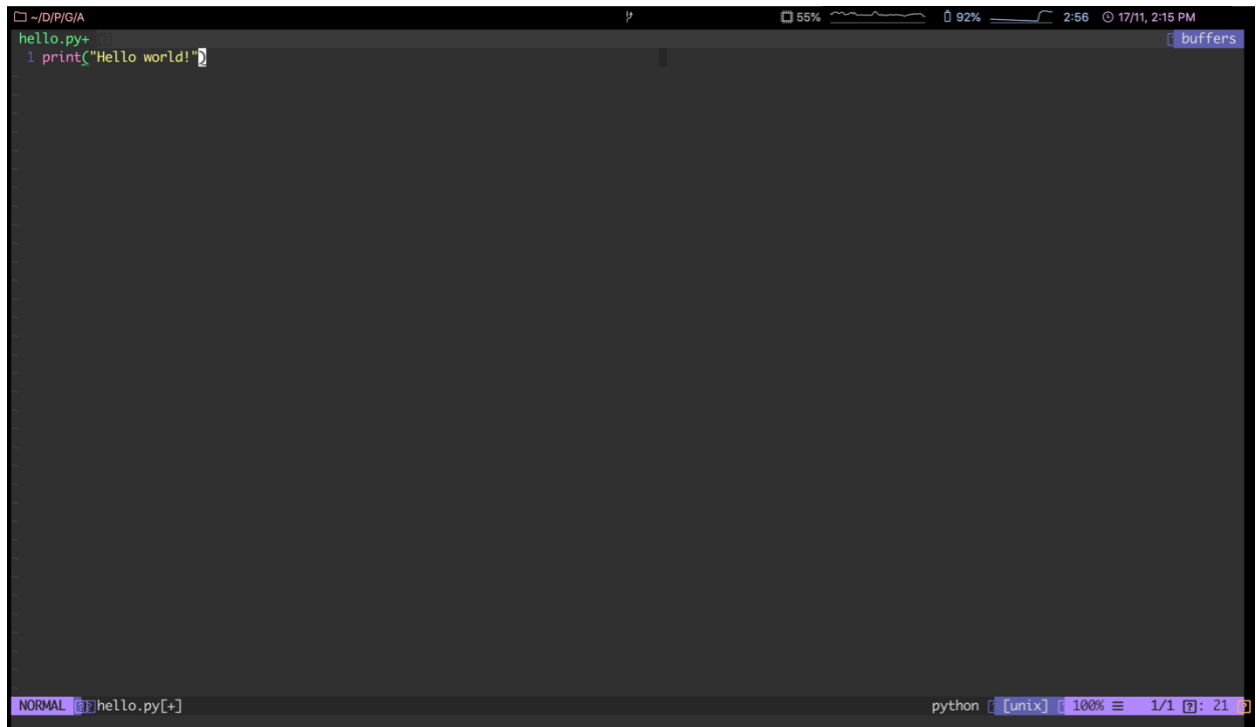


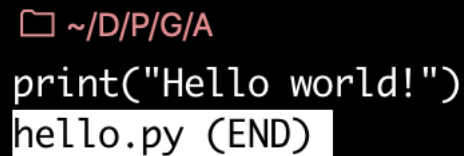
1. С помощью текстового редактора Vim создать файл с программой на Python, выводящей текст Hello, world!



The screenshot shows a Vim editor window with a dark background. The top status bar indicates the file path is ~/D/P/G/A, the file is named hello.py, and it is in normal mode. The main editing area shows a single line of Python code: `1 print("Hello world!")`. The bottom status bar shows the file is encoded in utf-8, has 1 line and 21 characters, and is a Python file.

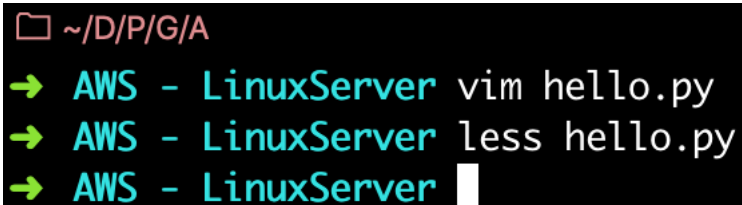
Готово.

Содержимое файла:



The screenshot shows the output of the `less` command in a terminal. It displays the file path `~/D/P/G/A`, followed by the Python code `print("Hello world!")`, and ends with `hello.py (END)`.

Команды в консоли:



The screenshot shows a terminal window with three commands entered: `vim hello.py`, `less hello.py`, and a prompt for another command. The terminal output shows the file path `~/D/P/G/A` and the command prompt `AWS - LinuxServer`.

2. Запустить команду, определяющую число строк в файле.

Поскольку в этом файле только одна строка – это не интересно, возьмем другой файл на Python'е, например быструю сортировку Хоара, часть его покажу на скрине:

```

def quick_sort_no_memory(array: list, fst: int, lst: int):

    if fst >= lst:
        return

    print(array)
    pivot = array[random.randint(fst, lst)]
    i, j = fst, lst

    while i <= j:

        while array[i] < pivot:
            i += 1

        while array[j] > pivot:
            j -= 1

        if i <= j:
            array[i], array[j] = array[j], array[i]
            i, j = i + 1, j - 1

    quick_sort_no_memory(array, fst, j)
    quick_sort_no_memory(array, i, lst)

quick_sort_no_memory(array, 0, len(array) - 1)
print(array)

```

И того – задание – посмотреть количество строк:

```

❏ ~/D/P/G/D/src
➔ src git:(Lesson_7) less Sort_Hoar.py
➔ src git:(Lesson_7) wc -l Sort_Hoar.py
  60 Sort_Hoar.py
➔ src git:(Lesson_7) █

```

Того – 60 строк,

Более подробная статистика:

```
→ src git:(Lesson_7) wc Sort_Hoar.py
60      137      1165 Sort_Hoar.py
→ src git:(Lesson_7) █
```

60 – строк, 137 слов и 1165 символов

3. Создать еще один файл с командой на Python, выводящей текст Linear regression.

```
→ AWS - LinuxServer vim Regression
→ AWS - LinuxServer python3 Regression
Linear regression
→ AWS - LinuxServer █
```

4. Объединить эти два файла с помощью команды cat.

```
→ AWS - LinuxServer cat hello.py Regression > new.py
```

```
→ AWS - LinuxServer less new.py
```

Посмотрим содержимое

```
❏ ~/D/P/G/A
print("Hello world!")
print("Linear regression")
new.py (END)
```

5. Придумать три случая применения команды cat для работы с текстовыми файлами.

5.1 – посмотреть содержимое

```
→ AWS - LinuxServer cat new.py
print("Hello world!")
print("Linear regression")
→ AWS - LinuxServer █
```

5.2 объединить два файла

```
→ AWS - LinuxServer cat hello.py Regression > new.py
```

5.3 Перенаправить вывод

```
→ AWS - LinuxServer cat > new_file.text  
This is new text line  
an done more  
and more
```

Посмотреть содержимое:

```
→ AWS - LinuxServer cat new_file.text  
This is new text line  
an done more  
and more
```