DAY 2

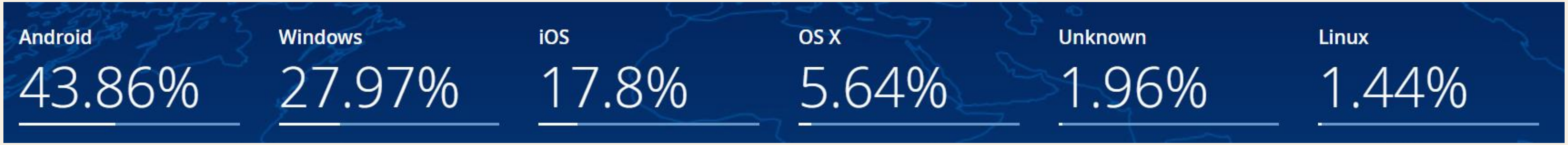
OS INSTALLATION

OS → Desktop Version & Server Version

Open-Source Operating System:

The open-source operating system permits the use of publicly accessible, freely distributable code for commercial purposes. An open-source OS's source code is accessible since it is an open-source application or programmed. According to their needs, the user can alter or edit those programs and create new apps.

Open-source operating systems include Linux, Open Solaris, Open BDS, Free BSD, Minix, etc.



Android	Windows	iOS	OS X	Unknown	Linux
43.86%	27.97%	17.8%	5.64%	1.96%	1.44%

Some of the best Open-Source Operating Systems:

*** Linux Mint * Linux Lite * Ubuntu * Chrome OS**

Advantages:

Reliable: The open-source operating systems are most reliable.

Cost-efficient: Most of the open-source operating systems are free and some of them are less expensive.

Flexibility: We can customize it as per our requirement.

Disadvantages:

Complicated: It is not as user-friendly as the ones that are closed. To use this software, we must have a basic understanding of technology.

Risk to Safety: Even if the flaws have been found, there is still a threat of attacks because the attackers have access to the source code.

Lack of support: There is no customer service accessible to help you if you encounter a problem.

The commercial operating system is not publicly accessible, not freely distributable code for commercial purposes.

For commercial, the users need to spend moderate to expensive cost. It is available under high licensing cost.

A commercial OS's source code is not accessible since it is a commercial or closed-source application or programmed.

According to their needs, the user can't alter or edit those programs and create new apps.

Some of the best Commercial Operating Systems:

Windows OS
Mac OS
Unix
Oracle

Advantages:

A single point of contact for any issues: Once paid for a piece of software, we have a real-world counterparty to deal with if a problem does occur.

Clearly defined licensing and usage conditions: Commercial software for organization has the advantage of coming with a well-documented promise that you will have support behind it.

A well-defined plan for the software and upgrade: Companies can avoid the costs associated with technological failures by performing timely upgrades.

