 15%
IP address: 192.168.110.246
Users online: 3

[root@openEuler ~]# uname -a
Linux openEuler 4.19.90-2003.4.0.0036.oe1.x86_64 #1 SMP Mon Mar 23 19:06:43 UTC 2020 x86_64 x86_64
x86_64 GNU/Linux
```

Check whether the software source is installed successfully. You can search for the software.

```
[root@openEuler ~]# dnf search samba
Last metadata expiration check: 0:14:40 ago on Mon 14 Sep 2020 04:47:11 PM CST.
=====
Name Exactly Matched: samba
=====
samba.x86_64 : A suite for Linux to interoperate with Windows
=====
Name & Summary Matched: samba
=====
samba-libs.x86_64 : Libraries for samba
samba-help.x86_64 : Help package for samba
pcp-pmda-samba.x86_64 : PCP metrics for Samba
samba-client.x86_64 : Client package for samba
samba-common-tools.x86_64 : Tools package for samba
samba-winbind.x86_64 : The winbind package for samba
samba-winbind-modules.x86_64 : The winbind modules for samba
samba-common.x86_64 : Common package for samba client and server
samba-winbind-clients.x86_64 : The winbind client package for samba
=====
Summary Matched: samba
=====
libwbclient.x86_64 : The winbind client library for samba
```

Step 3 Install the file sharing service.

Run the following command to install the Samba service and related components:

```
[root@openEuler ~]# dnf -y install samba samba-client samba-common
```

Start the Samba service and set it to start upon system startup.

```
[root@openEuler ~]# systemctl start smb;systemctl enable smb
Created symlink /etc/systemd/system/multi-user.target.wants/smb.service →
/usr/lib/systemd/system/smb.service.
```

Check the listening status of the server. The server is listening on TCP ports 139 and 445.

```
[root@openEuler ~]# netstat -lantp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 0.0.0.0:111              0.0.0.0:*           LISTEN     1276/rpcbind
```

tcp	0	0 192.168.122.1:53	0.0.0.0:*	LISTEN	3674/dnsmasq
tcp	0	0 0.0.0.0:22	0.0.0.0:*	LISTEN	5632/sshd
tcp	0	0 0.0.0.0:445	0.0.0.0:*	LISTEN	755432/smbd
tcp	0	0 0.0.0.0:44321	0.0.0.0:*	LISTEN	2585/pmcld
tcp	0	0 0.0.0.0:4330	0.0.0.0:*	LISTEN	739788/pmllogger
tcp	0	0 0.0.0.0:139	0.0.0.0:*	LISTEN	755432/smbd
tcp	0	0 192.168.110.246:22	172.19.130.180:60842	ESTABLISHED	753195/sshd:
root [
tcp	0	64 192.168.110.246:22	172.19.130.180:56950	ESTABLISHED	686088/sshd:
root [
tcp6	0	0 ::111	::*	LISTEN	1276/rpcbind
tcp6	0	0 ::22	::*	LISTEN	5632/sshd
tcp6	0	0 ::445	::*	LISTEN	755432/smbd
tcp6	0	0 ::44321	::*	LISTEN	2585/pmcld
tcp6	0	0 ::4330	::*	LISTEN	739788/pmllogger
tcp6	0	0 ::139	::*	LISTEN	755432/smbd

Check whether the firewall is enabled. If yes, disable the firewall.

```
[root@openEuler ~]# systemctl stop firewalld; systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Fri 2020-07-31 10:51:17 CST; 16ms ago
     Docs: man:firewalld(1)
   Process: 1908 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited,
status=0/SUCCESS)
   Main PID: 1908 (code=exited, status=0/SUCCESS)

Jul 08 11:19:39 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 08 11:19:40 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 31 10:51:15 openEuler systemd[1]: Stopping firewalld - dynamic firewall daemon...
Jul 31 10:51:17 openEuler systemd[1]: firewalld.service: Succeeded.
Jul 31 10:51:17 openEuler systemd[1]: Stopped firewalld - dynamic firewall daemon.
[root@openEuler ~]# setenforce 0    # Temporarily disable seLinux.
```

Run the **smbclient** command to check the service sharing status. Enter the password of the **root** user. The current service is running.

