

**mWriter:** Philip Wilkinson

**Related Content:** We have a published post on [grep case insensitive](#), and an in-flight one on [grep exclude](#). Link to these if there's some relevant bridge between them. Other topics we'll do separately from this post include: grep multiple strings, grep multiple lines, grep multiple files (so don't cover them in this post in depth). We'll also eventually have a post on "grep" in general, so don't spend more than a sentence max introducing "what is grep"-like context.

**Topic Context:** Subtopics you *should* cover in this post include:

- ``grep count occurrences``
- ``grep count lines`` (slightly different from counting occurrences - if you have multiple occurrences on a single line, that would only count as +1 here)
- Any other useful angles you can think of related to counting with grep
- Show examples of edge cases and nuances

**Target Search intent:** *If I am an engineer that just typed ``grep count`` into Google, I would be most happy to find a page that shows the command to count the number of occurrences, prints the number of occurrences or counts the number of lines that the string occurs on.*

**Page Title / H1:** Grep Count Lines or Occurrences

**Description:** Using `grep -c` or `grep -o` combined with the `wc` command can help you efficiently count lines or occurrences in a file.

---

While `grep` is most often used for finding strings and neighboring text, it can be used for counting lines and occurrences as well.

## `#count-lines` `grep` `count` lines `#count-lines`

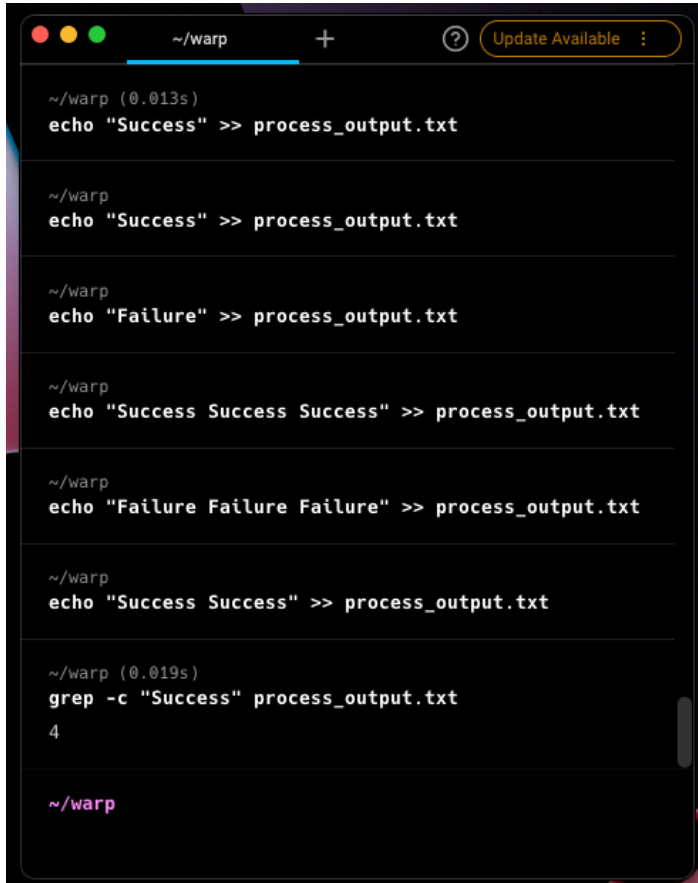
To count the number of lines that a string appears in using `grep`:

...

```
$ grep -c "string" <file_name>
```

...

For which the `-c` flag is used to count the number of lines that are matched and print out the number. This can be useful, for example, when you want to search through log files for the number of entries from a particular IP, endpoint or other identifier where you only care about the number of lines, not the full number of matches.

A screenshot of a terminal window titled '~ /warp'. The window shows a sequence of commands and their outputs. The first command is 'echo "Success" >> process\_output.txt' which outputs '~ /warp (0.013s)'. This is followed by another 'echo "Success" >> process\_output.txt' command. Then, an 'echo "Failure" >> process\_output.txt' command. Next, an 'echo "Success Success Success" >> process\_output.txt' command. Then, an 'echo "Failure Failure Failure" >> process\_output.txt' command. This is followed by an 'echo "Success Success" >> process\_output.txt' command. Finally, a 'grep -c "Success" process\_output.txt' command which outputs '4'. The prompt '~ /warp' is visible at the bottom.

```
~/warp (0.013s)
echo "Success" >> process_output.txt

~/warp
echo "Success" >> process_output.txt

~/warp
echo "Failure" >> process_output.txt

~/warp
echo "Success Success Success" >> process_output.txt

~/warp
echo "Failure Failure Failure" >> process_output.txt

~/warp
echo "Success Success" >> process_output.txt

~/warp (0.019s)
grep -c "Success" process_output.txt
4

~/warp
```

