Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Вступ до операційної системи Linux**

Лабораторна робота № 2

«Вивчення інструментів командної строки Linux»

Виконав:

студент групи ІО-64

Андрійчук Д. А.

Залікова книжка №6401

Перевірив Габінет А. В.

Київ

2016 р.

(Далі весь текст буде російсько)

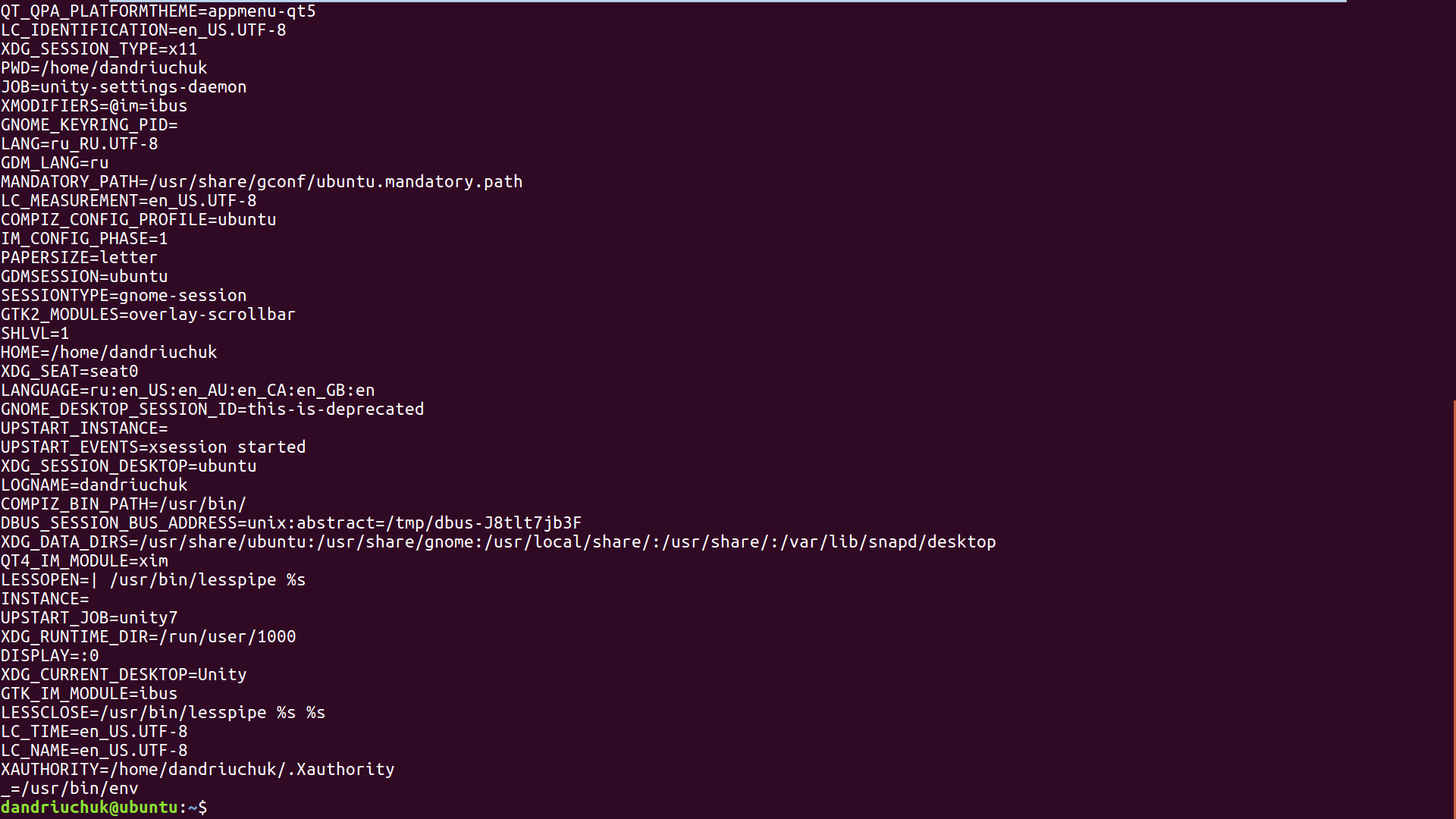
Тема: Изучение инструментов командной строки Linux.

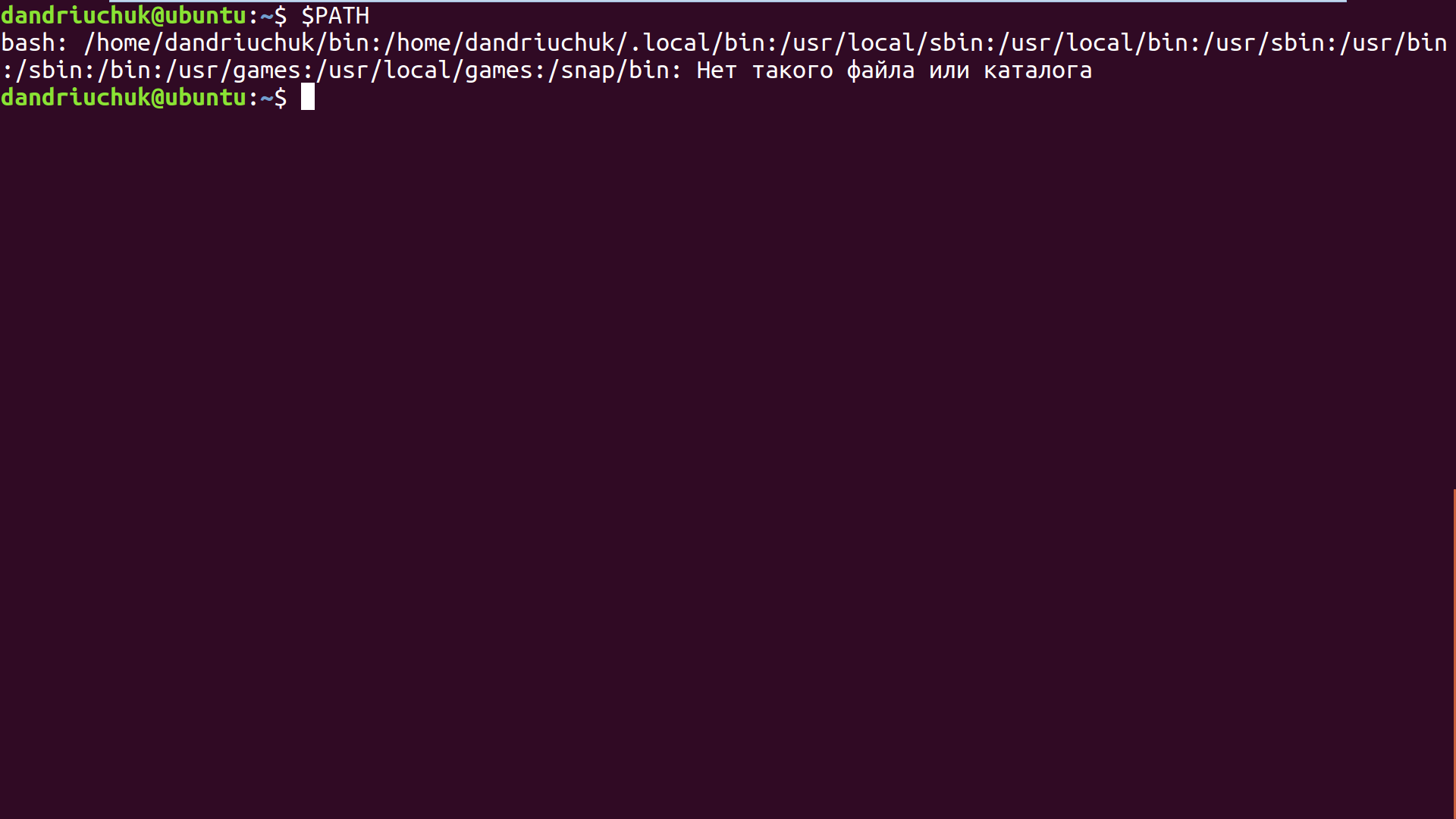
Цель: Ознакомиться с некоторыми инструментами командной строки Linux, использовать их при редактировании текстовых файлов.

Ход работы:

1. Работа в командной строке.
   1. Познакомиться и попрактиковаться с командами env и echo. Проанализировать вывод команды echo $PATH:

**env** — UNIX‐утилита, исполняющая команду с изменением окружения, выводит список переменных окружения.



$PATH – выведёт на экран значение переменной PATH. То есть список абсолютных путей к директориям в которых находятся используемые оболочкой элементы (файлы). 

* 1. Попрактиковаться с запуском команд, используя относительные и абсолютные пути:

**Абсолютный путь** — это путь, который указывает на одно и то же место в файловой системе, вне зависимости от текущей рабочей директории или других обстоятельств. Полный путь всегда начинается с корневого каталога.

Как вы уже могли заметить, все эти пути объединяет то, что они начинаются с /.



Указывая путь /usr/local/bin в качестве аргумента команде cd мы говорим ей перейти в корневую директорию /, затем в директорию usr, потом в local и bin. Абсолютные пути всегда начинаются с /

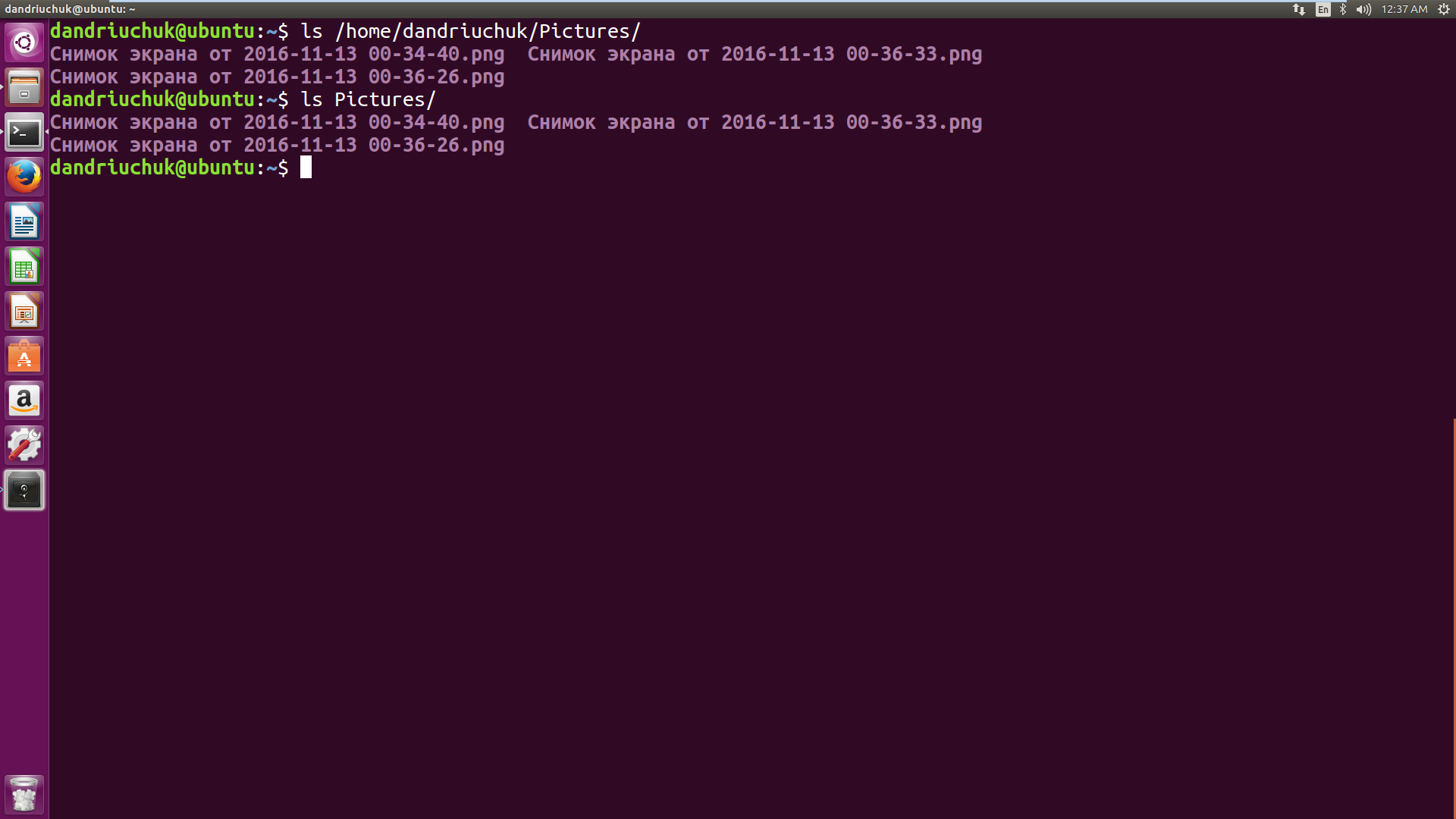
**Относительный путь** представляет собой путь по отношению к текущему рабочему каталогу пользователя или активных приложений

Команда cd и другие команды отсчитывают эти пути относительно текущей директории. Относительные пути никогда не начинаются с /.

Например, если мы находимся в /usr и хотим перейти в /usr/local/bin.



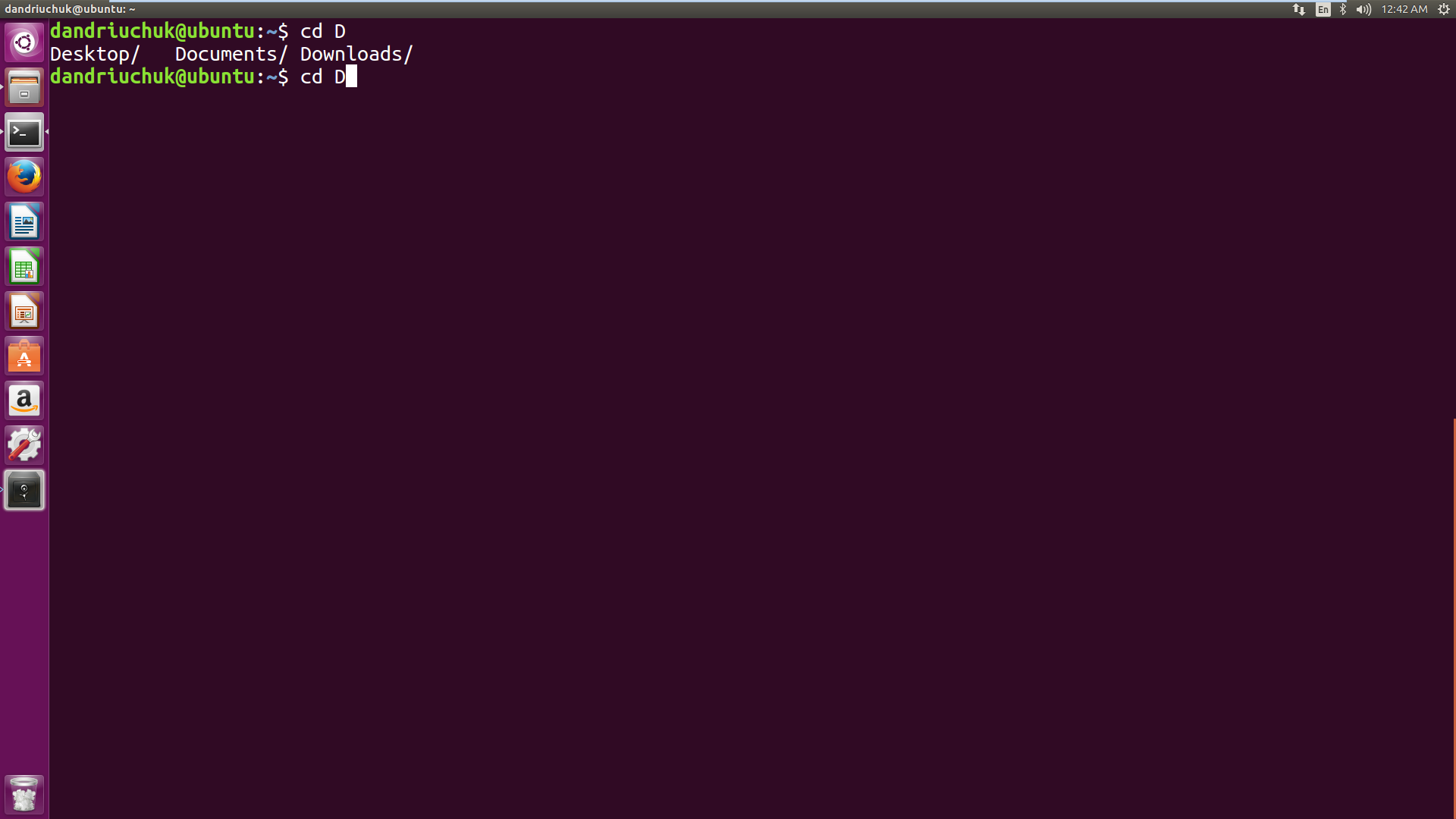
Пример: (Выведем содержимое каталога «Изображения» абсолютным и относительным путями)



* 1. Получить навыки с автозаполнением с помощью клавиши Tab

**Tab** — выполняет автозаполнение, если есть несколько вариантов автозаполнения, после повторного нажатия выведет все возможные варианты продолжения. Если файл один то он будет дописан даже без ввода её первых символов.

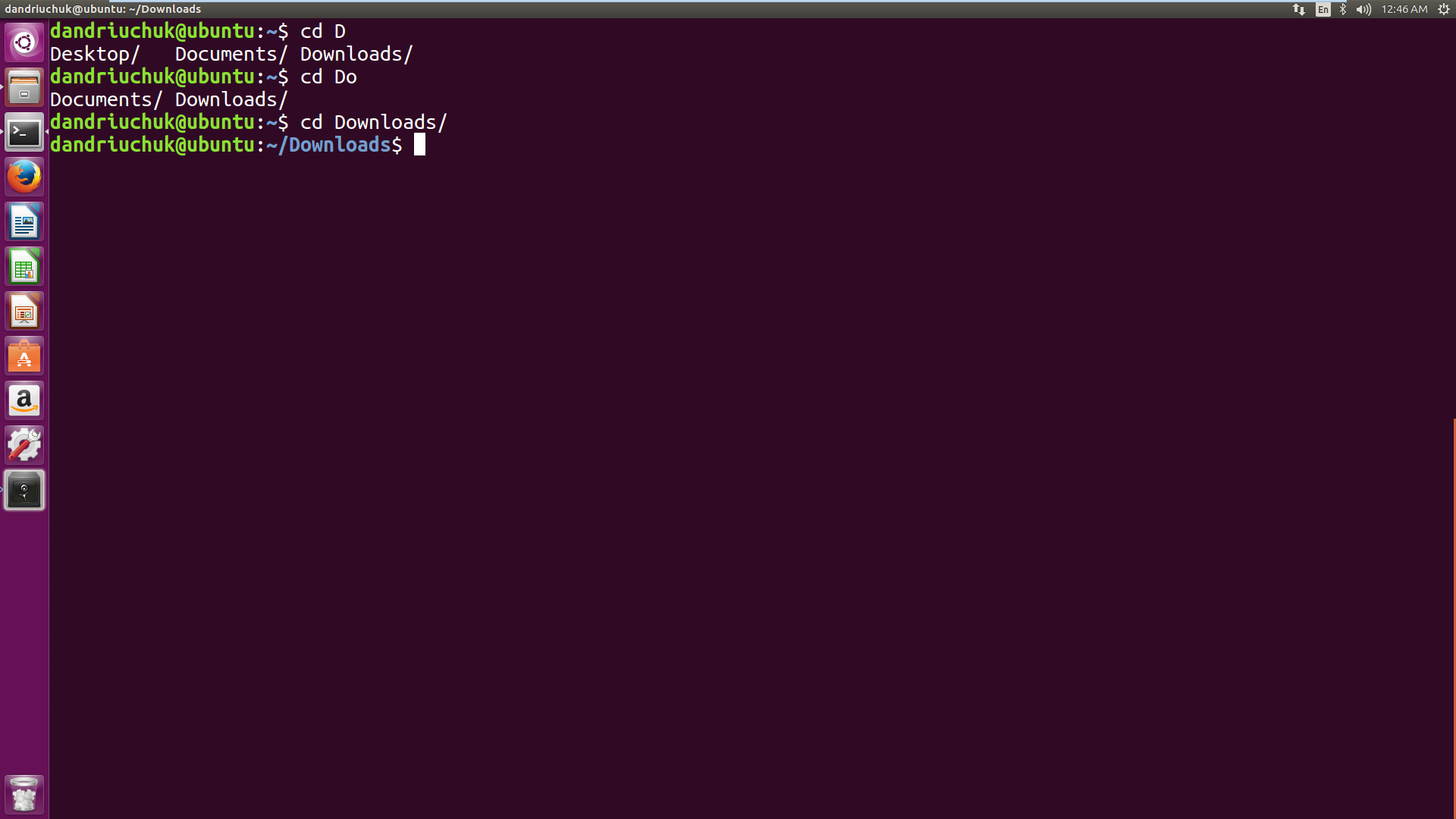
Пример: (Попробуем перейти в папку Downloads написав только её первую букву.)



