**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра технологий программирования

Жуковский Павел Сергеевич

Технологии Программирования

Отчет по лабораторной работе №2

«Работа в командной строке и с системой контроля версии git» студента 2 курса 13 группы

**Преподаватель**

**Давидовская Мария Ивановна**

**Минск 2019**

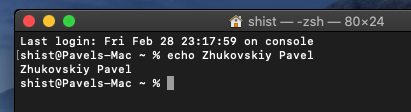
**ЗАДАНИЕ 1. РАБОТА В КОМАНДНОЙ СТРОКЕ**

1. Выведите сообщение «Имя Фамилия», указав Ваше имя и фамилию, в виде

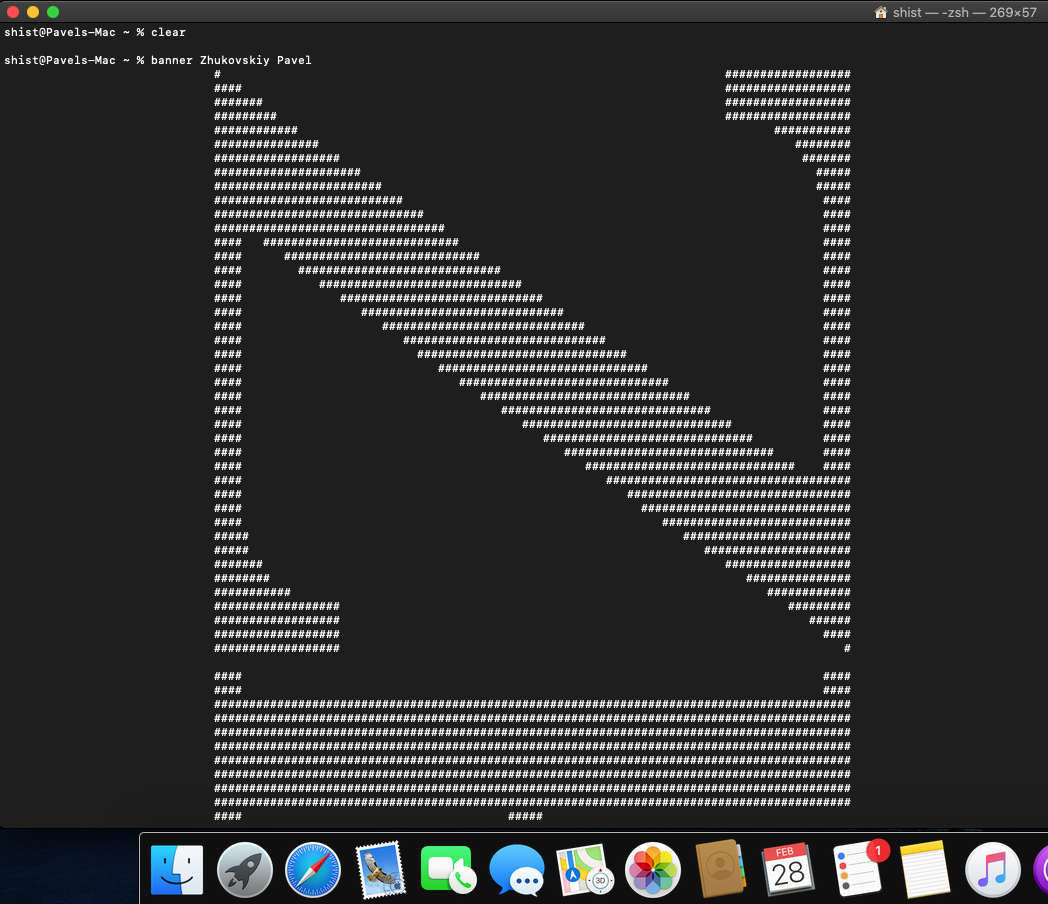
нескольких строк с помощью команд echo и banner. При необходимости

установите требуемое приложение.

Результат работы команды echo:



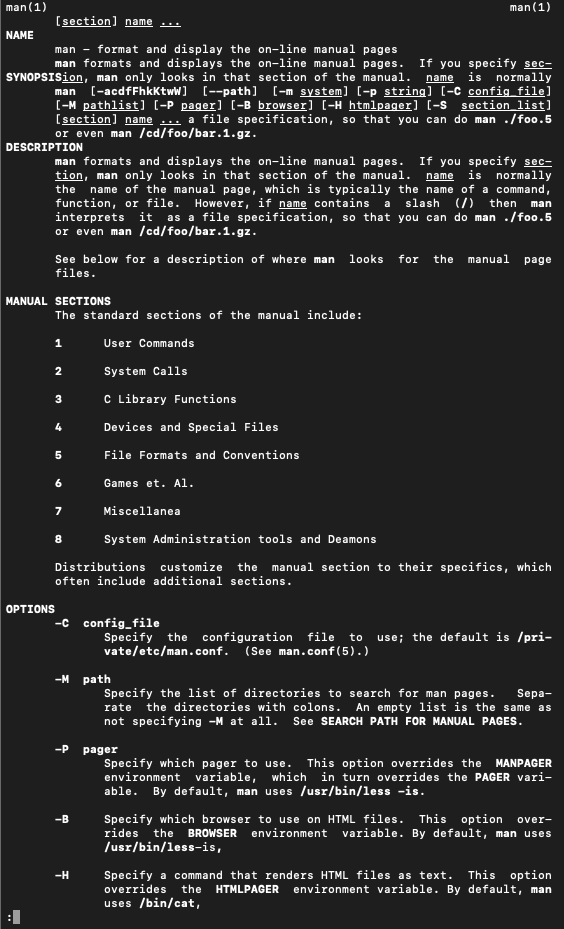
Результат работы команды banner (большая часть не вместилась в экран):



2. Прочитайте статью справочной системы man о пользовании справочной

системой.

Для этого ввожу в консоли команду % man man, после чего на консоли появляется следующая информация:

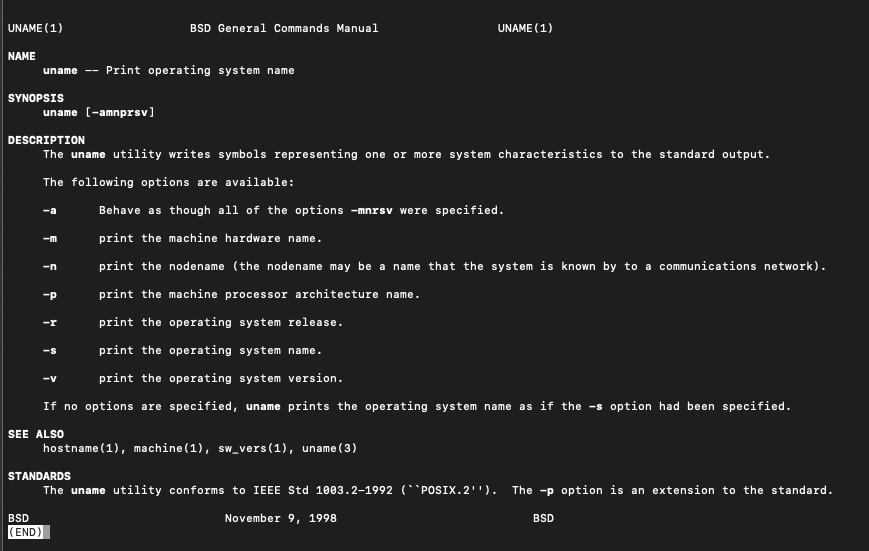


P.S. Чтобы ввести новую команду, пришлось нажать сочетание клавиш ctrl+Z.

3. Прочитайте статью справочной системы о команде uname. Из какого раздела

справочника Вы прочитали статью?

