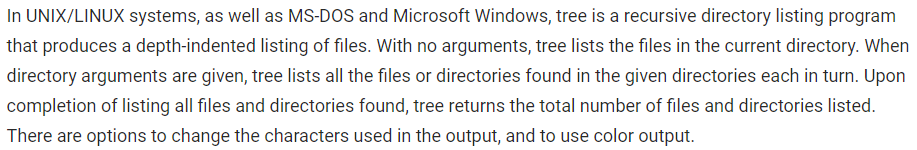
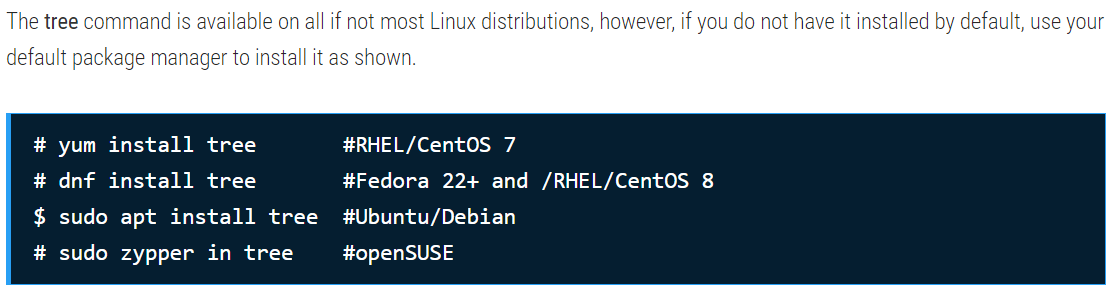
**tree command:**



**Installation:**



**-a :** All files are printed. By default, tree does not print hidden files (those beginning with a dot `.’). In no event does tree print the file system constructs `.’ (current directory) and `..’ (previous directory)

* **tree -a**

**Note :** you must use the -a option to also consider those files beginning with a dot `.’ for matching. Valid wildcard operators are `\*’ (any zero or more characters), `?’ (any single character), `[…]’ (any single character listed between brackets (optional – (dash) for character range may be used: ex: [A-Z]), and `[^…]’ (any single character not listed in brackets) and `|’ separates alternate patterns.

**-f :** Prints the full path prefix for each file.

* **tree -f**

**-d :** List directories only.

* **tree -d**
* **tree -df 🡪 to display the directories with full path**

You can specify the maximum display depth of the directory tree using the **-L** option. For example, if you want a depth of 2, run the following command.

* **tree -L 2**
* **tree -f -L 2**

To display only those files that match the wild-card pattern, use the **-P** flag and specify your pattern. In this example, the command will only list files that match cata\*, so files such as Catalina.sh, catalina.bat, etc. will be listed.

* **tree -P cata\***
* **tree -f -P cata\***

You can also tell the tree to prune empty directories from the output by adding the **--prune** option, as shown.

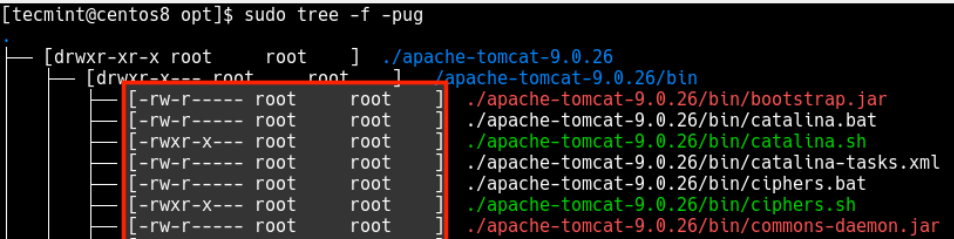
* **tree -f –prune**

There are also some useful file options supported by tree such as **-p** which prints the file type and permissions for each file in a similar way as the ls -l command.

* **tree -p**
* **tree -f -p**

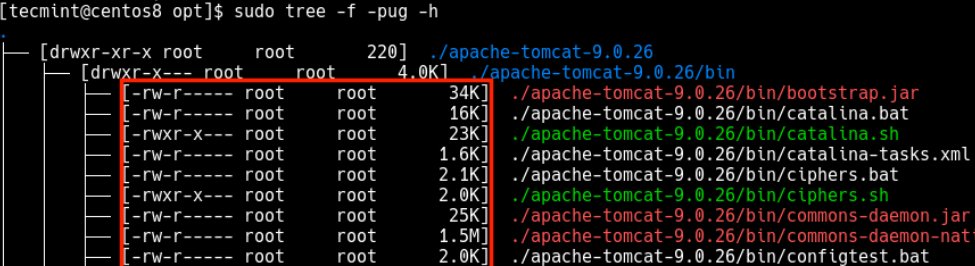
Besides, to print the username (or UID if no username is available), of each file, use the **-u** option, and the -g option prints the group name (or GID if no group name is available). You can combine the **-p, -u and -g** options to do a long listing similar to ls -l command.

* **tree -f -pug**



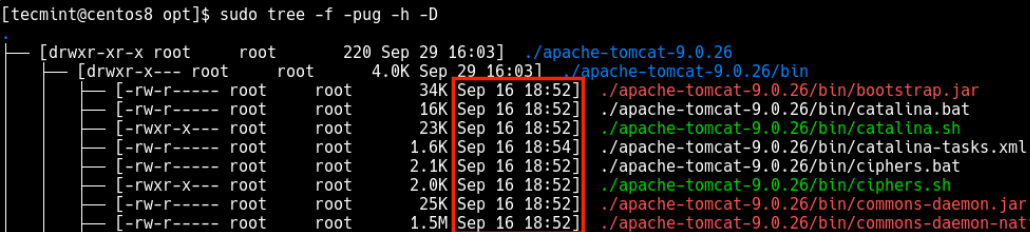
You can also print the size of each file in bytes along with the name using the **-s** option. To print the size of each file but in a more human-readable format, use the -h flag and specify a size letter for kilobytes (K), megabytes (M), gigabytes (G), terabytes (T), etc..

* **tree -f -s**
* **tree -f -h**



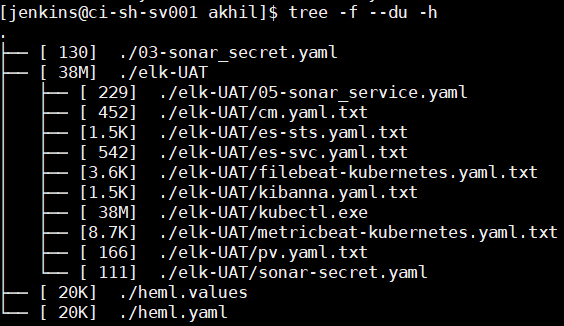
To display the date of the last modification time for each sub-directory or file, use the **-D** options as follows.

* **tree -f -pug -h -D**



Another useful option is **--du**, which reports the size of each sub-directory as the accumulation of sizes of all its files and subdirectories (and their files, and so on).

* **tree -f –du**
* **tree -f --du -h**



you can send or redirect the tree’s output to filename for later analysis using the **-o** option.

* **tree -o direc\_tree.txt**

if you want to run tree command for any particular directory other than the current one

* **tree [directory name]**
* **tree ../**

List those directories which have greater ‘N’ number of files/directories

* **tree --filelimit 3 ./GFG**