Неудача. Попробуем через две буквы.



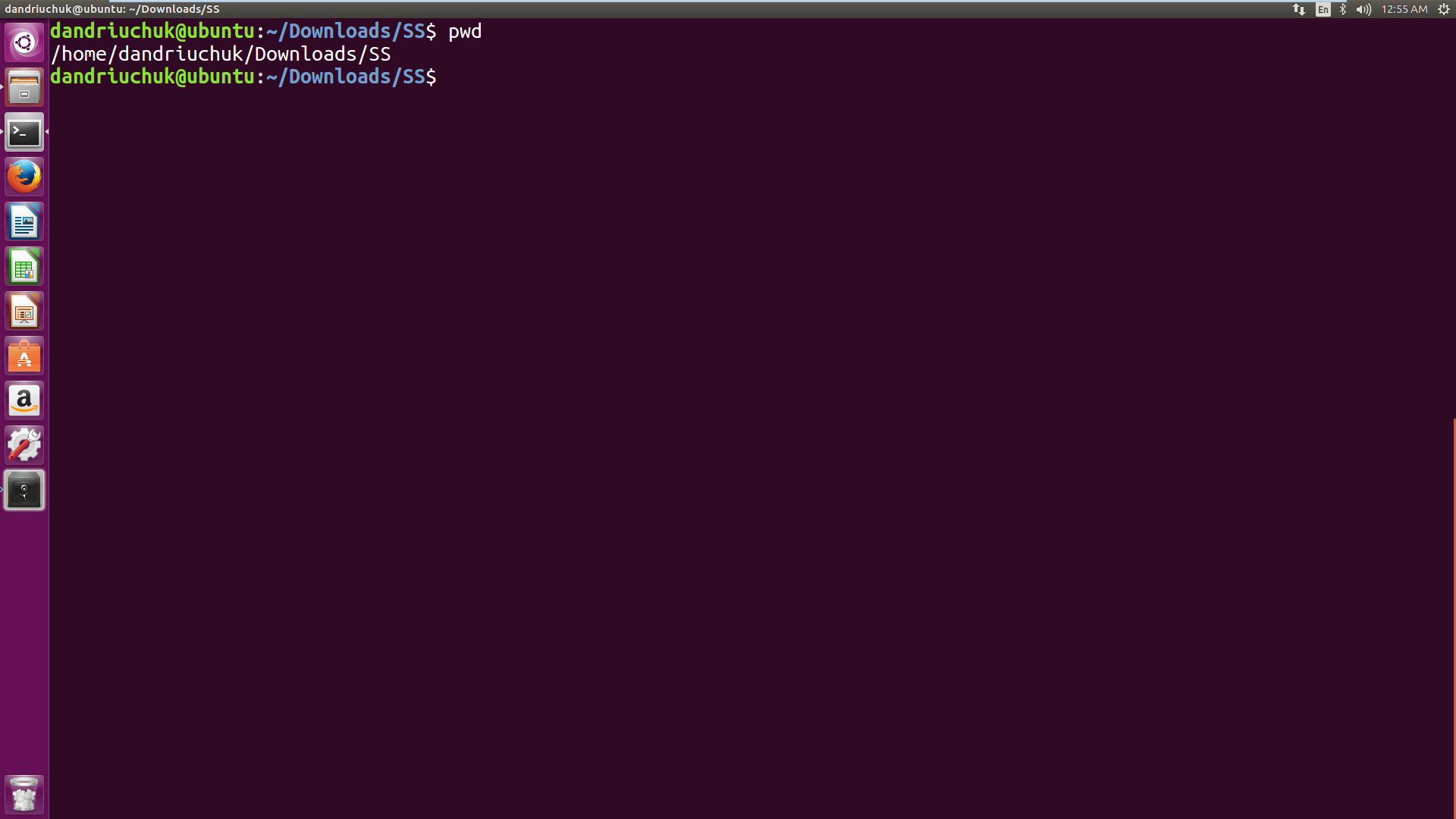
Может три.



Доказать что я не написал слово самостоятельно я не могу. Надеюсь вы мне поверите, что имя папки было дописано терминалом.

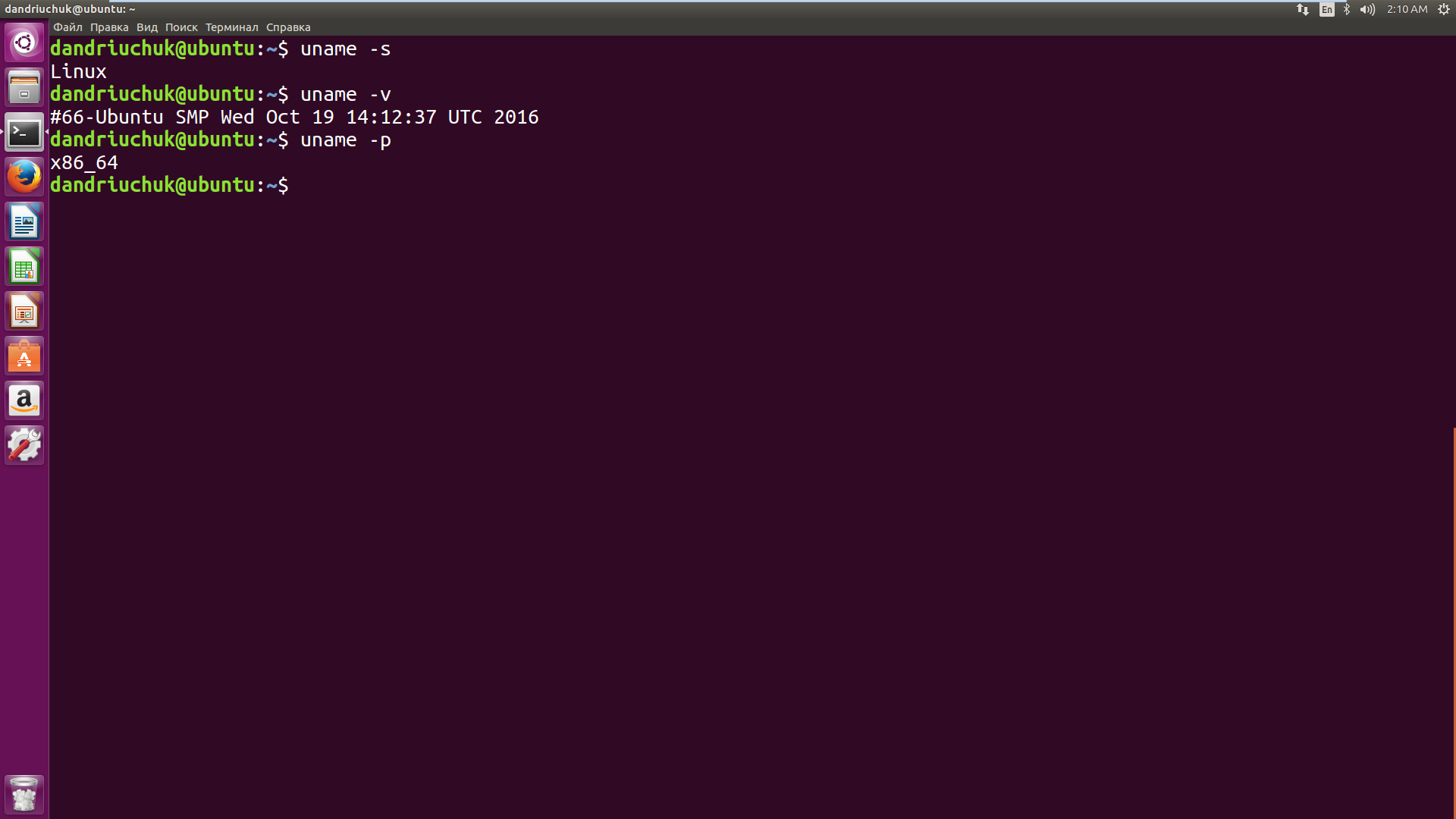
* 1. Познакомиться и попрактиковаться с командами pwd, uname

**pwd** — указывает текущее местоположение.



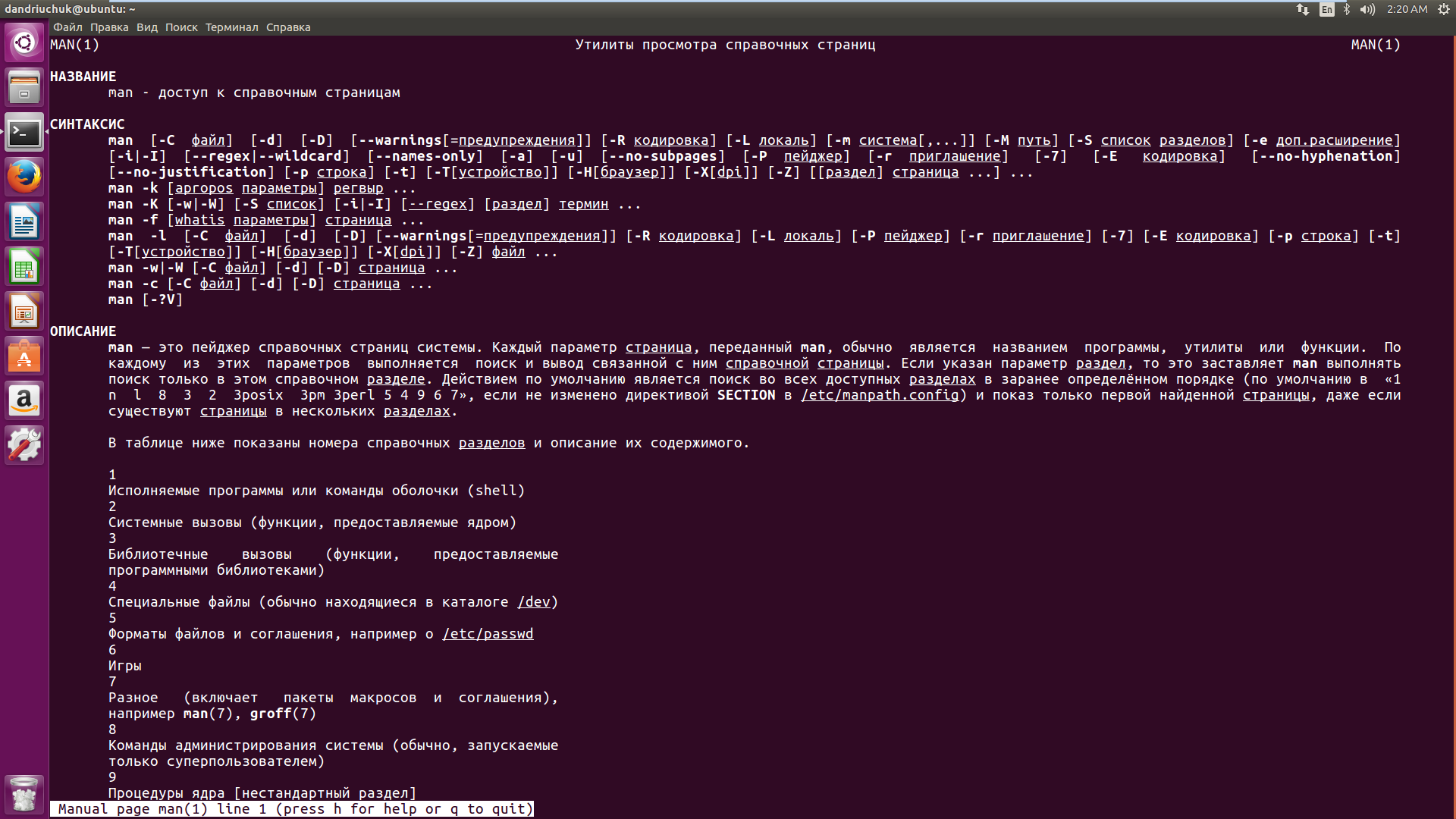
**uname -** сообщает информацию о данном компьютере и операционной системе.

Пример: (Выведем на экран название, версию ядра, тип процессора для данного компьютера)



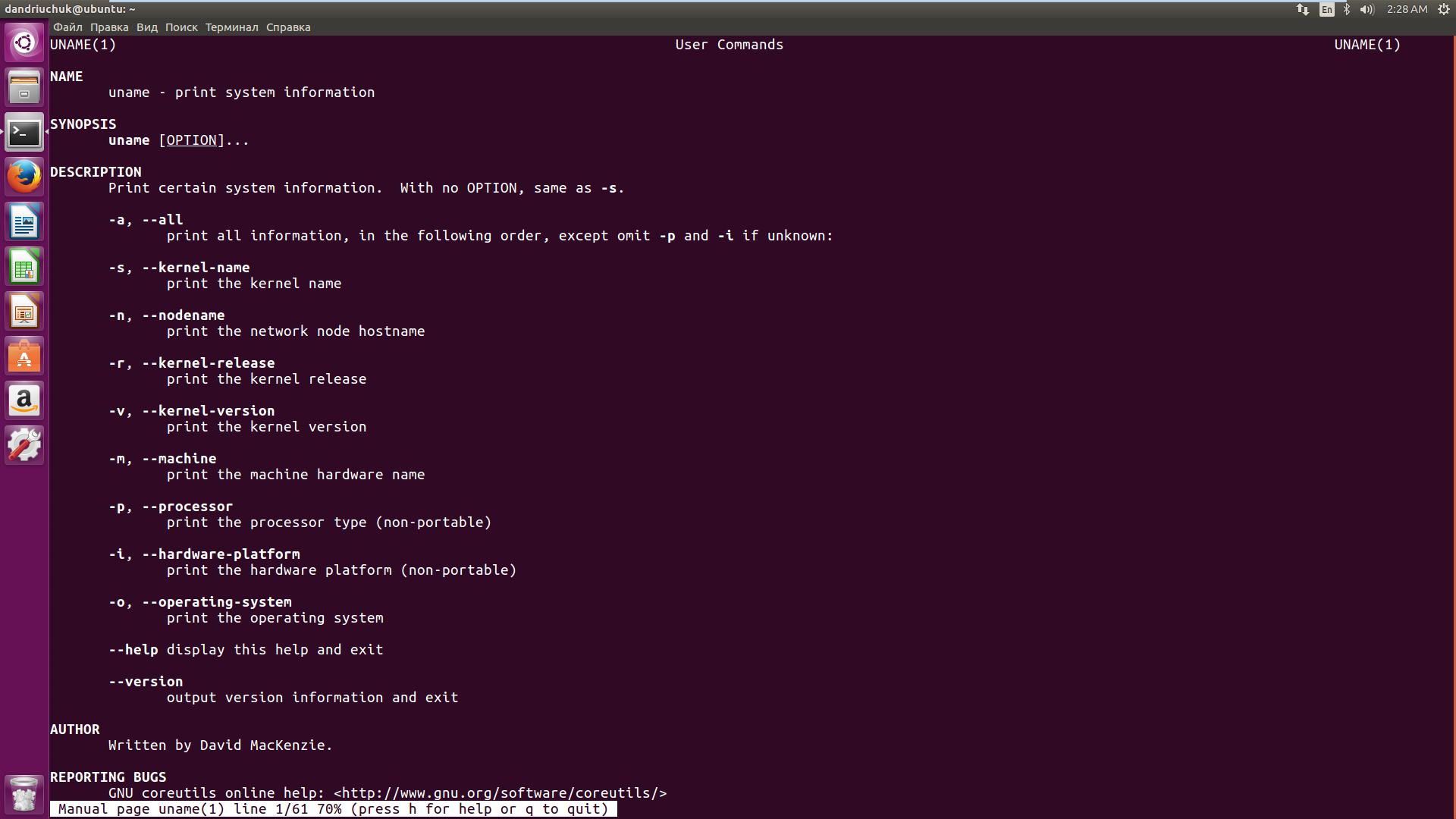
1.5 Изучите, как использовать man pages и различные секции, связанные с man pages . Попробуйте использовать различные опции команды man

**man** — (от слова manual - руководство) выводит справочное руководство по какой-либо команде.



Руководство по руководству. (Слишком большое для примера, но это первая команда которую следует выучить начинающему пользователю)

Выведем информацию об уже представленой команде uname.

