

### Graded Assessments

## Linux File System Repair

A computer file system is software that provides structure for storing the operating system (OS) and all other software installed on system hard drives. A hard drive must be formatted with a file system before the operating system can be installed. Since Linux is an open source OS, innovators have created nearly 100 file systems that support Linux OS installations. Several common file systems that are used for Linux systems include ext, ext2, ext3, ext4, JFS, XFS, ZFS, F2FS, and more.

Symptoms of data corruption can include:

Symptoms of data corruption can include:

- System suddenly shuts down
- Software program will not launch or it crashes when opening a corrupted file. May also give an error message saying:
  - "File format not recognized" or
  - "[file name] is not recognized"
- Corrupted files and folders may no longer appear in the file system.
- The operating system (OS) may report bad sectors when failing to execute commands.
- Damaged platter-based hard drives can make clicking sounds or unusual vibrations.

Data corruption on system hard drives and file systems can be caused by

- **Software errors** -
  - Software errors can be any software event that interferes with normal hard disk read/write operations.
  - Viruses and malware can be designed to intentionally cause corruption to data.
  - Antivirus software can damage files if the software experiences problems while scanning or repairing the files.
- **Hardware malfunctions** -
  - Larger files are more likely to experience corruption than smaller files. Large files occupy more disk space, making them statistically more likely to cross a bad sector on the hard drive.
  - Hard drives that contain platters are at risk of experiencing malfunctioning read/write heads. Damaged heads can corrupt multiple files and directories in a single read/write transaction. Hard drives with moving mechanical parts are more likely to experience failures from moving parts that wear out over time.
- **Electrical damage** - Can happen when a power failure occurs while the system is writing data to a hard drive.

The most critical first step, after data corruption has been identified or suspected, is to shut down the affected hard drive(s). The reason for this step is to stop the cause of the corruption from writing to the hard drives. The longer the corruption activity continues, the more difficult recovering the data becomes.

Precautions should be taken before powering up a corrupted hard drive to run repair tools. It is important to minimize any read/write operations on the disk other than those produced by data recovery tools. One method to prevent further damage could be to have a corrupted Linux system boot from an external device or network (PXE boot). An alternative method might be to attach the corrupted hard drive as an external hard drive to a healthy system running Linux. A hard drive adapter or drive docking station can be used to convert an internal drive into an external device.

Before connecting a corrupted drive to a healthy system, the **automount** service must be disabled. The **fsck** command will not repair corruption on a mounted file system. In fact, mounting a corrupted file system can cause the healthy Linux system to crash. Although the corrupted file system should not be mounted, the device file for the corrupted hard drive in the **/dev** directory must be readable for the **fsck** command to access the drive.

**Important Warning:** The `fsck` command should NOT be used

- on a hard drive that was a member of a RAID array.
- on a mounted file system (must be unmounted).

An important command line recovery tool offered in the Linux operating system is the **fsck** command. It should be run anytime a Linux system malfunctions. The **fsck** command can check the file system and repair some, but not all, inconsistencies found. In some cases, **fsck** may recommend deleting a corrupted file or directory. The default setting for the **fsck** command is to prompt the user to approve or deny the repair of any problems found. The user running the **fsck** command must have write permissions for the corrupted file or directory to be able to approve a repair. If the user does not choose to repair inconsistencies found, the file system will remain in a corrupted state.

The **fsck** command will check for inconsistencies and and prompt the user to make decisions on whether or **fsck** should repair for the following problems:

- Block count is not correct
- Blocks and/or fragments that are:
  - allocated to multiple files
  - illegally allocated
- Block and/or fragment numbers listed in i-node metadata that are:
  - overlapping
  - out of range
  - marked free in the disk map
  - corrupted
- Map inconsistencies on the disk or in the i-node.
- Directory:
  - contains references to a file but that number does not equal the number of links listed in the same file's i-node metadata.
  - sizes are not multiples of 512

The following checks are not run on compressed file systems.

- **Directory checks:**
  - Directories or files that cannot be located or read.
  - The i-node map has an i-node number marked as being free in the directory entry.
  - The i-node number in the metadata is out of range.
  - The . (current directory) or .. (parent directory) link is missing or does not point to itself.
- **Fragments found in files larger than 32KB.**
- **Any fragments that are not listed as the last address of the file in an i-node metadata file.**

1. Enter **fsck** as a command line instruction. Syntax

```
fsck [n] [:p] [:y] [f] [FileSystem1name - FileSystem2name ...]
```

- The **-n flag**: Sends a "no" response to all **fsck** questions and does not allow **fsck** to write to the drive.
  - The **-p flag**: Presents error messages for kernel problems from displaying while automatically fixing those minor errors. Outside of recovering from data corruption, it is a best practice to run the **fsck -p** command regularly at startup as a preventative measure.
  - The **-y flag**: Sends a "yes" response to all **fsck** questions to automatically attempt to repair all inconsistencies found. This flag should be used only for severely corrupt file systems only.
  - The **-f flag**: Runs a **fsck** check that **excludes** file systems that were successfully unmounted for shutdown before the system crashed.
- **Filesystemname**: If you do not specify a file system, the **fsck** command checks all file systems in **/etc/fstab**, where the **check attribute** is set to **true**.
- For some advanced flags, use the **man fsck** command.

a. To have the **fsck** command check all of the default file systems and prompt the user on how to handle each inconsistency found, simply enter at a command line:

b. For ext, ext2, ext3, and ext4 file systems, the **e2fsck** command can be used:

c. To have the **fsck** command check specific file system(s) and automatically fix any inconsistencies found, enter:

- 0 = All scanned file systems have been restored to a functional state.
- 2 = **fsck** did not finish checks or repairs due to an interruption.
- 4 = File system has changed and the computer needs to be rebooted.
- 8 = **fsck** could not repair some or all file system damage.

In many Linux OS distributions, the **fsck** utility will automatically run at boot under certain circumstances, including:

- When a file system has been labeled as “dirty”, meaning that data scheduled to be written to the file system is different from what was actually written or not written to the disk. This could occur if the system shut down during a write operation.
- When a file system has been mounted multiple times (can be set to a specific value) without a file system check.

Configuring the **fsck** command to run automatically on boot and reboot differs depending on which brand and version of Linux is installed on the system. As a root or sudo user, use vi (visual instrument) to add the **fsck** command to the boot sequence.

1. In Debian and Ubuntu,
  - a. Edit the **rcS** file.
  - b. Add the following command to the **rcS** file:
2. In CentOS,
  - a. Create or edit a file named **autofsck**.
  - b. Add the following command to the **autofsck** file:

**Mark as completed**