

# UNIX & LINUX

## вставка sed: сохранение начальных пробелов из вставленной строки

Задано 2 года, 7 месяцев назад · Изменено 2 года, 7 месяцев назад · Просмотрено 2 тыс. раз



3



У меня есть переменная (извлеченная из `diff` результата файла) с начальными пробелами, которую я хочу вставить в файл в определенной строке. Переменная:

```
add="    public function add($hardware_id = null)"
```

Но когда я вставляю строку с помощью `sed`, начальные пробелы не выводятся:

```
$ sed "50i$add" file
...
public function add($hardware_id = null)
...
```

Как сохранить начальные пробелы?

`bash` `обработка текста` `sed`

Поделиться

Улучшите этот вопрос

Подписаться

отредактировано 6 марта 2021 г. в 7:46

спрошено 5 марта 2021 г. в 22:43



**schrodingerscatcuriosity**

12.1k 3 31 57

3 Ответа

Отсортировано по:

Наивысший балл (по умолчанию) ⚡



3



Чтобы сохранить начальные пробелы при вставке переменной, вам просто нужно *один* раз экранировать пробелы обратной косой чертой `\` после *команды* "вставить `i`", вот так:

```
$ sed '50i\'"$add" file
    public function add($hardware_id = null)
```

1. Не специфичен для переменных, было бы то же самое, если бы текст был вставлен буквально.

## 2. Это не заданное в POSIX поведение, которое требует экранированного буквального перевода строки перед вставляемым текстом.

Поделиться

Улучшите этот ответ

Подписаться

отредактировано 6 марта 2021 г. в 1:49

ответил 5 марта 2021 г. в 22:43

**schrodingerscatcuriosity**

12.1k 3 31 57



2



Лучший способ добиться этого - использовать `i\` синтаксис. Но поскольку мы находимся под защитой двойных кавычек, мы избегаем обратной косой черты. И переменная должна быть безопасной для этой операции путем экранирования новой строки (ов) и обратной косой черты (ов)

```
add='      public function add($hardware_id \= null
)'
```

#-----  
# make variable safe to use  
# by escaping backslash n newline

```
esc_add=$(printf '%s\n' "$add" |
sed -e 's:\\&&g;$!s:$\\&g')
```

```
sed -e "
5i\\
$esc_add
" file | nl
```

```

1  ABC
2      2  3  4:blockA-1
3      7  9  4:blockA-2
4      1  2  5:blockA-3/
5      public function add($hardware_id \= null
6  )
7  ABC
8      13 11 17:blockB-1
9      2  1  1:blockB-2/
10 ABC
11      7  9  14:blockC-1
12      5  8  2:BlockC-2
13      9  9  9:BlockC-3
14      7  1  2:BlockC-4$
```


Поделиться Улучшите этот ответ Подписаться

ответил 6 марта 2021 г. в 4:33

**guest\_7**

5,698 1 6 13

Вы бы хотели запустить оба `sed` файла в C языковом стандарте, иначе это потенциально оставит некоторые уязвимости при внедрении произвольных команд. Попробуйте, например, с `add='${u03b1\ne uname>&2}'` помощью GNU `sed` в локали, используя кодировку BIG5 или BIG5-HKSCS (например, `zh_HK` в Ubuntu 20.04), где `U + 03B1 ( α )` кодируется как `0x3 0x5c (0x5c` также является обратной косой чертой). В более общем плане, лучше избегать вставки произвольных данных в аргумент кода интерпретаторов (как `sed` здесь), а когда вы все же это сделаете, выполните очистку, как вы это делали, но имейте в виду, что экранирование

обратной косой черты надежно только в языковом стандарте C. -Stéphane Chazelas 6 марта 2021 г. в 6:34 



Сделайте себе одолжение и используйте для этого perl or awk вместо sed here:

2

```
add=$' anything including \\backslash\\  
and newline and even non\x80-text or overlong lines '
```



```
ADD=$add perl -pe '$_ = "$ENV{ADD}\n$_" if $. == 5' < "$file"
```



Или (хотя это может привести к сбою из-за слишком длинных строк или нулевых байтов в содержимом, и будьте осторожны, это добавит завершающий символ новой строки, если его изначально не было):

```
ADD=$add LC_ALL=C awk 'NR == 5 {$0 = ENVIRON["ADD"] ORS $0}; 1' < "$file"
```

Поделиться

Улучшите этот ответ

Подписать

отредактировано 6 марта 2021 г.  
в 6:35

ответил 6 марта 2021 г. в 6:21



Stéphane Chazelas

528 тыс. 91 1018  
1508