```
[root@openEuler ~]# smbclient -L localhost
Enter SAMBA\root's password:
Anonymous login successful

      Sharename          Type          Comment
      -----          ----          -----
      print$            Disk          Printer Drivers
      IPC$              IPC           IPC Service (Samba 4.11.6)
SMB1 disabled -- no workgroup available
```

----End

9.3 Setting File Sharing Users and Permissions

9.3.1 Overview

9.3.1.1 About This Exercise

In this exercise, you can create files and users on the openEuler OS and configure the file sharing service to enable clients to access the file sharing server.

9.3.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Understand how to configure the file sharing server.
- Understand how to control permissions on the file sharing service.

9.3.2 Configuring the Exercise

9.3.2.1 Configuration Roadmap

1. On the openEuler OS, you can add users and files to provide user authentication and storage space for the Samba file sharing server.
2. After the Samba file sharing service is installed and deployed, configure the Samba service. Then, users can access the Samba service.

9.3.2.2 Procedure

Step 1 Add a Samba user.

Run the **useradd** command to add the **smb** user and set shell-free login for the user.

```
[root@openEuler ~]# useradd -s /sbin/nologin -M smb
```

Set the Samba server password of the **smb** user, for example, **Huawei12#\$**.

```
[root@openEuler02 samba]# smbpasswd -a smb
New SMB password:
Retype new SMB password:
```

Added user smb.

Step 2 Create a shared directory.

Create a shared file directory **share** for public sharing and create a user file directory **smb**.

```
[root@openEuler ~]# mkdir /var/share /var/smb
```

Change the permission on the **share** and **smb** directories to **777**.

```
[root@openEuler ~]# chmod 777 /var/share /var/smb
```

Step 3 Add the Samba public sharing configuration.

```
[root@openEuler ~]# vim /etc/samba/smb.conf
```

Add the following information to the **global** file:

```
[global]
workgroup = SAMBA
security = user
map to guest = Bad User    # Add this line.
passdb backend = tdbsam

printing = cups
printcap name = cups
load printers = yes
cups options = raw
```

Add the **share** directory for public sharing and allow anonymous access to the directory.

```
[share]
comment = share
path = /var/share
guest ok = yes
writeable = yes
browseable = yes
```

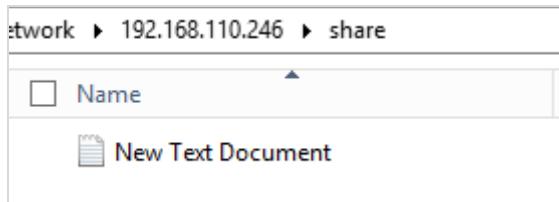
Save the configuration and exit. Then, restart the Samba service.

```
[root@openEuler ~]# systemctl restart smb
```

Use a Windows-based computer to access the Samba server. Open a file on the computer and enter the shared path **** to access the file sharing service.



Go to the directory and create a text file. If the text file can be created, the public file sharing directory is set successfully.



Step 4 Add the user sharing configuration.

Run the following command to set the owner of the shared directory **smb** on the openEuler OS.

```
[root@openEuler ~]# chown smb:smb /var/smb
```

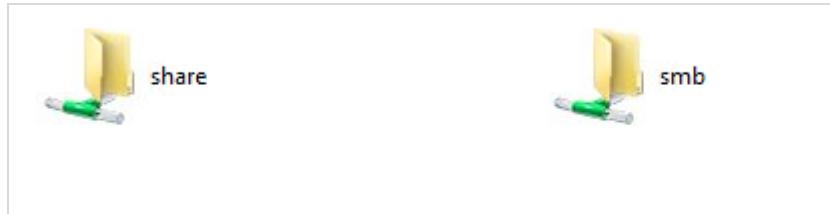
Edit the Samba configuration file **/var/samba/smb.conf**, add the user sharing configuration, save the modification, and exit.