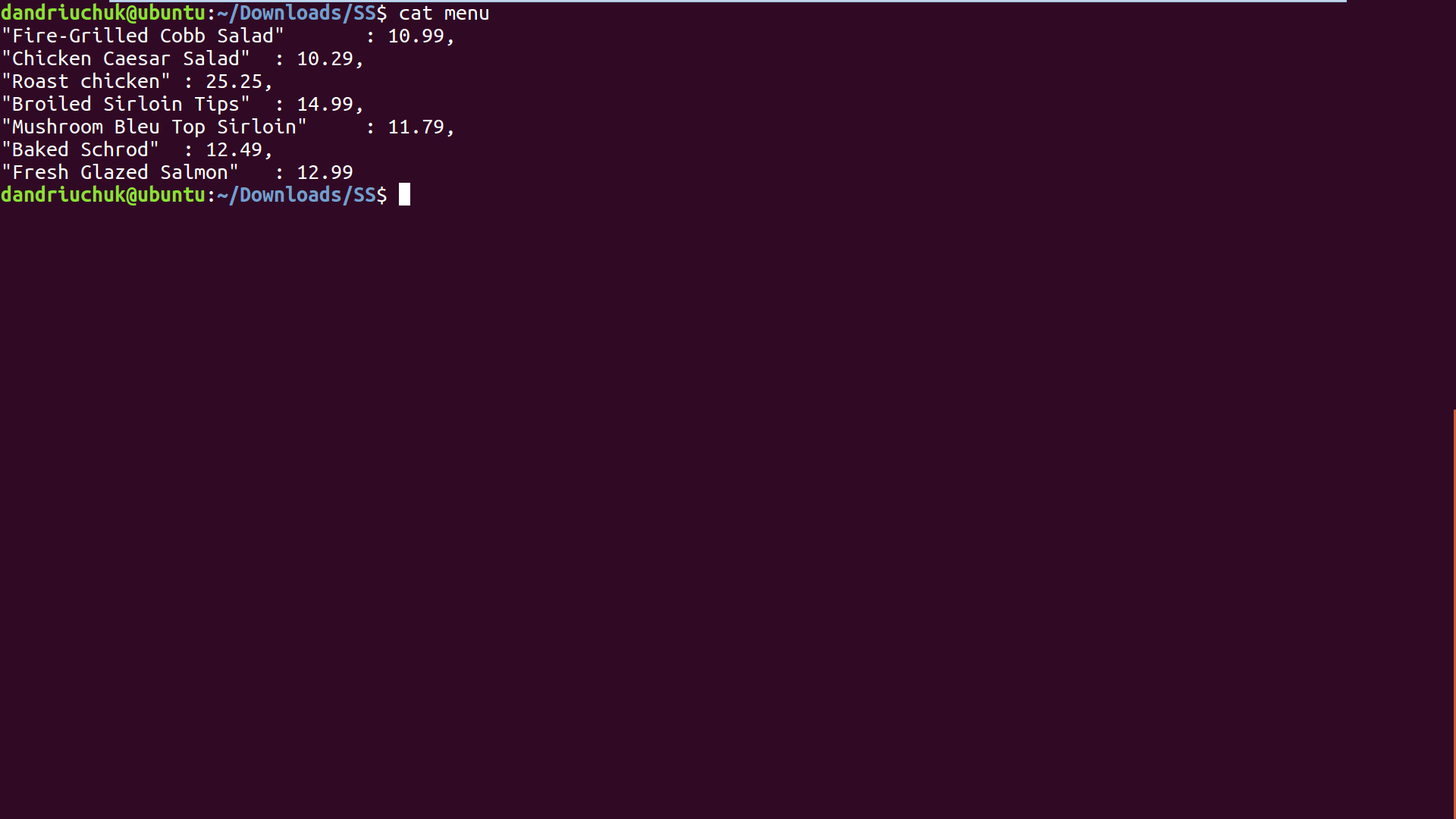


1. Обработка текстовых потоков с помощью фильтров.
   1. Познакомиться и попрактиковаться с командами cat, cut, expand, unexpand, fmt и tac.

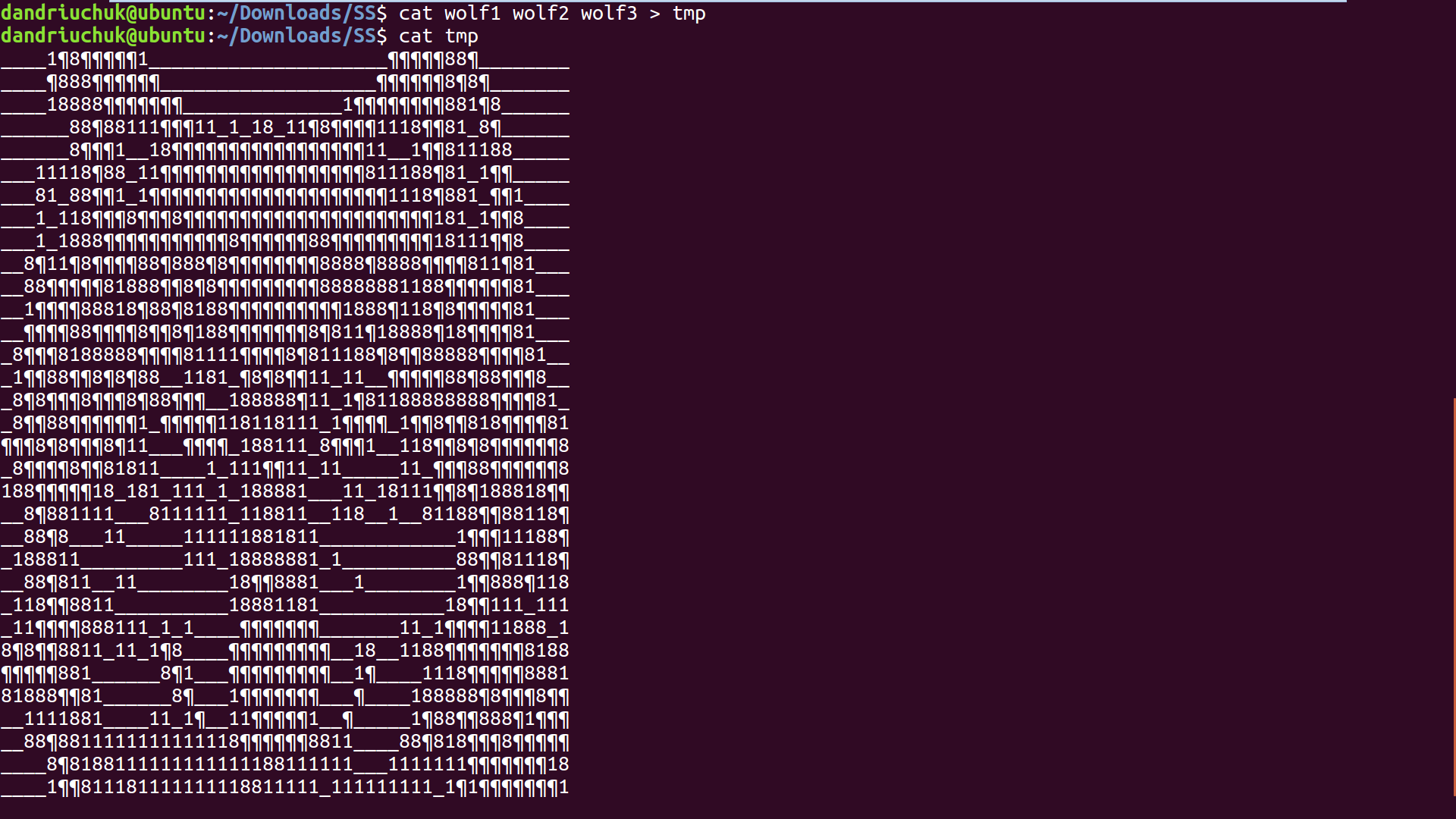
**cat** – (от слова concatenate – конкатенация, объединение) при задавании только одного файла выводит содержимое на стандартный вывод, иначе последовательно выводит содержимое файлов в заданном порядке (выглядит будто файлы были объединены).

Все последующие файлы для примеров находятся в директории Downloads/SS.

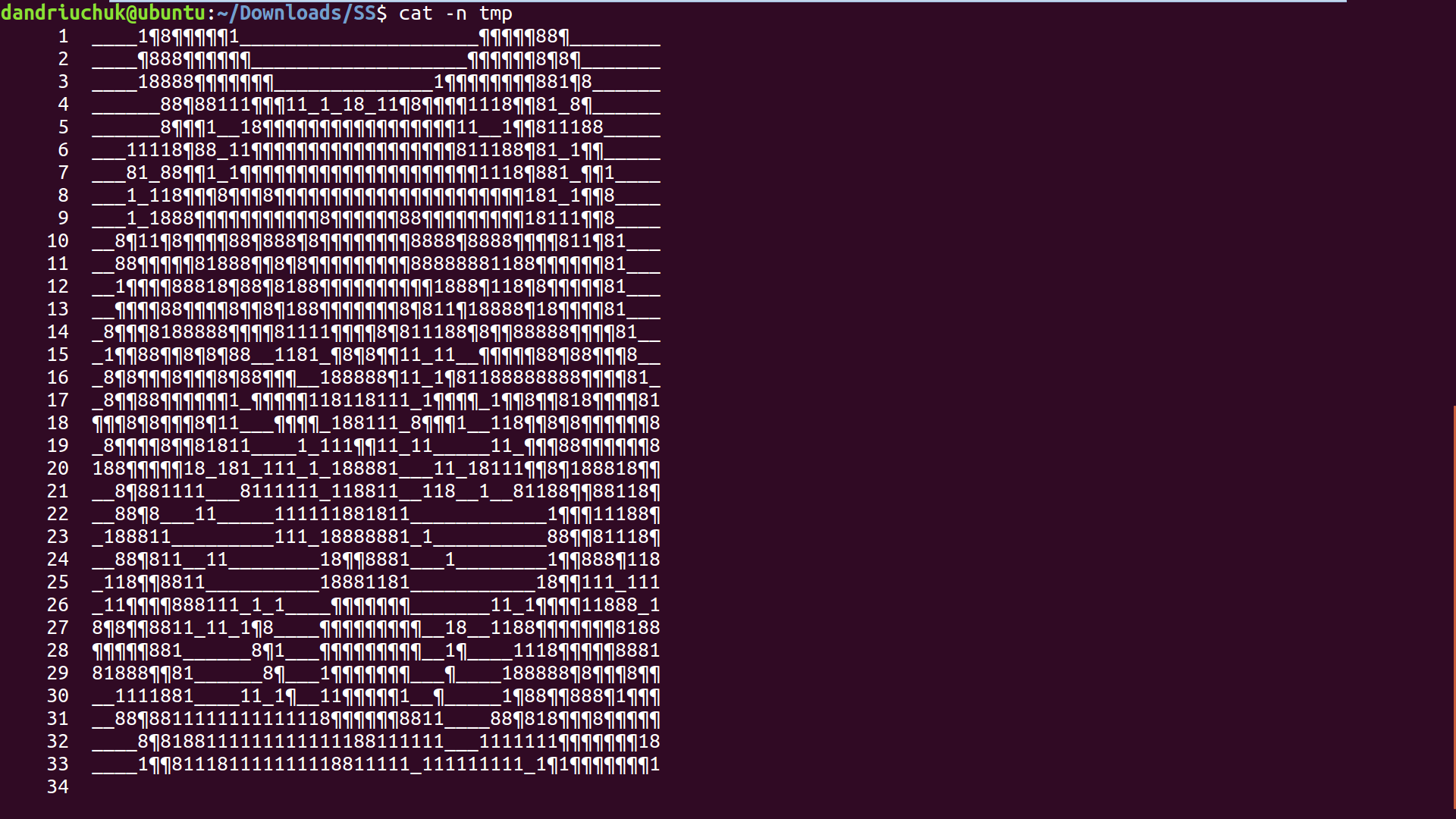
Пример.(Выведем содержимое файла menu.)



Пример(Содержимое файла wolf было порезано на 3 части: wolf1, wolf2, wolf3. Нужно их соединить не открывая сами файлы и записать в файл tmp, вывести содержимое tmp на экран)

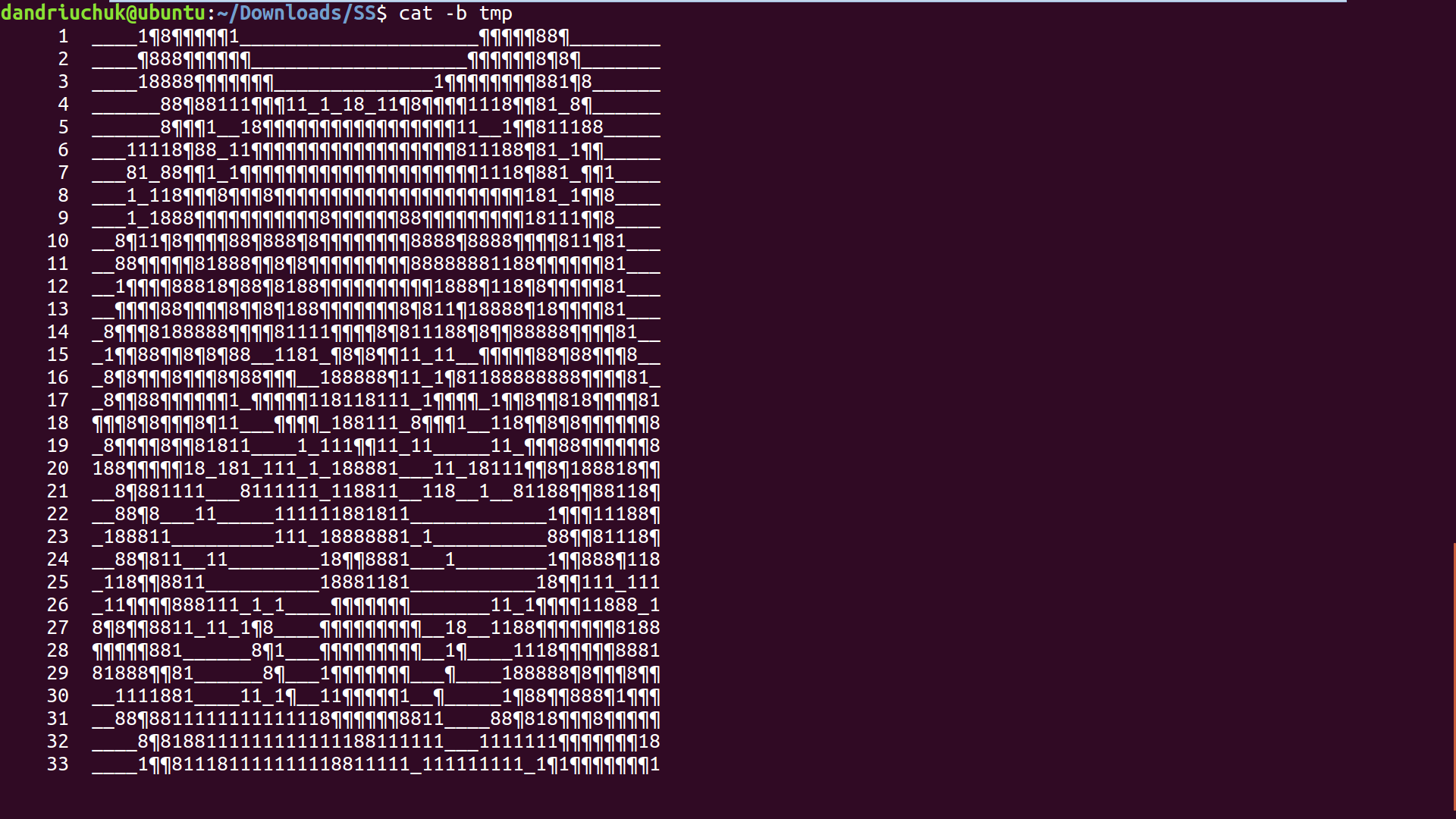


Пример(Дополнительно давайте пронумеруем строки в нашем новосозданом файле )



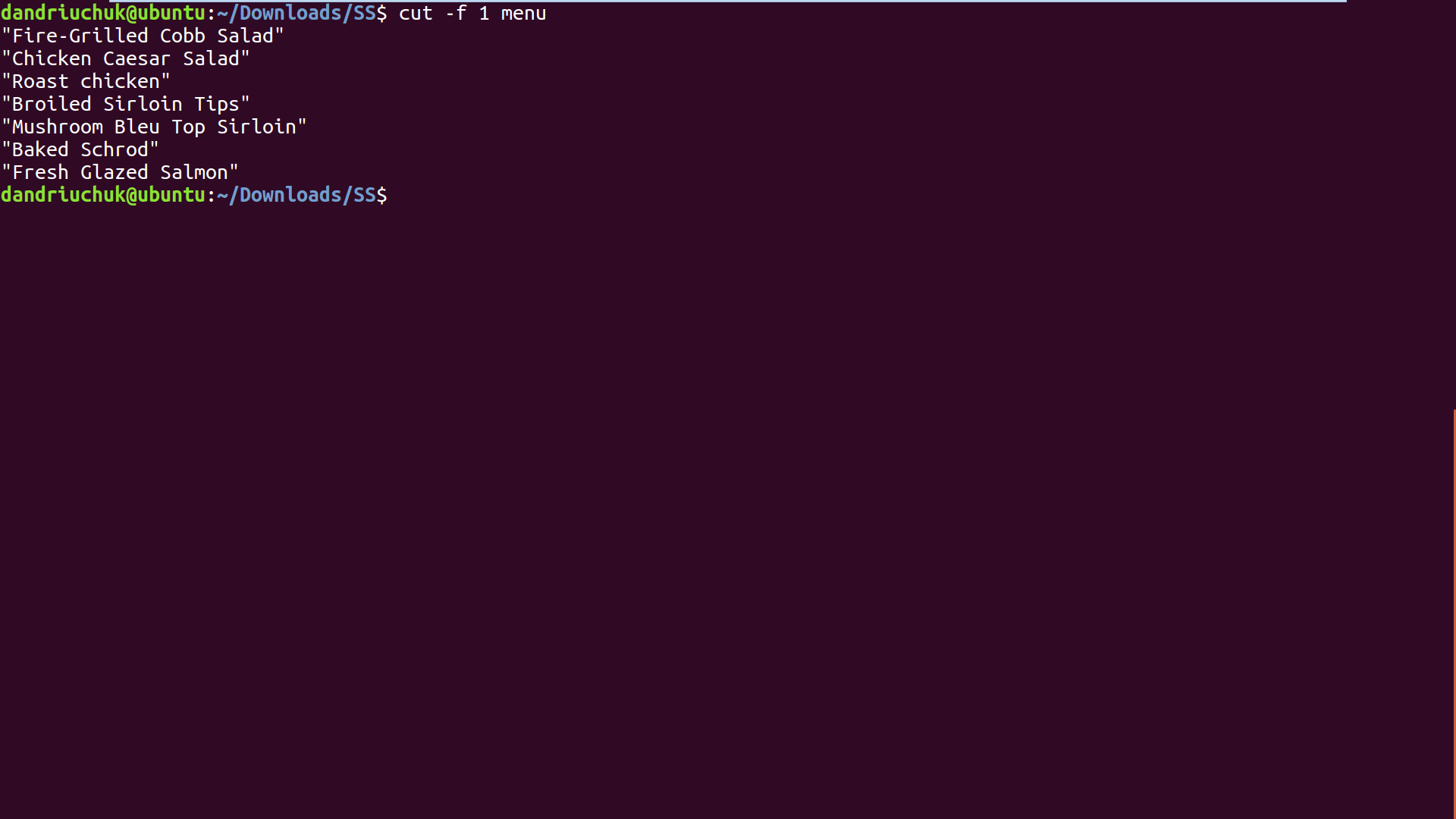
Была пронумерована даже пустая строка.

