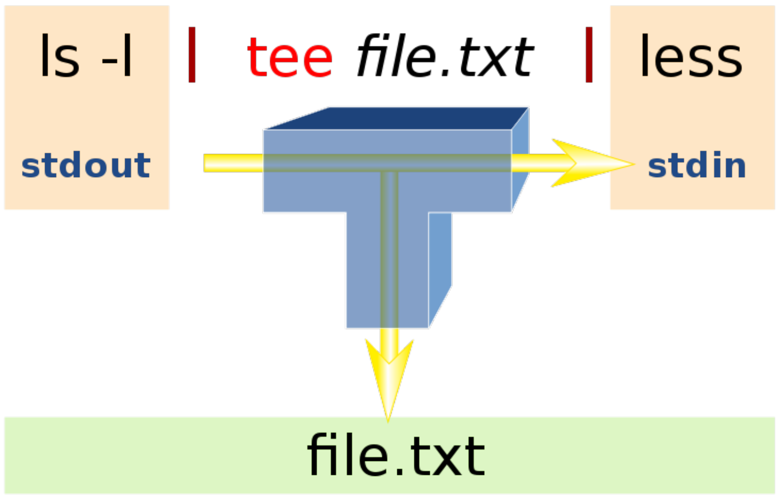
WIKIPEDIA

tee (command)

In computing, **tee** is a command in command-line interpreters (shells) using standard streams which reads standard input and writes it to both standard output and one or more files, effectively duplicating its input.[1] It is primarily used in conjunction with pipes and filters. The command is named after the T-splitter used in plumbing.[2]



Example usage of *tee*: The output of `ls -l` is redirected to *tee* which copies them to the file *file.txt* and to the pager `less`. The name *tee* comes from this scheme - it looks like the capital letter *T*

Contents

Description and syntax

tee is normally used to *split* the output of a program so that it can be both displayed and saved in a file. The command can be used to capture intermediate output before the data is altered by another command or program. The tee command reads standard input, then writes its content to standard output. It simultaneously copies the result into the specified file(s) or variables. The syntax differs depending on the command's implementation:

Unix-like

```
tee [ -a ] [ -i ] [ File ... ]
```

Arguments:

- **`File`** One or more files that will receive the "tee-d" output.

Flags:

- **`-a`** Appends the output to the end of `File` instead of writing over it.
- **`-i`** Ignores interrupts.

The command returns the following exit values (exit status):

- 0 The standard input was successfully copied to all output files.
- >0 An error occurred.

Using *process substitution* lets more than one process read the *standard output* of the originating process. Read this example from GNU Coreutils, tee invocation (https://www.gnu.org/software/coreutils/manual/html_node/tee-invocation.html).

Note: If a write to any successfully opened File operand is not successful, writes to other successfully opened File operands and standard output will continue, but the exit value will be >0.

4DOS and 4NT

```
TEE [/A] file...
```

Arguments:

- **`file`** One or more files that will receive the "tee'd" output.

Flags:

- **`/A`** Append the pipeline content to the output file(s) rather than overwriting them.

Note: When *tee* is used with a pipe, the output of the previous command is written to a temporary file. When that command finishes, *tee* reads the temporary file, displays the output, and writes it to the file(s) given as command-line argument.

Windows PowerShell

```
tee [-FilePath] <String> [-InputObject <PSObject>]
tee -Variable <String> [-InputObject <PSObject>]
```

Arguments:

- **`-InputObject <PSObject>`** Specifies the object input to the cmdlet. The parameter accepts variables that contain the objects and commands or expression that return the objects.
- **`-FilePath <String>`** Specifies the file where the cmdlet stores the object. The parameter accepts wildcard characters that resolve to a single file.

- **`-Variable <String>`** A reference to the input objects will be assigned to the specified variable.

Note: *tee* is implemented as a `ReadOnly` command alias. The internal cmdlet name is `Microsoft.PowerShell.Utility\Tee-Object`.

Examples

Unix-like

- To view and save the output from a command (lint) at the same time:

```
lint program.c | tee program.lint
```

This displays the standard output of the command `lint program.c` at the computer, and at the same time saves a copy of it in the file `program.lint`. If a file named `program.lint` already exists, it is deleted and replaced.

- To view and append the output from a command to an existing file:

```
lint program.c | tee -a program.lint
```

This displays the standard output of the `lint program.c` command at the computer and at the same time appends a copy of it to the end of the `program.lint` file. If the `program.lint` file does not exist, it is created.

- To allow escalation of permissions:

```
cat ~/.ssh/id_rsa.pub | ssh admin@server "sudo tee -a /root/.ssh/authorized_keys2 > /dev/null"
```

This example shows *tee* being used to bypass an inherent limitation in the sudo command. *sudo* is unable to pipe the standard output to a file. By dumping its stdout stream into `/dev/null`, we also suppress the mirrored output in the console. The command above gives the current user root access to a server over ssh, by installing the user's public key to the server's key authorization list.

In Bash, the output can be filtered before being written to the file—without affecting the output displayed—by using process substitution. For example, `ls --color=always | tee >(sed "s/\x1b[^m]*m//g" > ls.txt)` removes common ANSI escape codes before writing to `ls.txt`, but retains them for display.[3]

4DOS and 4NT

This example searches the file `wikipedia.txt` for any lines containing the string "4DOS", makes a copy of the matching lines in `4DOS.txt`, sorts the lines, and writes them to the output file `4DOSsorted.txt`:

```
find "4DOS" wikipedia.txt | tee 4DOS.txt | sort > 4DOSsorted.txt
```

Windows PowerShell

- To view and save the output from a command at the same time:

```
ipconfig | tee OutputFile.txt
```

This displays the standard output of the command `ipconfig` at the console window, and simultaneously saves a copy of it in the file `OutputFile.txt`.

- To display and save all running processes, filtered so that only programs starting with svc and owning more than 1000 handles are output:

```
Get-Process | Where-Object { $_.Name -like "svc*" } | Tee-Object ABC.txt | Where-Object { $_.Handles -gt
1000 }
```

This example shows that the piped input for *tee* can be filtered and that *tee* is used to display that output, which is filtered again so that only processes owning more than 1000 handles are displayed, and writes the unfiltered output to the file `ABC.txt`.

See also

- GNU Core Utilities
- Pipeline (Unix)
- List of Unix programs

References

- `tee (http://www.opengroup.org/onlinepubs/9699919799/utilities/tee.html)`: duplicate standard input – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group
- GNU tee manual (https://www.gnu.org/software/coreutils/manual/html_node/tee-invocation.html)

1. "Man Page for tee (posix Section 1)" (http://www.unix.com/man-page/POSIX/1/tee/). IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6. Retrieved 1 December 2013.
2. "In Unix, what do some obscurely named commands stand for?" (http://kb.iu.edu/data/abnd.html). Retrieved 2012-02-03.
3. "GNU Coreutils: tee invocation" (https://www.gnu.org/software/coreutils/manual/html_node/tee-invocation.html#index-_002d_002doutput_002derror). Retrieved 3 February 2016.

External links

- An introduction on Linux I/O Redirection "Linux I/O Redirection" (http://wadhavankar.org/tn/linux/user/standardIo.php) with tee

Retrieved from "https://en.wikipedia.org/w/index.php?title=Tee_(command)&oldid=802558036"