**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Кафедра технологии программирования

**Харченко Роман Кириллович**

**РАБОТА В КОМАНДНОЙ СТРОКЕ И С GIT-РЕПОЗИТОРИЕМ**

Отчет по лабораторной работе №2

“Технологии программирования”

студента 2 курса 12 группы

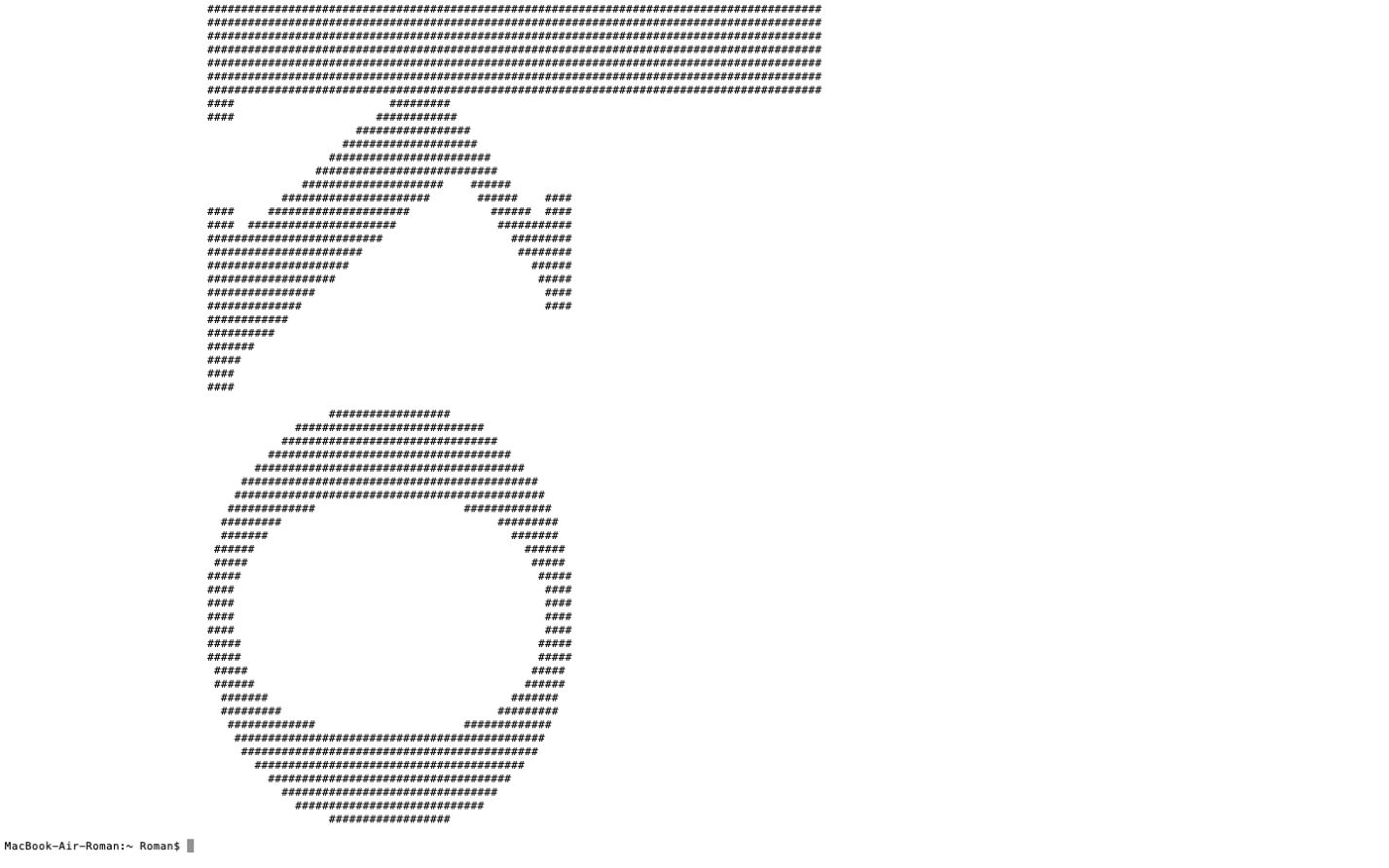
**Преподаватель**

Белый А. А.

