PassQueen
a.k.a dumpsqueen.com

LPIC-2 Exam 201, Part 1 of 2, version 4.5

LPI 201-450

Version Demo

Total Demo Questions: 20

Total Premium Questions: 204

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Linux Kernel	23
Topic 2, System Startup	42
Topic 3, Filesystem and Devices	39
Topic 4, Advanced Storage Device Administration	25
Topic 5, Network Configuration	31
Topic 6, System Maintenance	36
Topic 7, Domain Name Server	5
Topic 8, HTTP Services	3
Total	204

QUESTION NO: 1

After the downloading patch-4.6.4.xz from http://kernel.org, what are the next steps to prepare the build of a version 4.6.4 Linux kernel? (Choose two.)

A. Uncompress the file and move the resulting directory to /usr/src/linux

B. Apply the patch file to the kernel source directory containing kernel version 4.6.0

C. Apply the patch file to the kernel source directory containing kernel version 4.6.3

D. Uncompress the file using xz to get the uncompressed patch file

E. Use patch to apply the uncompressed patch file to the source directory of any previous kernel version

ANSWER: C D

Explanation:

`patch-4.6.4.xz` is an xz-compressed incremental kernel patch, so it must first be decompressed to make the patch data available to the `patch` utility. This can be done with `unxz patch-4.6.4.xz`, or equivalently by streaming its contents with `xzcat` to `patch`. The resulting patch is intended to update the immediately preceding stable source release in the same series: kernel 4.6.3. From within a clean Linux 4.6.3 source tree, a typical command is `patch -p1 < ../patch-4.6.4`. This transforms that source tree into the 4.6.4 source state, after which the usual kernel configuration and build process can proceed. Kernel.org publishes both complete source tarballs and incremental patches; the latter depend on the specific preceding source version rather than being suitable for arbitrary prior versions. The kernel.org archive documents the available patch artifacts, while the GNU patch manual describes applying unified patches and pathname stripping with `-p`. See kernel.org and GNU patch documentation.

QUESTION NO: 2

A regular user, joe, has just run:

./configure && make && make install

to build and install a program. However, the installation fails. What could be done to install the program? (Choose TWO correct answers.)

A. Install the binaries manually with suinstall.

B. Run make install with root privileges.

C. Do not run ./configure in order to maintain the default configuration for correct installation.

D. Rerun ./configure with a --prefix option where the user has permissions to write.

E. Run make install_local to install into /usr/local/.

ANSWER: B D

Explanation:

Running `make install` with root privileges is appropriate when the software has been configured to install into a system-owned directory, such as `/usr/local`, `/usr`, or another location that a regular user cannot modify. The configure and build phases generally do not need elevated privileges; using privilege only for the installation step follows the principle of least privilege. A typical approach is to build as the regular user and then use `sudo make install`, subject to the system's administrative policy.

Rerunning `./configure` with a `--prefix` option pointing to a directory writable by the user is also valid. Autoconf-based build systems use `--prefix` to select the base installation directory. For example, configuring with `./configure --prefix="$HOME/.local"` commonly allows installation without administrative access, placing executables in `$HOME/.local/bin` and related files beneath that directory. The user may then need to add that bin directory to `PATH`.

GNU Autoconf documents installation directory options and their effect on generated Makefiles at GNU Autoconf: Installation Directory Variables. Guidance on using `sudo` for controlled privilege elevation is available in the sudo manual.

QUESTION NO: 3

What should be done after updating the configuration file for syslogd in order to make the changes become effective? (Choose TWO correct answers.)

A. No action is required, syslogd will notice the updated configuration file after a few minutes.

B. Send the HUP signal to the syslogd process.

C. Restart the syslogd service.

D. Run the command syslogd -u.

ANSWER: B C

Explanation:

Sending the HUP signal to the syslogd process causes the daemon to reload its configuration without fully stopping and starting the logging service. Traditionally, a SIGHUP is the standard signal used by syslog implementations to instruct the running daemon to reread its configuration file and apply updated selector, facility, priority, and destination rules. This permits logging to continue with minimal interruption while the revised configuration takes effect.

Restarting the syslogd service is also valid because a newly started daemon reads its configuration during initialization. Administrators can restart the service through the init system in use, such as a SysV init script or a systemd unit where one is provided. A restart is particularly useful when a reload operation is unavailable, unsuccessful, or when changes require a complete daemon reinitialization. Both approaches ensure that the running logging daemon uses the updated configuration rather than retaining rules loaded at its earlier startup. The traditional syslogd documentation describes SIGHUP as reinitializing the daemon, including rereading configuration: syslogd(8) manual page. For modern rsyslog-based systems, configuration reload behavior is documented by the rsyslog documentation.