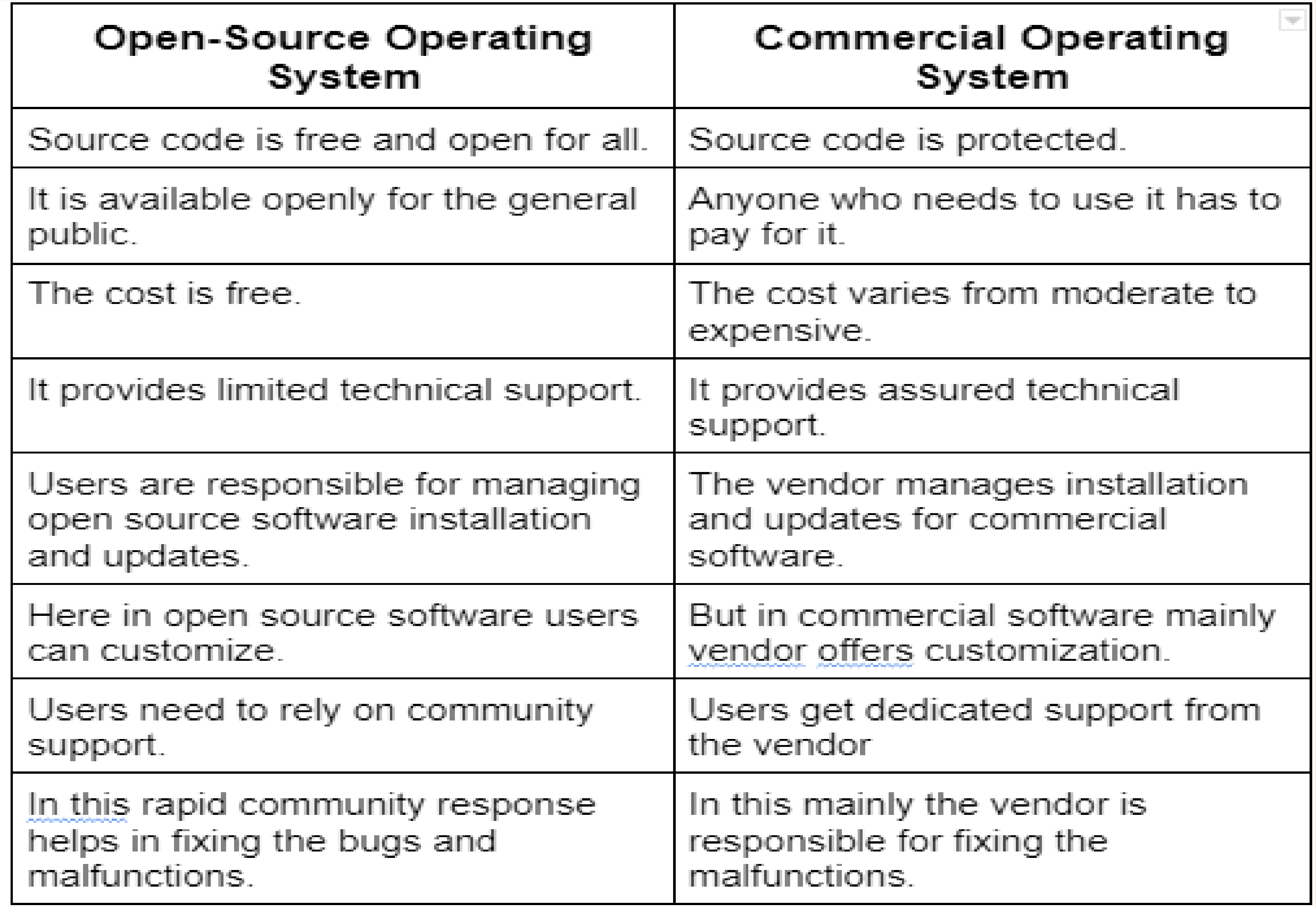
Disadvantages:

Strict guidelines for licensing: In an effort to avoid software audits, more than a quarter of companies admit to over-licensing.

Exclusive source code: You won’t be able to modify the source code to for your requirements. You have to subscribe to a more costly plan to get any extra functionality.

Risk of losing a software license: There’s a good possibility that the app will be idle unless it’s software required for the operation of your business. Various studies suggest that between 30 and 37% of commercial software licenses owned by businesses are unused or highly overused.

TAKEAWAY



Open-Source Operating System	Commercial Operating System
Source code is free and open for all.	Source code is protected.
It is available openly for the general public.	Anyone who needs to use it has to pay for it.
The cost is free.	The cost varies from moderate to expensive.
It provides limited technical support.	It provides assured technical support.
Users are responsible for managing open source software installation and updates.	The vendor manages installation and updates for commercial software.
Here in open source software users can customize.	But in commercial software mainly vendor offers customization.
Users need to rely on community support.	Users get dedicated support from the vendor
In this rapid community response helps in fixing the bugs and malfunctions.	In this mainly the vendor is responsible for fixing the malfunctions.

Get Ubuntu Server | Download | Ubuntu

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-simple-install

Installation Quick Start - SUSE Linux Enterprise Server 15 SP4

LOG IN TO A LOCAL SYSTEM

A **terminal** is a text-based interface to enter commands into and print output from a computer system. To run the shell, you must log in to the computer on a terminal.

Many system administrators choose not to run a graphical environment on their servers, because users do not log in to servers as a desktop workspace. A server's workload can more effectively use the significant resources that a graphical environment uses.

