

DUMPS ARENA

Linux Professional Institute BSD Specialist

LPI 702-100

Version Demo

Total Demo Questions: 8

Total Premium Questions: 84

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, BSD Installation and Software Management	14
Topic 2, BSD System Administration	51
Topic 3, BSD Networking	14
Topic 4, BSD Security	5
Total	84

QUESTION NO: 1

Which of the following features are provided by the command `date`? (Choose TWO correct answers.)

- A. It outputs the time in a specific format
- B. It displays a calendar
- C. It sends NTP queries
- D. It changes the system's timezone.
- E. It sets the clock to a specific time

ANSWER: A E

Explanation:

The `date` utility provides both display and clock-setting functionality on BSD systems. “It outputs the time in a specific format” is correct because `date` prints the current date and time by default and accepts a format string beginning with `+` to control its output. Conversion specifications such as `+%Y-%m-%d`, `+%H:%M:%S`, and `+%a` allow scripts and administrators to produce machine-readable dates, times, and weekday representations. The output is interpreted using the active locale and time-zone environment unless an option such as `-u` is used for UTC.

“It sets the clock to a specific time” is also correct. When supplied with a supported numeric date-and-time operand, `date` can set the system clock. This operation changes a system-wide resource and therefore requires appropriate privilege, normally the superuser account. The utility also supports setting or displaying values relative to UTC where implemented through its UTC option. These capabilities make `date` useful both for formatted reporting and for controlled manual clock adjustment. BSD-specific syntax and supported formatting conversions are documented in the relevant system manual page; consult the documentation for the particular BSD release in use. See the [FreeBSD date\(1\) manual page](#) and the [POSIX date specification](#).

QUESTION NO: 2

Which statement about the difference between compiling an application from source code and installing a pre-compiled binary is true?

- A. Compiling source code is faster than installing a pre-compiled package.
- B. An application compiled from source code is more secure than a pre-compiled program
- C. Building an application from source code allows setting options which cannot be changed when executing the application.
- D. Pre-compiled binaries offer more configuration possibilities than those built from source.
- E. Applications that require the user to accept a license are only available as pre-compiled packages.

ANSWER: C

Explanation:

Building an application from source code allows the person performing the build to select compile-time options before the executable is created. These options can control which optional components are included, which libraries or features are enabled, what target architecture or optimization settings are used, and sometimes which default paths or behaviors are built into the program. Once those decisions have been compiled into the resulting binary, they generally cannot be changed merely by invoking the application with different runtime arguments; changing them requires rebuilding the software with different build settings. This is a key benefit of source-based installation for administrators who need a package tailored to a particular system or use case.

On BSD systems, the FreeBSD Ports Collection provides a practical example: ports can present build options that determine features included in the package or application being built. The FreeBSD Handbook describes configuring such options in its

[Ports Collection](#) chapter. More generally, the GNU build system documents how `configure` accepts build-time options that shape the generated build configuration in the [GNU Autoconf manual](#).

QUESTION NO: 3

In an interactive BSD shell with job control enabled, job number 1 has been stopped with the suspend character. Which TWO commands resume that job? (Choose TWO correct answers.)

- A. `bg %1`
- B. `fg %1`
- C. `jobs %1`
- D. `wait %1`
- E. `kill -STOP %1`

ANSWER: A B

Explanation:

`bg %1` resumes stopped job 1 in the background. `fg %1` resumes the same stopped job in the foreground and attaches it to the terminal. Both commands use the shell job identifier prefixed by `%`.

`jobs` only displays job status, and `wait %1` waits for a job rather than resuming a stopped one. `kill -STOP` explicitly stops a process.

QUESTION NO: 4

Which configuration file defines when log files are rotated'?

- A. `/etc/logs.conf`
- B. `/etc/rc.logd`
- C. `/etc/newsyslog.conf`
- D. `/etc/rc.d/syslogd`
- E. `/etc/syslog.conf`

ANSWER: C

Explanation:

`/etc/newsyslog.conf` is the standard configuration file read by `newsyslog` to determine log-rotation policy on BSD systems, including FreeBSD. Each configuration entry identifies a log file and specifies such controls as permissions, retention count, size threshold, rotation schedule, compression, and an optional process PID file. The `when` field is specifically used to establish time-based rotation intervals or schedules; for example, `@T00` requests rotation at midnight. A size value can additionally trigger rotation when a log reaches its configured limit. The `newsyslog` utility is normally invoked periodically by the system's scheduled maintenance facilities, evaluates the rules in this file, archives qualifying logs, and can signal the relevant daemon to reopen its log file. This makes `/etc/newsyslog.conf` the file that defines when BSD log files are rotated. FreeBSD documents the file format, including the schedule-related `when` field and the available rotation parameters, in the [newsyslog.conf\(5\) manual page](#). Operational behavior and command-line usage are documented in the [newsyslog\(8\) manual page](#).

QUESTION NO: 5 - (SIMULATION)

Which file on a BSD system defines the printers which are available by the lpr printing system'? (Specify the full name of the file including path)

ANSWER: See the explanation for the answer

Explanation:

The file is:

`/etc/printcap`

On BSD systems using the LPR/LPD printing system, `/etc/printcap` defines the available printers and their capabilities/configuration entries.

QUESTION NO: 6 - (SIMULATION)

What symbol is used in the vi editor to start the forward search for a string? (Specify ONLY the single character symbol without any parameters)

ANSWER: See the explanation for the answer

Explanation:

`/`

In `vi`, the slash character starts a forward search (for example, `/text`).

QUESTION NO: 7

Which command finds all directories within the current user's home directory?

- A. `find "$HOME" -type d`
- B. `find. -dir`
- C. `find / -d | home`
- D. `find -uid myname`
- E. `find -d home`

ANSWER: A

Explanation:

`find "$HOME" -type d` is correct because `$HOME` expands to the pathname of the current user's home directory, making that directory the starting point for the search. The `find` utility recursively walks the directory tree beneath its starting path. Its `-type d` primary restricts matches to entries whose file type is a directory, so the result contains the home directory itself

and every directory located below it, subject to filesystem permissions. Quoting "\$HOME" is good shell practice because it preserves the home-directory pathname as one argument if it contains spaces or other shell-significant characters. This form is also preferable to relying on the current working directory: the question specifically requires searching the current user's home directory, which may not be the directory from which the command is executed. The BSD `find` manual documents that a pathname begins each search and that `-type` evaluates the file type; `d` denotes a directory. See the [FreeBSD find\(1\) manual](#) and the [POSIX find specification](#).

QUESTION NO: 8

A shared project directory must ensure that newly created files inherit the directory's group, while preventing ordinary group members from deleting each other's files. The directory is assigned mode `3770`. Which TWO effects of this mode are correct? (Choose TWO correct answers.)

- A. New files created in the directory inherit the directory's group.
- B. Ordinary group members cannot remove files owned by other users in the directory.
- C. Users outside the directory's group receive read and execute access.
- D. Existing files in the directory become immutable.
- E. The sticky bit grants write access to every local user.

ANSWER: A B

Explanation:

The leading `3` sets both the set-group-ID and sticky bits. On a directory, the set-group-ID bit causes newly created entries to inherit the directory's group. The sticky bit restricts removal and renaming of entries to the file owner, directory owner, or superuser.

Mode `770` grants no permissions to users outside the owner and group classes. The sticky bit does not make files immutable, and it does not itself grant write access.