Попробуем пронумеровать непустые строки



**cut –** выводит указаные поля(байты) в кажной строке файла на стандартный поток вывода.

Пример.( Вывести блюда файла menu без их стоимости)



Пример.(Похоже содержание файла lab не соответствует истине. Удалите 58,59 символы)

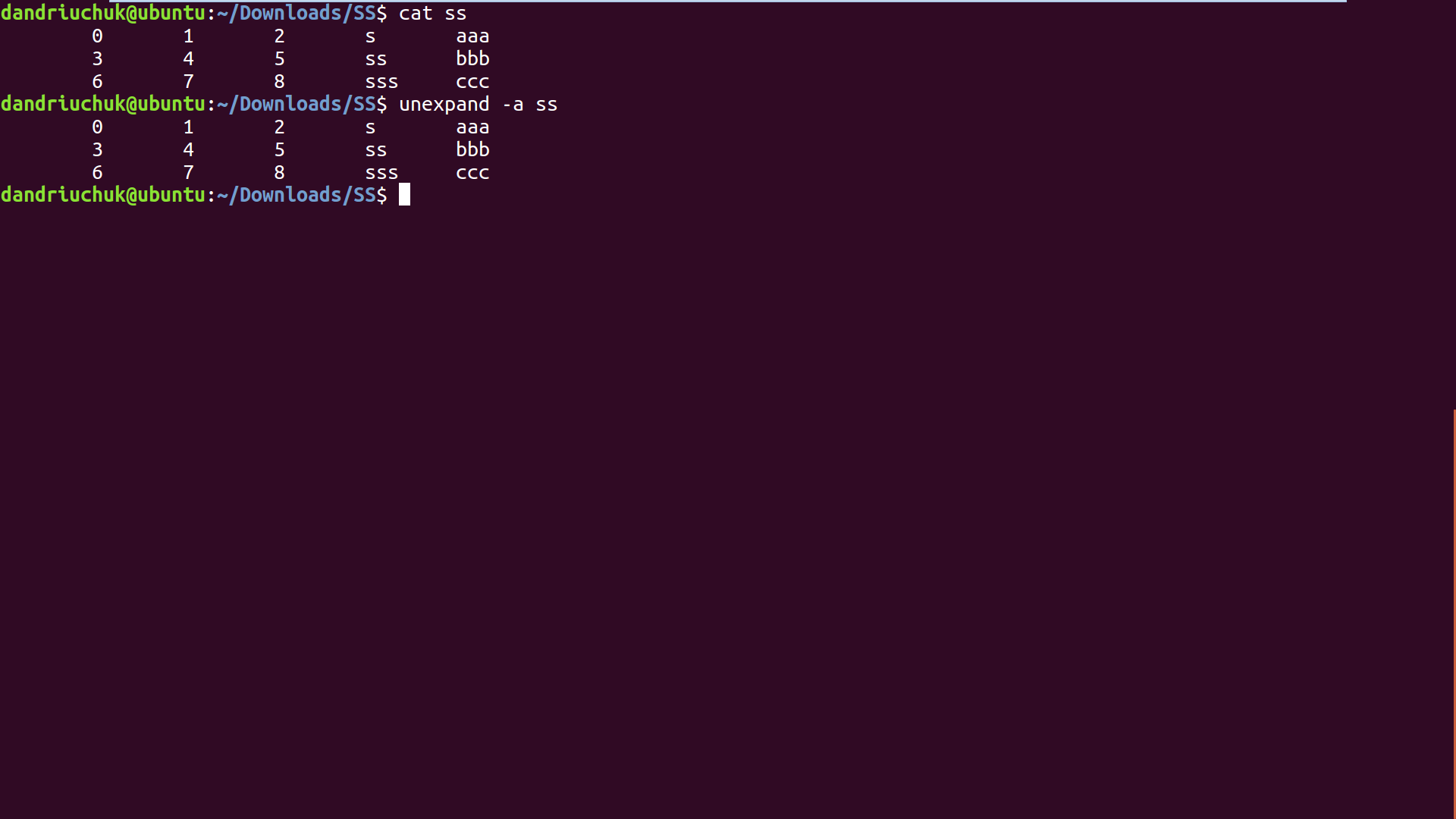


**expand**, **unexpand** — утилити преобразовывающие табуляции в пробелы и пробелы в табуляции соответственно.

Пример(файл Kiev неудобно читать. Замените все табуляции на единичные пробелы и запишите в файл Киев)

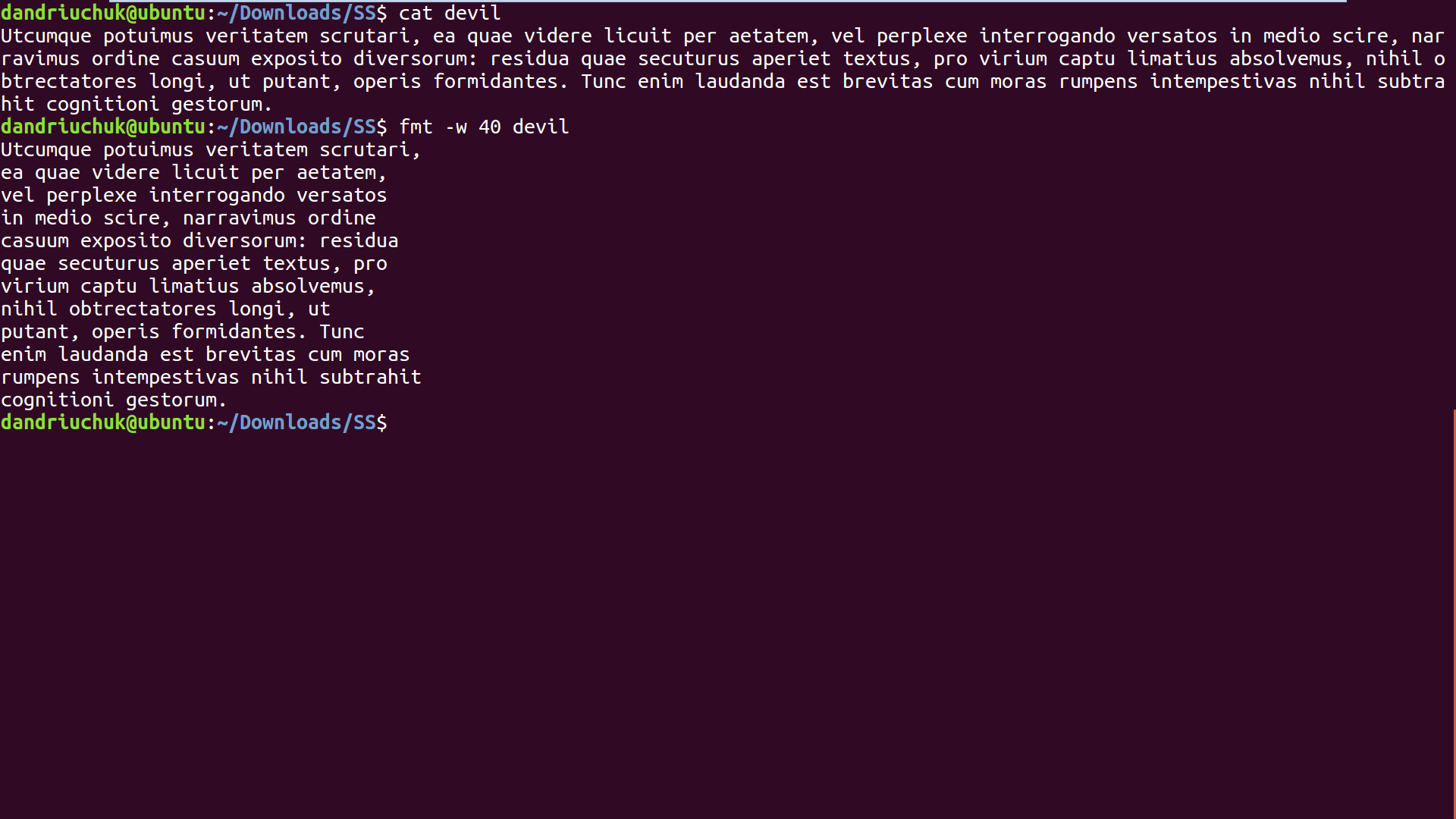


Пример(превратите пробелы между символами в файле ss на табы)



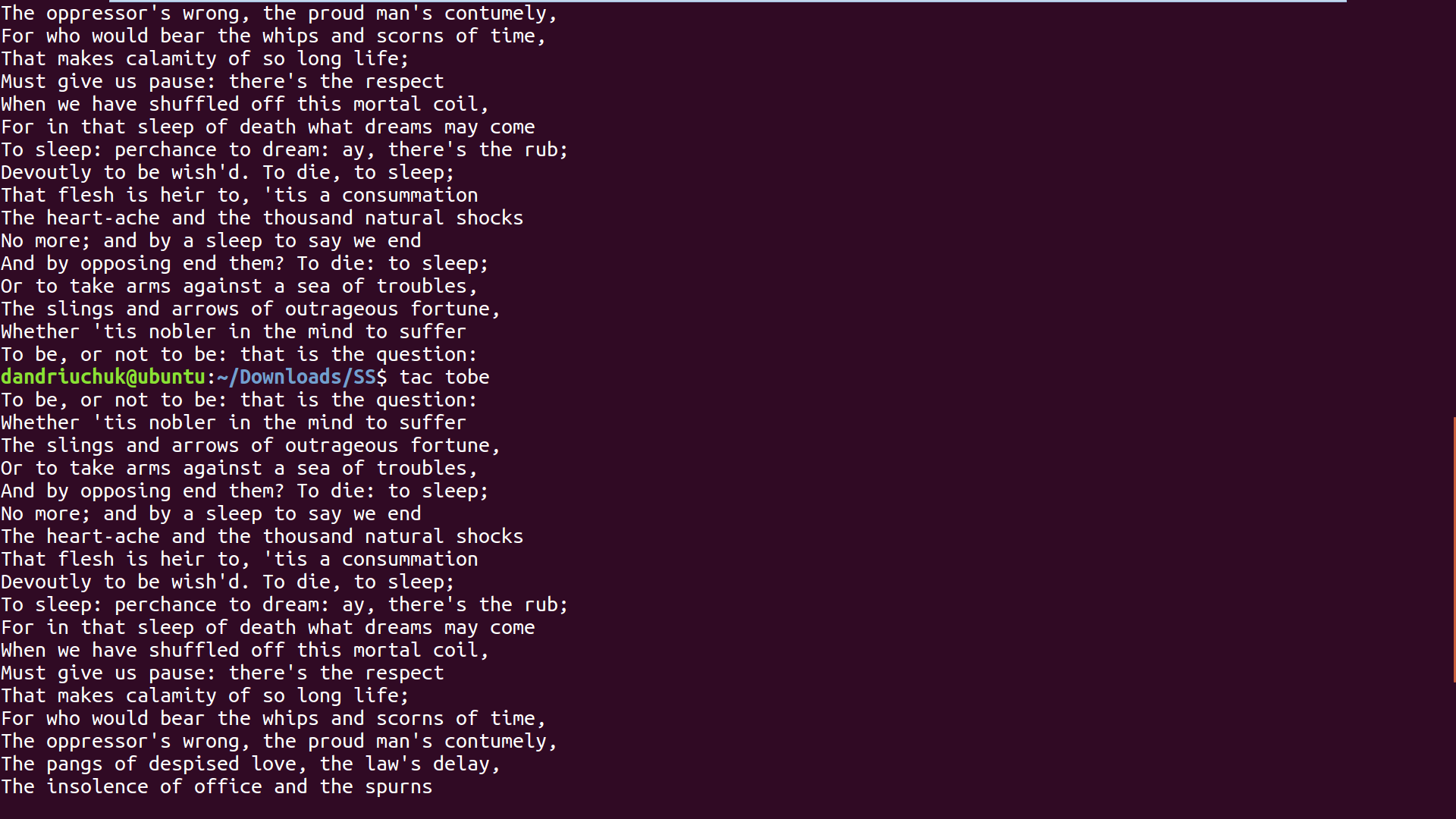
**fmt** — форматирует каждый абзац в файле и выводит на стандартный вывод.

Пример (Файл devil написан всего в несколько строк, из-за чего некоторые слова обрезаются терминалом, придайте ему вид стиха, ширина строки 40)



**tac –** делает тоже что и cat но выводит строки в обратном прядке.

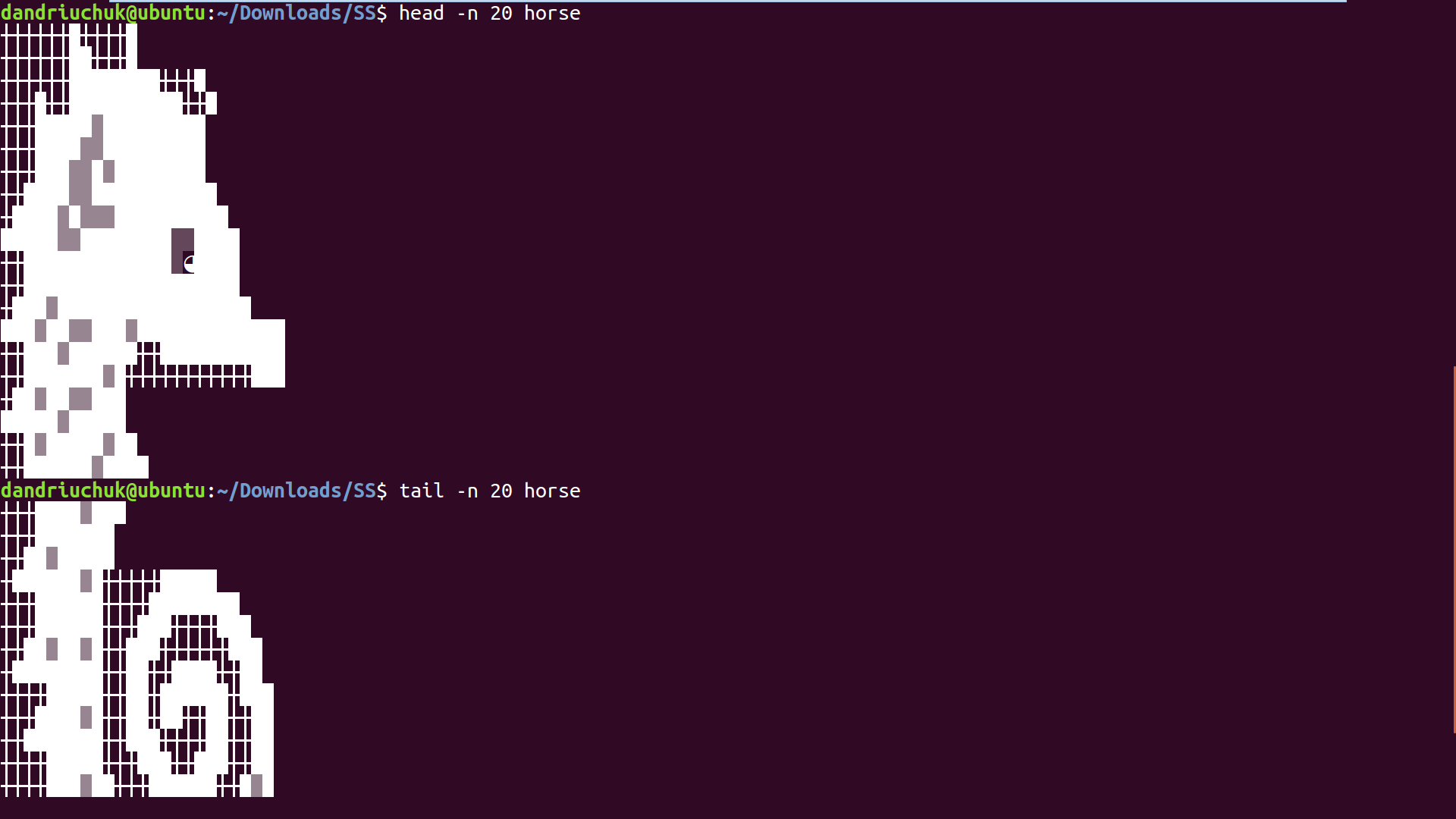
Пример(Обрадуйте Шекспира, верните перевёрнутый монолог Гамлета (файл tobe) в нормальное состояние)



* 1. Познакомиться и попрактиковаться с командами head, tail, od, join, paste и pr

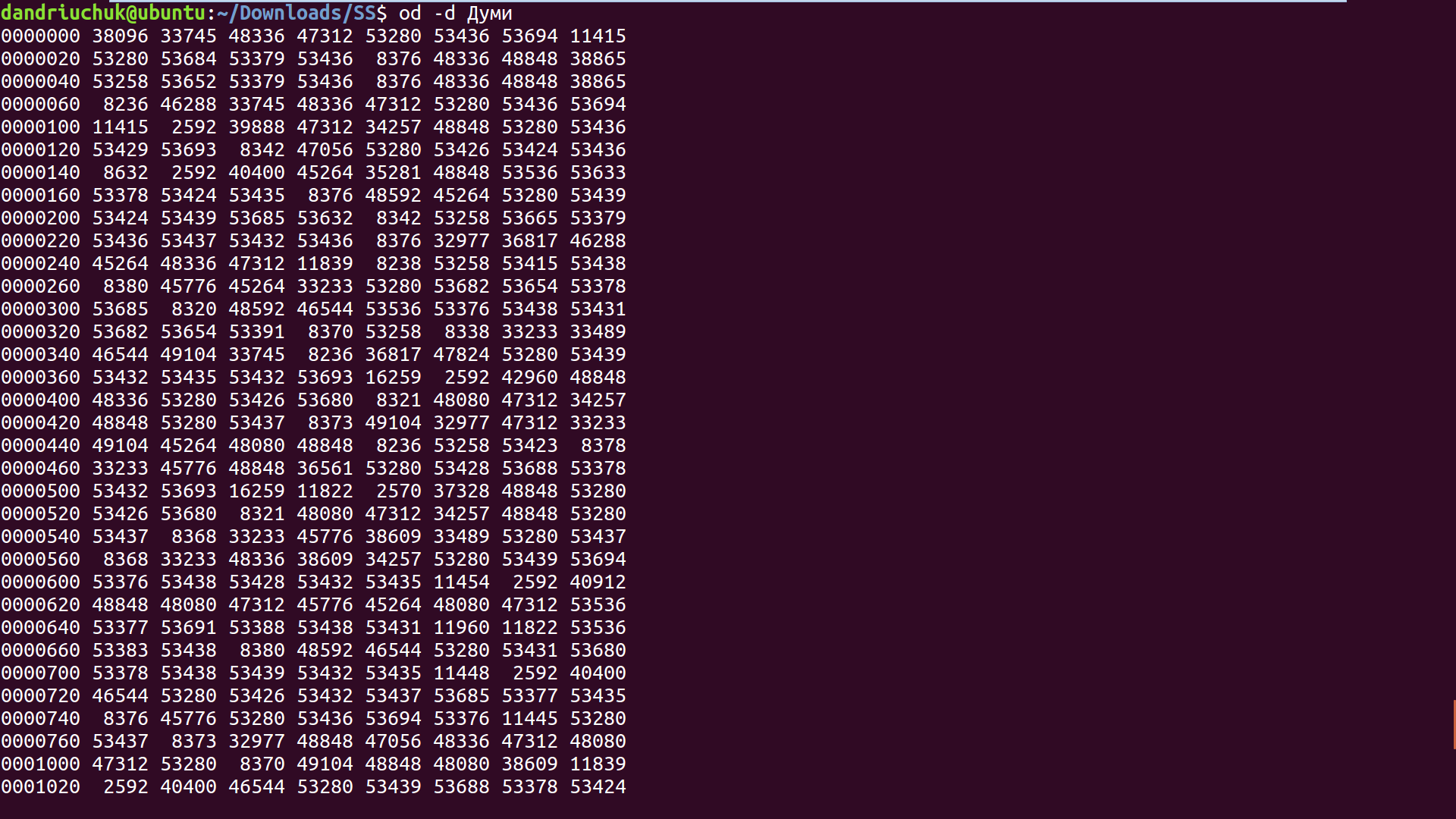
**head, (tail)** — выводит первые (последние) 10 строк по умолчанию, с помощью ключа –n можно изменять количество строк на вывод .