**Минск 2020**

**Задание №1**

**1.1 ( echo текст, banner текст )**



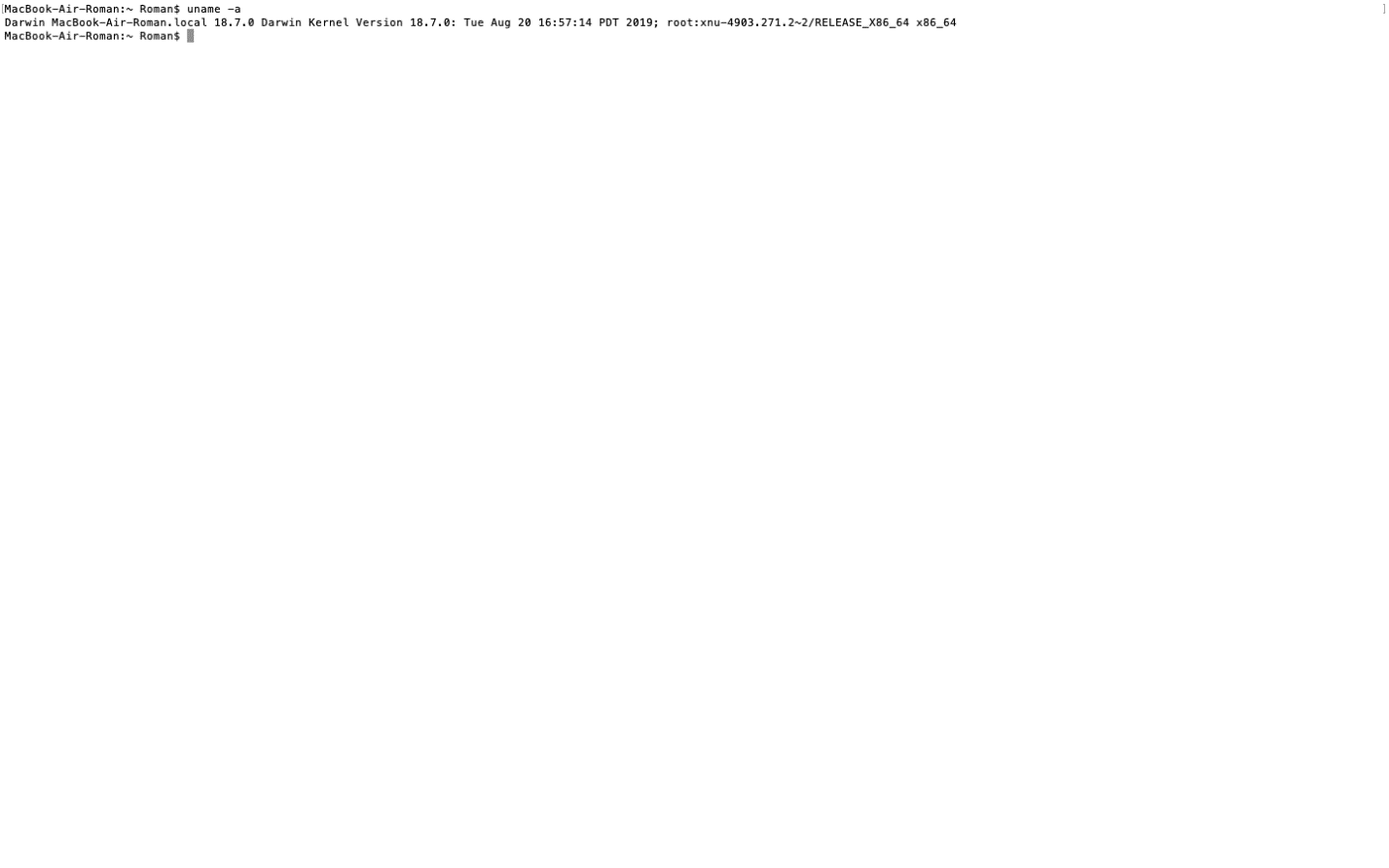
**1.2**

**Выполнено**

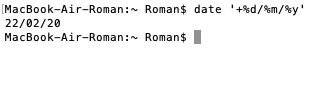
**1.3**

<http://manpages.ubuntu.com/manpages/precise/ru/man2/uname.2.html>

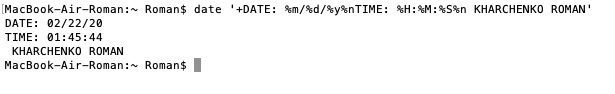
