OSCP Linux Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Table of Contents

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- **Practice answering questions under realistic conditions,**
- **Improve accuracy and speed,**
- **Review explanations to strengthen weak areas, and**
- **Approach the exam with greater confidence.**

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here’s how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don’t just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There’s no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

1. What are the symbols for boolean AND, OR, NOT in Bash and what is notable about their behavior?

A. AND = &&, OR = ||, NOT = !; In the shell, && runs the second command only if the first one succeeds, and || runs the second command only if the first one fails.

B. AND = and, OR = or, NOT = not; They always evaluate strictly left-to-right.

C. AND = &, OR = |, NOT = ~; They have no special semantics.

D. AND = +, OR = -, NOT = !; They are used to perform arithmetic.

2. Which command shows the users currently logged in?

A. date

B. who

C. passwd

D. uptime

3. What exit value does the command 'true' align with and what does that mean?

A. 1 — It always fails.

B. 0 — It always runs successfully.

C. 2 — It sometimes fails.

D. 255 — It terminates with an error.

4. Which command would you use to view PATH directly?

A. echo $PATH

B. printenv PATH

C. env | grep PATH

D. set PATH

5. Which directory provides kernel information via a virtual filesystem?

A. /proc

B. /sys

C. /dev

D. /run

6. What does the shell do when you use > file.txt before a command?

A. Redirects standard input to file.txt

B. Redirects standard output to file.txt

C. Redirects standard error to file.txt

D. Redirects standard output to file.txt

7. Given the Bash array animals defined as ("a dog" "a cat" "a fish"), what does echo ${animals[1]} print?

A. a dog

B. a cat

C. a fish

D. a cow

8. What is the effect of quoting the end marker in a here-document?

A. Prevents parameter expansion

B. Prevents redirection

C. Prevents file globbing

D. Prevents expansion of variables and command substitution, preserving literal text

9. Which of the following is a valid alternative method to locate PATH without using printenv?

A. printenv PATH

B. declare | grep PATH

C. echo $PATH

D. ls /proc/self/environ

10. Which command outputs the target path of a symbolic link?

A. ls -l

B. readlink

C. cat

D. touch

Answers

1. A
2. B
3. B
4. B
5. A
6. D
7. B
8. D
9. B
10. B

Explanations

1. What are the symbols for boolean AND, OR, NOT in Bash and what is notable about their behavior?

A. AND = &&, OR = ||, NOT = !; In the shell, && runs the second command only if the first one succeeds, and || runs the second command only if the first one fails.

B. AND = and, OR = or, NOT = not; They always evaluate strictly left-to-right.

C. AND = &, OR = |, NOT = ~; They have no special semantics.

D. AND = +, OR = -, NOT = !; They are used to perform arithmetic.

In Bash, the boolean-like control operators are &&, ||, and !. They aren't just true/false operators; they act on the exit status of commands. A command that finishes with exit status 0 is considered success (true), while a non-zero exit status is failure (false). Using this, && runs the second command only if the first one succeeds, enabling short-circuiting. Using || runs the second command only if the first one fails, also short-circuiting. The NOT operator negates the exit status of the following command, so it flips success to failure and vice versa, which is handy in conditionals. Other symbols like words such as and/or/not, or symbols like & (background), | (pipe), or arithmetic operators don't represent the same boolean control behavior in this context.

2. Which command shows the users currently logged in?

A. date

B. who

C. passwd

D. uptime

The key idea is identifying a command that queries the system's user accounting data and prints who is actively logged in. That command reads the utmp records and shows one line per active session, typically including the username, the terminal (TTY), and the login time (and sometimes the remote host). This directly answers who is currently signed into the system. The other commands serve different purposes: date shows the current date and time; passwd is used to change a user's password; uptime reports how long the system has been running and, in some outputs, how many users are logged in but not who they are. So they don't provide a direct list of the users currently logged in, whereas this command does.

3. What exit value does the command 'true' align with and what does that mean?

A. 1 — It always fails.

B. 0 — It always runs successfully.

C. 2 — It sometimes fails.

D. 255 — It terminates with an error.

Exit status signaling is what this question tests. In Unix-like systems, exit code 0 means success, while any non-zero value indicates some kind of error or failure. The command true is a deliberate no-op that simply exits with status 0, so it always signals success. That's why the correct interpretation is that the exit value is 0 and it ran successfully. The other codes described assume either failure or an undefined meaning for this command, which doesn't align with true's defined behavior.