QUESTION NO: 4

In this example output, which descriptions match the purpose of the free, buff and cache columns? (Choose THREE correct answers.)

vmstat 1 100

procs ----------memory---------- ---swap-- -----io---- --system-- ----cpu----

r b swpd free buff cache si so bi bo in cs us sy id wa

0 0 0 282120 134108 5797012 0 0 0 2 0 0 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1007 359 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1117 577 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1007 366 0 0 100 0

A. Used swap space

B. RAM available for filesystem buffers

C. Available free RAM

D. RAM used for buffers

E. RAM used for filesystem cache

ANSWER: C D E

Explanation:

The `free`, `buff`, and `cache` fields are memory-accounting values reported by `vmstat`. **Available free RAM** matches `free`: it is memory that is currently unused by the kernel and can be allocated without first reclaiming cached data. This is distinct from the newer "available memory" estimate shown by tools such as `free`, which also considers reclaimable cache.

RAM used for buffers matches `buff`. Buffers are kernel-managed memory used for block-device-related metadata and buffering. **RAM used for filesystem cache** matches `cache`, which represents cached file data (the page cache) retained in RAM to make subsequent filesystem reads faster. Buffer and cache memory are normally reclaimable when applications require RAM, so large values in these fields are generally expected on an active Linux system rather than automatically indicating memory pressure.

The procps-ng `vmstat` manual documents these memory columns and defines `free` as idle memory, `buff` as memory used as buffers, and `cache` as memory used as cache. See the vmstat(8) manual page and the Linux kernel filesystem caching documentation.

QUESTION NO: 5

Which of the following commands shows capabilities and usable frequencies for the wireless interface wlan0?

A. iw phy phy0 info

B. iw dev wlan0 info

C. iw dev wlan0 show

D. iw phy wlan0 show

E. iw phy0 show

ANSWER: A

Explanation:

`iw phy phy0 info` displays detailed information about the wireless physical device named `phy0`, including its supported interface modes, bands, channels, frequencies, and transmit-power capabilities. A wireless network interface such as `wlan0` is a logical interface attached to a physical wireless device (a wiphy); on systems where `wlan0` is associated with `phy0`, querying that PHY provides the radio-level capabilities and available frequencies relevant to the interface. The `info` subcommand is specifically intended to show detailed PHY properties, including band sections with frequency and channel data. If the underlying PHY number is not already known, it can be identified from the output of `iw dev wlan0 info`, which reports the interface's wiphy number. The command syntax and PHY-information behavior are documented in the iw manual page. Additional background on the Linux wireless configuration interface is available from the Linux Wireless iw documentation.

QUESTION NO: 6 - (SIMULATION)

Which directory in /dev/disk/ can be used to determine the UUID of a connected hard disk?

ANSWER: See the explanation for the answer

Explanation:

The directory is `/dev/disk/by-uuid/`.

It contains symbolic links named after filesystem UUIDs, pointing to the corresponding device or partition. For example:

```
ls -l /dev/disk/by-uuid/
```

To see a device UUID directly, use:

```
blkid /dev/sda1
```

QUESTION NO: 7

A Linux server is running in single user mode for regular maintenance. Which commands are used to restore the server to its usual runlevel? (Choose TWO correct answers.)

A. telinit 0

B. shutdown -r now

C. sync

D. shutdown -h now

E. reboot

ANSWER: B E

Explanation:

`shutdown -r now` is correct because the `-r` action requests an immediate reboot. During the subsequent boot process, the init system starts the configured default target or runlevel, returning the host from its maintenance-oriented single-user environment to its normal operational state. The `now` argument schedules that reboot without delay, which is appropriate once maintenance is complete and the administrator has confirmed that the system can be restarted.

`reboot` is also correct because it initiates a system reboot. As with a reboot requested through `shutdown`, startup resumes through the normal boot sequence and reaches the system's configured default runlevel or, on systemd-based distributions, default target. This makes both commands practical ways to leave single-user mode when the intended outcome is to restart the server normally rather than merely change its current execution state. See the shutdown(8) manual page and the systemd.target documentation for the reboot action and default-target behavior.

QUESTION NO: 8

Which of the following commands can be used to script interactions with various TCP or UDP services?

