Отчет по прохождению внешнего курса

Часть 3. Продвинутые темы

Скобеева А.А.

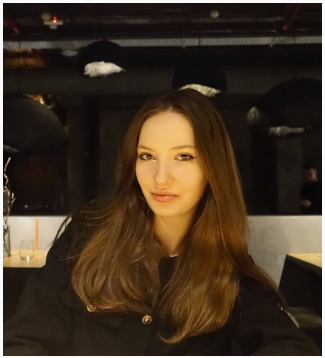
07 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Скобеева Алиса Алексеевна
- студентка 1-го курса направления “Прикладная информатика”
- Российский университет дружбы народов
- 1132246836@pfur.ru



Вводная часть

Актуальность

- Данная презентация актуальна для всех, кто хочет пройти внешний курс “Введение в Linux”

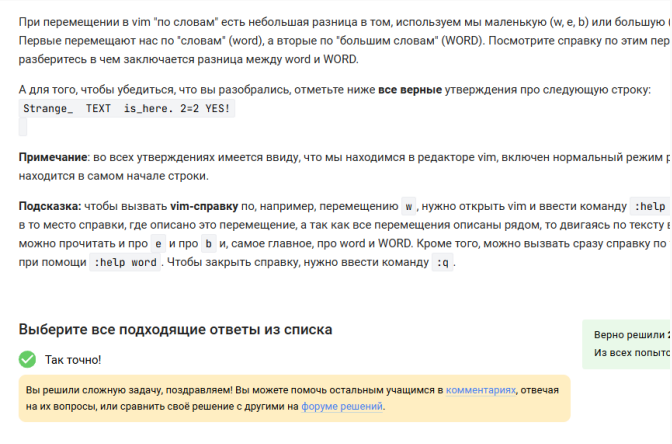
Цели и задачи

- Изучить текстовые и видеоматериалы, а также выполнить все практические задания 3 раздела

Основная часть

Текстовый редактор vim

- Изучив материалы данного раздела и выполнив все практические задания мы научились пользоваться текстовым редактором vim.



При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (w, e, b) или большую (
Первые перемещают нас по "словам" (word), а вторые по "большим словам" (WORD). Посмотрите справку по этим пер
разберитесь в чем заключается разница между word и WORD.

А для того, чтобы убедиться, что вы разобрались, отметьте ниже **все верные** утверждения про следующую строку:

```
Strange_  TEXT  is_here. 2=2 YES!
```

Примечание: во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим р
находится в самом начале строки.

Подсказка: чтобы вызвать **vim-справку** по, например, перемещению `w`, нужно открыть vim и ввести команду `:help`
в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаясь по тексту в
можно прочитать и про `e` и про `b` и, самое главное, про word и WORD. Кроме того, можно вызвать сразу справку по
при помощи `:help word`. Чтобы закрыть справку, нужно ввести команду `:q`.

Выберите все подходящие ответы из списка

Верно решили
Из всех попыт

Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в комментариях, отвечая на их вопросы, или сравнить своё решение с другими на форуме решений.

- [ ] Чтобы попасть в конец строки, нужно одинаковое число нажатий, что на W, что на w
- [ ] В этой строке 5 "слов" (word)
- [ ] В этой строке 9 "больших слов" (WORD)

Скрипты на bash: основы

- Изучив материалы данного раздела и выполнив все практические задания мы научились писать небольшие скрипты и запускать их.

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: arguments.sh.

Напишите скрипт на bash, который принимает на вход два аргумента и выводит на экран строку следующего вида:

```
Arguments are: $1=первый_аргумент $2=второй_аргумент
```

Например, если ваш скрипт называется `./script.sh`, то при запуске его `./script.sh one two` на экране должно появиться:

```
Arguments are: $1=one $2=two
```

а при запуске `./script.sh three four` будет:

```
Arguments are: $1=three $2=four
```

Подсказка: в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов.

Напишите программу. Тестируется через stdin → stdout

Верно решили **25 053** учащихся
Из всех попыток **41%** верных

Абсолютно точно.