[#count-occurrences][.inline-code]grep[.inline-code] count occurrences[#count-occurrences]

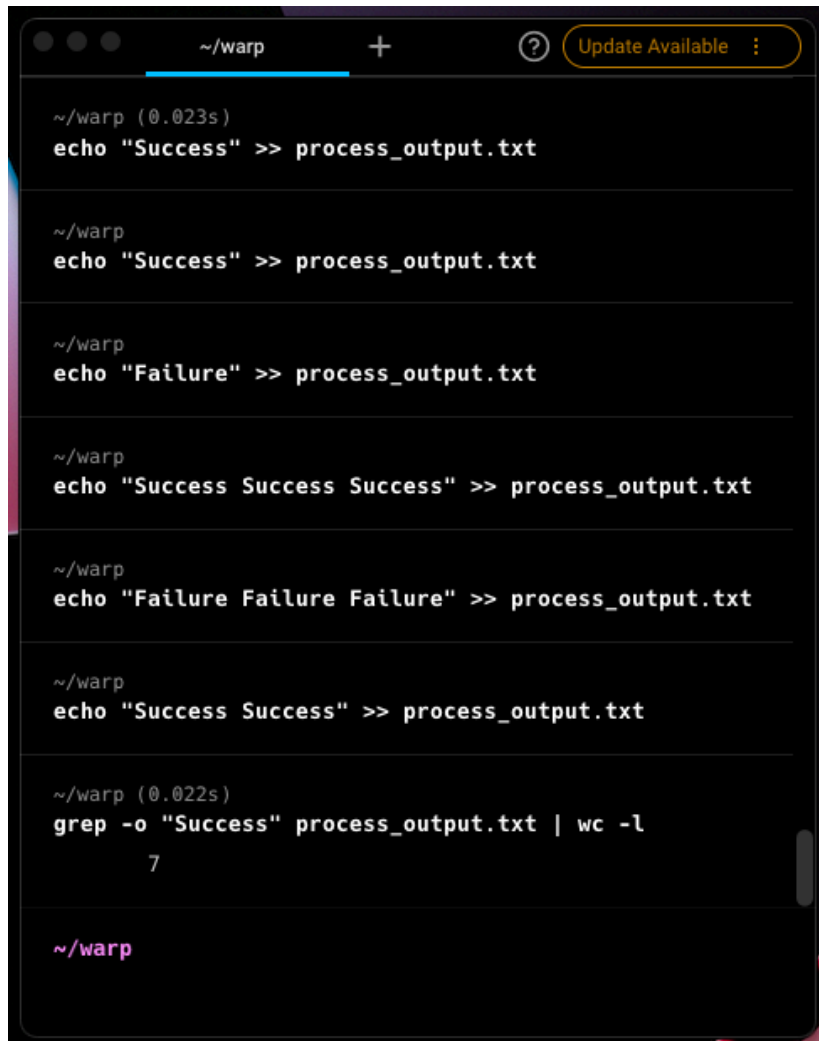
If however you want to count the number of occurrences of a string, beyond simply the number of lines, then the command can be used:

...

```
$ grep -o "string" <file_name> | wc -l
```

...