A. ftp

B. nc

C. tcpdump

D. strings

E. wget

ANSWER: B

Explanation:

`nc` is the correct command because Netcat is a general-purpose networking utility designed to create connections to arbitrary network endpoints using either TCP or UDP. It reads data from standard input and writes received data to standard output, making it particularly suitable for shell scripts, pipelines, automated probes, and sending or receiving application-protocol requests. A script can use `nc` to connect to a host and port, provide a request as input, collect the response, or listen locally for incoming traffic. Its support for both connection-oriented TCP and datagram-oriented UDP is central to its usefulness when interacting with many different services rather than one specific protocol. Common implementations also provide useful scripting controls such as connection timeouts, source or destination port selection, listening mode, and optional verbose diagnostics. The OpenBSD `nc` manual describes it as a utility for arbitrary TCP and UDP connections and

listens, and documents its input/output behavior and protocol-selection options. See the OpenBSD nc manual page and the Debian nc(1) manual page.

QUESTION NO: 9

Which commands below are useful to collect data about remote filesystem connections? (Choose TWO correct answers.)

A. pidstat

B. nfsiostat

C. sadf

D. cifsiostat

ANSWER: B D

Explanation:

`nfsiostat` is designed to report I/O statistics for mounted Network File System filesystems. It obtains client-side NFS statistics and presents useful measures for each mount, including operation rates, retransmissions, RPC-related timing, and read/write throughput. These data help an administrator evaluate the behavior and performance of remote NFS storage connections from the client system.

`cifsiostat` provides corresponding statistics for mounted CIFS/SMB filesystems. It reports activity and performance information for CIFS mounts, such as operation counts and read/write data, making it useful when investigating access to remote SMB file servers. Both utilities are part of the sysstat toolset and focus specifically on network-mounted filesystem activity rather than general CPU, process, or local-disk statistics. Their reports can be sampled repeatedly over a specified interval, which supports observing trends and correlating remote filesystem behavior with application workloads. See the sysstat manual pages and the cifsiostat manual page for command usage and reported metrics.

QUESTION NO: 10

The main configuration file for autofs has this entry:

/home /etc/auto.home

What is the meaning of the /etc/auto.home file?

A. It has the indirect maps for the mounting of file systems.

B. It has configuration information, such as passwords and keys, for the remote file server.

C. It has configuration information on settings for the /home mount point.

D. It is the holds the SSL key to allow authentication to the remote file server.

ANSWER: A

Explanation:

`/etc/auto.home` has the indirect maps for the mounting of file systems. In an autofs master map, each entry associates a mount point with a map source. Here, `/home` is the indirect-mount base directory and `/etc/auto.home` is the local map file consulted by the automounter. Entries in that file normally provide a key, optional mount options, and a location. For example, an entry with the key `alice` can cause autofs to mount the specified remote filesystem at `/home/alice` when that path is accessed. The mount is created on demand rather than being mounted permanently at boot, and autofs can later unmount it after it has been idle for the configured timeout.

The term “indirect” is important: the map does not describe one filesystem mounted directly on `/home`. Instead, it describes potential child mount points below `/home`, selected by map keys. This master-map structure is documented in the

auto.master(5) manual page, while the key, options, and location syntax used by automount maps is described in the autofs(5) manual page.

QUESTION NO: 11

Which /etc/hosts.allow entries will permit access to sshd for users from the 192.168.1.0/24 subnet? (Choose TWO correct answers.)

A. sshd : 192.168.1.

B. sshd : 192.168.1

C. sshd : 192.168.1.0 netmask 255.255.255.0

D. sshd : 192.168.1.0/255.255.255.0

E. sshd : 192.168.1.0

ANSWER: A D

Explanation:

`sshd : 192.168.1.` is valid TCP Wrappers client-pattern syntax. A client address pattern ending with a period is interpreted as an IPv4 address prefix. Consequently, it matches every address whose first three octets are `192.168.1`, which is precisely the range from `192.168.1.0` through `192.168.1.255`. This represents the `192.168.1.0/24` network for the purpose of an access-control rule.

`sshd : 192.168.1.0/255.255.255.0` is also valid. TCP Wrappers supports an IPv4 network specification formed from a network address and a slash-separated dotted-decimal netmask. The mask `255.255.255.0` fixes the first 24 bits of the client address, so any client in the `192.168.1.0/24` subnet matches this `sshd` rule. Entries in `/etc/hosts.allow` grant access to matching clients, provided that the service is built with TCP Wrappers support and no preceding access-control rule changes the applicable result. The TCP Wrappers hosts access documentation describes both address-prefix matching and network/mask client patterns: hosts_access(5). See also the configuration-file details in hosts.allow(5).

QUESTION NO: 12

Which of the following tools are used to measure memory usage? (Choose THREE correct answers.)