Теперь вам доступен Форум решений, где вы можете сравнить свое решение с другими или спросить совета.

```
var1=$1
var2=$2

echo "Arguments are: \$1=$var1 \$2=$var2"
```

Скрипты на bash: ветвления и циклы

- Изучив материалы данного раздела и выполнив все практические задания мы научились использовать управляющие конструкции языка bash, которые позволяют писать скрипты, где часть инструкций выполняется только при опр. условиях(ветвления), а часть инструкций выполняется по много раз подряд(циклы).

```
while [[ 1==1 ]]
do
        group=""
        echo "enter your name:"
        read name
        if [[ -z $name ]]
        then
                break
        fi
        echo "enter your age:"
        read age
        if [[ $age -eq 0 ]]
        then
                break
        fi
        if [[ $age -le 16 ]]
        then
                group="child"
        elif [[ $age -le 25 ]]
        then
```

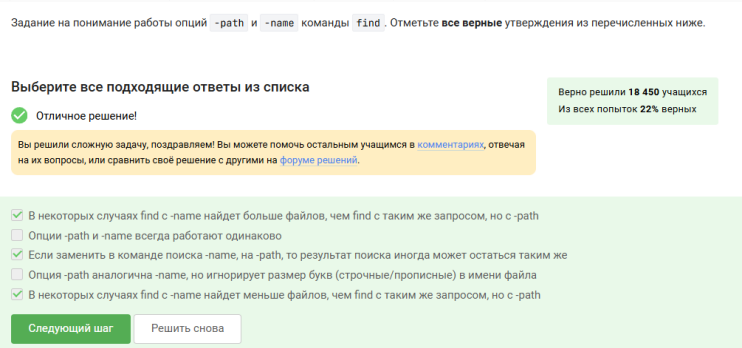
Скрипты на bash: разное

- Изучив материалы данного раздела и выполнив все практические задания мы научились писать довольно сложные и полезные скрипты на bash. Также мы изучили несколько тем: арифметические операции, запуск внешних программ и обработка результатов их работы; понятие функций в языке bash и их использование.

```
#!/bin/bash
while [[ True ]]
do
  read birinchi amal ikkinchi
  if [[ $birinchi == "exit" ]]
  then
    echo "bye"
    break
  elif [[ "$birinchi" =~ "^[0-9]+$" && "$ikkinchi" =~ "^[0-9]+$" ]]
  then
    echo "error"
    break
  else
    case $amal in
"+") let "result = birinchi + ikkinchi";;
"-") let "result = birinchi - ikkinchi";;
"/") let "result = birinchi / ikkinchi";;
"*") let "result = birinchi * ikkinchi";;
"%") let "result = birinchi % ikkinchi";;
"**") let "result = birinchi ** ikkinchi";;
*) echo "error" ; break ;;
```

Продвинутый поиск и редактирование

- Изучив материалы данного раздела и выполнив все задания мы научились работать с потоковым текстовым редактором sed, который позволяет не только искать слова в файлах, но и сразу же эти файлы редактировать.

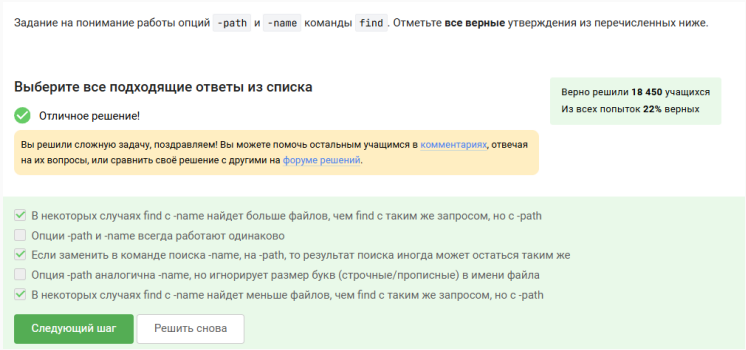


Строим графики в gnuplot

- Изучив материалы данного раздела и выполнив все практические задания мы научились использовать базовые команды gnuplot для работы в интерактивном режиме, а также научились писать gnuplot-скрипты.
- 

Разное

- Изучив материалы данного раздела и выполнив все практические задания мы узнали много нового об еще нескольких важных темах.
-



Результаты

- Мы изучили все материалы 3-го раздела и успешно выполнили все практические задания.