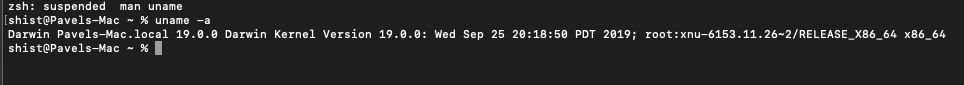
Чтобы прочитать данную статью, требуется ввести в терминал команду % man uname, после чего выведется следующее сообщение:



4. Определите имя машины, название и версию операционной системы, с

которой Вы работаете. Каков аппаратный тип системы?

Все это можно узнать с помощью команды uname –a:



5. Выведите дату в форматах dd-mm-yy, mm-dd-yy hh:mm:ss.

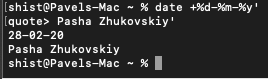
Вывожу с помощью команды date ‘+%d-%m-%y’



6. Выведите дату в две строки: на первой – день, месяц, год, на второй –

текущее время, снабдив вывод комментарием с Вашим именем и фамилией.

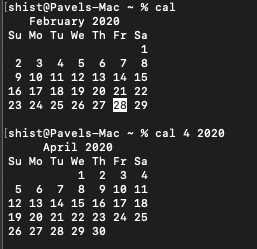
Для этого нужно не вводить открывающую кавычку перед датой, а ввести её во второй строке после указания имени:



7. Выведите календарь на текущий месяц, на месяц Вашего рождения текущего

года.

Для этого ввожу команду % cal, выводится календарь текущего месяца, после этого ввожу команду % cal 4 2020, чтобы вывести календарь апреля этого года (в апреле мой День Рождения).

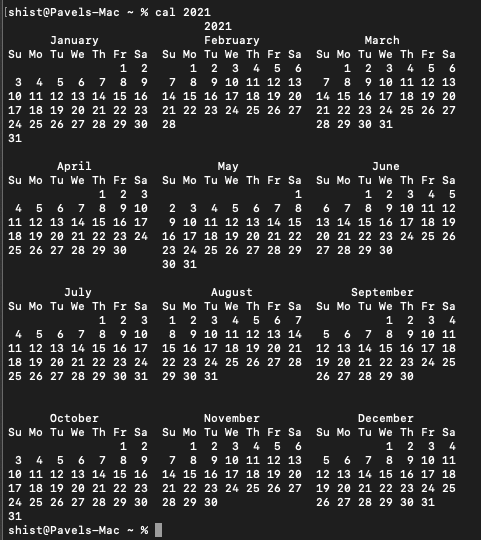


8. Выведите календарь на будущий год. В каком столбце отображаются

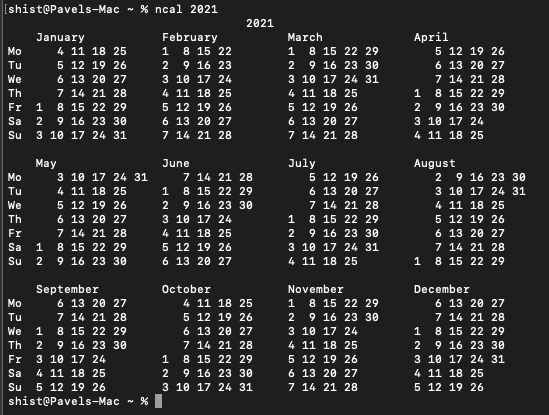
воскресенья? Как сделать, чтобы неделя начиналась с понедельника? С

воскресенья?

Календарь на будущий год можно вывести командой % cal 2021

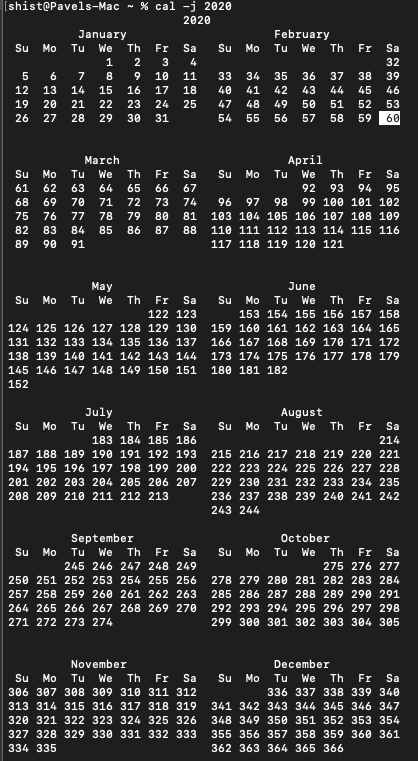


Видно, что каждая неделя начинается с воскресенья (т.е. воскресенье находится в первом столбце). Соответственно, чтобы неделя начиналась с воскресенья, нужно ничего не делать, а так и вводить команду % cal (2021). Если же нужно, чтобы календарь начинался с понедельника, можно ввести команду % ncal (год):



9. Определите порядковый номер текущего дня с начала года.

Это можно сделать с помощью команды % cal –j 2020:

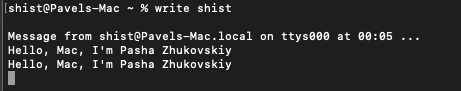


Видно, что текущий день является 60-ым по счету. Также видно, что в году 366 дней, но это потому что в этом году в феврале 29 дней.

10. Мария Ивановна, этот пункт у вас почему-то был пустой.

11.Используя команду write, пошлите сообщение на консоль.

Посылаю сообщение на консоль командой write (перед этим нужно залогиниться своим именем в операционной системе):



P.S. Выходим с помощью ctrl+Z

12.Используя команду mesg, определите, разрешены ли сообщения на Ваш терминал. Запретите сообщения. Какова будет реакция системы, если кто-

нибудь попытается передать Вам сообщение?



Так как консоль написала «is y», это означает, что сообщения разрешены, но их можно запретить командой % mesg n:



Консоль написала «is n», что означает, что сообщения запрещены, соответственно, если кто-то попытается передать мне сообщение, у него это не получится.

13.Прочитайте свою почту в консоли.

Это можно сделать командой % mail:

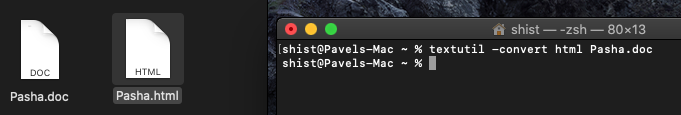


P.S. Как видно, у меня нет писем.

14.Привести несколько примеров команды конвертации файла \*.doc в формат

\*.html.

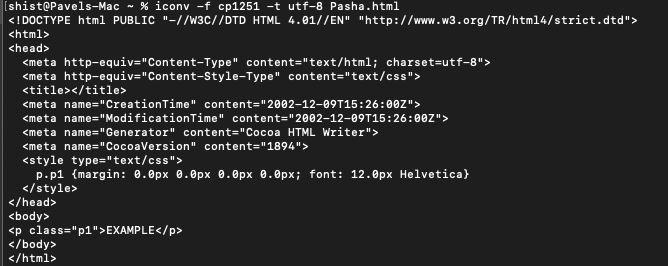
В данном примере конвертирую \*.doc файл в файл формата \*.html



15.Привести несколько примеров изменения кодировки файла, например из

cp1251 в utf-8.

Например, iconv -f cp1251 -t utf-8 test.html test.html :



**ЗАДАНИЕ 2. РАБОТА С ФАЙЛАМИ**

*cat*

1. Создавать файл dirlist.txt, состоящий из 10 строк. Нечётные строки —

пустые, чётные — содержат 2) Ваше ФИО, 4) место (город/посёлок)

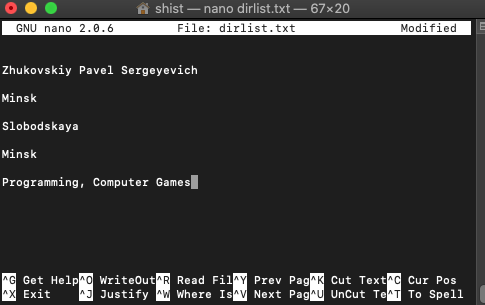
проживания, 6) улица проживания, 8) область проживания, 10) список Ваших

хобби и интересов.