```
[smb]
comment = smb
path = /var/smb
write list = smb
browseable = yes
writable = yes
read list = smb
valid users = smb
create mask = 0777
directory mask = 0777
```

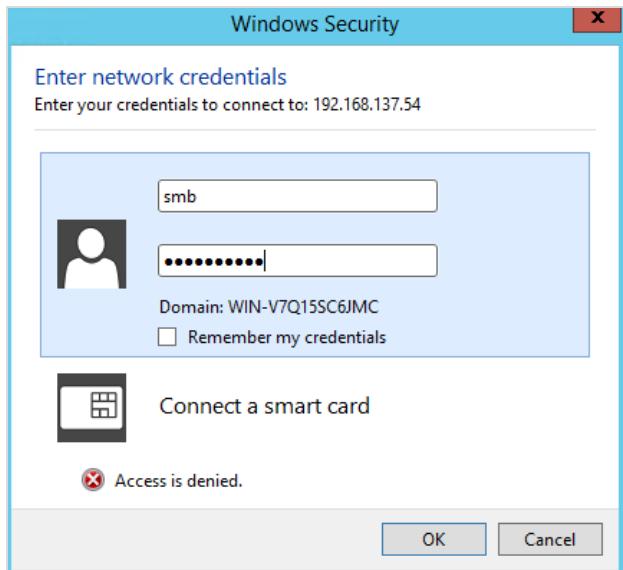
Run the following command to restart the Samba service:

```
[root@openEuler ~]# systemctl restart smb
```

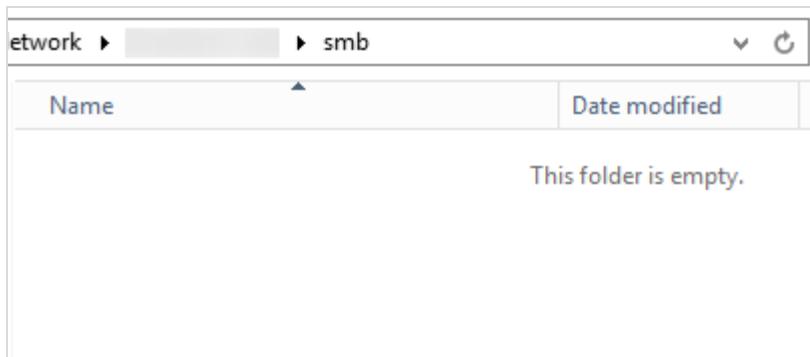
On the computer, refresh the file sharing access path <\\ip\\>. The **smb** directory is displayed.



Click **smb** to access the directory. Enter the username and password created in step 1 and the authentication information. Access the shared directory.



After the authentication information is correctly entered, the directory can be accessed.



Create a folder and a file in the file directory. The folder and file can be created successfully, and the user can share the file server successfully.

----End

9.4 O&M File Sharing Server

9.4.1 Overview

9.4.1.1 About This Exercise

In this exercise, the file sharing service is periodically backed up on the openEuler OS, and common troubleshooting methods for Samba file sharing services are introduced.

9.4.1.2 Objectives

Upon completion of this exercise, you will be able to:

- Understand how to configure scheduled tasks based on scripts.
- Understand how to locate common problems of the file sharing server.

9.4.2 Configuring the Exercise

9.4.2.1 Configuration Roadmap

1. After logging in to the openEuler OS, compile a script to back up the files on the file sharing server and save the data of the **/var/share/** directory to the **/var/smb** directory.
2. Set the data backup script to be executed at a specified time every day.
3. Check the Samba server logs and rectify the fault.

9.4.2.2 Procedure

Step 1 Edit the data backup script.

Log in to the openEuler OS as the **root** user, and then use the text editor to edit the backup script.