A. mpstat

B. pstree

C. sar

D. top

E. vmstat

ANSWER: C D E

Explanation:

`sar`, `top`, and `vmstat` are standard Linux tools that provide memory-related performance data. `sar`, from the sysstat suite, can report memory and swap statistics over time; for example, its memory reports show free and available memory, memory committed as a percentage, active and inactive memory, and page-cache-related values. This makes it particularly useful for observing historical or periodically collected memory behavior. The `sar` documentation describes these reports in detail at the sysstat sar manual.

`top` provides an interactive, continuously refreshed view of system memory totals and per-process memory consumption. Its display includes physical-memory and swap summaries, while process entries expose measures such as resident memory,

virtual memory, and memory percentage. This supports both an overall assessment and identification of processes using substantial RAM. `vmstat` reports virtual-memory activity and system-wide memory figures, including free memory, buffer and cache values, swap activity, paging, and process run or block counts. Its periodic output is useful for recognizing memory pressure and paging patterns. See the vmstat manual page for the reported fields and usage.

QUESTION NO: 13

Which directory contains system-specific systemd unit files? (Specify the full path to the directory.)

A. /lib/systemd/system

B. /etc/systemd/system

ANSWER: B

Explanation:

`/etc/systemd/system` is the directory for system-specific unit files and administrator-managed systemd configuration. It is the appropriate location for units created locally by an administrator, as well as for overrides, replacement units, and symbolic links that enable units. Files in this directory have precedence over vendor-provided unit files, allowing a system administrator to tailor service behavior for the particular host without modifying package-owned files.

This separation is important for reliable package management: distribution or package-installed unit definitions belong in the vendor unit directory, while host-specific policy and customizations belong under `/etc`. For example, a locally written service definition can be placed in `/etc/systemd/system/example.service`, followed by `systemctl daemon-reload` so the system manager reloads unit definitions. systemd's documented unit load path lists `/etc/systemd/system` as the configuration directory with the highest persistent priority for system units. See the systemd.unit manual and the systemd system configuration documentation.

QUESTION NO: 14

In the below example output, which columns detail the percent of time the CPU spent running non-kernel code and the percent of time the CPU spent running kernel code? (Choose TWO correct answers.)

vmstat 1 100

procs ----------memory---------- ---swap-- -----io---- --system-- ----cpu----

r b swpd free buff cache si so bi bo in cs us sy id wa

0 0 0 282120 134108 5797012 0 0 0 2 0 0 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1007 359 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1117 577 0 0 100 0

0 0 0 282120 134108 5797012 0 0 0 0 1007 366 0 0 100 0

A. id

B. us

C. wa

D. sy

ANSWER: B D

Explanation:

In `vmstat` output, `us` reports the percentage of CPU time spent executing user-space processes. User-space work is non-kernel code, such as applications, command-line programs, and other processes running outside the kernel. This metric is useful when determining whether processor demand is primarily caused by application workloads.

The `sy` column reports the percentage of CPU time spent executing system, or kernel, code. Kernel time includes operating-system work performed on behalf of processes, such as system-call handling, process scheduling, memory management, and device-related operations. Together, `us` and `sy` describe the active CPU time divided between user-space execution and kernel execution during each reporting interval. In the displayed samples, both values are zero, indicating that no measurable time in those intervals was spent running either user-space or kernel code; the CPU is instead shown as idle.

The procps-ng `vmstat` manual defines these CPU fields explicitly: vmstat(8) manual page. Kernel CPU accounting terminology is also documented in Linux kernel CPU time accounting documentation.

QUESTION NO: 15

What is the full path to the directory which contains the scripts (or links to the original scripts) to run while the system boots to SysV-init runlevel 2?

A. /etc/rc2.d/

ANSWER: A

Explanation:

`/etc/rc2.d/` is the conventional SysV-init runlevel-specific directory on Debian-derived systems for runlevel 2. Its contents are normally symbolic links to service-control scripts stored in `/etc/init.d/`, rather than independent copies of those scripts. During a transition into runlevel 2, init processes entries in this directory according to their names: links beginning with `S` start services, while those beginning with `K` stop services. The numeric portion of a link name establishes the ordering in which those actions occur. For example, an entry such as `S01service` is handled before `S99service`. This organization lets administrators maintain each service's actual start/stop implementation in one central init-script directory while controlling which services run at each runlevel through the per-runlevel link directories. The Debian policy manual documents the use of `/etc/rc?.d/` directories and their start/stop link naming conventions. See the Debian Policy Manual section on system run levels and init.d scripts and the update-rc.d manual page.

QUESTION NO: 16

Which of the following init systems comes along with an own UEFI boot loader?

