11 Essential Linux Interview Questions *

- 1.6Kshares

Submit an interview questionSubmit a question
Looking for experts? Check out Toptal's Linux developers.

How would you swap the `stdout` and `stderr` of a command?

View the answer →Hide answer

```
$ command 3>&2 2>&1 1>&3
```

To swap `stdout` and `stderr` of a command, a third file descriptor is being created (in this case `3`), which is assigned to the same target that `stderr` is pointed to (referenced by `&2`). Then `stderr` is pointed to the same target `stdout` is pointed to (`&1`). Finally, `stdout` is pointed back to where the newly created file descriptor is pointed (which is the same target `stderr` originally pointed to.)

How would you count every occurrence of the term “potato” in all the files appearing under the current directory, and its subdirectories, recursively?

View the answer →Hide answer

```
$ grep -orI potato . | wc -l
```

To list every occurrence of the term “potato” on a separate line, one must run `grep -o potato <path>`. Adding the `r` flag to the command makes the search recursively process every file under the given path, and the `I` flag ensures that matches in binary files are ignored. In addition, the `w` flag can be included to match the exact term only, and ignore superstrings such as “potatoes”, and to make the search case-insensitive, the `i` flag can be added as well:

```
$ grep -iworI potato . | wc -l
```

The number of lines yielded by this `grep` command is the number of occurrences of the desired term, which can then be counted by piping it into the `wc -l` command.

How would you write a shell script that prints all the additional arguments passed to it in reverse order?

View the answer →Hide answer

```
for (( i = ${#}; i > 0; i-- )); do
        echo ${!i}
done
```

The arguments are available as `$<n>`, where `n` is the position of the argument. For example, `$0` would give the name of the script, `$1` would give the first additional argument, `$2` the second, and so on. The total number of additional arguments is found in `$#`.

A loop that starts at `$#` and ends at 1 can be used to print each additional argument in reverse order.

Find top Linux developers today. Toptal can match you with the best engineers to finish your project.

Hire Toptal’s Linux developers

How would you write a shell script and ensure that only one instance of the script may run for every user? Strong atomicity is not required.

View the answer →Hide answer

In Bash:

```
LOCKFILE=/tmp/lock-`whoami`
if [ -e ${LOCKFILE} ] && kill -0 `cat ${LOCKFILE}`; then
    echo "Already running!"
    exit 1
fi
trap "rm -f ${LOCKFILE}; exit" INT TERM EXIT
echo $$ > ${LOCKFILE}
```

Start by determining a name for the lock file. In this case, the lock file is generated by suffixing a common name with the username of the current user.

Then, check if the lock file exists and if the PID contained within the lock file is running. If it is, exit with a message.

Create a trap to remove the lock file on a clean exit, or unclean exits (any exit with the signal `INT` or `TERM`).

Finally, if the script has not exited yet, create the lock file, and store the PID of the current process (`$$`) in it.

What are shared, slave, private, and unbindable mountpoints?

View the answer →Hide answer

A mount point that is shared may be replicated as many times as needed, and each copy will continue to be the exact same. Other mount points that appear under a shared mount point in some subdirectory will appear in all the other replicated mount points as it is.

A slave mount point is similar to a shared mount point with the small exception that the "sharing" of mount point information happens in one direction. A mount point that is slave will only receive mount and unmount events. Anything that is mounted under this replicated mount point will not move towards the original mount point.

A private mount point is exactly what the name implies: private. Mount points that appear under a private mount point will not be shown elsewhere in the other replicated mount points unless they are explicitly mounted there as well.

An unbindable mount point, which by definition is also private, cannot be replicated elsewhere through the use of the bind flag of the mount system call or command.

What are some basic measures that you would take to harden a server's SSH service?

View the answer →Hide answer

There are a some very simple steps that can be taken to initially harden the SSH service, such as:

- Forcing the service to use only version 2 of the protocol will introduce both security and feature enhancement.
- Disabling root login, and even password-based logins, will further reinforce the security of the server.
- The whitelist approach can be taken, where only the users that belong to a certain list can login via SSH to the server.
- Disabling password-based login will require you to then allow key based logins, which is secure, but can be taken further by restricting their use from only certain IP addresses.
- Changing the port to something other than 22 significantly decreases random brute force attempts from the internet.

Sometimes the use of having an SSH service on a server may just be transferring files to and from the server (typically using tools like `scp`). In such a case, it is possible to change the shell of the user to something restrictive, such as rssh.

Finally it is often desirable to know exactly what is going on while you are not logged into the server. The logging verbosity may be increased if needed. Often, it is the logs that allow one to figure out if a key has indeed been stolen and is being abused.

What is a Unix shell? Is Bash the only Unix shell?

View the answer →Hide answer

A Unix shell is a software that provides a user interface for the underlying operating system. Unix shells typically provide a textual user interface - a command line interpreter - that may be used for entering and running commands, or create scripts that run a series of commands and can be used to express more advanced behavior.

Bash is not the only Unix shell, but just one of many. Short for Bourne-Again Shell, it is also one of the many Bourne-compatible shells. However, Bash is arguably one of the most popular shells around. There are other, modern shells available that often retain backwards compatibility with Bash but provide more functionality and features, such as the Z Shell (zsh).