**1.4 ( подробнее по ссылке выше )**

****

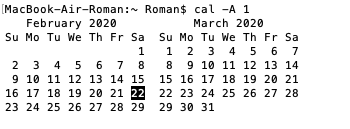
**1.5 ( аналогично для всех, например %m/%d/%y и тд )**

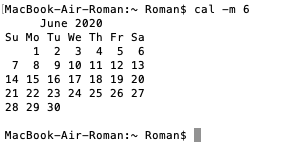
****

**1.6**

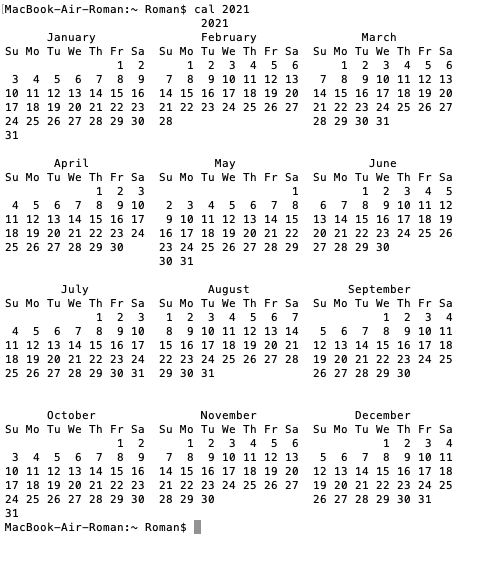
****

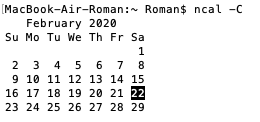
**1.7 ( cal –A кол-во месяцев, cal –m номер месяца для вывода )**