Для начала прописываю команду nano dirlist.txt:

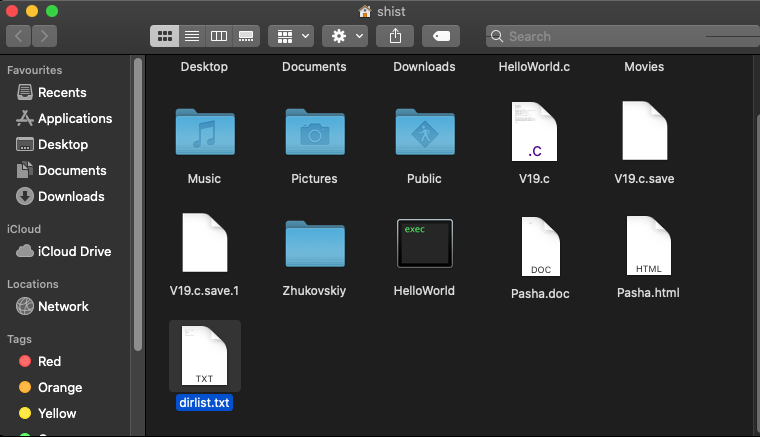


Затем заполняю его данными:



После чего сохраняю с помощью сочетанием клавиш ctrl+O, затем закрываю окошко сочетанием клавиш ctrl+X, после чего возвращаюсь в терминал.

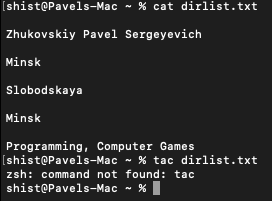
Файл появился:



2. вывести содержимое файла dirlist.txt:

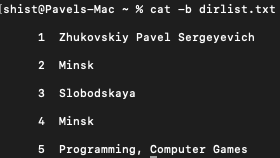
a) в обратном порядке:

Вообще, чтобы просто вывести содержимое файла, используется команда % cat, а чтобы вывести в обратном порядке, используется команда % tac, но у меня не видит команду % tac, так как её нету в Mac OS:



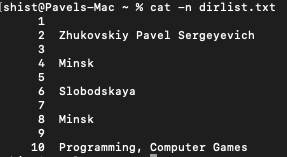
b) с нумерацией непустых строк:

Команда % cat –b (имя файла):



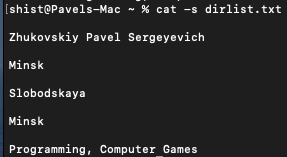
c) с нумерацией всех строк:

Команда % cat –n (имя файла):



d) схлопывая подряд идущие пустые строки в одну:

Команда % cat –s (имя файла):

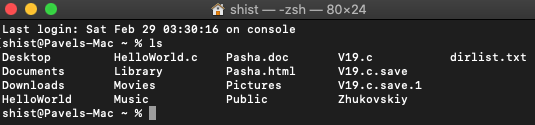


P.S. Только здесь эта функция не видна, т.к. более одной пустой строки подряд у нас в файлике нету.

*Работа с файлами*

3. Просмотрите содержимое Вашего домашнего каталога.

Для этого используем команду % ls



4. Определите имя своего домашнего каталога и его родительского каталога.

Для этого нужно прописать команду % echo ~



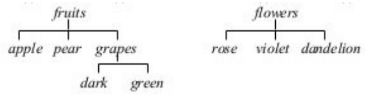
5. Определите полный путь до Вашего домашнего каталога.

Он определяется командой % pwd

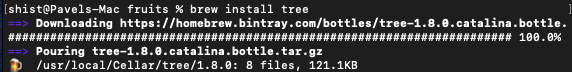


6. Создайте в своём домашнем каталоге подкаталоги вида и выведите их

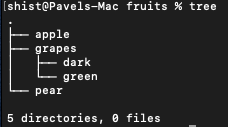
структуру в виде дерева:

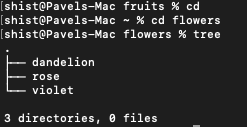
  
Для этого понадобится команда (нужна папка) % tree.

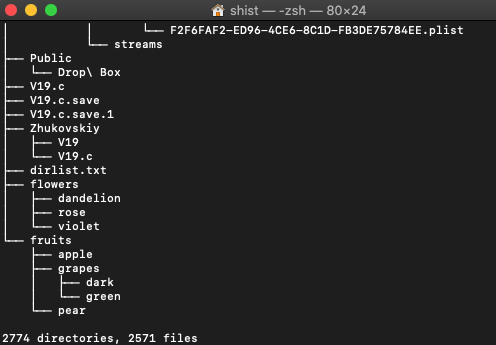
Но прежде всего эту команду нужно было установить в систему:



После чего я ей воспользовался и вывел деревья (перед этим нужно было зайти в папки fruits и flowers командой % cd (имя папки)):



  
Я даже попробовал вывести дерево всех моих папок, после чего система вывела мне огромное дерево, включая папки fruits и flowers:

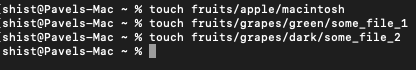


7. Находясь в домашнем каталоге, с помощью команды touch создайте файл

macintosh в имеющемся каталоге apple и несколько файлов в каталогах

dark и green.

Создал:

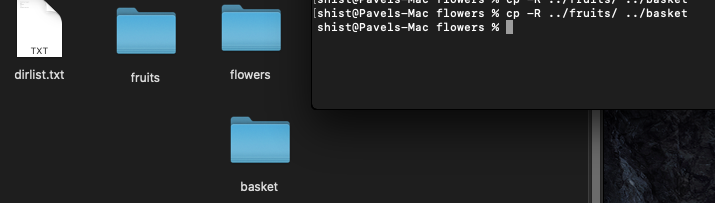


8. Войдите в каталог flowers. Находясь в каталоге flowers, скопируйте все

подкаталоги fruits вместе с находящимися в них файлами в специально

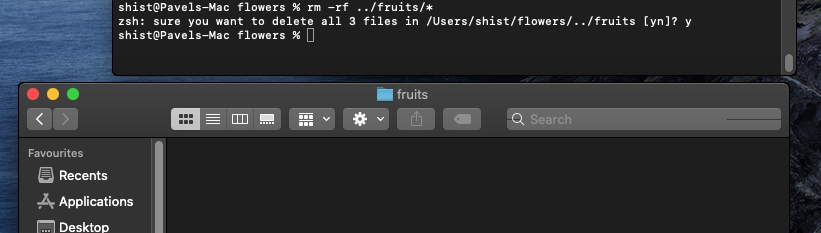
созданный каталог basket.

Делаю это командой % cp –R ../fruits/ ../basket



9. Находясь в каталоге flowers, удалите каталог fruits.

Делаю это командой % rm –rf ../fruits/\*



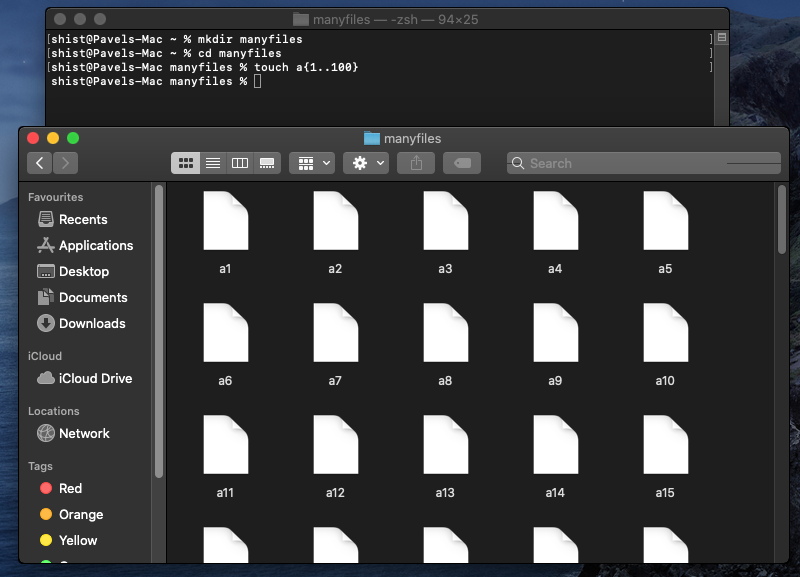
Видно, что после этого папка fruits стала пуста.