Where is the target path of a symlink stored? How are permission settings for symlinks handled?

View the answer →Hide answer

The target path of a symlink is stored in an inode - the data structure used to store file information on disk.

Typically, the permission settings of the symlink itself only control the renaming and removal operations performed on the symlink itself. Any operation that deals with the contents of the file linked to are controlled by the permission settings of the target file.

What are terminal multiplexers? What are some of their key features? What are some of the more popular ones currently available?

View the answer →Hide answer

Terminal multiplexers enable several terminals to be created and controlled from a single screen or from a single remote session. The terminals and sessions can be detached and left running, even with the user logging off.

Two of the more common ones available today are GNU Screen and tmux.

`Screen` enables you to connect to multiple remote servers without needing to open multiple terminal shells. Work can be preserved and a session detached, f
example, to wait for the output of a long-running command. On subsequent reconnection, users can reattach to existing sessions or run new sessions. Sessio
also be shared among different users, which may be useful in audit or training scenarios.

Both `Screen` and `tmux` support split-screen functionality (to be more precise, `tmux` supports this and `Screen` supports it via a plugin). This allows, for example, running`tail` on a service's log file in one part of the screen, and editing the configuration of that service, and restarting it if necessary, in another.

What would be a simple way to continuously monitor the log file for a service that is running?

View the answer →Hide answer

Probably the simplest and most common way to do this would be by using the command:

```
tail -F $LOGFILE
```

where `$LOGFILE` is an environment variable corresponding to the path to the log file to be monitored.

By default, the Linux `tail` command prints the last 10 lines of a given file to standard output. The `-F` option causes additional file content to be displayed in realtime as the file continues to grow. This yields a simple mechanism for monitoring services via their log files in close to realtime.

Two other specific command line options of interest in this context are:

- The `-s` option causes `tail` to sleep for a specified number of seconds between updates (e.g., `tail -F -s 10` will update the displayed file contents roughly every 10 seconds rather than in close to realtime as the file is updated).
- The `-n` option can be used to specify a number of lines other than 10 to initially display (e.g., `tail -n 20 -F` will first display the last 20 lines of the file and will then continue updating the output in realtime).

What is a Linux null (or Blackhole) route? How can it be used to mitigate unwanted incoming connections?

View the answer →Hide answer

A [Linux null (or Blackhole) route] is a type of routing table entry which, upon matching a packet, discards it without forwarding the packet any further or sending any ICMP.

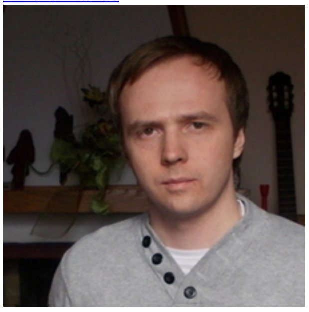
Using this technique, it is possible to block an IP (or range of IP addresses) by running a simple command. For example, blocking 192.168.0.1 can simply be done with the following command:

```
# ip route add blackhole 192.168.0.1/32
```

* There is more to interviewing than tricky technical questions, so these are intended merely as a guide. Not every "A" candidate worth hiring will be able to answer them all, nor does answering them all guarantee an "A" candidate. At the end of the day, [hiring remains an art, a science — and a lot of work].
Submit an interview question
Submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.

Name
Email
Enter your question here
Enter your answer here
All fields are required
☐ I agree with the Terms and Conditions of Toptal, LLC's [Privacy Policy]
Submit a Question

Thanks for submitting your question.
Our editorial staff will review it shortly. Please note that submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.
Looking for Linux experts? Check out Toptal's [Linux developers].

View full profile »
Simanas Venčkauskas
Lithuania
Simanas is a passionate developer, entrepreneur, and business analyst who excels at utilizing latest artificial intelligence techniques to deliver top-quality software solutions in the shortest period of time.
LinuxPythonArtificial Intelligence (AI)+4 more
Hire Simanas

View full profile »
Sebastian Nanek
United Kingdom
Sebastian is a seasoned developer who has been crafting true gems with passion since 2008. He has extensive experience both as a team leader and working in a solo environment. He emphasizes maintaining excellent communications with clients.
LinuxRubyRuby on Rails (RoR)+2 more
Hire Sebastian

View full profile »
Mahmud Ridwan
Bangladesh
Mahmud is a software developer with a knack for efficiency, scalability, and stable solutions. With years of experience working with a wide range of technologies, he is still interested in exploring, encountering, and solving new and interesting programming problems.
LinuxExpress.jsGoCoffeeScript+3 more
Hire Mahmud
Toptal connects the top 3% of freelance talent all over the world.

Join the Toptal community.

Hire a developer
or
Apply as a developer

Highest In-Demand Talent

- iOS Developer
- Front-End Developer
- UX Designer
- UI Designer
- Financial Modeling Consultants
- Interim CFOs

About

- Top 3%
- Clients
- Freelance Developers
- Freelance Designers
- Freelance Finance Experts
- About Us

Contact

- Contact Us
- Press Center
- Careers
- FAQ

Social

- Facebook
- Twitter
- Google+
- LinkedIn

Toptal

Hire the top 3% of freelance talent™

- Privacy Policy
- Website Terms

Find a world-class Linux developer for your team.Hire Toptal’s Linux developers