A. systemd

B. SysVinit

C. Upstart

D. OpenRC

E. launchd

ANSWER: A

Explanation:

systemd is correct because the systemd project includes `systemd-boot`, a lightweight UEFI boot manager. It is installed as an EFI executable, typically `systemd-bootx64.efi` on 64-bit x86 systems, and is started directly by UEFI firmware or through another UEFI boot entry. systemd-boot reads boot loader specification entries from the EFI System Partition and can present installed Linux kernels, unified kernel images, and other EFI bootable operating systems. This capability is part of

the systemd project and is distinct from systemd's core role as the userspace init system and service manager after the kernel has started. In an LPIC-2 context, this is the relevant association: systemd provides both the init/service-management framework and its accompanying UEFI boot-loader component. The systemd Boot Loader Specification also defines a standard layout for boot-entry files, allowing boot loaders such as systemd-boot to discover and display bootable entries consistently. See the systemd-boot documentation and the Boot Loader Specification.

QUESTION NO: 17

A system has one hard disk and one CD writer which are both connected to SATA controllers. Which device represents the CD writer?

A. /dev/hdb

B. /dev/sdd

C. /dev/scd1

D. /dev/sr0

E. /dev/sr1

ANSWER: D

Explanation:

`/dev/sr0` is the device node that represents the CD writer. In modern Linux systems, SATA storage devices are handled through the SCSI subsystem, regardless of whether the physical hardware uses SATA rather than parallel SCSI. Optical drives, including CD writers, are therefore exposed by the kernel using the `sr` driver naming convention, where `sr` denotes a SCSI CD-ROM device. The numeric suffix identifies the drive's order among detected optical devices, starting at zero. Because the system contains one CD writer, its expected device node is `/dev/sr0`.

A CD writer is accessed as a whole block device through this node. Disc-burning tools can use it directly, subject to the required permissions and the appropriate media being present. Linux may additionally create convenient symbolic links under paths such as `/dev/cdrom`, but `/dev/sr0` is the standard kernel device name for the first detected SCSI/SATA optical drive. The Linux kernel's device-number documentation identifies the `sr` family as SCSI CD-ROM devices, and the kernel `sr` driver documentation describes its role for SCSI-compatible CD-ROM hardware. See Linux allocated device numbers and Linux SCSI subsystem documentation.

QUESTION NO: 18

After installing a compiled kernel, it can not find any modules that are needed to be loaded. What make target was likely missed while installing the kernel?

A. make modules_install, modules_install

ANSWER: A

Explanation:

The required target is `make modules_install`. Building a Linux kernel produces the kernel image and, separately, any kernel features configured as loadable modules. The module object files are not usable by the newly installed kernel until the `modules_install` target copies them into the standard module tree, normally `/lib/modules/<kernel-release>/`. This directory is named for the exact kernel release, so the running kernel can locate modules matching its version through tools such as `modprobe` and the kernel's module-loading mechanism. A typical installation sequence therefore includes building the kernel, running `make modules_install`, and installing the kernel image and related boot files. After module installation, running `depmod` for the new kernel release generates module dependency metadata; kernel installation tooling

often performs this automatically, but it can also be run explicitly if needed. The kernel build system documents `modules_install` as the installation target for kernel modules and identifies `INSTALL_MOD_PATH` as the optional staging-root setting. See the Linux kernel documentation on Kbuild and the depmod manual page.

QUESTION NO: 19

Please enter the complete path to the main SysV init process configuration file.

A. /etc/inittab

ANSWER: A

Explanation:

The complete path is `/etc/inittab`. In the traditional System V init architecture, the kernel starts `init` as the first userspace process, and init reads `/etc/inittab` as its principal configuration file. This file defines the system's default runlevel through the `id` entry and specifies actions for init to perform at startup, on entering or leaving runlevels, in response to power events, and for terminal login processes such as getty. Administrators historically modified this file to configure which services and consoles were started for particular runlevels. Although many current Linux distributions use systemd and may retain `/etc/inittab` only for compatibility or omit it entirely, the SysV init configuration file requested by the question is specifically `/etc/inittab`. The Linux man-pages documentation describes inittab as the configuration file used by init, including its entry format and action field. See the inittab(5) manual page and the Linux Standard Base inittab specification.

QUESTION NO: 20 - (SIMULATION)

After configuring a new kernel, what file under /usr/src/linux/ contains the configuration?

ANSWER: See the explanation for the answer

Explanation:

The kernel configuration is stored in:

`/usr/src/linux/.config`

The hidden `.config` file is created or updated by kernel configuration tools such as `make menuconfig`, `make xconfig`, or `make oldconfig`.