LOG IN TO A REMOTE SYSTEM

Linux users and administrators often need to get shell access to a remote system by connecting to it over the network. In a modern computing environment, many headless servers are virtual machines or are running as public or private cloud instances.

In Linux, the most common way to get a shell prompt on a remote system is to use **Secure Shell (SSH)**. Most Linux systems (including Red Hat Enterprise Linux) and macOS provide the OpenSSH command-line program ssh for this purpose.

```
[user@host ~]$ ssh remoteuser@remotehost
remoteuser@remotehost's password: password
[remoteuser@remotehost ~]$
```

Some systems, such as new cloud instances, for tighter security do not allow users to use a password to log in with ssh. An alternative way to authenticate to a remote machine without entering a password is through **public key** authentication.

```
ssh -i mylab.pem remoteuser@remotehost
```

LOG OUT FROM A REMOTE SYSTEM

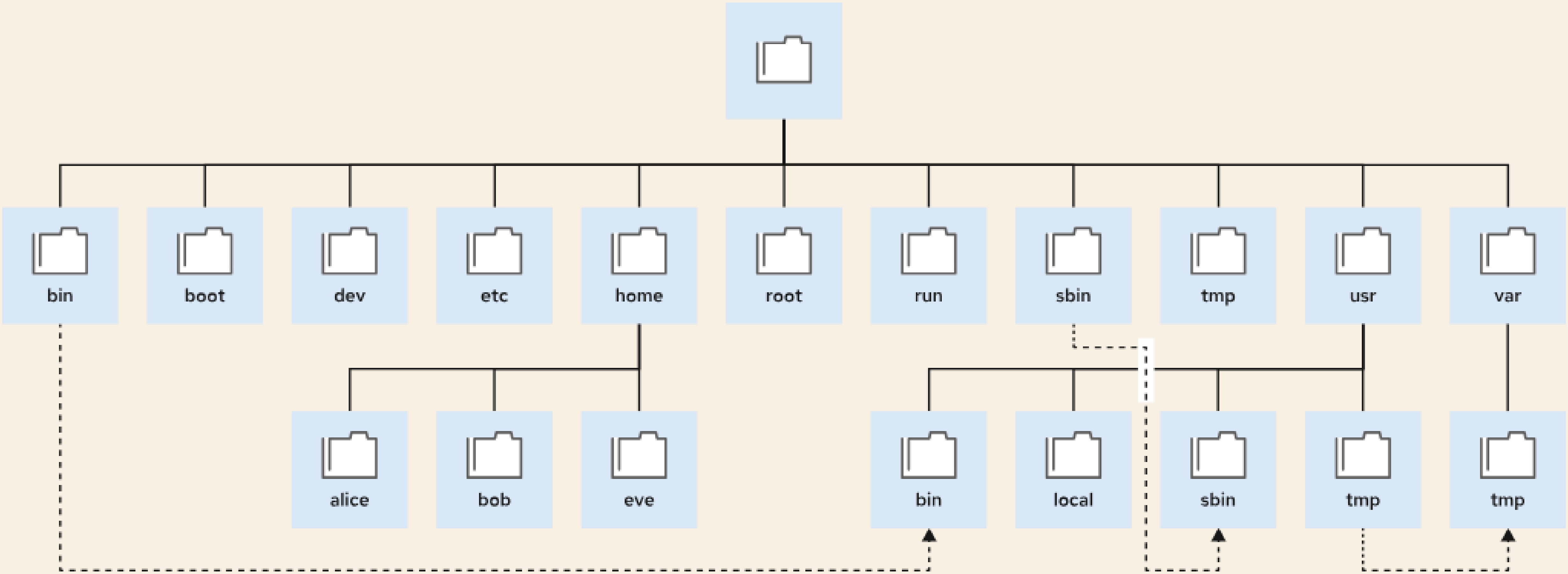
When you are finished with the shell and want to quit, you can choose one of several ways to end the session. You can enter the **exit** command to terminate the current shell session. Alternatively, finish a session by pressing Ctrl+D.