10.Создайте в домашнем каталоге директорию manyfiles. Создайте в ней с

помощью команды и регулярных выражений 100 файлов с именами a1,

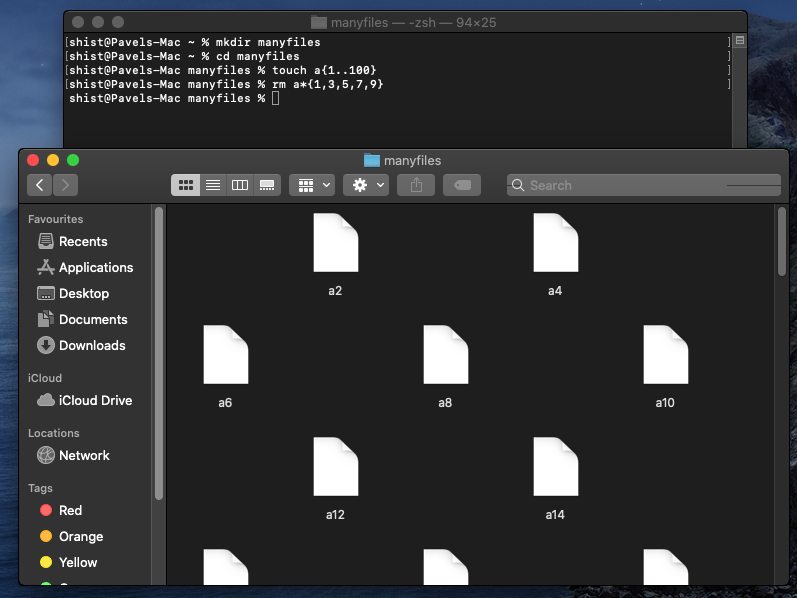
a2, a3, .... a100. Цикл не используем!

Для начала создаем директорию manyfiles с помощью команды % mkdir. Затем открываем эту папочку командой cd manyfiles. После чего вводим команду % touch a{1..100}, которая создаст нам в папке manyfiles сотню файлов с именами от a1 вплоть до a100:



Удалите только файлы с нечётными номерами.

Удаляем все файлы, имена которых заканчиваются на 1, 3, 5, 7, 9. Это и будут файлы с нечетными номера. Для этого можно прописать команду rm a\*{1,3,5,7,9}:

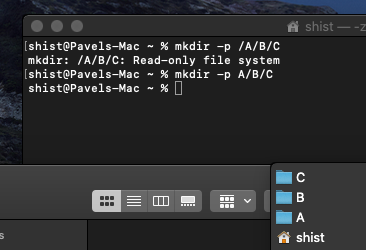


11.Находясь в своём домашнем каталоге, создайте следующие подкаталоги,

используя регулярные выражения и одну командную строку:

◦ A/B/C;

Создаём с помощью команды % mkdir –p /A/B/C (параметр –p отвечает за то, чтобы при отсутствии внешних папок они создавались автоматически):



◦ A/B;

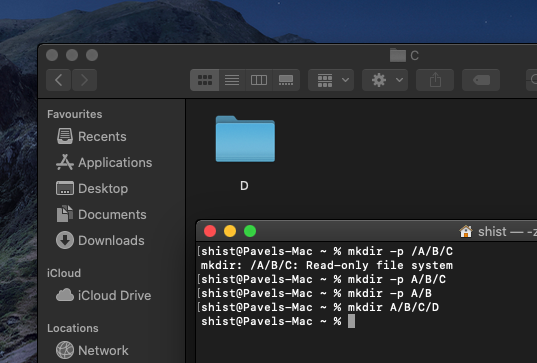
Ввожу % mkdir –p A/B:



Ничего не произошло, т.к. у нас уже есть папка A и папка B в ней.

◦ A/B/C/D;

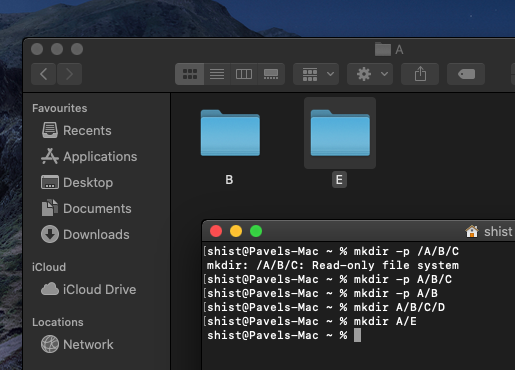
Ввёл % mkdir A/B/C/D (без –p, так как у нас уже есть папки A/B/C):



Видно, что в папке C появилась папка D.

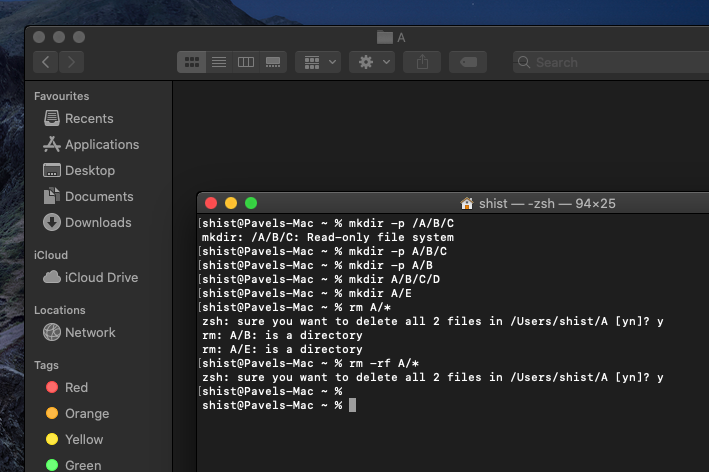
◦ A/E.

Ввёл % mkdir A/E (без –p, так как у нас уже есть папка A):



Видно, что в папке A создалась папка E.

Находясь в своем домашнем каталоге, удалите все подкаталоги каталога A.  
Сначала я ввел rm A/\*, чтобы удалить папки B и E. У меня не получилось, так как B и E являются целыми директориями, и потом, чтобы удалить их, мне пришлось добавить еще параметр –rf, итого команда выглядела % rm –rf A/\*:



*grep*

12.Обновите в редакторе nano текстовый файл dirlist.txt так, чтобы он

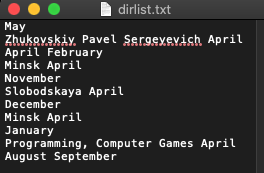
содержал в каждой строке как минимум одно название месяца. Вывести

строки файла dirlist.txt, содержащие строки с определенным месяцем

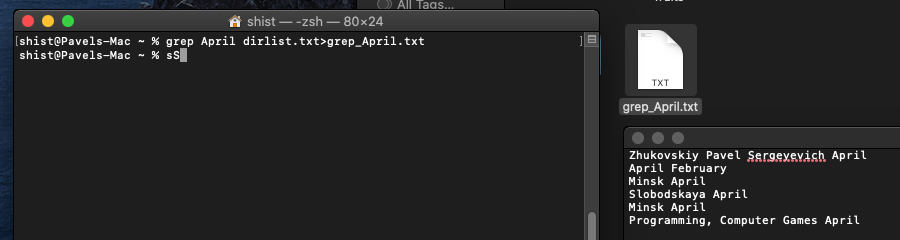
(месяц определяем по дате рождения) и записать их в файл

grep\_month\_name.txt

Обновил файл dirlist.txt следующим образом:



После чего ввёл команду % grep April dirlist.txt>grep\_April.txt

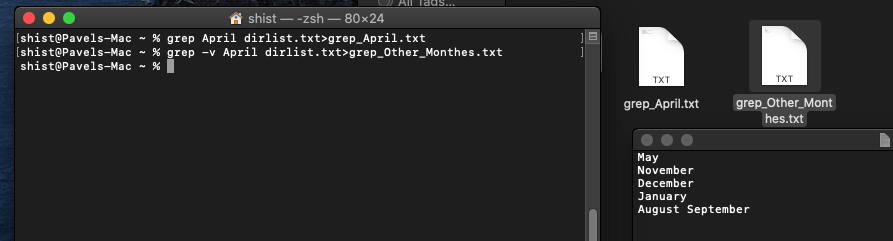


После этого у меня создался файл с именем grep\_April.txt, в котором действительно появились нужные строки.