```
[root@openEuler ~]# vim /root/backup.sh
```

Edit the following content:

```
#!/bin/sh
mkdir /var/backup # Create a temporary backup directory.
cp -r /var/share/ /var/backup/ # Copy the data in the shared folder to the backup directory.
tar -zcPvf /var/smb/backup$(date +%Y%m%d).tar.gz /var/backup # Package the data in the shared
directory to the /var/backup directory.
rm -rf /var/backup/ # Delete the temporary backup directory.
find /var/smb/ -mtime +30 -name "*.tar.gz" -exec rm -rf {} \; # Delete backup data generated 30 days
ago.
```

Save the file and exit. Grant the execute permission on the file.

```
[root@openEuler ~]# chmod +x /root/backup.sh
```

```
[root@openEuler ~]#
[root@openEuler ~]#
[root@openEuler ~]# chmod +x /root/backup.sh
[root@openEuler ~]#
```

Manually execute the script and check whether the script is normal.

```
[root@openEuler ~]# sh /root/backup.sh
```

```
[root@openEuler ~]#
[root@openEuler ~]# sh /root/backup.sh
/var/backup/
/var/backup/share/
/var/backup/share/newfile.txt
[root@openEuler ~]# ls -l /var/smb/
total 8
-rw----- 1 root root 165 Jun  6 13:50 backup20200606.tar.gz
-rwxrw-rw- 1 smb  smb      0 Jun  6 13:18 newfile.txt
drwxrwxrwx 2 smb  smb  4096 Jun  6 13:17 newfolder
[root@openEuler ~]#
[root@openEuler ~]#
```

Step 2 Set a scheduled backup task.

Set a periodic task to execute the preceding backup script at 22:00 every day. Run the **crontab -e** command to edit the backup task.

```
[root@openEuler ~]# crontab -e  
0 22 * * * /root/backup.sh
```

```
0 22 * * * /root/backup.sh  
~  
~  
~
```

Save the file and exit. Run the **crontab -l** command to view the scheduled task.

```
[root@openEuler ~]# crontab -l
```

```
[root@openEuler ~]#  
[root@openEuler ~]# crontab -l  
0 22 * * * /root/backup.sh  
[root@openEuler ~]#
```

Step 3 View the Samba log.

Log files are stored in the **/var/log/** directory. You can run the **ls** command to check the directory for storing log files.

```
[root@openEuler ~]# ls /var/log
```

```
[root@openEuler ~]#  
[root@openEuler ~]# ls /var/log  
anaconda      cloud-init.log      dnf.log      hawkey.log-20200605  messages      rhsm          spooler  
audit         cloud-init-output.log  dnf.rpm.log  httpd           multi-queue-hw.log  rpmpkgs       systemtap.log  
bttmp         cron             dracut.log   journal        openEuler-security.log  rpmpkgs-20200606 tallylog  
bttmp-20200605 dnf.librepo.log    firewalld  lastlog       private          samba          tuned  
chrony        dnf.librepo.log-20200605 hawkey.log maillog      README          secure          wtmp  
[root@openEuler ~]#  
[root@openEuler ~]#
```

View the Samba log file name.

```
[root@openEuler ~]# ls -l /var/log/samba/
```

```
[root@openEuler ~]#  
[root@openEuler ~]# ls /var/log/samba/ -l  
total 16  
drwx----- 3 root root 4096 Jun  5 18:37 cores  
-rw-r--r--  1 root root 4907 Jun  6 14:00 log.smbd  
drwx----- 2 root root 4096 Mar 24 05:26 old  
[root@openEuler ~]#
```

View Samba file sharing service logs.