The following example shows a user who logs out of an SSH session:

```
[remoteuser@remotehost ~]$ exit
logout
Connection to remotehost closed.
[user@host ~]$
```

THE FILE-SYSTEM HIERARCHY

The Linux system stores all files on file systems, which are organized into a single inverted tree known as a file-system hierarchy. This hierarchy is an inverted tree because the tree root is at the top, and the branches of directories and subdirectories stretch below the root.



The / directory is the root directory at the top of the file-system hierarchy. The / character is also used as a directory separator in file names.

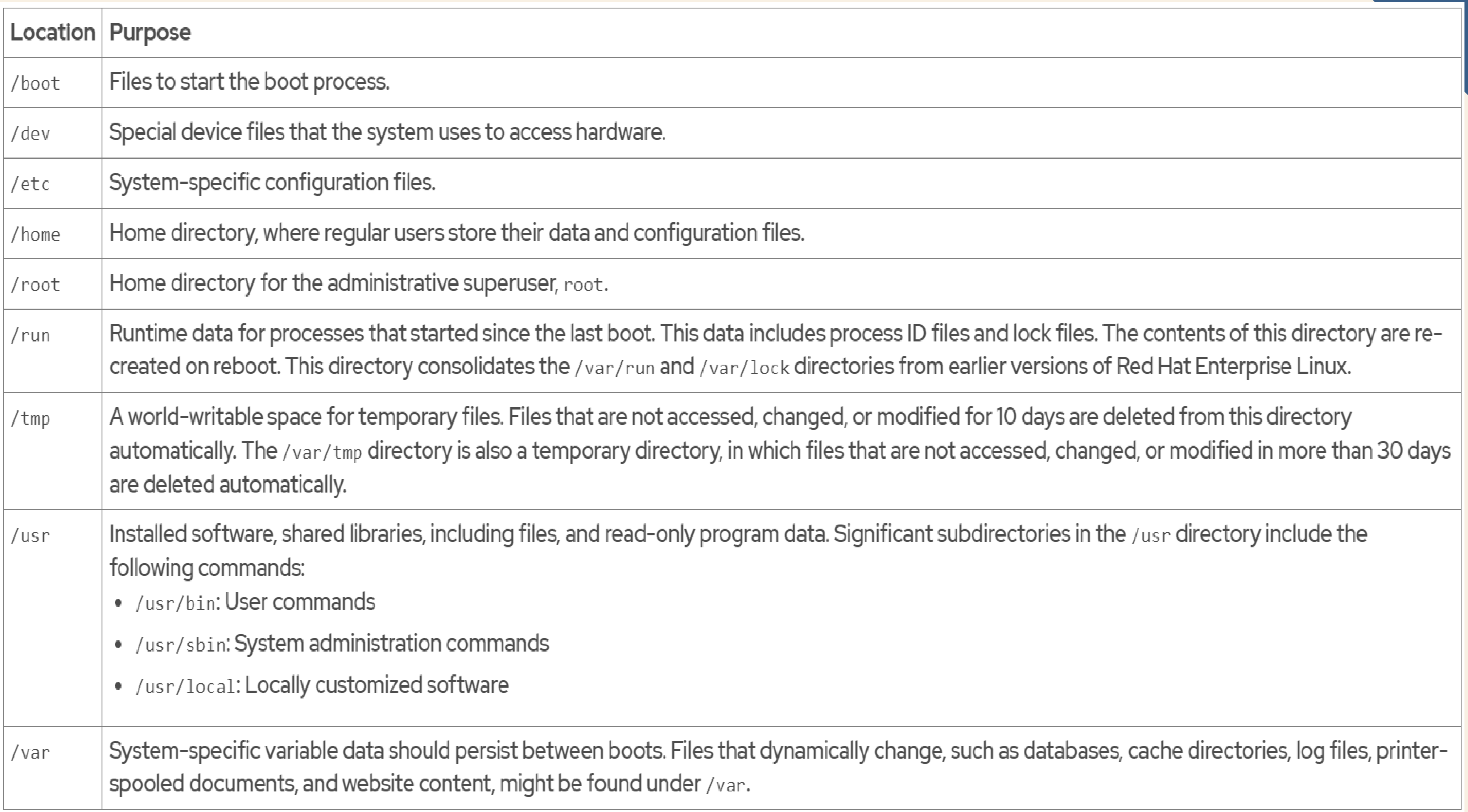
The following terms help to describe file-system directory contents:

Static content remains unchanged until explicitly edited or reconfigured.