13.Записать строки, не содержащие этот месяц, в файл

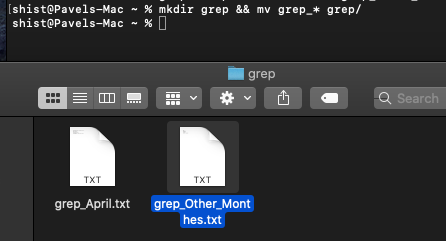
grep\_other\_monthes.txt.

Ввёл grep –v April dirlist.txt>grep\_Other\_Monthes.txt, после чего создался файл grep\_Other\_Monthes.txt, в котором вывелись все строки, не содержащие April:



14.Создать папку grep, переместить в нее файлы созданные в пунктах 12 и 13.

Создал папку командой mkdir grep и переместил туда файлы, начинающиеся с grep\_, вот таким образом (команда && объединяет две команды; команда mv перемещает что-то куда-то):



15.Создать папку mac\_os\_lab и вложенные папки (не менее 3-х уровней

вложения и 5 вложенных папок). Создать текстовые файлы в каждой из

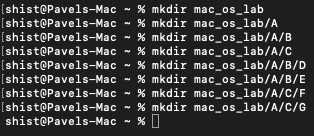
директорий. Ряд файлов должны содержать слово «root». Находясь в

папке mac\_os\_lab найти все файлы в этой директории и ее

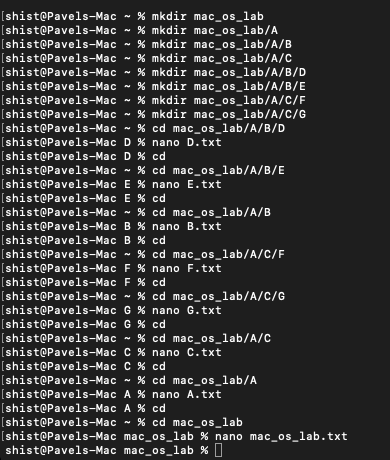
поддиректориях, в которых встречается подстрока root, вывести строки с

указанием их номеров.

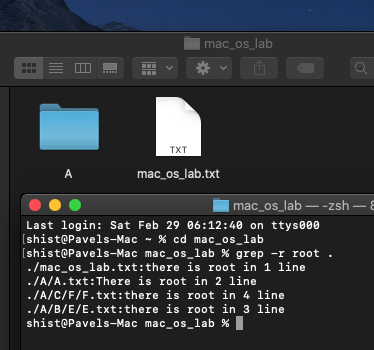
Для начала я создал 7 папок на трёх уровнях (две папки D и E внутри папки B; две папки F и G внутри папки C; сами же папки C и B внутри папки A; папка A лежит внутри mac\_os\_lab):



Затем я создал текстовые файлы (по одному в каждой папке), и в каждом из них кое-что написал (в некоторых файлах я записывал подстроку «root»):



После всех вышеперечисленных действий я прописал команду grep –r root ., после чего терминал вывел мне всю нужную информацию.



*find*

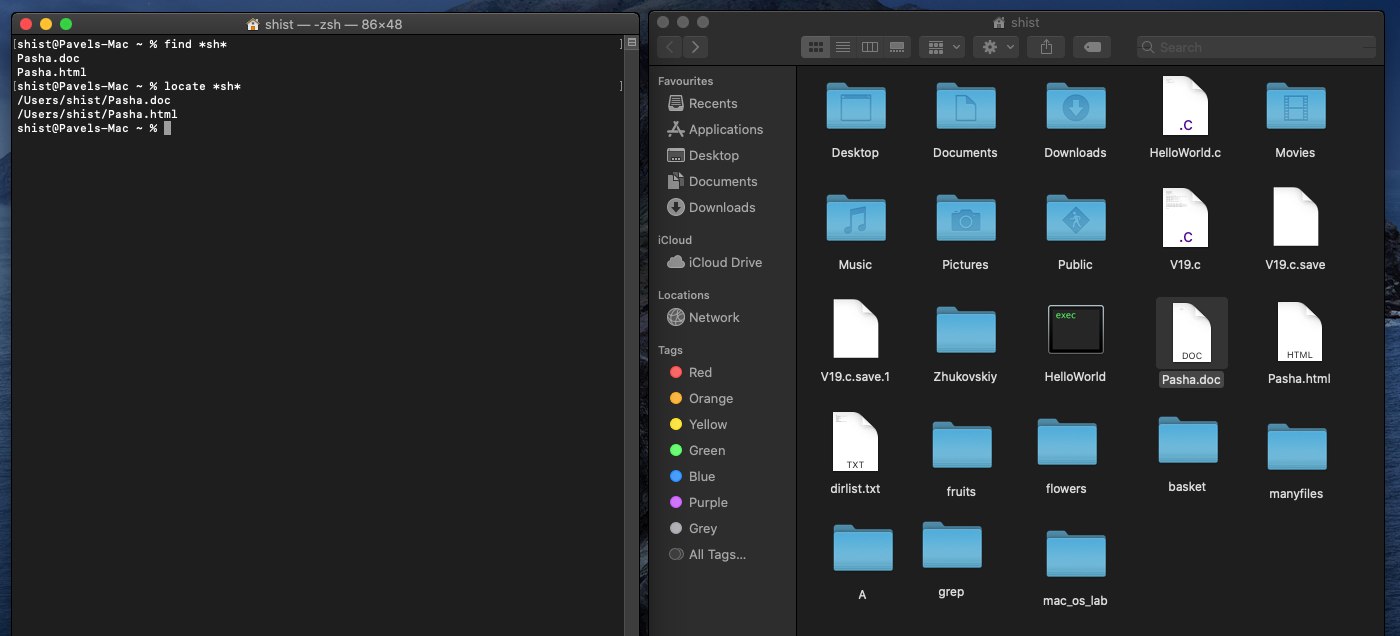
16.Найти все файлы в системе, содержащие в имени «sh» (с помощью команд

find и locate). Какая из команд работает быстрее, почему? Доп:

Проверить совпадают ли результаты выполнения обеих команд, чем они

отличаются, почему?

