Bash: Как посчитать количество слов в файле и вывести слово в зависимости от того, где оно находится в файле.

Category: awk,bash & sh,bash программирование,GNU/Linux,tr,wc
2026-01-19

Задача.

Задача: В примере есть простой файл *.txt. Требуется написать **bash** для вывода определенного слова, в зависимости от того, где оно находится в файле.

Выполнение задачи.

К примеру есть текстовый файл с содержимым:

```
# cd ~

# mcedit mice_and_mystics.txt
```

> *«О мышах и тайнах» (Mice and Mystics) — это кооперативная приключенческая настольная игра. Ключевой особенностью является сказочный сюжет, который создаётся прямо у нас на глазах. Дело в том, что в этой игре мы постоянно сталкиваемся с различными выборами, а от принятых решений зависит, где мы окажемся и как себя поведут сказочные персонажи. Никогда не знаешь, куда тебя заведёт выбранный путь! Итак, давным-давно, в незапамятные времена, существовало на свете Королевство людей. Правил ими добрый король Эндон. Он был хорошим, щедрым человеком, и все подданные любили его. К несчастью, его супруга рано скончалась, и своего единственного сына, принца Коллина, королю приходилось растить одному… По словам разработчика игры Джерри Хоторна, «О мышах и тайнах» — это не игра в сказку, а сказка, в которую можно играть на столе. Изначально Джерри создавал игру для своей маленькой дочери, которая только училась читать. На тот момент её любимыми*

животными были мышки, и Джерри пытался сделать настолько увлекательную игру про мышей, чтобы дочке самой захотелось бы проходить её дальше и читать всё больше игровых художественных вставок. Но Джерри немного перестарался. Потому что игра увлекла не только дочку, но и весь мир.

Вы можете посчитать количество слов в файле с помощью команды:

```
# wc -w < mice_and_mystics.txt
```

Ответ:

```
[root@emachines ~]# wc -w < ~/mice_and_mystics.txt
184
[root@emachines ~]#
```

Способ простого вывода слова по указанному порядковому номеру.

Команда **awk** разбивает строку на пробелы и печатает $wordnumber строковую часть, а tr используется для удаления новых строк.

Выбирая значение wordnumber= вы можете указывать порядковый номер слова и его выведет на экран:

```
# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=5 '{
print $wordnumber }'
# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=55 '{
print $wordnumber }'
# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=155 '{
print $wordnumber }'
```

Ответ:

```
[root@emachines ~]# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=5 '{ print $wordnumber }'
(Mice
[root@emachines ~]# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=55 '{ print $wordnumber }'
знаешь,
[root@emachines ~]# cat mice_and_mystics.txt | tr -d '\n' | awk -v wordnumber=155 '{ print $wordnumber }'
чтобы
[root@emachines ~]#
```

Способ указания диапазона поиска в тексте.

Команда **awk** ищет первую строку FNR==1 и в substr берет слово по номеру расположения слова в диапазоне, который вы могли бы увеличить или сохранить в соответствии с вашими потребностями.

Так как у нас в тексте файла 184 слова, то диапазон поиска будет от 1 (первого) до 184 (крайнего) слова:

```
# awk 'FNR==1{print substr($5,1,184); next}' mice_and_mystics.txt
# awk 'FNR==1{print substr($55,1,184); next}' mice_and_mystics.txt
# awk 'FNR==1{print substr($155,1,184); next}' mice_and_mystics.txt
```

Ответ:

```
[root@emachines ~]# awk 'FNR==1{print substr($5,1,184); next}' mice_and_mystics.txt
(Mice
[root@emachines ~]# awk 'FNR==1{print substr($55,1,184); next}' mice_and_mystics.txt
знаешь,
[root@emachines ~]# awk 'FNR==1{print substr($155,1,184); next}' mice_and_mystics.txt
чтобы
[root@emachines ~]# mcedit mice_and_mystics.txt
```

Способ отслеживания пробелов перед словами.

С пристальным отчетом пробелов:

```
# awk 'BEGIN{RS="[[:space:]]+"} NR==5' mice_and_mystics.txt
# awk 'BEGIN{RS="[[:space:]]+"} NR==55' mice_and_mystics.txt
# awk 'BEGIN{RS="[[:space:]]+"} NR==155' mice_and_mystics.txt
```

Ответ:

```
[root@emachines ~]# awk 'BEGIN{RS="[[:space:]]+"} NR==5' mice_and_mystics.txt
(Mice
[root@emachines ~]# awk 'BEGIN{RS="[[:space:]]+"} NR==55' mice_and_mystics.txt
знаешь,
[root@emachines ~]# awk 'BEGIN{RS="[[:space:]]+"} NR==155' mice_and_mystics.txt
чтобы
[root@emachines ~]#
```

или

```
# gawk 'NR==5' RS="[[:space:]]+" mice_and_mystics.txt
# gawk 'NR==55' RS="[[:space:]]+" mice_and_mystics.txt
# gawk 'NR==155' RS="[[:space:]]+" mice_and_mystics.txt
```

Ответ:

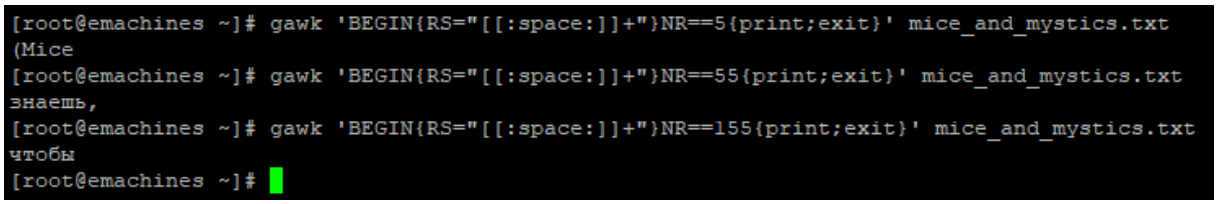
```
[root@emachines ~]# gawk 'NR==5' RS="[[:space:]]+" mice_and_mystics.txt
(Mice
[root@emachines ~]# gawk 'NR==55' RS="[[:space:]]+" mice_and_mystics.txt
знаешь,
[root@emachines ~]# gawk 'NR==155' RS="[[:space:]]+" mice_and_mystics.txt
чтобы
[root@emachines ~]#
```

Чтобы улучшить производительность, вы можете выйти из скрипта, когда совпадение найдено:

```
# gawk 'BEGIN{RS="[[:space:]]+"}NR==5{print;exit}'
mice_and_mystics.txt
```

```
# gawk 'BEGIN{RS="[[:space:]]+"}NR==55{print;exit}'
mice_and_mystics.txt
# gawk 'BEGIN{RS="[[:space:]]+"}NR==155{print;exit}'
mice_and_mystics.txt
```

Ответ:



Решение подходит для всех дистрибутивов **GNU/Linux**.

Оригиналы источников информации.

1. question-it.com «Заставить bash вывести определенное слово из файла *.txt».