Linux uniq command

Updated: 11/06/2021 by Computer Hope

```
[~]$ uniq
```

On Unix-like operating systems, the **uniq** command reports or filters out repeated lines in a file.

This page covers the GNU/Linux version of **uniq**.

Description

uniq filters out adjacent, matching lines from input file *INPUT*, writing the filtered data to output file *OUTPUT*. A matching line is "adjacent" if it's immediately before or after another matching line.

For example, consider a file **fruits.txt** with the following lines of text:

```
apple
apple
apple
pear
apple
pear
pear
```

If you run **uniq** on these lines of text, every line that matches an adjacent line is omitted:

```
uniq fruits.txt
```

```
apple
pear
apple
pear
```

Syntax

```
uniq [OPTION]... [INPUT [OUTPUT]]
```

Options

-c, --count	Prefix lines with a number representing how many times they occurred.
-d, --repeated	Only print duplicated lines.

<table>
<tr><td>**-D**,
--all-repeated[**=***delimit-method*]</td><td>Print all duplicate lines. *delimit-method* may be one of the following:
<table>
<tr><td>**none**</td><td>Do not delimit duplicate lines at all. This is the default.</td></tr>
<tr><td>**prepend**</td><td>Insert a blank line before each set of duplicated lines.</td></tr>
<tr><td>**separate**</td><td>Insert a blank line between each set of duplicated lines.</td></tr>
</table>
The **-D** option is the same as specifying **--all-repeated=none**.</td></tr>
<tr><td>**-f** *N*, **--skip-fields=***N*</td><td>Avoid comparing the first *N* fields of a line before determining uniqueness. A field is a group of characters, delimited by whitespace.
This option is useful, for instance, if your document's lines are numbered, and you want to compare everything in the line except the line number. If the option **-f 1** were specified, the adjacent lines
`1 This is a line. 2 This is a line.`
would be considered identical. If no **-f** option were specified, they would be considered unique.</td></tr>
<tr><td>**-i**, **--ignore-case**</td><td>Normally, comparisons are case-sensitive. This option performs case-insensitive comparisons instead.</td></tr>
<tr><td>**-s** *N*, **--skip-chars=***N*</td><td>Avoid comparing the first *N* characters of each line when determining uniqueness. This is like the **-f** option, but it skips individual characters rather than fields.</td></tr>
<tr><td>**-u**, **--unique**</td><td>Only print unique lines.</td></tr>
<tr><td>**-z**, **--zero-terminated**</td><td>End lines with 0 byte (null), instead of a newline.</td></tr>
<tr><td>**-w**, **--check-chars=***N*</td><td>Compare no more than *N* characters in lines.</td></tr>
<tr><td>**--help**</td><td>Display a help message and exit.</td></tr>
<tr><td>**--version**</td><td>Output version information and exit.</td></tr>
</table>

Notes

If *INPUT* is not specified, **uniq** reads from the standard input.

If *OUTPUT* is not specified, **uniq** writes to the standard output.

If no options are specified, matching lines are merged to the first occurrence.

uniq does not detect repeated lines unless they are adjacent. To omit ALL occurrences of identical lines, sort the input first, or use **sort -u** instead of **uniq**.

Examples

In the following examples, we have a text file, **myfruit.txt**, with five lines of text:

```
I have an apple.
I have an apple.
I also have two pears.
I have an apple.
I have three fruits total.
```

Here are several ways to run **uniq** on this file to omit repeated lines.

If adjacent lines are identical, display the line once

```
uniq myfruit.txt
```

```
I have an apple.
I also have two pears.
I have an apple.
I have three fruits total.
```

Same as above, but prefix each line with the number of times repeated

```
uniq -c myfruit.txt
```

```
   2 I have an apple.
   1 I also have two pears.
   1 I have an apple.
   1 I have three fruits total.
```

Show only duplicates (adjacent identical lines)

```
uniq -d myfruit.txt
```

```
I have an apple.
```

Show only unique lines (with no adjacent identical lines)

```
uniq -u myfruit.txt
```

```
I also have two pears.
I have an apple.
I have three fruits total.
```

Sort the lines alphabetically

The sort command sorts the lines of text alphabetically.

```
sort myfruit.txt
```

```
I also have two pears.
I have an apple.
I have an apple.
I have an apple.
I have three fruits total.
```

Now all identical lines are adjacent.

Show unique lines only once

The output of **sort** can be piped to **uniq**. Because the input is sorted, all identical lines will be adjacent. As a result, all identical lines in the file are displayed only once.

```
sort myfruit.txt | uniq
```

```
I also have two pears.
I have an apple.
I have three fruits total.
```

The same result can be accomplished by running **sort -u**.

```
sort -u myfruit.txt
```

```
I also have two pears.
I have an apple.
I have three fruits total.
```

Related commands

comm — Compare two sorted files line by line.

pack — Compress files using a Huffman algorithm.

pcat — Print the uncompressed contents of a compressed file.

sort — Sort the lines in a text file.

uncompress — Extract files from compressed archives.