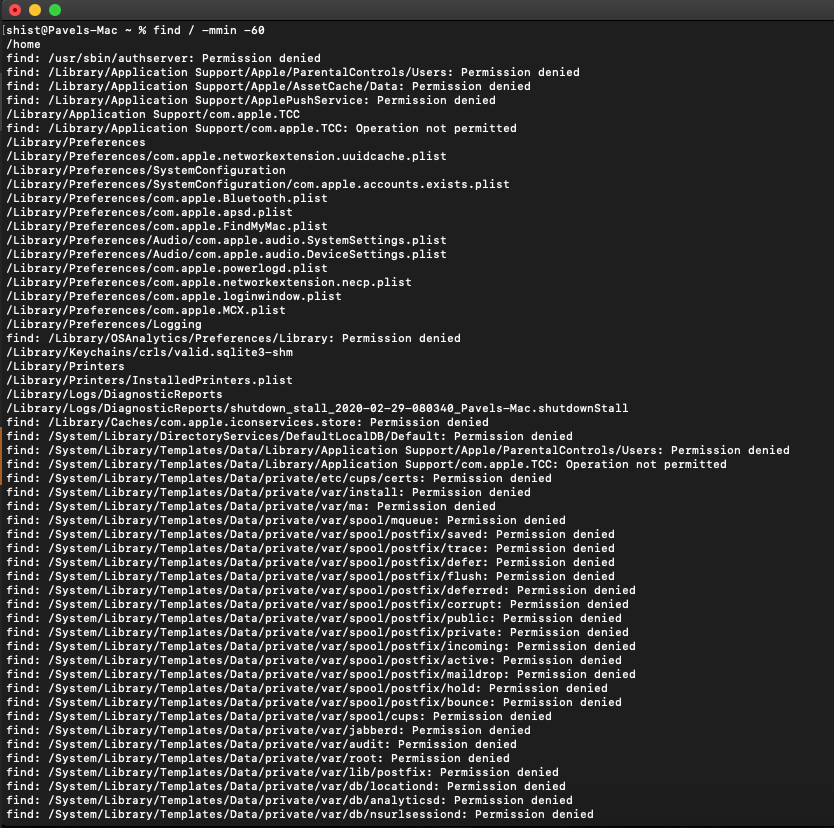
Ищу с помощью команд % find \*sh\* и % locate \*sh\*:



Команда find работает медленнее, так как ей приходится искать по всей системе, в то время как команда locate работает быстрее, ибо ей приходится работать с базой данных для поиска файлов. Отличие этих команд заключается в том, что команда locate, в отличие от команды find, выводит не только имя файла, но и весь путь до него. Это связано с особенностями БД, с которой работает команда locate.

17.Найти файлы, изменённые за последний час.

Делаю это командой % find / -mmin -60:



18.Удалить весь каталог manyfiles со всеми файлами.

Делаю это командой % rm –rf manyfiles:



После этого папка manyfiles исчезла.

*Математические операции в консоли*

19. Используя редактор nano, создайте текстовый файл следующего

содержания:

25+117

86\*124

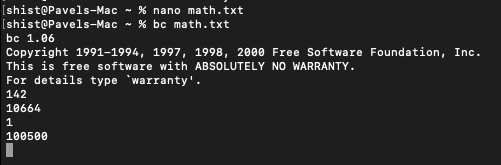
157%12

45225/45\*100

Посчитайте все примеры из файла с помощью одной команды. (Вариантов команды

существует несколько, засчитывается каждый из них). Использовать встроенную команду!

Можно воспользоваться командой % bc (имя файла с примерами):



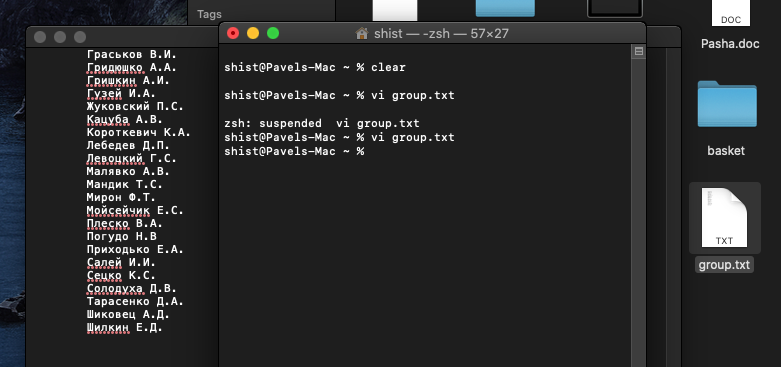
*Команды для работы с текстом*

20.Используя редактор Vi/Vim, создайте текстовый документ, содержащий

список студентов группы. Структура каждой строки: символ табуляции,

фамилия, пробел, инициалы.

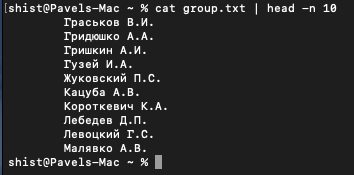
Создал командой % vi group.txt



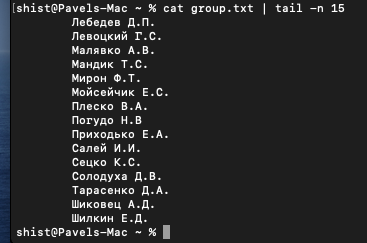
P.S. Чтобы выйти, сохранив, нужно нажать ESC : wq ENTER

21.Выведите первые 10 строк файла, последние 15 строк файла.

Выводу первые 10 строк командой % cat group.txt | head –n 10:



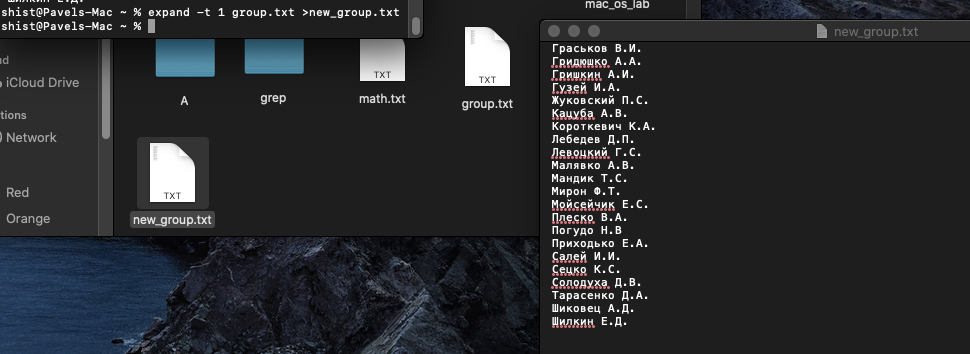
Вывожу последние 15 строк командой % cat group.txt | tail –n 15:



22.Используя команду expand, замените символы табуляции в начале строки

на соответствующее количество пробелов и сохраните с новым именем.

С помощью команды команды expand –t 1 group.txt >new\_group.txt я заменил символы табуляции на пробелы и сохранил в новый файл new\_group.txt:



23.Используя команды diff, diff3 сравните файлы. В каких случаях

используется каждая из команд.

Сравнил:



Команда diff используется тогда, когда нам нужно сравнить 2 файла. Если же нам понадобится сравнить, например, 3 файла, то тут мы уже воспользуемся командой diff3.

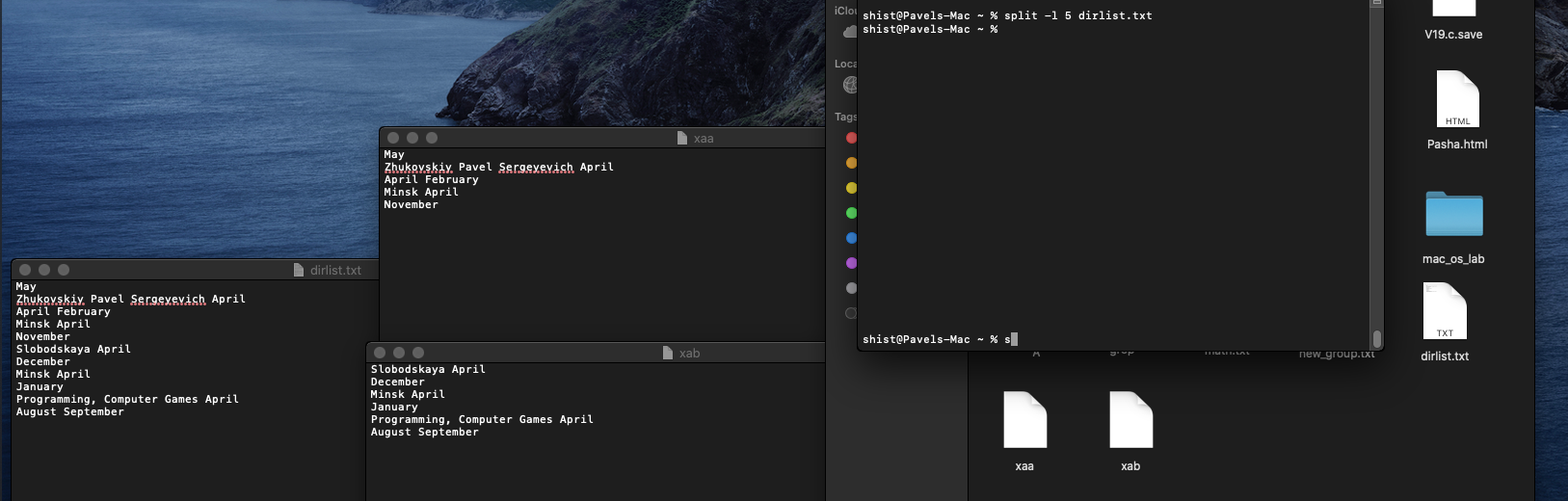
24.Для каких задач используется команды cmp, comm