4. Which command would you use to view PATH directly?

A. echo $PATH

B. printenv PATH

C. env | grep PATH

D. set PATH

PATH is the environment variable that lists directories the shell searches for executables. To view its value directly, using printenv PATH queries the environment and prints only the value of PATH, cleanly and predictably across shells. This avoids any shell expansion or extra text, giving a straightforward one-line result. Echoing $PATH also works, but it relies on the shell to expand the variable and can behave differently in different contexts. The other approaches either pull more of the environment or perform a different operation (like listing or setting variables), so they aren't as direct for retrieving PATH.

5. Which directory provides kernel information via a virtual filesystem?

A. /proc

B. /sys

C. /dev

D. /run

Kernel information is exposed through /proc, a procfs virtual filesystem. It presents kernel and process data as files, not as real disk files, so reading those paths queries the running kernel for current state. For example, /proc/cpuinfo shows CPU details, /proc/meminfo shows memory statistics, and /proc/version reveals the kernel version. This interface exists specifically to let user-space programs access kernel information in a simple, uniform way. Other virtual filesystems serve related but different purposes: /sys (sysfs) exposes device and driver attributes and objects, /dev provides device nodes for interacting with hardware, and /run holds runtime data. None of these are the standard source for broad kernel information in the same way as /proc.

6. What does the shell do when you use > file.txt before a command?

A. Redirects standard input to file.txt

B. Redirects standard output to file.txt

C. Redirects standard error to file.txt

D. Redirects standard output to file.txt

Redirecting output with the > operator sends the command's standard output to a file instead of the screen. Before the command runs, the shell opens file.txt for writing, truncates it if it already exists (or creates it if it doesn't), and connects the process's stdout (file descriptor 1) to that file. As a result, any normal output from the command goes into file.txt. If you want to append instead of overwriting, use >>; to redirect errors as well, use 2> for standard error or combine streams with >file 2>&1.

7. Given the Bash array animals defined as ("a dog" "a cat" "a fish"), what does echo ${animals[1]} print?

A. a dog

B. a cat

C. a fish

D. a cow

In Bash, array indices start at zero. The array defined has three elements: index 0 holds "a dog", index 1 holds "a cat", and index 2 holds "a fish". When you reference the element at index 1, you're selecting the second item. Echoing that value prints "a cat". Note that the value contains a space, but echo prints it as that two-word string.

8. What is the effect of quoting the end marker in a here-document?

A. Prevents parameter expansion

B. Prevents redirection

C. Prevents file globbing

D. Prevents expansion of variables and command substitution, preserving literal text

Quoting the end marker makes the here-document literal. The shell stops performing expansions on the content, so anything like variables or command substitutions stays exactly as written and is not executed or substituted. This means lines with $VAR or $(cmd) are passed to the command unchanged, preserving literal text. That's why this option is the best choice: it captures both the prevention of expansions and the exact preservation of the text.

9. Which of the following is a valid alternative method to locate PATH without using printenv?

A. printenv PATH

B. declare | grep PATH

C. echo $PATH

D. ls /proc/self/environ

Locating PATH can be done by inspecting the shell's own variable declarations. In Bash, the declare builtin lists all declared variables along with their attributes. Piping that output to grep for PATH isolates the line that defines PATH, typically showing its value and offering a hint about whether it's exported. For example, you might see a line like declare -x PATH="/usr/local/sbin:/usr/local/bin:/usr/bin:/bin" which reveals both the value and the export status. This method doesn't call printenv, and it uses the shell's own state to reveal how PATH is set. While echoing $PATH will display the directories in PATH, declare | grep PATH specifically demonstrates the variable's declaration and export, which can be informative in contexts where you care about how the shell handles it. The other approaches either duplicate the value output in a simple way or delve into low-level environment representations, which aren't as direct for understanding how PATH is defined in the shell.

10. Which command outputs the target path of a symbolic link?

A. ls -l

B. readlink

C. cat

D. touch

Revealing where a symbolic link points. A symbolic link stores the path to its target, and readlink prints that path exactly as stored in the link, without following it. This makes readlink the clear choice for viewing the destination of a symlink, and you can use readlink -f if you need the fully resolved absolute path. Other commands don't give the destination directly: ls -l shows a listing that may include the target after an arrow but is not a dedicated path print; cat follows the link to show the contents of the target file; touch simply creates or updates a timestamp on a file.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://oscplinux.examzify.com

We wish you the very best on your exam journey. You've got this!