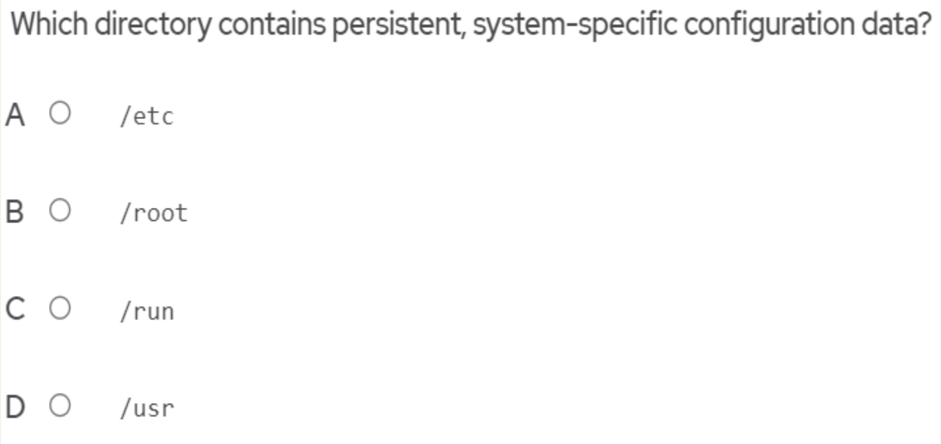
Dynamic or variable content might be modified or appended by active processes.

Persistent content remains after a reboot, such as configuration settings.

Runtime content from a process or from the system is deleted on reboot.



Location	Purpose
`/boot`	Files to start the boot process.
`/dev`	Special device files that the system uses to access hardware.
`/etc`	System-specific configuration files.
`/home`	Home directory, where regular users store their data and configuration files.
`/root`	Home directory for the administrative superuser, `root`.
`/run`	Runtime data for processes that started since the last boot. This data includes process ID files and lock files. The contents of this directory are re-created on reboot. This directory consolidates the `/var/run` and `/var/lock` directories from earlier versions of Red Hat Enterprise Linux.
`/tmp`	A world-writable space for temporary files. Files that are not accessed, changed, or modified for 10 days are deleted from this directory automatically. The `/var/tmp` directory is also a temporary directory, in which files that are not accessed, changed, or modified in more than 30 days are deleted automatically.
`/usr`	Installed software, shared libraries, including files, and read-only program data. Significant subdirectories in the `/usr` directory include the following commands: • `/usr/bin`: User commands • `/usr/sbin`: System administration commands • `/usr/local`: Locally customized software
`/var`	System-specific variable data should persist between boots. Files that dynamically change, such as databases, cache directories, log files, printer-spooled documents, and website content, might be found under `/var`.



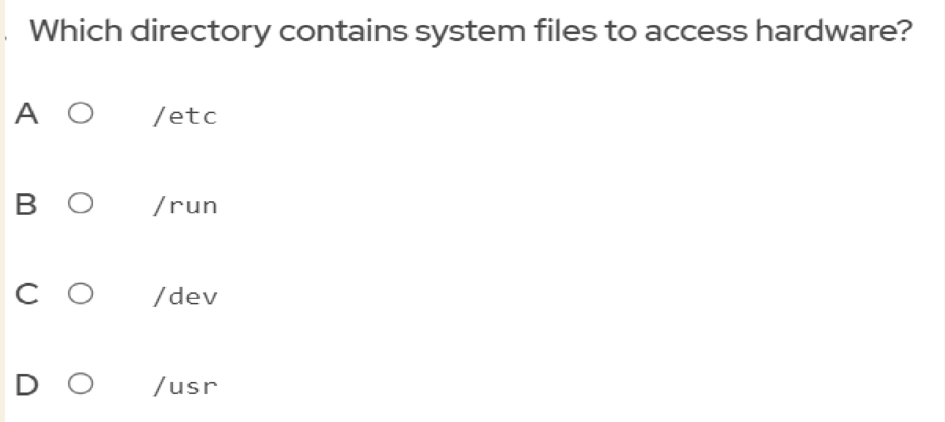
Which directory contains persistent, system-specific configuration data?