Пример: (Выведем первые и последние 20 строк в файле horse)



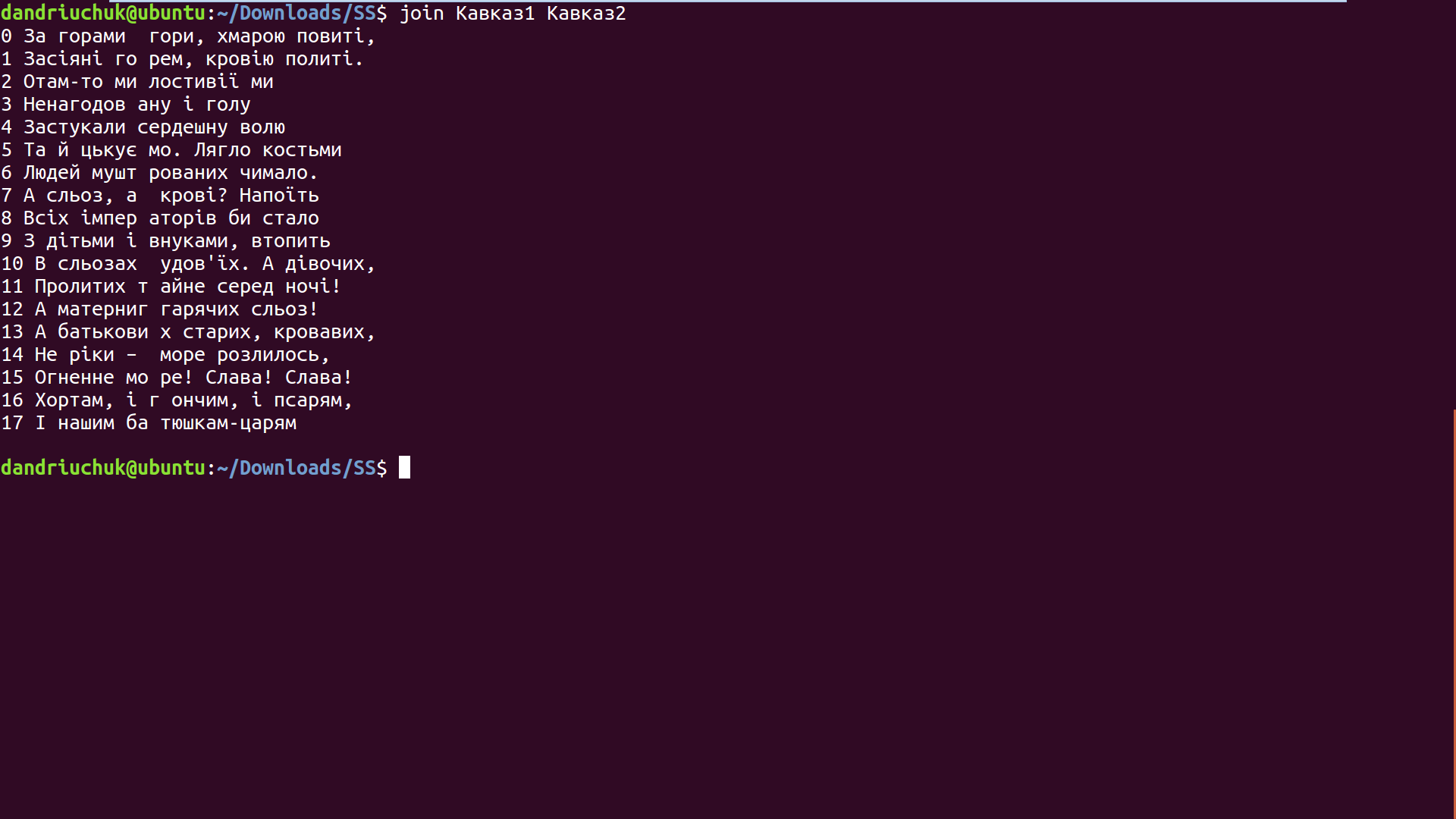
**оd** — вывод различных форм(форматов) представления файла.

Пример: (Выведем данные в файле “Думи” в виде беззнаковых десятичных двухбайтовых единиц)



**join** — для каждой пары входных строк с одинаковыми общими полями выводит строку на стандартный вывод. По умолчанию общее поле считается первым, поля разделяются пробелами.

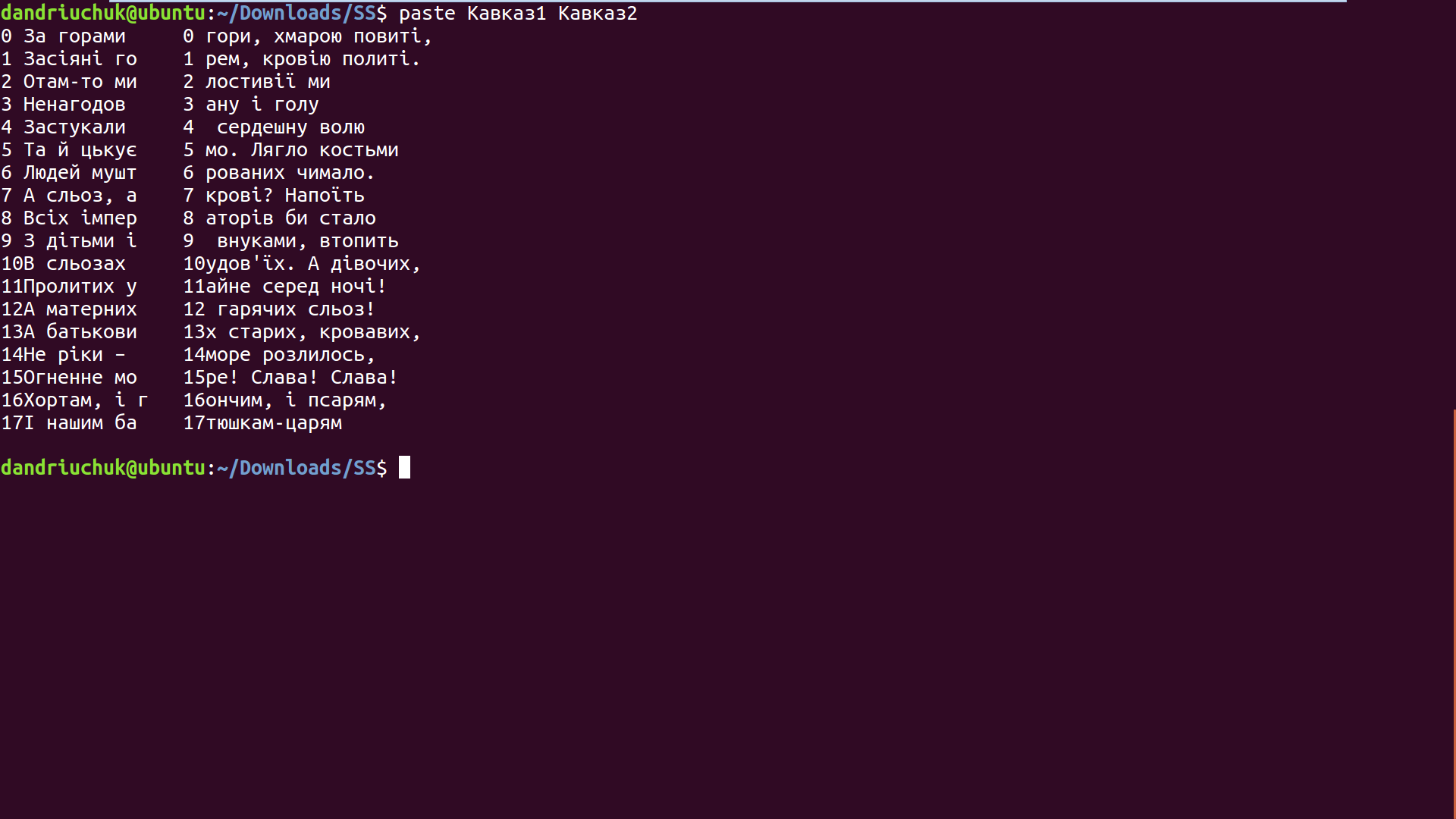
Пример(Обединим файлы Кавказ1 и Кавказ2)



На месте деления стоит пробел.

**paste** — объединение строк файлов, выводит в виде колонок

Пример(Обединим файлы Кавказ1 и Кавказ2)



**pr** — разбивает ФАЙЛ(ы) на страницы или колонки для печати.

Пример(на файле Wolf)

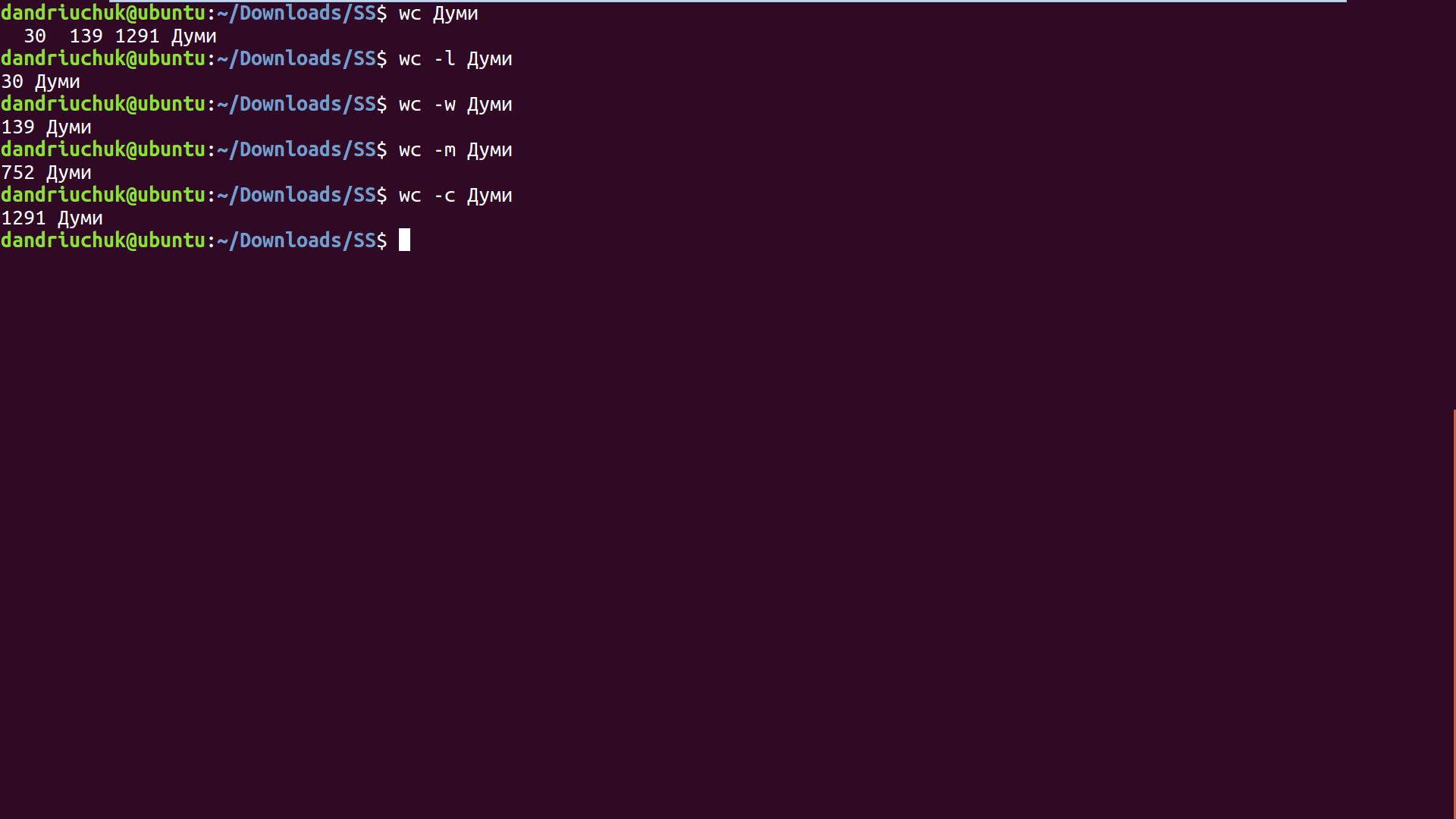


**wc -** Печатает число символов новой строки, слов и байт для каждого ФАЙЛА и

итоговую строку, если было задано несколько ФАЙЛОВ. Словом считается

последовательность символов ненулевой длины, отделённая пробельным символом.

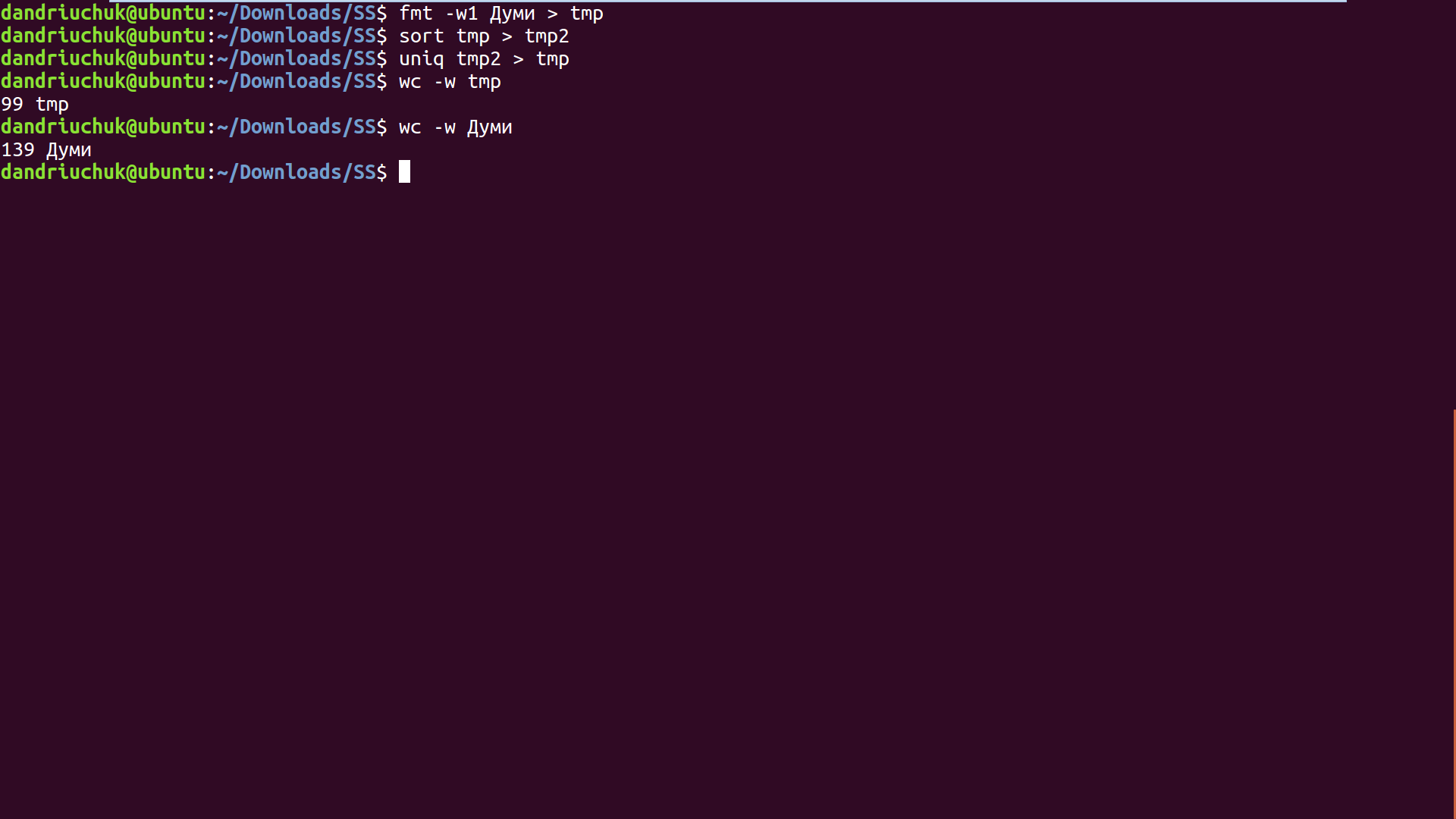
Пример( Используя wc вывести отдельно количество строк, количество слов, количество букв, количество байт



**sort** — сортирует данные в файле. (По умолчанию в алфавитном порядке или по первому числу)

**uniq** — утилита которая позволяет отфильтровать одинаковые строки, при условии что файл отсортирован.

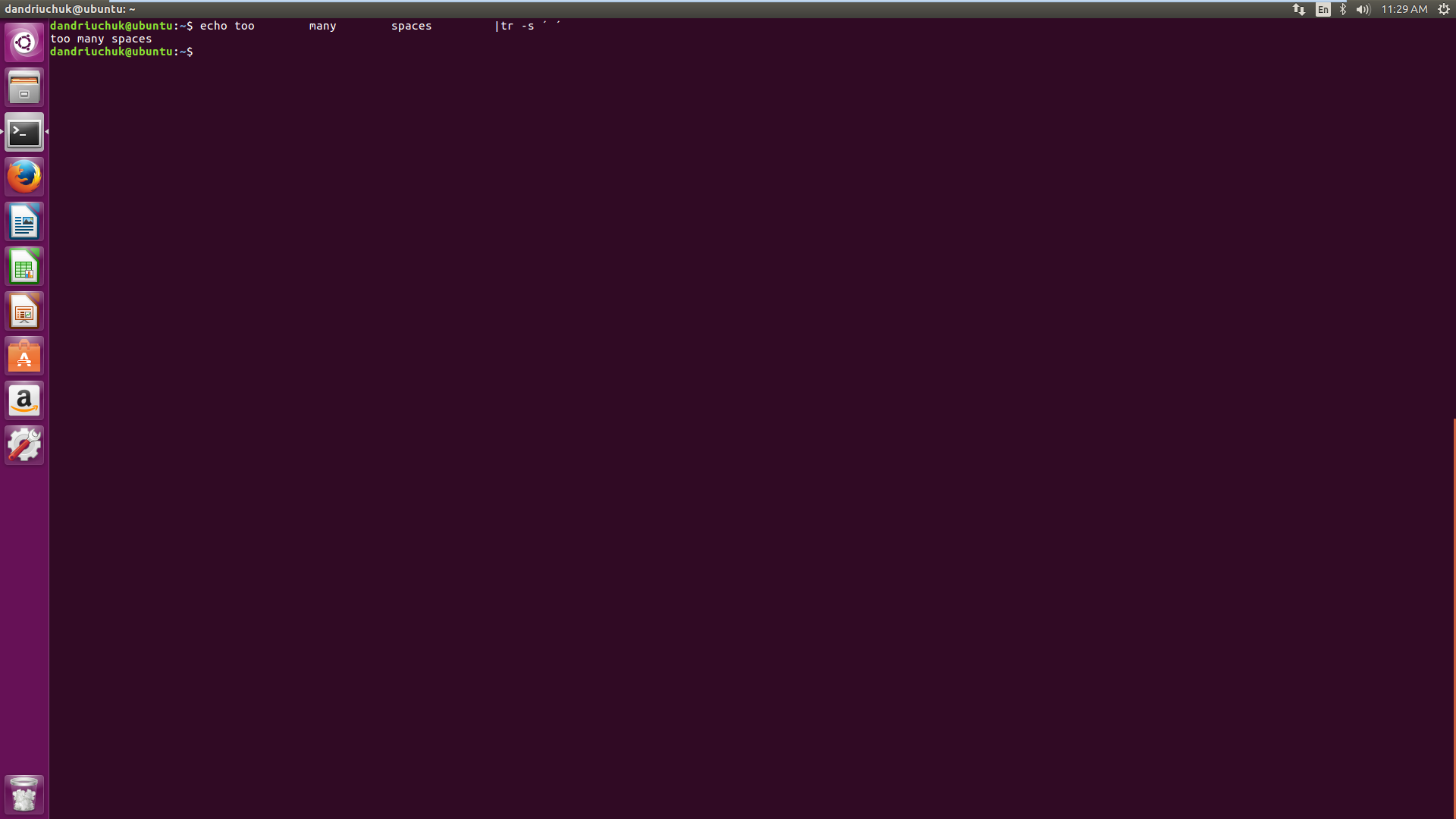
Пример( Найти сколько слов в файле Думы не повторяются, для этого отформатируем текст что б в строке было только одно слово, отсортируем полученный вывод, найдем уникальные строки-слова, подщитаем количество слов утилитой wc в файле)



**tr** — преобразует, уплотняет и/или удаляет символы со стандартного ввода и

печатает на стандартный вывод.

