Linux Administration

Comparing files, searching data

Xavier Belanger

This work is licensed under
a Creative Commons Attribution-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0/

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

How to compare files

- Two options are available when you need to compare files:
  - checking if the files are identical
  - looking for actual differences in content
- Using one or the other way depends on what your end goal is and the level of detail that you need.

Checking files attributes

- If you are not expecting any difference or corruption, you can check if files are identical by size. This is not a reliable solution, but may be sufficient in some basic cases.
- A more efficient way is to use a checksum.

File checksum

- A hash function can be used to create a checksum for a file, then comparing the checksum for each file could prove that they are identicals.
- Hashes are computed the same way across systems.
- Checksums can provide proof of integrity, not of authenticity.

File checksum commands

- Various functions are available: MD5, SHA-1, SHA-2 (224, 256, 384 and 512).
- *md5sum <file>*
- *sha1sum <file>*
- *sha256sum <file>*

Checking for differences

- Two commands can be used to compare files: *comm* and *diff*.
- *comm* is more limited and specialized, *diff* is usually the go-to solution.

comm

- This command works best with sorted files; it will compare two files line by line and display the lines that are common or unique between the two.
- *comm -12 <file1> <file2>*
  Print only lines present in both file1 and file2.
- *comm -3 <file1> <file2>*
  Print lines in file1 not in file2, and vice versa.

diff

- This command provides details about lines added, deleted, modified between two files.
- The output is truncated to the focus on where the changes are made; identical blocks are not displayed.
- The file used as reference should be listed first:
  diff <file1> <file2>

Searching for data

- *grep* is the most versatile command to search and extract text from a file.
- *strings* is another command to search for character strings, mostly in non-text files.

strings

- When looking for character strings in a file, run the strings command against it:
- *strings <file>*
- This is often used to identify binaries from unknown origin.

grep, egrep, zgrep

- grep is search for a pattern in files.
- *grep <pattern> <file1> <file2> …*
- *grep -E* (or *egrep*) can be used to access extended functions.
- *zgrep* can be used to search directly into compressed files.

grep useful options

- ignoring case: *grep -i*
- inverting matches: *grep -v*
- recursive search: *grep -R*
- counting matches: *grep -c*
- colorizing the output: *grep --color*
- printing results only: *grep -o*
- printing filename: *grep -H*