```
[root@openEuler ~]# tail /var/log/samba/log.smbd -n 20
```

```
[root@openEuler ~]# tail /var/log/samba/log.smbd -n 20
[2020/06/06 13:17:47.298399,  0] ../../lib/param/loadparm.c:800(lpcfg_map_parameter)
    Unknown parameter encountered: "browsable"
[2020/06/06 13:17:47.298466,  0] ../../lib/param/loadparm.c:1859(lpcfg_do_service_parameter)
    Ignoring unknown parameter "browsable"
[2020/06/06 13:18:16.108318,  0] ../../lib/param/loadparm.c:800(lpcfg_map_parameter)
    Unknown parameter encountered: "browsable"
[2020/06/06 13:18:16.108388,  0] ../../lib/param/loadparm.c:1859(lpcfg_do_service_parameter)
    Ignoring unknown parameter "browsable"
[2020/06/06 13:19:24.216682,  0] ../../lib/param/loadparm.c:800(lpcfg_map_parameter)
    Unknown parameter encountered: "browsable"
[2020/06/06 13:19:24.218293,  0] ../../lib/param/loadparm.c:1859(lpcfg_do_service_parameter)
    Ignoring unknown parameter "browsable"
[2020/06/06 13:58:41.257215,  0] ../../lib/param/loadparm.c:800(lpcfg_map_parameter)
    Unknown parameter encountered: "browsable"
[2020/06/06 13:58:41.257288,  0] ../../lib/param/loadparm.c:1859(lpcfg_do_service_parameter)
    Ignoring unknown parameter "browsable"
[2020/06/06 14:00:49.612262,  0] ../../lib/param/loadparm.c:800(lpcfg_map_parameter)
    Unknown parameter encountered: "browsable"
[2020/06/06 14:00:49.614458,  0] ../../lib/param/loadparm.c:1859(lpcfg_do_service_parameter)
    Ignoring unknown parameter "browsable"
[root@openEuler ~]#
```

Step 4 Refer to the following common faults of the Samba server.

- The Samba server cannot be accessed.

Error: Failed to connect to the network in file sharing mode.

Solution:

Run the **smbclient** command on the Samba server to check whether the Samba service is normal.

```
[root@openEuler ~]# smbclient -L localhost
```

```
[root@openEuler ~]#
[root@openEuler ~]# smbclient -L localhost
do_connect: Connection to localhost failed (Error NT_STATUS_CONNECTION_REFUSED)
[root@openEuler ~]#
```

The service is not started properly. Run the service restart command to restart the service.

```
[root@openEuler ~]# systemctl restart smb
```

```
[root@openEuler ~]#
[root@openEuler ~]# systemctl restart smb
[root@openEuler ~]#
```

Run the **smbclient** command to access the Samba server again. The access is successful.

```
[root@openEuler ~]#
[root@openEuler ~]# smbclient -L localhost
Enter SAMBA\root's password:

      Sharename      Type      Comment
      -----      ----
      print$        Disk      Printer Drivers
      share         Disk      share
      smb           Disk      smb
      IPC$          IPC       IPC Service (Samba 4.11.6)
SMB1 disabled -- no workgroup available
[root@openEuler ~]#
```

- The Samba server is incorrectly configured and cannot be started.

Error: After the Samba configuration is modified, an error occurs when the Samba service is restarted.

```
[root@openEuler ~]#  
[root@openEuler ~]# systemctl restart smb  
Job for smb.service failed because the control process exited with error code.  
See "systemctl status smb.service" and "journalctl -xe" for details.  
[root@openEuler ~]#
```

Solution: In the configuration directory of the Samba server, run the **testparm** command to check the configuration.

```
[root@openEuler ~]#  
[root@openEuler ~]# cd /etc/samba/  
[root@openEuler samba]# testparm  
Load smb config files from /etc/samba/smb.conf  
WARNING: Ignoring invalid value 'all' for parameter 'security'  
Error loading services.  
[root@openEuler samba]#
```

Check the configuration file.

```
[global]  
workgroup = SAMBA  
[redacted]  
security = all  
map to guest = Bad User  
passdb backend = tdbsam
```

Change the value to a correct one.

```
[global]  
[redacted]  
workgroup = SAMBA  
[redacted]  
security = user  
map to guest = Bad User  
passdb backend = tdbsam
```

Restart the Samba service.

```
[root@openEuler ~]# systemctl restart smb
```

- You do not have the permission to access or create a file.

Problem: A file fails to be created under the shared directory.

Solution: On the openEuler OS, check whether the permission on the shared file directory is correct.

```
[root@openEuler ~]# ls /var/share
```

----End