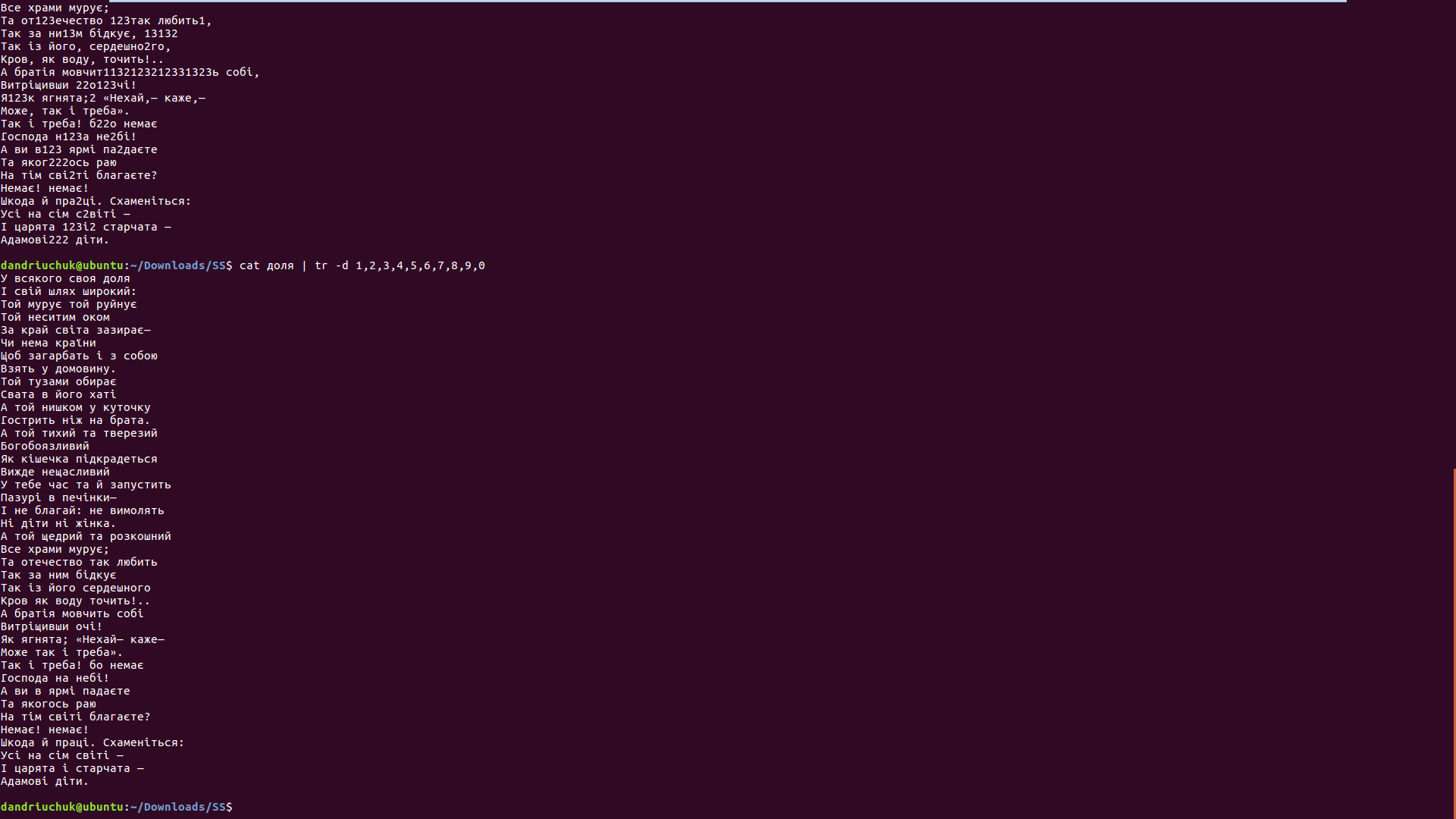
Пример(удалим часто повторяющиеся пробелы.)



Заменим о на 0

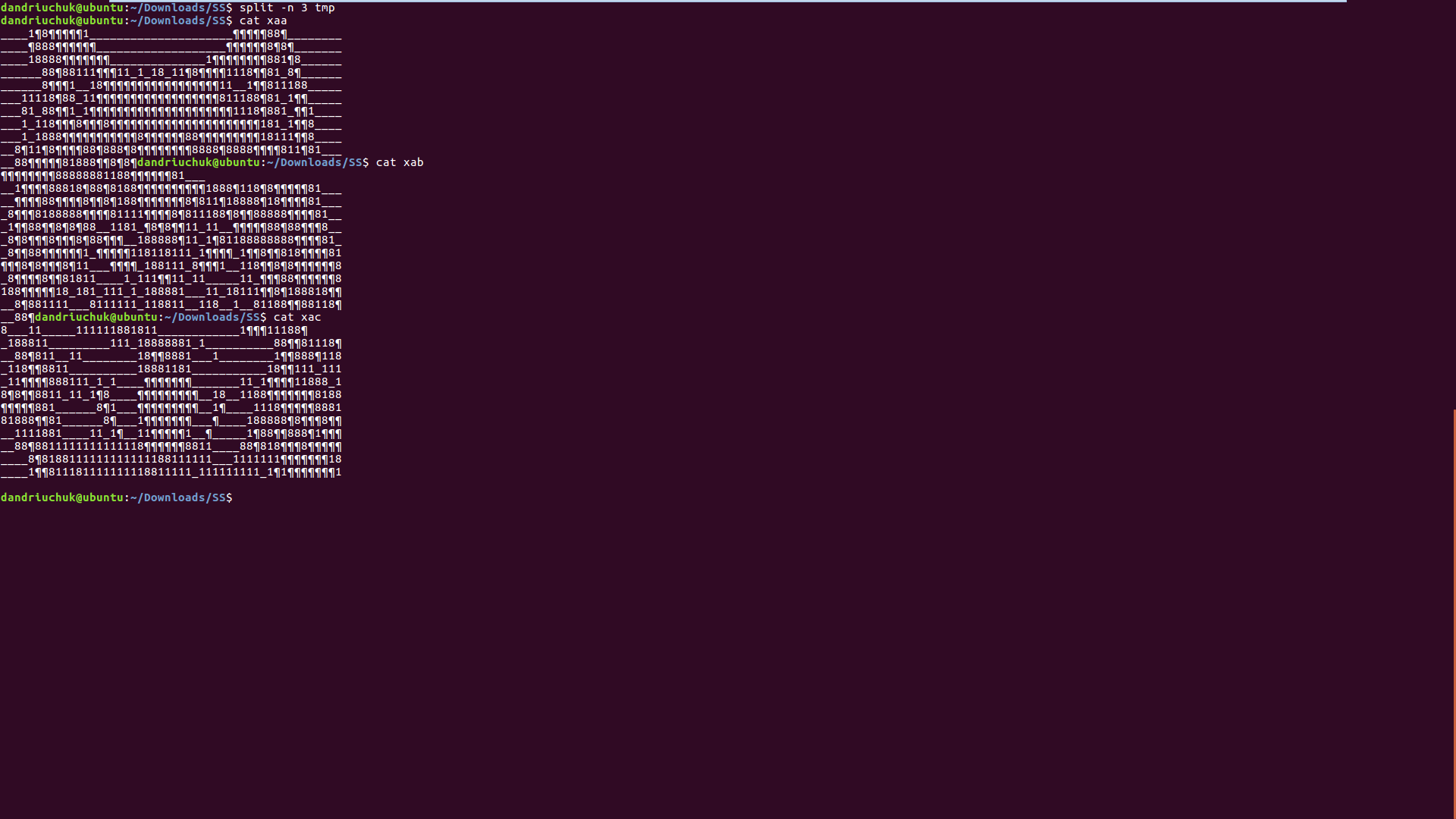


Пример(удалим все цыфры)



**split** — разбитие файла на меньшие, в каждом файле задается указаное количество символов, или строк, при этом исходный файл остается/

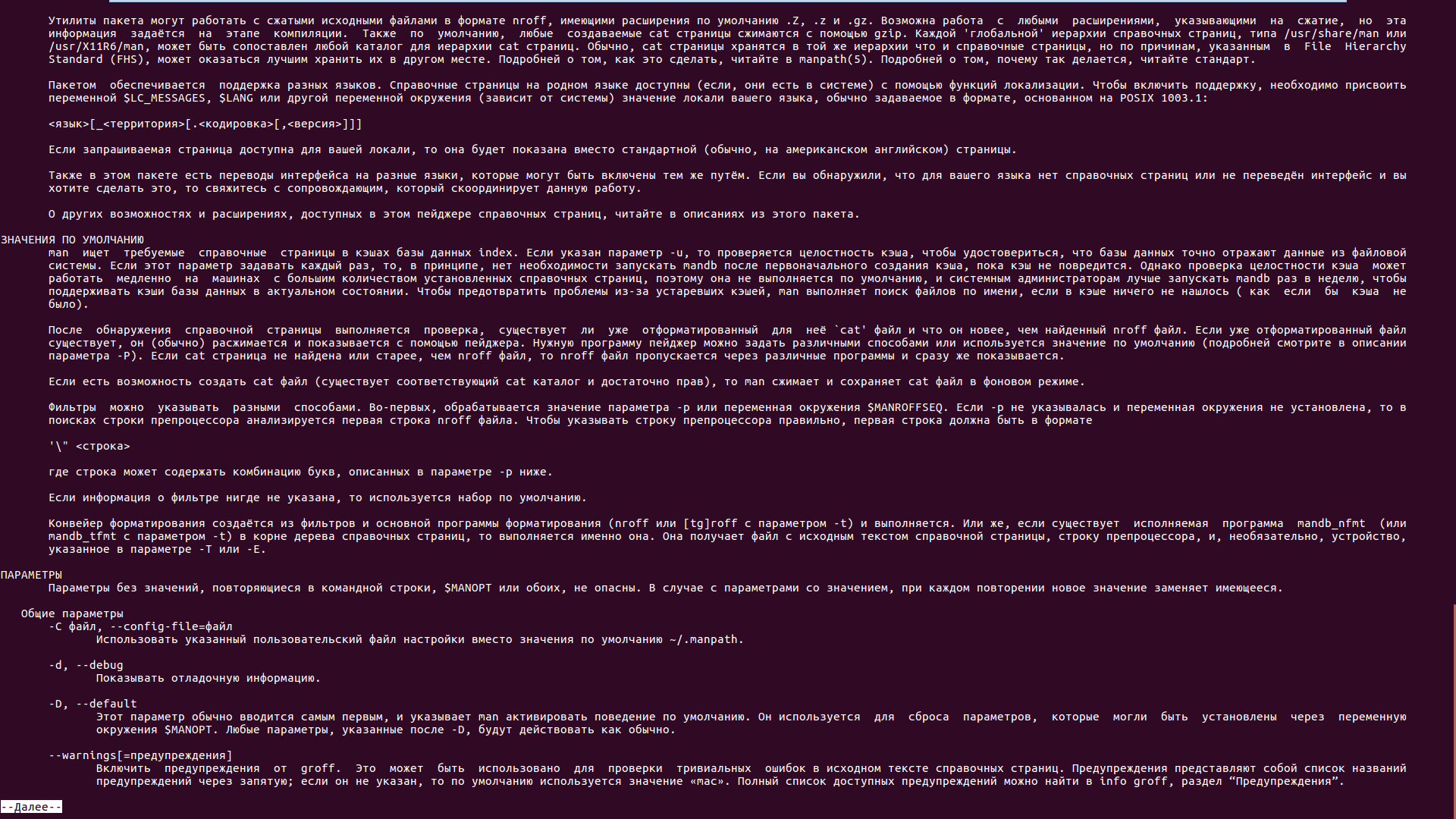
Пример(Разделим tmp на 3 части)



**more** — команда постраничной или построчной прокрутки большого текста на экране и поиска в этом тексте

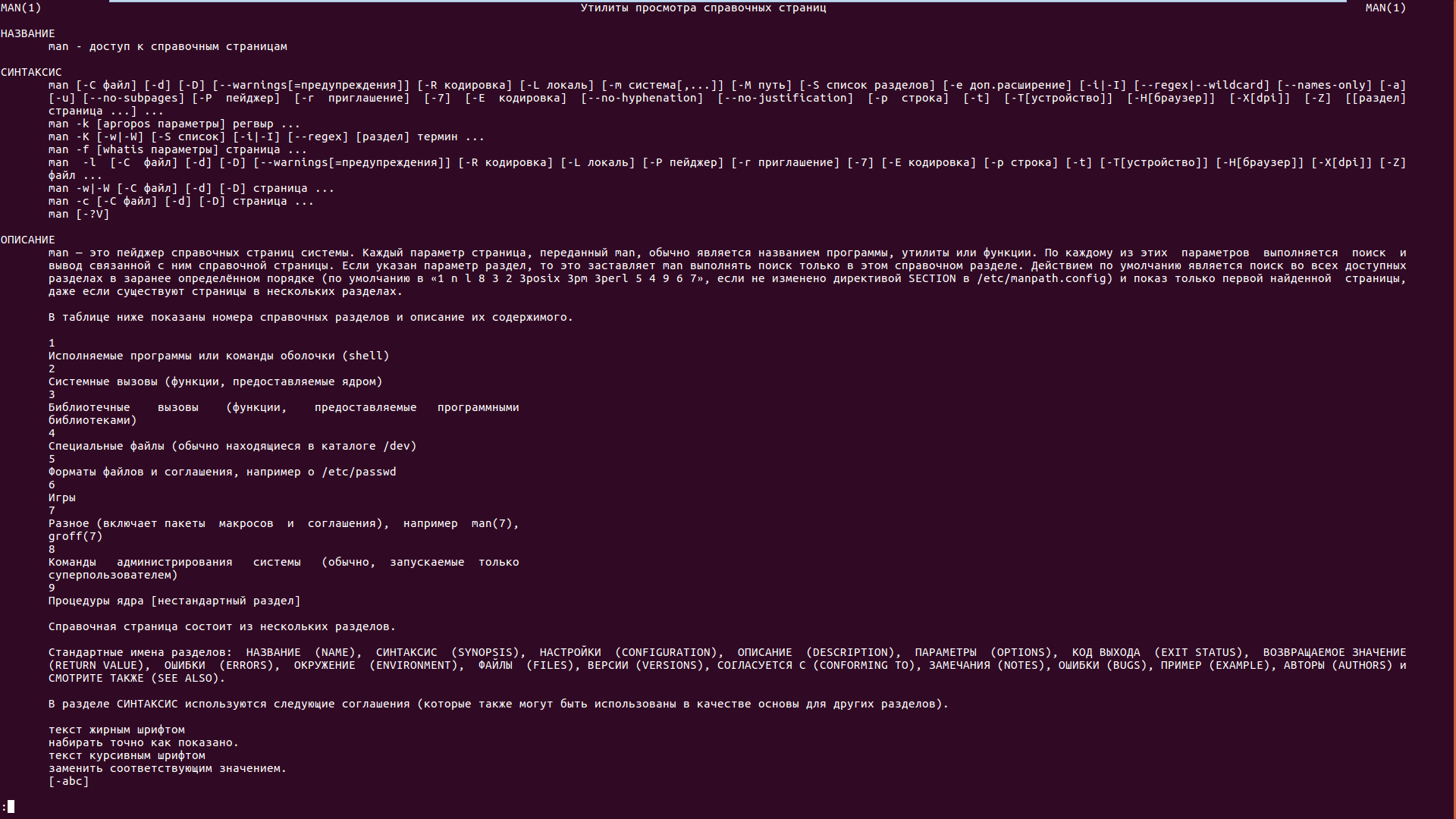
**less** — усовершенствованный вариант команды more. В дополнение к функциям команды more команда less позволяет выполнять следующие операции:

Пример(откроем man man в виде more)



Всё ещё сложно ориентироватся

Пример(откроем man man в виде less)

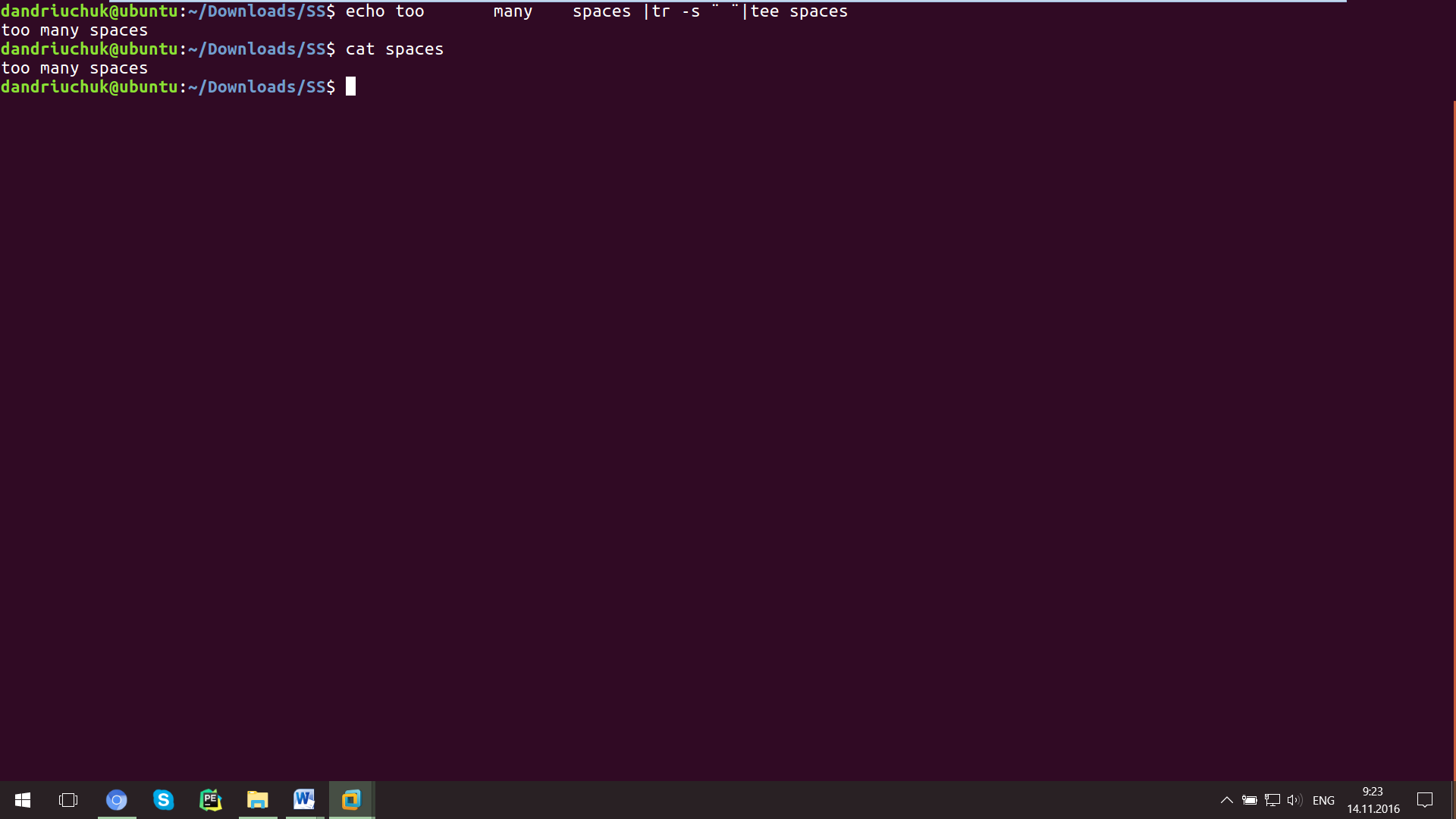


Наиболее удобный так как не засоряет терминал.

