

UNIX & LINUX

Как проверить, содержит ли переменная буквы и цифры

Спросил 5 лет, 4 месяца назад Изменено 5 лет, 4 месяца назад Просмотрено 3 тыс. раз



0



Я пытаюсь сделать проверку в скрипте оболочки на Linux, которая завершается, если STR содержит «только алфавиты» или содержит «только цифры». Она должна пройти, если STR содержит и цифры, и алфавиты. И не содержит специальных символов.

Я пытаюсь сделать что-то подобное, но это работает лишь частично.

```
#!/bin/sh
```

```
STR=$1
```

```
if [[ ! $STR =~ ^[[:alnum:]]*[:alpha:][[:alnum:]]*$ ]]; then
    echo "The input must contain both digits and alphabets like abc123"
    exit 1
fi
```

```
echo "Success"
exit 0
```

Проверка работает частично, то есть если STR содержит только цифры. Но не работает, если STR содержит все алфавиты. Я хочу exit 1 не выполнять только если STR содержит и алфавиты, и цифры, например abc123 или ABC123 .

Проверка не пройдена, если STR содержит abc .

Как сделать так, чтобы проверка работала и когда STR содержит abc только, и когда STR содержит 123 только?

[линукс](#)[Баш](#)[оболочка](#)

Делиться Улучшить этот вопрос
Следовать

отредактировано 6 марта 2019 г. в 12:23

спросил 6 марта 2019 г. в 11:54



TheWaterProgrammer
159 3 10

профессиональный совет: не называйте свой скрипт «тестом», если только вы не уверены в том, как его вызываете :) – Джефф Шаллер ♦ 6 мар. 2019 г. в 12:06

обратите внимание, что «alnum» включает в себя «альфа» и «числа», поэтому, если вы действительно хотите увидеть число, «alnum» – не лучший выбор. – Джефф Шаллер ♦ 6

Отсортировано по:

Наивысший балл (по умолчанию)

3 ответа



3



Здесь вам не нужно использовать сложные регулярные выражения. У вас есть два условия, поэтому используйте два теста:

```
if [[ $str == *[:digit:]* ]] &&
    [[ $str == *[:alpha:]* ]]
then
    printf "%s" contains both letters and digits\n' "$str"
else
    printf "%s" lacks either letters or digits\n' "$str"
fi
```

Вы также говорите что-то о "специальных символах", но не уточняете, что это такое. Предполагая, что вы имеете в виду символы, соответствующие `[:punct:]`, и что вы *не* хотите их в строке, вы можете использовать

```
if [[ $str == *[:digit:]* ]] &&
    [[ $str == *[:alpha:]* ]] &&
    [[ $str != *[:punct:]* ]]
then
    printf "%s" contains both letters and digits, and no specials\n' "$str"
else
    printf "%s" lacks either letters or digits, or contains specials\n' "$str"
fi
```

`[:punct:]` будет соответствовать любому из символов в строке

```
!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~
```

[Делиться](#) [Улучшить этот ответ](#)[Следовать](#)

отредактировано 6 марта 2019 г. в 12:55

ответил 6 марта 2019 г. в 12:50



Все еще жду ♦
339 тыс. 37 681
990



2



Регулярное выражение `^[[:alnum:]]*[:alpha:][[:alnum:]]*$` не проверяет цифры отдельно, так как `[:alnum:]` соответствует также буквам. `[:alpha:]` должно быть подмножеством `[:alnum:]`, поэтому шаблон соответствует любой строке, которая состоит только из букв и цифр, содержащей хотя бы одну букву.

(Предполагая обычное жадное сопоставление, если входные данные равны `abcd`, то `abc` будет сопоставлено с `[:alnum:]*`, final `d` будет сопоставлено с `[:alpha:]`, а last `[:alnum:]*` не будет (необходимо) соответствовать чему-либо.)



Если вы хотите проверить, что строка содержит хотя бы одну букву и одну цифру, проще разделить это на два теста, то есть тест на `[[[:alpha:]]]` и `[[[:digit:]]]`:

```
if [[ $str =~ [[[:alpha:]]] && $str =~ [[[:digit:]] ]]; then
    echo "contains a letter and a digit"
```

Если вы также хотите отклонить строки, содержащие что-либо еще (например, знаки препинания), добавьте проверку для этого:

```
if [[ $str =~ [[[:alpha:]]] && $str =~ [[[:digit:]]] && ! $str =~ [^[:alnum:]] ];
then
    echo "contains a letter and a digit, but no non-alphanumerics"
```

Чтобы найти случаи сбоя, инвертируйте весь тест, добавив `!` в начало (`if ! [[...]]`).

(Чтобы сделать это с помощью одного регулярного выражения, можно использовать что-то вроде этого:

```
^([[:alnum:]]*[[[:alpha:]]][[:alnum:]]*[[[:digit:]]][[:alnum:]]*|[[[:alnum:]]*[[[:digit:]]][[:alnum:]]*[[[:alpha:]]][[:alnum:]]*)$
```

но это немного ужасно)

Делиться Улучшить этот ответ

Следовать

отредактировано 6 марта 2019 г. в 13:07

ответил 6 марта 2019 г. в 12:55



илккачу

141 тыс. 16 249 417



0



Попробуйте сделать это проще. Разделите вопрос на два:

```
...
if [[ $STR =~ ([a-zA-Z]+) && $STR =~ ([0-9]+) ]]; then
...
```



Делиться Улучшить этот ответ Следовать



ответил 6 марта 2019 г. в 12:32



Хуан

251 3 10

Вы пропустили оператор `not`, который есть `!`. Предложение `not` важно для меня здесь, чтобы понять, как это работает – [TheWaterProgrammer](#) 6 мар. 2019 г. в 12:37

Разобрался, что нет. `if [[!($STR =~ ([a-zA-Z]+) && $STR =~ ([0-9]+))]]`; тогда должно быть – [TheWaterProgrammer](#) 6 мар. 2019 г. в 12:40

Насколько я понимаю ваш вопрос, вам НЕ нужно "не". "Если", которое я написал, означает: "есть ли у меня здесь какая-нибудь буква (обычная или заглавная)? ... и есть ли у меня здесь какая-нибудь цифра?", что в точности соответствует тому, что вы написали в описании: "Я хочу выход 1 ... только если STR содержит и буквы, и цифры" – [Хуан](#) 6 мар. 2019 г. в 12:42

На самом деле, если вы поставите это «не», вы разрушите логику. – [Хуан](#) 6 мар. 2019 г. в 13:14