A ○ /etc

B ○ /root

C ○ /run

D ○ /usr



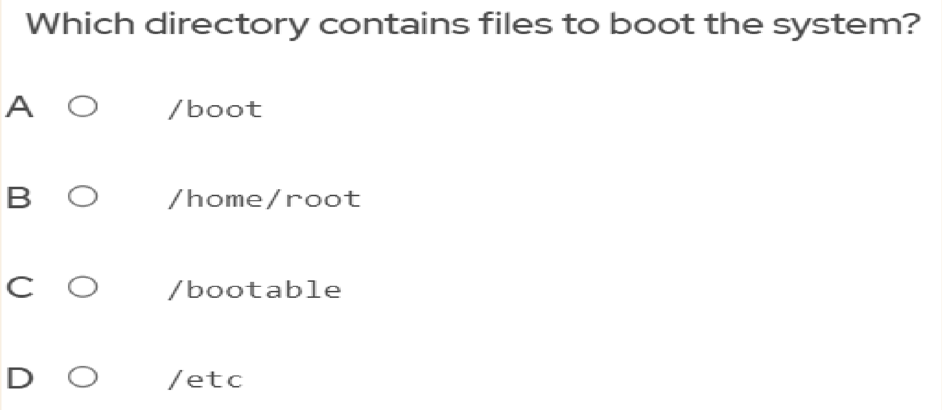
Which directory contains system files to access hardware?

A ○ /etc

B ○ /run

C ○ /dev

D ○ /usr



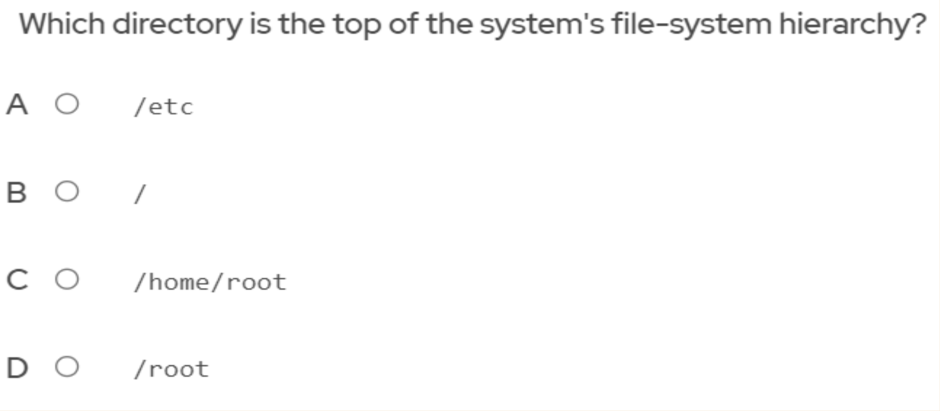
Which directory contains files to boot the system?

A ○ /boot

B ○ /home/root

C ○ /bootable

D ○ /etc



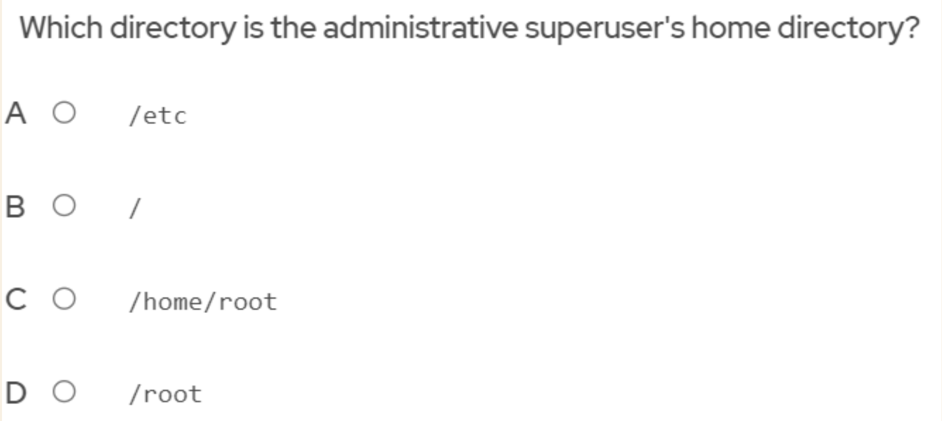
Which directory is the top of the system's file-system hierarchy?