1. Использование потоков, конвейеров и перенаправления.
   1. Познакомиться и попрактиковаться с командами tee, xargs

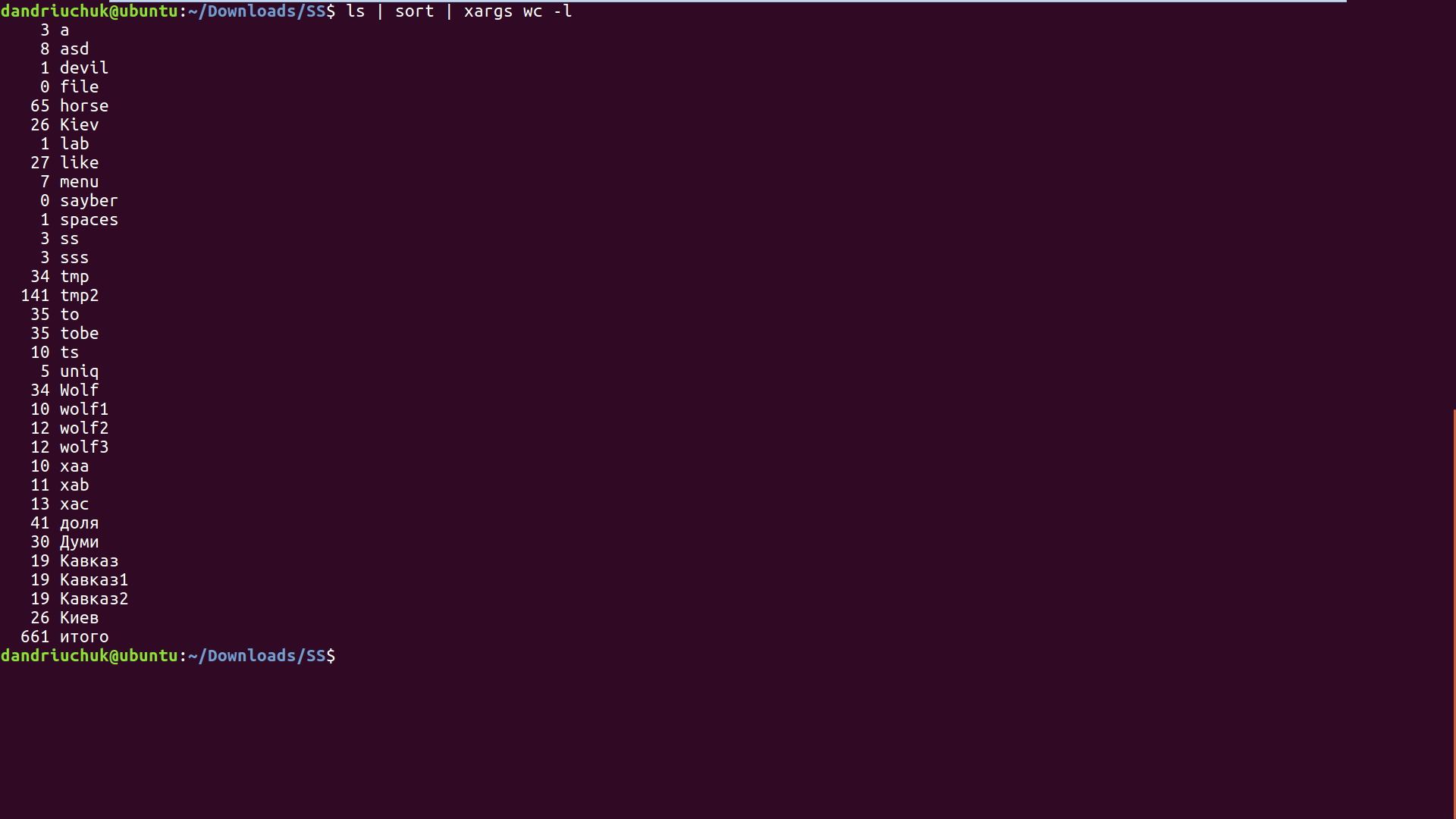
**tee** — используется для разделения потока.

Пример(удалим часто повторяющиеся пробелы и запишем вывод в файл spaces)



**xargs** — утилита для формирования списка аргументов. Команда xargs объединяет зафиксированный набор заданных в командной строке начальных аргументов с аргументами, прочитанными со стандартного ввода, и выполняет указанную команду один или несколько раз.

Пример(Подсчитаем количество строк во всех файлах папки /Downloads/SS)



* 1. Попрактиковаться с перенаправлением данных с помощью символов: >, >>, 2>>, 2>&1, |

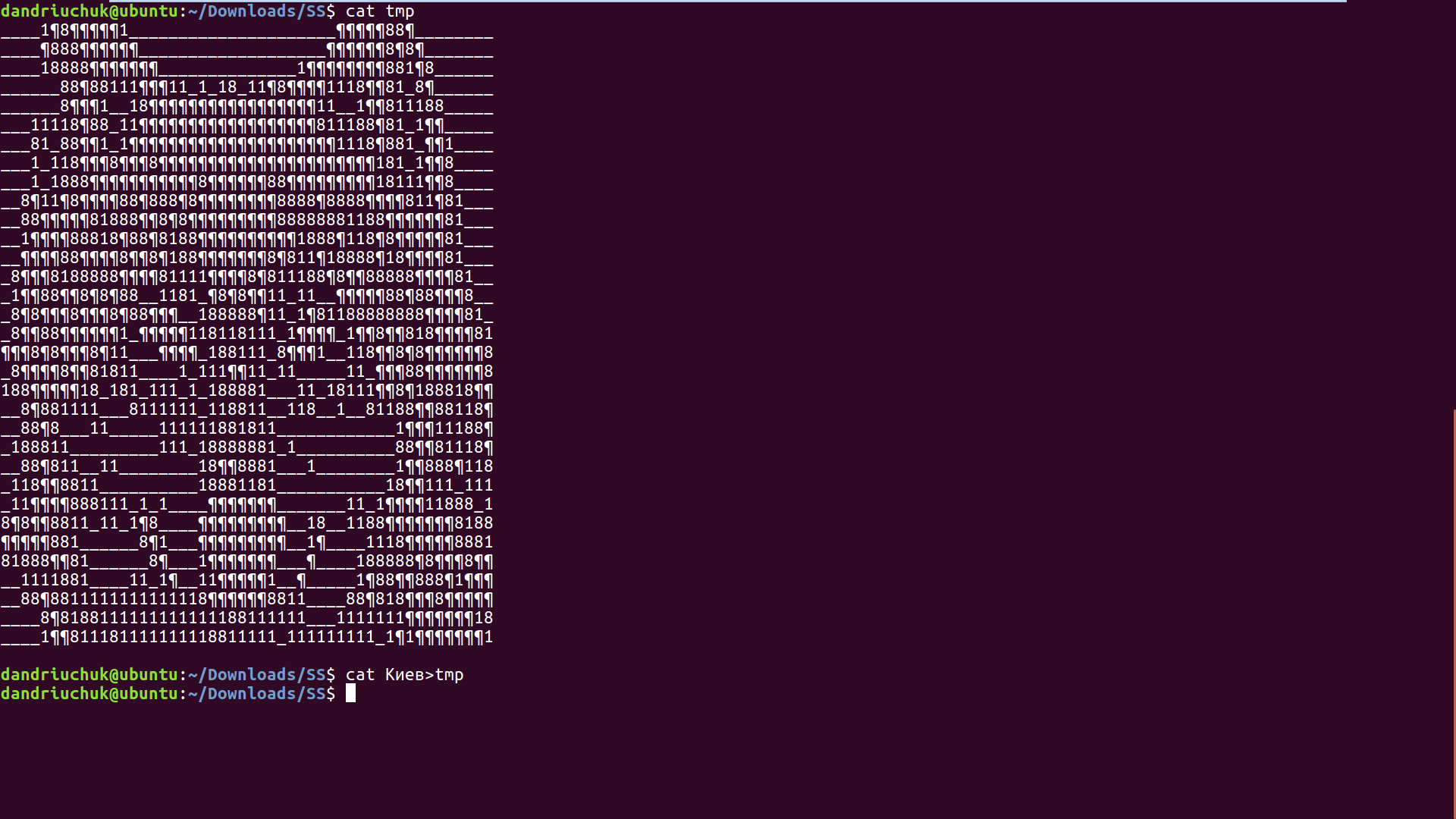
> — запись данных в файл при этом начальные данные в файле стираются.

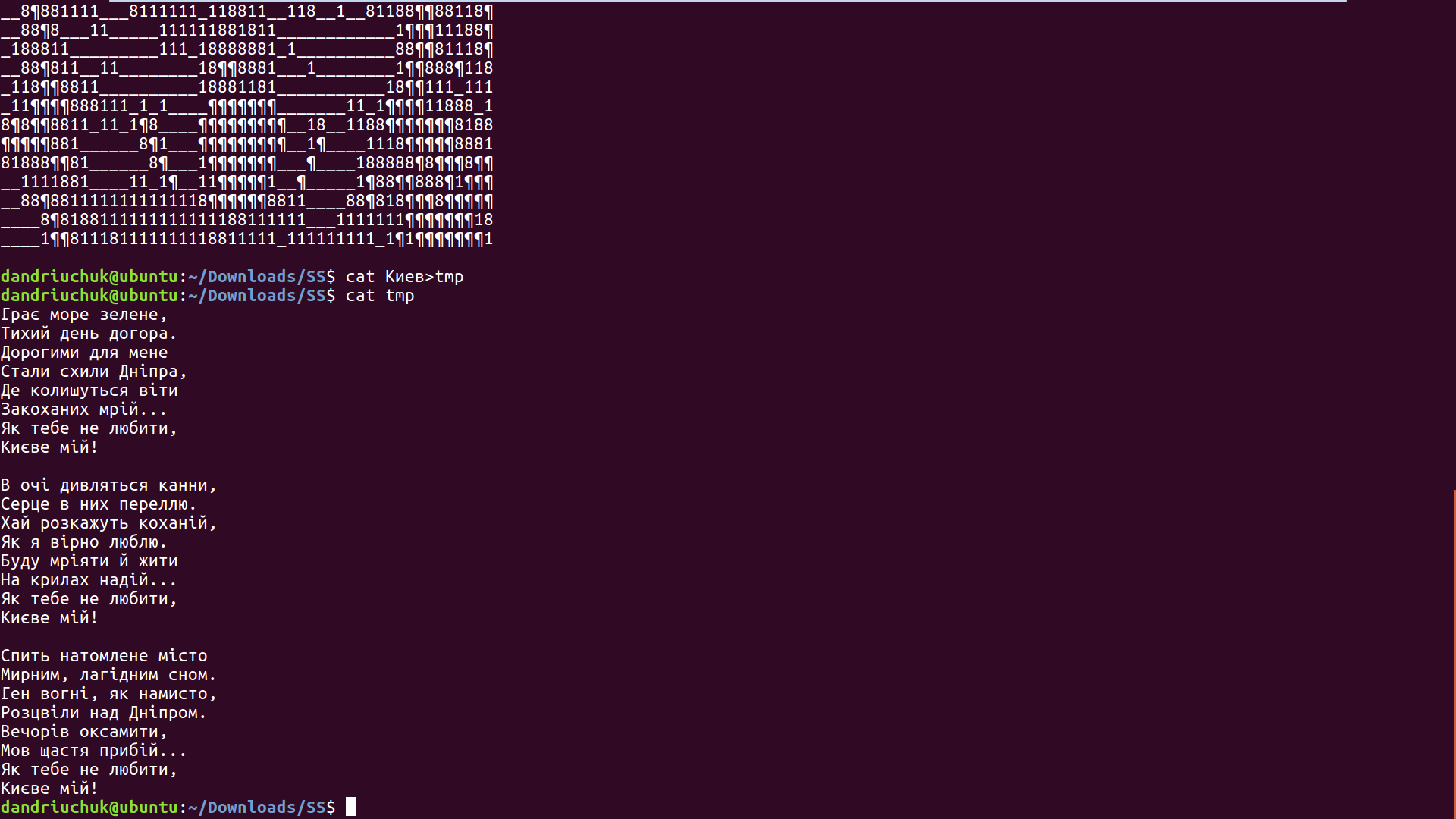
>> — запись данных в файл при этом начальные данные в файле остаются.

2> — перенаправляет стандартный поток ошибок в файл.

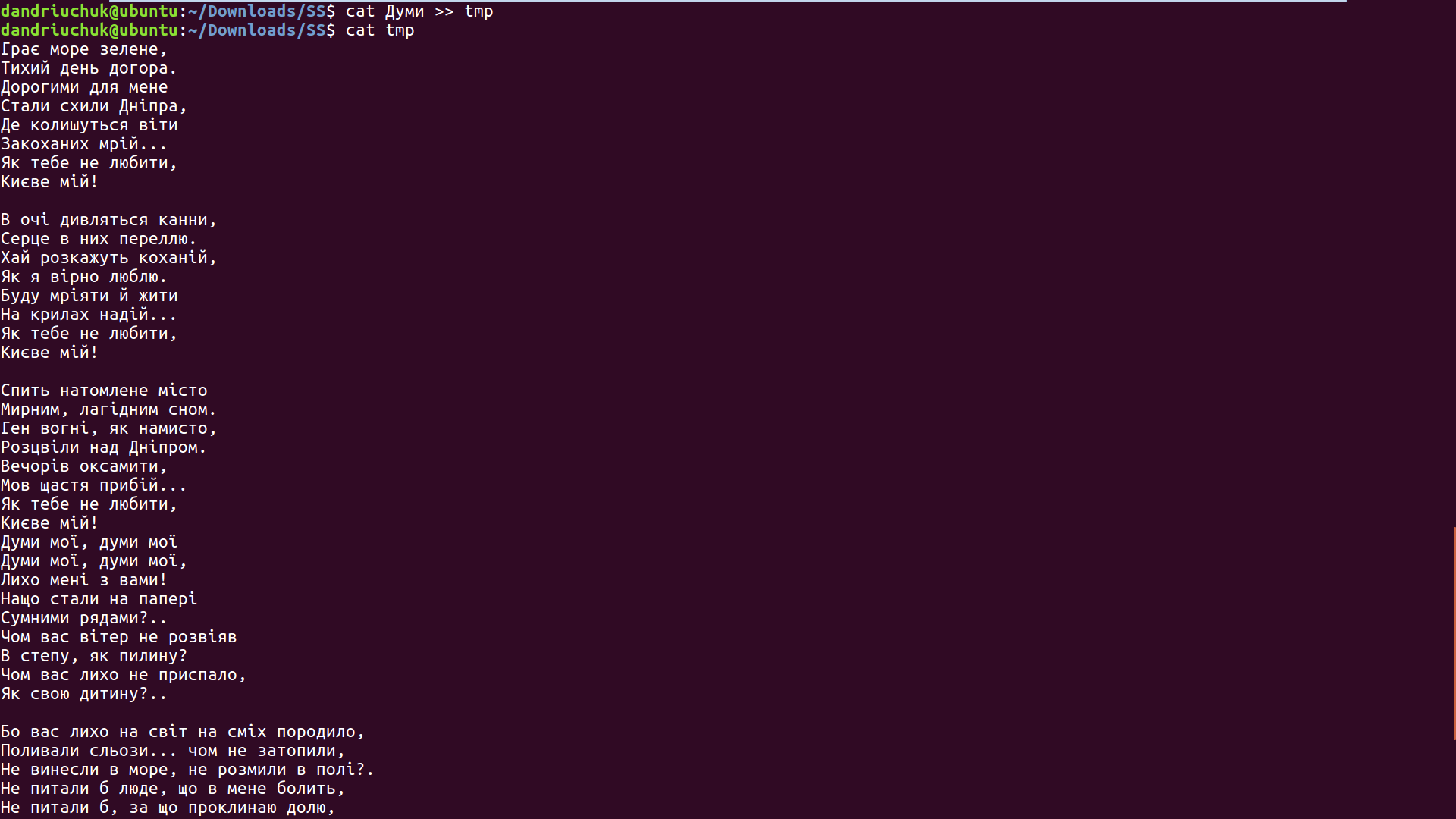
2>> — перенаправляет стандартный поток ошибок в файл не удаляя прежних данных.