Команда cmp используется тогда, когда нам требуется сравнить какие-то два файла побитово.  
Команда comm используется тогда, когда нам нужно сравнить два файла на совпадение строк и/или на уникальные строки.

25.Используя команду split разбейте файл dirlist.txt из 10 строк на два

файла по 5 строк.

Разбил командой % split –l 5 dirlist.txt:



**ЗАДАНИЕ 3. РАБОТА С GIT-РЕПОЗИТОРИЕМ**

1. Для работы с Git-репозиторием необходимо установить утилиту git, если

она не установлена.

Установил.

2. Задать данные пользователя (имя, адрес электронной почты), в консоли на

глобальном уровне.

Сделал это, используя команды % git config --global user.name «ZhukovskiyPavel» и % git config --global user.email shist.pupust@mail.ru:



3. Изучить основные команды работы с git в консоли (см. СТРУКТУРА GIT).

Изучил.

4. Создать учётную запись на проекте github.

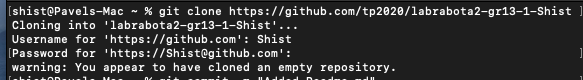
Создал. (ссылка: https://github.com/Shist)

5. Перейти по ссылке-приглашению, опубликованному для Вашей подгруппы в

теме 2 курса на сайте <https://edufpmi.bsu.by/>.

Перешел.

6. Клонировать репозиторий на свою машину в консоли.

Клонировал командой % git clone [https://github.com/tp2020/labrabota2-gr13-1-Shist](https://vk.com/away.php?to=https%3A%2F%2Fgithub.com%2Ftp2020%2Flabrabota2-gr13-2-EgorS2000&cc_key=)  


7. Создать файл Readme. Добавить в список отслеживаемых (git add) и

зафиксировать в репозитории (закоммитить) (git commit). Данная

последовательность команд обеспечивает создание ветки master. Если не

опубликовтаь первый коммит и попробовать создать ветку, то ветка master не

будет создана! Опубликовать изменения из локального репозитория во

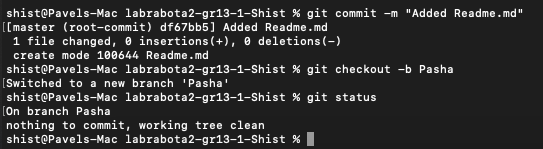
внешнем репозитории на github, выполнив команду git push, т. е. запушить.



8. Создайте в локальном репозитории ветку Development (у меня это будет папка Pasha) в консоли и

переключитесь в неё.

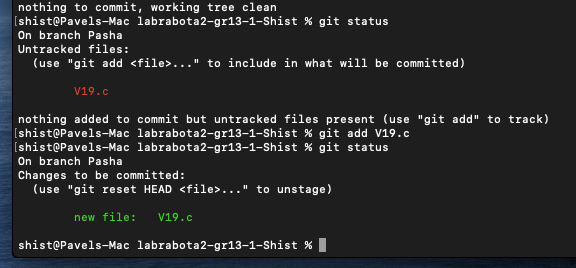
Создаю ветку командой % git commit –m «Added Readme.md». Можно переключиться и сразу перейти на ветку командой % git checkout –b Pasha (можно еще пользоваться branch). Проверяем состояние командой % git status.



9. Добавьте в локальный репозиторий проект из 6 задания из лабораторной

работы 1.

Добавил с помощью команды % git add V19.c, после чего проверил состояние командой % git status:



10. Выполните несколько изменений файлов (такими изменениями может

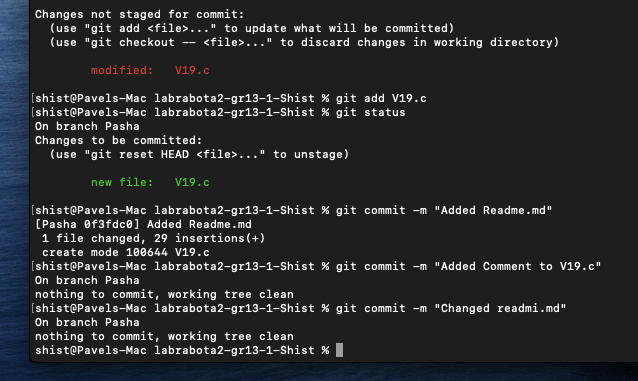
быть улучшение читаемости кода, добавление комментариев и т. д.). После

каждого атомарного изменения выполняйте команду commit в репозиторий в

консоли. К каждому коммиту добавить осмысленный комментарий, чтобы при

необходимости можно было опознать, какие были выполнены изменения.

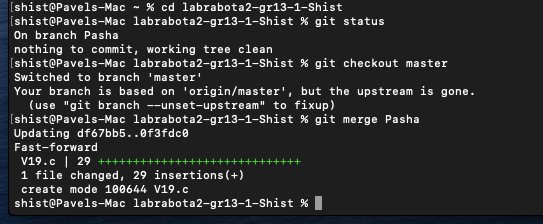
Сначала прописал % git commit –m с комментарием «Добавлен Readme.md». После этого прописал % git commit –m с комментарием «Добавлен комментарий в файле V19.c». Затем прописал % git commit –m с комментарием «Изменён файл readmi.md».



11. После внесения изменений слейте ветку Development (у меня это будет ветка Pasha) и главную ветку

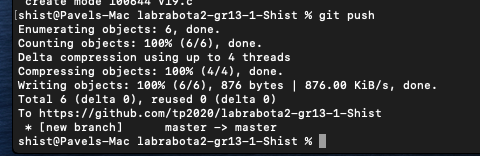
master.

Для этого я, во-первых, захожу в папку командой % cd labrabota2-gr-13-1-Shist. Затем проверяю статус командой % git status. Командой % git checkout master переходим в ветку ‘master’. Сливаем ветку master с веткой Pasha.



12.Отправьте изменений во внешний репозиторий.