For which, the [.inline-code]-o[.inline-code] flag gets the occurrence of that string, while [.inline-code]wc -l[.inline-code] will count the number of times the occurrence appears on each line. This is useful when you want to know how many times a particular string occurs in a document such as the number of times a name, variable or IP address appears.



```
~/warp (0.023s)
echo "Success" >> process_output.txt

~/warp
echo "Success" >> process_output.txt

~/warp
echo "Failure" >> process_output.txt

~/warp
echo "Success Success Success" >> process_output.txt

~/warp
echo "Failure Failure Failure" >> process_output.txt

~/warp
echo "Success Success" >> process_output.txt

~/warp (0.022s)
grep -o "Success" process_output.txt | wc -l
7

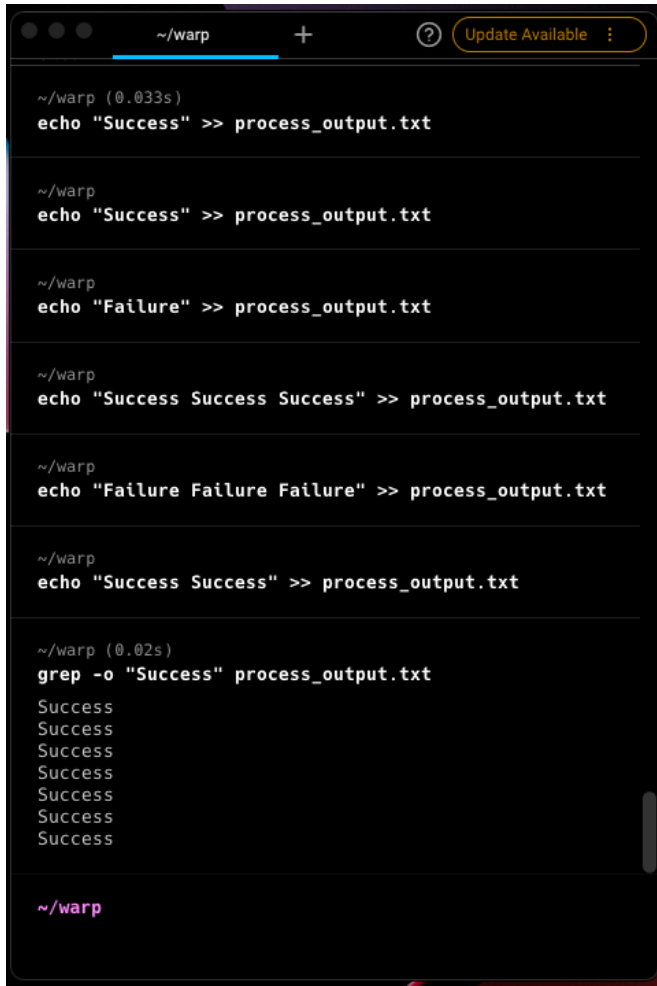
~/warp
```

[#print-occurrences][.inline-code]grep[.inline-code] print occurrences[#print-occurrences]

This command is simplified if you only want to print out the occurrences of that string rather than count the number of occurrences. This is done using:

```
...
$ grep -o "string" <file_name>
...
```

For which the [.inline-code]-o[.inline-code] flag is used to print out the matches on each line. This is useful when you want to see the context of the matches in the text they are part of, such as what strings the matches end up as.



```
~/warp (0.033s)
echo "Success" >> process_output.txt

~/warp
echo "Success" >> process_output.txt

~/warp
echo "Failure" >> process_output.txt

~/warp
echo "Success Success Success" >> process_output.txt

~/warp
echo "Failure Failure Failure" >> process_output.txt

~/warp
echo "Success Success" >> process_output.txt

~/warp (0.02s)
grep -o "Success" process_output.txt
Success
Success
Success
Success
Success
Success
Success
Success

~/warp
```

[#common-gotchas]Common “gotchas” when using  
[.inline-code]grep[.inline-code] to count[#common-gotchas]

[#using-regex][.inline-code]grep[.inline-code] uses regex  
standards[#using-regex]

It is important to know that the “string” following the [.inline-code]grep[.inline-code] command will match the document based on regular expression standards. This means that simply typing in [.inline-code]test[.inline-code] will match on longer strings containing that word such as “testing”. To match only the specific word use regex expressions or the [.inline-code]-w[.inline-code] flag. For example:

...

```
$ grep -o "\btest\b" <file_name> |wc -l
$ grep -ow "test" <file_name> | wc -l
```

...

[#multiple-words][.inline-code]grep[.inline-code] counting multiple words[#multiple-words]

If you want to search for multiple words at the same time, this command becomes more complicated. To print out the word and count you can use:

...

```
$ grep -o -E "string1|string2" <file_name> | sort | uniq -c
```

...

Read more on grep multiple strings to understand how the “or” part of this command came together.

[#counting-across-multiple-files][.inline-code]grep[.inline-code] counting across multiple files[#counting-across-multiple-files]

To match across multiple files and count the occurrences then this can become even more complicated. But the following command should print out the occurrences and which file they occur in:

...

```
$ grep -0 "string" <file_1> <file_2> | cut -d ':' -f 1 | uniq -c
```

...

All of these commands can also be combined with the [.inline-code]-i[.inline-code] flag so that the string match is [case insensitive](#)

[#using-awk-or-sed]Use [.inline-code]awk[.inline-code] or [.inline-code]sed[.inline-code] for text manipulation[#using-awk-or-sed]

If you want to manipulate text, or work with specific fields of a file, you will probably want to use a more specific tool such as [.inline-code]sed[.inline-code] or [.inline-code]awk[.inline-code]

[#learn-more]Find out more about  
[.inline-code]grep[.inline-code][#learn-more]

As always if you want to find out more about how to use the grep tool you can use:

...

```
$ man grep
```

'''

Which will print out all the options with explanations. Or:

'''

`$ grep --help`

'''

Which will print out a short page of all the available options.