A ○ /etc

B ○ /

C ○ /home/root

D ○ /root



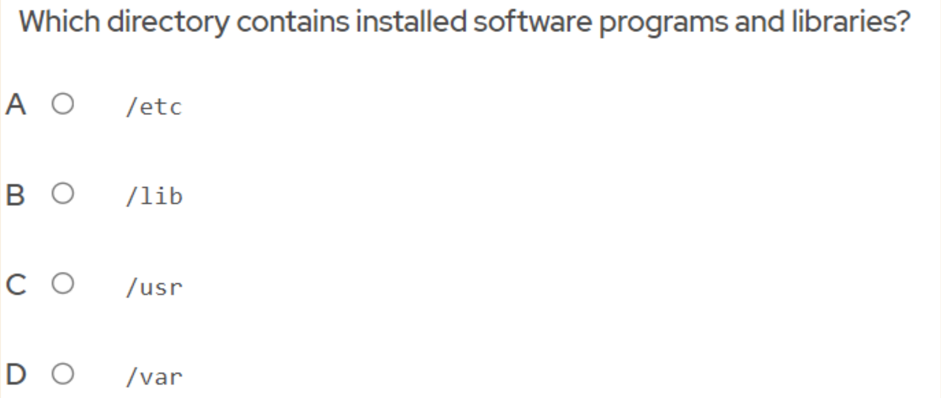
Which directory is the administrative superuser's home directory?

A ○ /etc

B ○ /

C ○ /home/root

D ○ /root



Which directory contains installed software programs and libraries?

A ○ /etc

B ○ /lib

C ○ /usr

D ○ /var

HELP SYSTEM

The help command is the simplest way to get information regarding a built-in shell command. It helps you fetch information from the shell’s internal documentation.

It takes a text string as the command line argument and looks for the provided string in the shell’s documents.

This saves the time spent in going through the entire documentation.

Help command itself offers three options:

-d: display only a brief description of the specified command.
-m: organize the available information just as the man command does.
-s: display the command syntax of the specified command.

```
help cd
help pwd
help help
```

MAN COMMAND

man command in Linux is used to display the user manual of any command that we can run on the terminal.

It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

```
man [command]
```

```
man ls
```

```
man cd
```

BASIC COMMAND SYNTAX

The GNU Bourne-Again Shell (bash) is a program that interprets commands that the user types.

Each string that is typed into the shell can have up to three parts: the command, options (which usually begin with a hyphen - or double hyphen -- characters), and arguments.

Each word that is typed into the shell is separated from other words with spaces.

Commands are the names of programs that are installed on the system. Each command has its options and arguments.

```
[user@host ~]$ whoami
```

```
[user@host ~]$ command1 ; command2
command1 output
command2 output
```

```
[user@host ~]$ date
Sun Jun 2 08:32:42 PM EST 2024
[user@host ~]$ date +%R     # plus sign (+) as an argument to specify a format string for the date command
20:33
[user@host ~]$ date +%x
06/02/2024
```

```
# Displays the current date and time
date

# Changes the current user's password
passwd

# Scans the compiled header of a file and displays its type
file /etc/passwd
file /bin/passwd
file /home

# To create single or multiple files, view the contents of files, concatenate the contents from various files, and redirect
contents of the file to a terminal or to files

cat /etc/passwd
cat file1 file2

# Display the beginning and the end of a file, respectively
head /etc/passwd
tail -n 3 /etc/passwd
```

```
# Counts lines, words, and characters in a file. Use the -l, -w, or -c options to display only the given number of lines, words, or characters, respectively.


wc /etc/passwd
wc -l /etc/passwd ; wc -l /etc/group
wc -c /etc/group /etc/hosts
```