Пример:(Перезаписи и дозаписи)



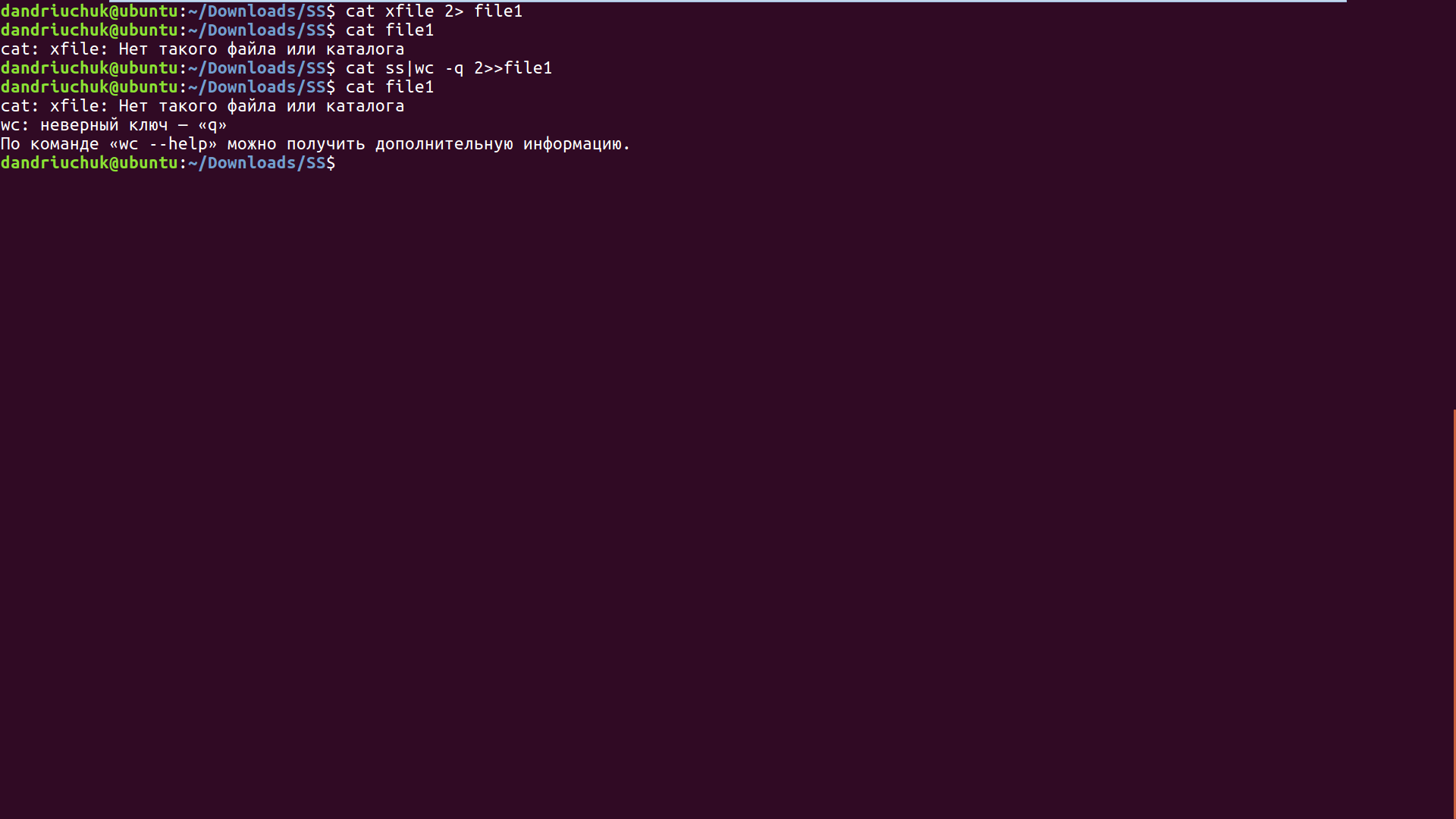


Файд tmp изменился.



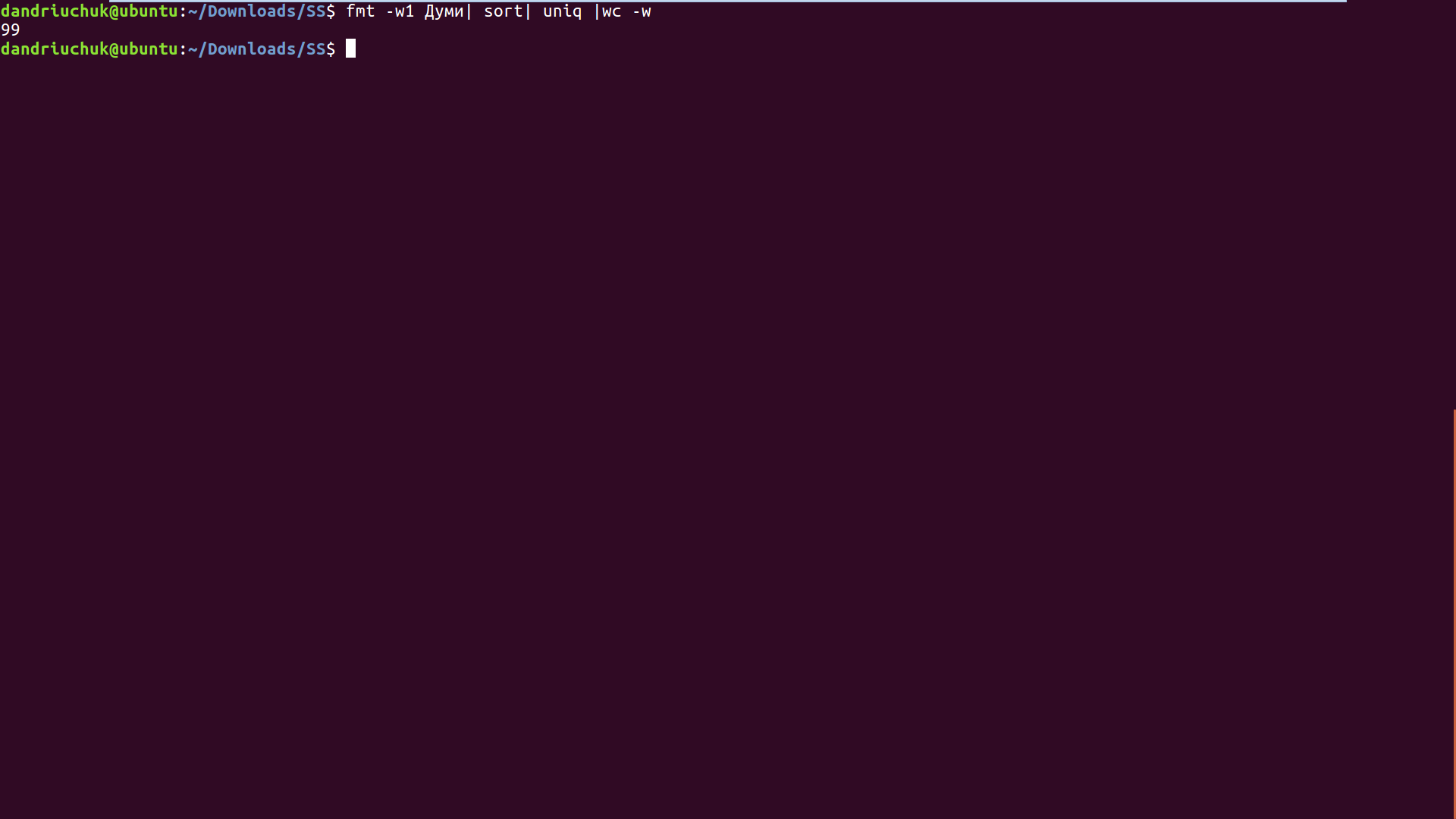
К файлу добавилось содержимое.

Пример перезаписи и дописования потока ошибок



| - весь вывод первой программы передастся второй, объединение в одну строку нескольких команд.

Пример(Зделаем пример для команды uniq одной строкой)



2>&1 – направить поток вывода ошибок в стандартный поток вывода

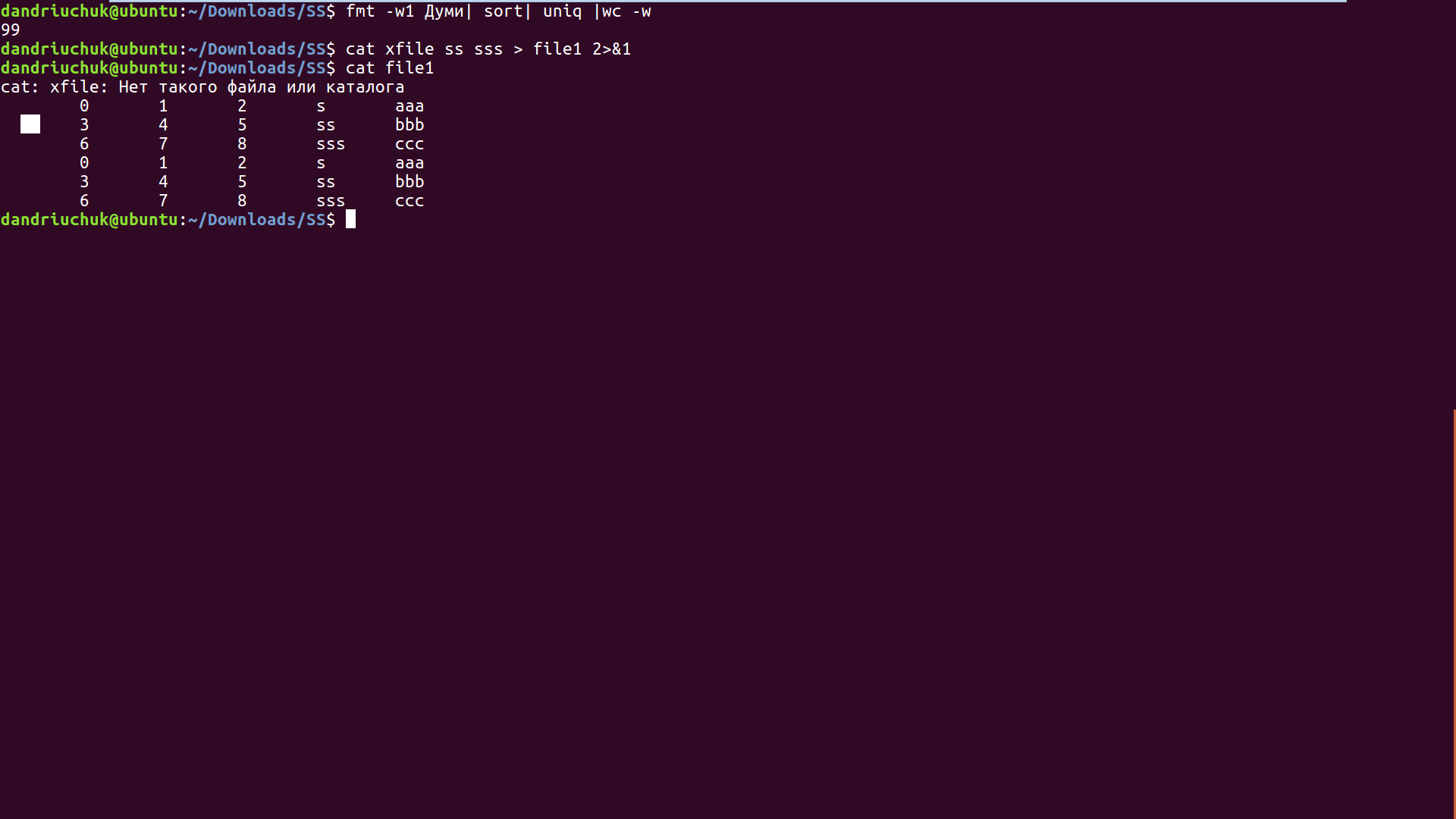
>file 2>&1

1. параметр ">file" — означает "перенаправить стандартный поток вывода (stdout) в файл c именем file";
2. параметр "2>&1" — означает "перенаправить стандартный поток ошибок (2) в стандартный поток вывода (1). Но, так как стандартный поток вывода уже перенаправлен в файл, то и стандартный поток ошибок перенаправляется туда же.

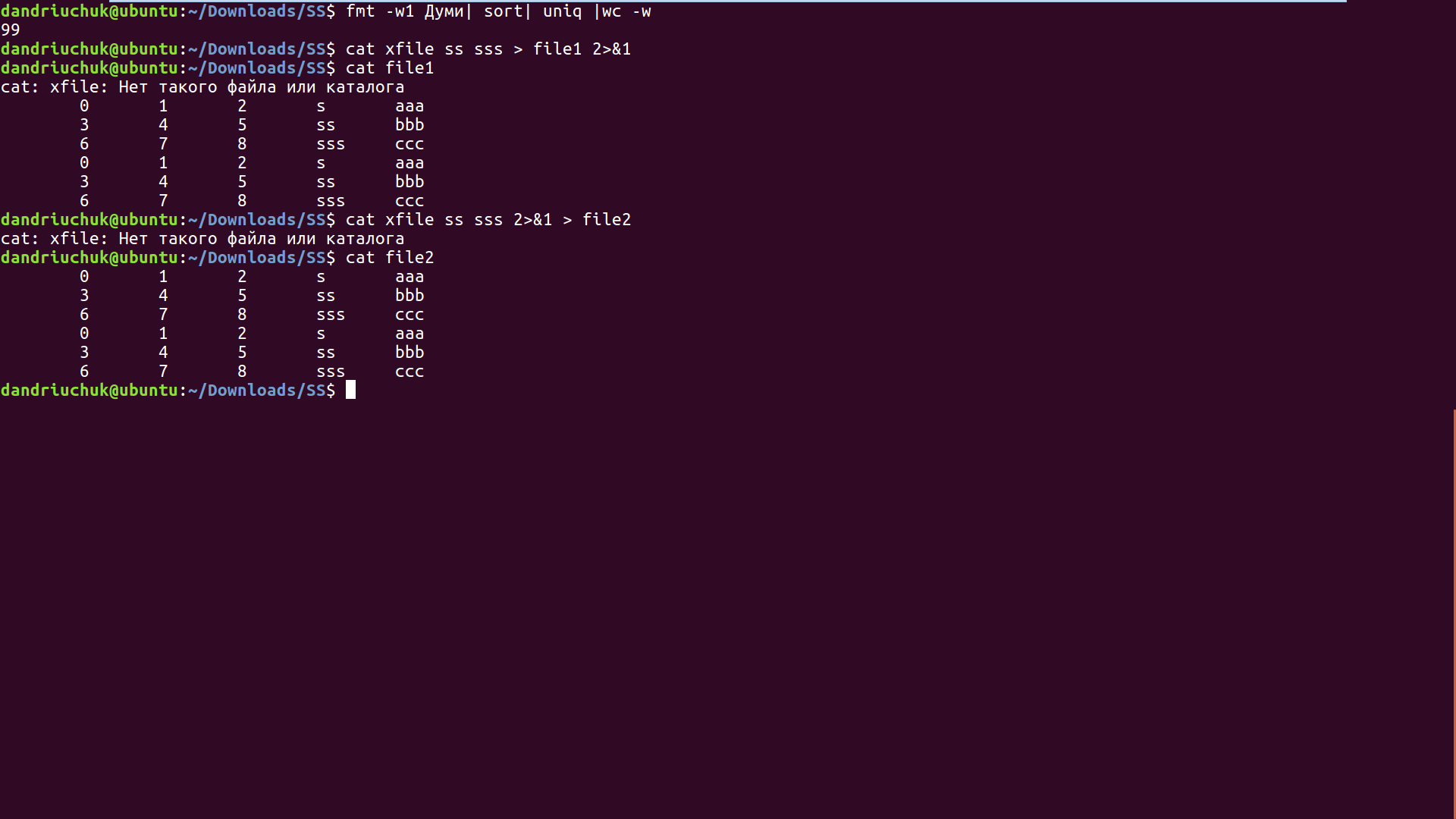
2>&1 >file

1. параметр "2>&1" — означает "перенаправить стандартный поток ошибок в стандартный поток вывода". На этот момент поток вывода (1) направляется в терминал, а значит, что поток ошибок тоже перенаправится в терминал.
2. параметр ">file" — означает "перенаправить стандартный поток вывода (stdout) в файл c именем file".

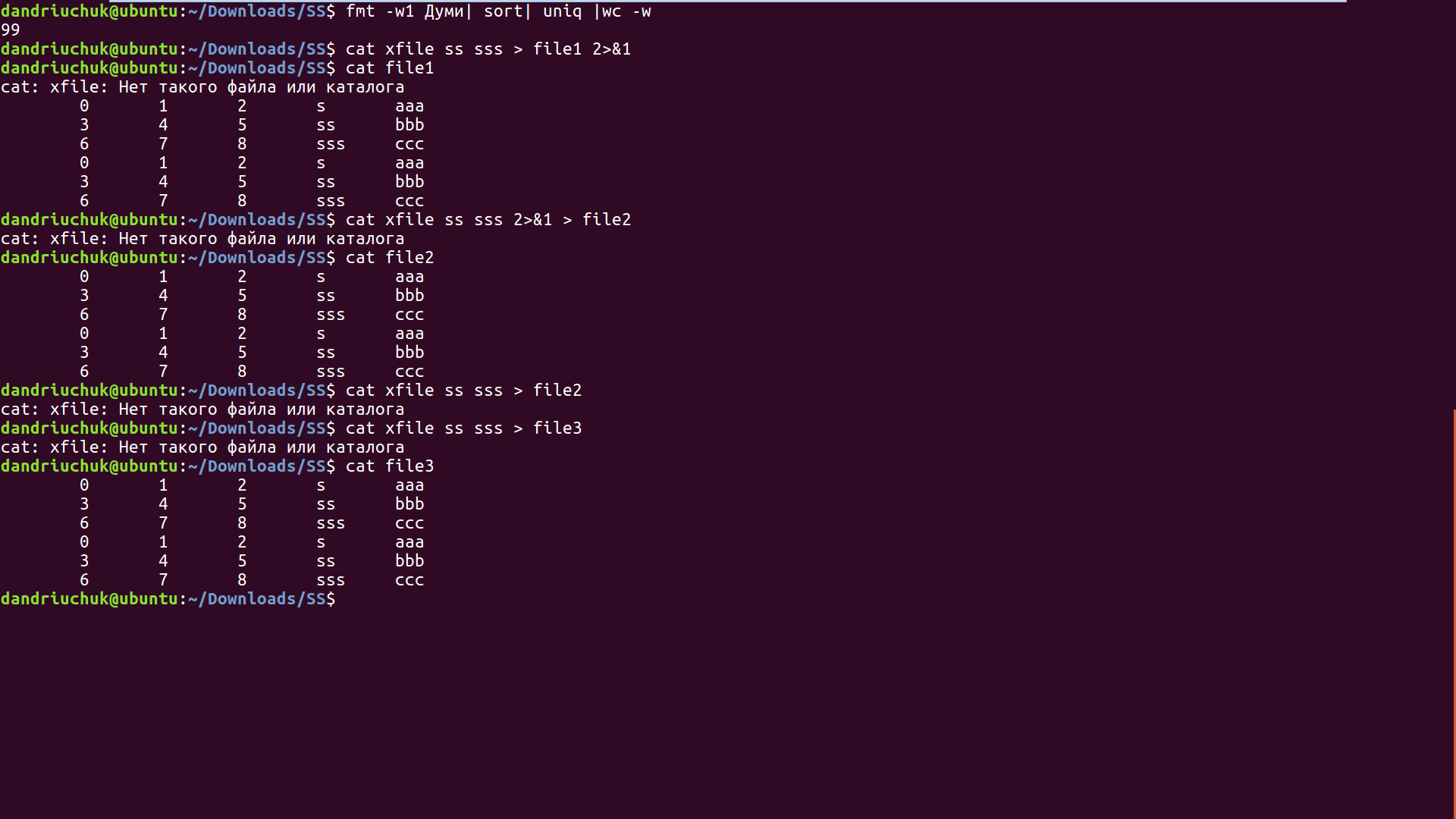
Пример (Выведем всё в один файл)



Пример (Выведем ошибки на экран, а команды в файл)



По сути тоже самое что и запись без 2>&1.



1. Поиск текстовых файлов с использованием регулярных выражений.
   1. Познакомится и попрактиковаться с утилитами grep, egrep, fgrep.

**grep, egrep. fgrep** — поиск указаных символов, комбинации символов.

Пример (Найдём ключ команды grep клторый отвечает за регистр)



Пример (Найдём все to файла to)