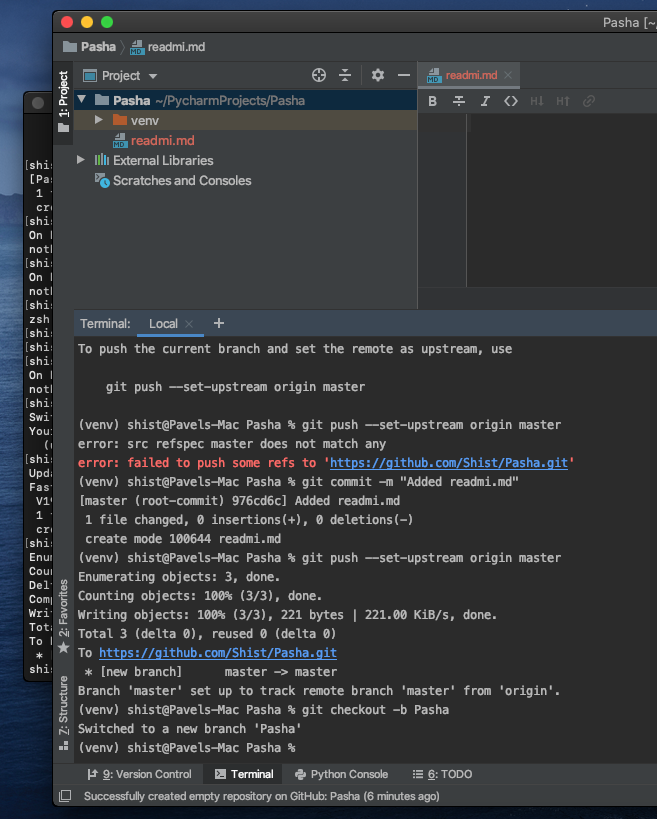
Любые изменения во внешний репозиторий вносим командой % git push.



Сделал пункты 13-19 в среде PyCharm (для этого предварительно установил ее на Mac OS).

15. Создайте ветку Live (у меня это будет ветка Pasha) и перейдите в неё.

Сначала делаю коммит командой % git commit –m "Added readmi.md". Создал репозиторий и пропушил туда readmi.md файл командой % git push --set-upstream origin master. Перехожу в ветку Pasha командой % git checkout –b Pasha.



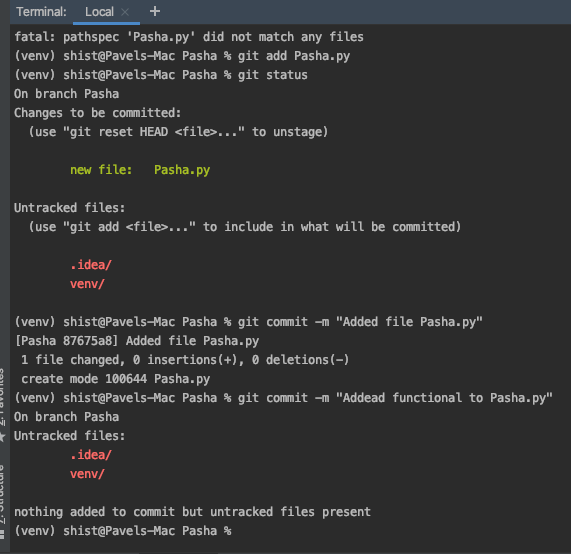
16. Внесите изменения в проект (добавьте новую функциональность) (см. п.

6) и после каждого атомарного изменения передавайте коммит в репозиторий

в SourceTree/SmartGit/GitKraken. К каждому коммиту добавлять

комментарий.

Добавил питоновский файл Pasha.py командой % git add Pasha.py. затем проверил состояние командой % git status. Затем сделал коммит комадной % git commit –m с комментарием «Добавлен файл Pasha.py». Затем сделал коммит командой % git commit –m с комментарием «Добавлен функционал в файл Pasha.py».

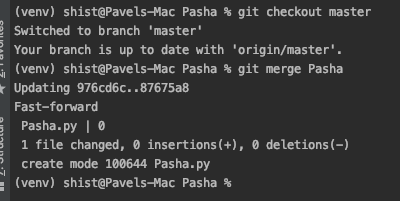


17. После внесения изменений слейте ветку Live и главную ветку master.

Проиллюстрируйте слияние двух веток в отчёте в

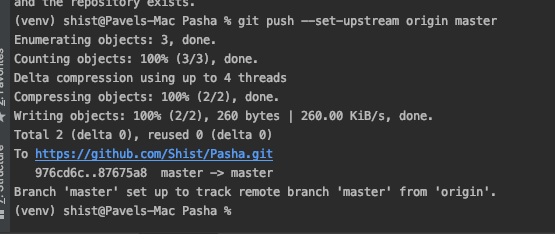
SourceTree/SmartGit/GitKraken.

Перехожу в главную ветку master командой % git checkout master. Затем сливаю ее с веткой Pasha с помощью команды % git merge Pasha.



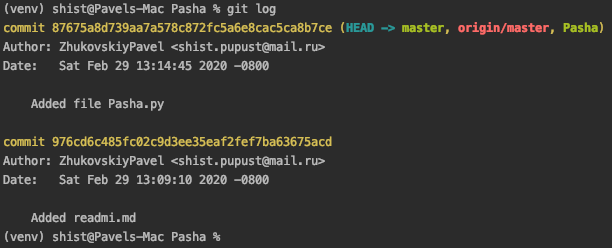
18.Отправьте изменения во внешний репозиторий.

Добавляю изменения во внешний репозиторий командой % git push –set-upstream origin master.



19.В отчёте проиллюстрируйте историю коммитов, работы с ветками.

Для этого прописываю команду % git log.



**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое «командная оболочка» / «командный интерпретатор»?

*Командная оболочка* – отдельный программный продукт, обеспечивающий прямую связь между пользователем и операционной системой. Текстовый пользовательский интерфейс командной строки предоставляет среду, в которой выполняются приложения и служебные программы с текстовым интерфейсом.   
*Командный интерпретатор* –компьютерная программа, часть операционной системы, обеспечивающая базовые возможности управления компьютером посредством интерактивного ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов.

2. Для каких задач предназначено приложение Terminal в macOS?

Приложение Terminal предназначено для выполнения консольных задач, а также для установки программ через brew. Отмечу, что в основном терминал используют опытные пользователи, чтобы через sudo выполнять различные операции.

3. Приведите полный путь к домашней папке пользователя, под которым вы

авторизованы в системе. Привести пример команды, позволяющей

определить путь к домашнему (текущему) каталогу в консоли.

Pwd – приводит до текущего.

~ – приводит до домашнего каталога

4. Какой командой можем вывести содержимое каталога системных

конфигурационных файлов?

Командой ls

5. С помощью какой команды, можем изменить командный интерпретатор с zsh

на bash?

/bin/bash /bin/zsh

6. В каких случаях выполняется инициализация репозитория? Какой командой?

git init

7. Что такое коммит?

Коммит - это команда, при выполнении которой данные, которые находятся в index'е переносятся в репозиторий и устанавливаются соответствующие ссылки на эти данные.

8. Почему перед созданием новой ветки важно создать коммит?

Создать коммит перед созданим новой ветки важно потому, что нам нужно, чтобы новая ветка проинициализировалась.

9. В каких состояниях может находиться файл в репозитории? Как происходит

изменение состояния файла?

Файл может находиться в одном из трёх состояний: зафиксированном, измененном и подготовленном.

10.Что такое ветка?

Ветка - полная копия проекта, изменяя которую вы никак не влияете на проект.

11.Какое условие является обязательным для формирования ветки master

после клонирования репозитория или его инициализации?

--set-upstream origin master, set-upstream чтобы создать эту ветку в нашем репозитории.

12.Что такое HEAD?

HEAD - это указатель на текущую ветку.

13.Способы создания веток.

Git --set-upstream origin Test

Git checkout –b Live

Git bench Live

14.Как узнать текущую ветку?

git branch

15.Как переключаться между ветками?

git checkout [name]

16.С помощью какой команды можно слить две ветки?

git merge [name]

17.Как подключить к инициализированному (созданному) локальному

репозиторию внешний репозиторий? Какой командой?

git remote set-url origin url-нового-репозитория

18.Какая команда позволяет отменить последний коммит?

Git revert, reset

19.Для чего предназначена команда rebase?

Rebase служит для перемещения. Собираются файлы в коммиты на одной ветке, и применяются к другой.

20.В каких случаях требуется создание форка? Приведите способы создания

форка, включая консольную команду.

Создание форка - это клонирование чужого репозитория. Можно осуществить командами:

git clone [github repo]

cd [github repo]

git fork