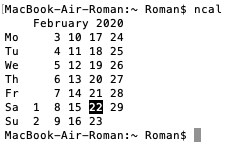
****

****

**1.8 ( ncal с пон-ка, ncal –C с воскресенья, cal год – вывод года )**

****

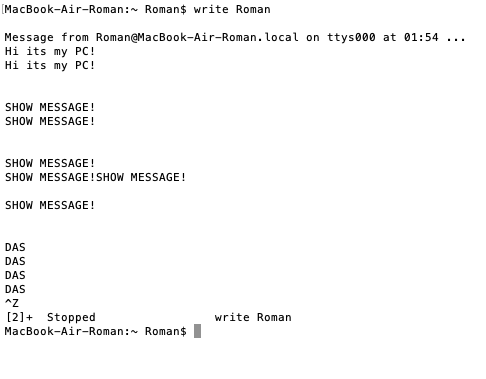
****

****

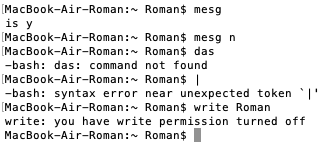
**1.9**

**Cal -j 2020 номер\_дня в\_году**

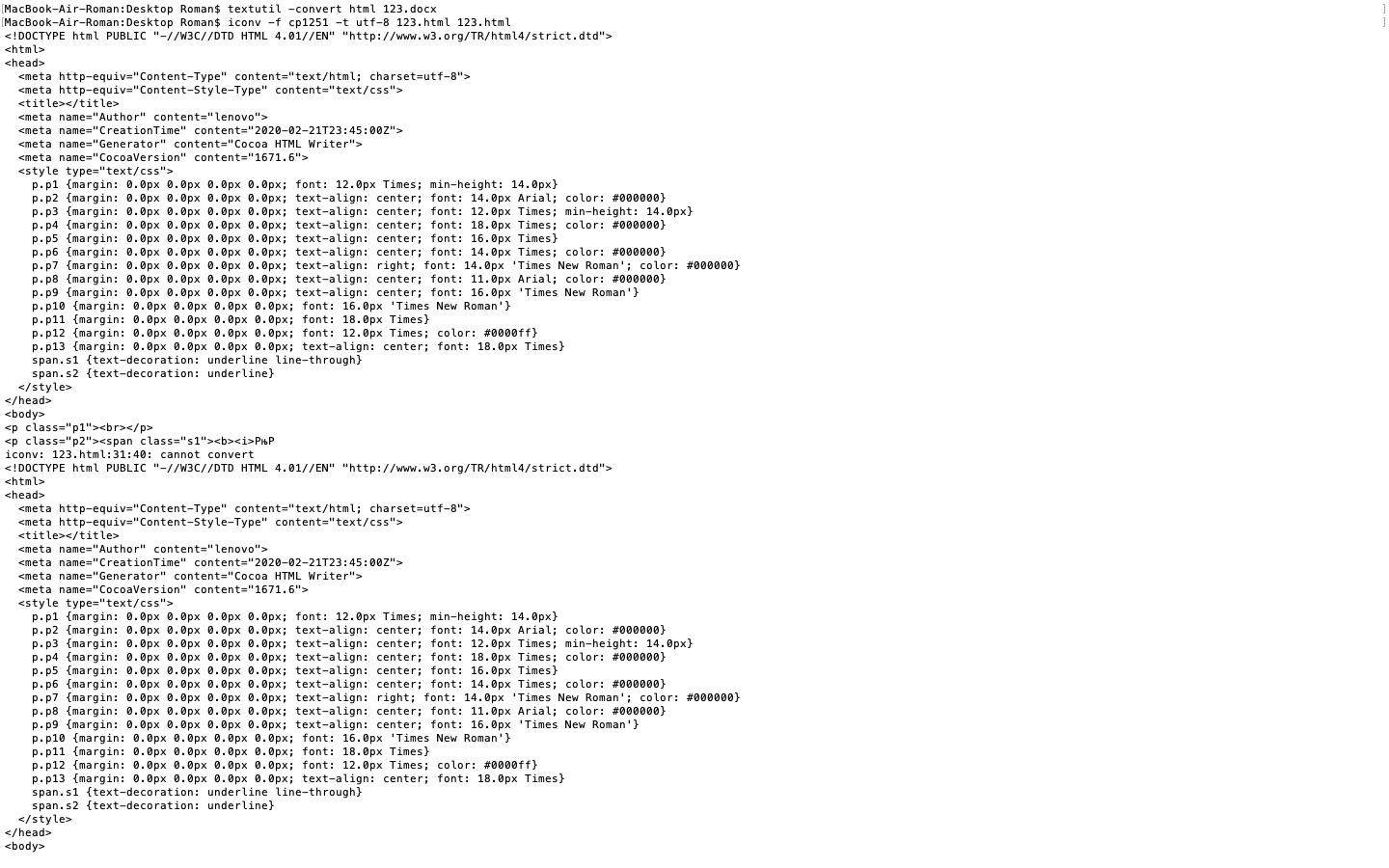
**1.11 ( write имя\_пользователя )**

****

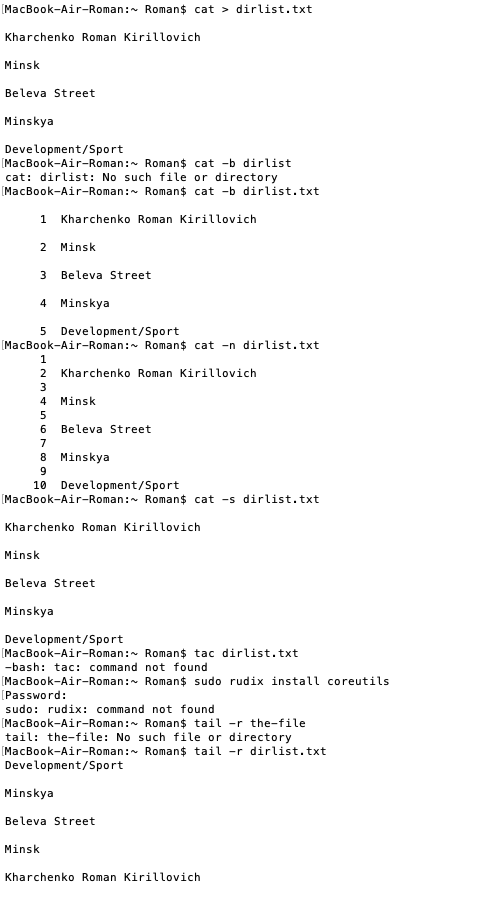
**1.12 ( terminal не дает отправить сообщение при mesg n )**

****

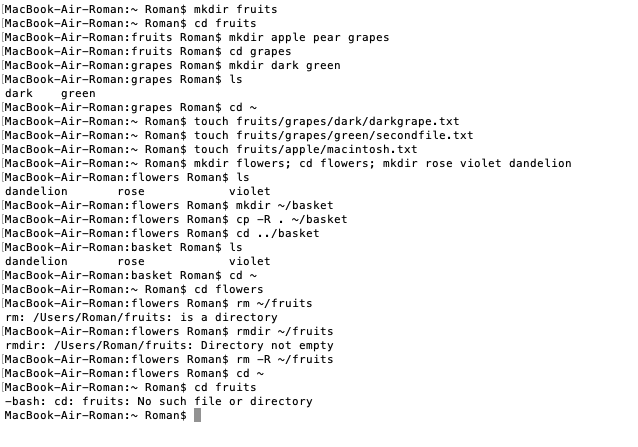
**1.13 - 1.15 ( mail, textutil, iconv )**

****

**Задание №2**

**2.1 – 2.2 ( tac не работает, поэтому используем tail -r)**

**2.3 – 2.9 ( Все команды были изучены в Лаб №1 )**

****

**2.9 - 2.25 ( также команды находятся в мануале, сократил для компактности отчета )**

**2.10**

touch a{1..100}

rm a\*{1,3,5,7,9}

**2.11 ( mkdir создает папку )**

mkdir –p /A/B/C/

mkdir -p A/B/

mkdir A/B/C/D/

mkdir A/E/

rm -rf ~/A/\*

**2.12**

grep june dirlist.txt > grep\_june\_month.txt

cat grep\_june\_month.txt

**2.13**

grep -v june dirlist.txt > grep\_another\_month.txt

cat grep\_another\_month.txt

january

february

march

april

may

july

september

october

november

december

**2.14**

mkdir grep &&mv grep\_\* grep/

cd grep &&ls

**2.15**

grep root mac\_os\_lab/

**2.16**

Locate - быстрее, find - больше возможностей

**2.17**

find / -mmin -60

**2.18**

rm -rf manyfiles

**2.19**

c=$((117+25))

echo $c

142

c=$((86+124))

echo $c

210

c=$((157%12))

echo $c

c=$((45225/45\*100))

echo $c

100500

**2.20**

Используем команду vi list.txt

**2.21**

cat list.txt | head -n 10

cat group.txt | tail -n 15

**2.22**

expand -t 1 list.txt

**2.23**

diff list.txt dirlist.txt

diff3 ( 3, то есть сравнение 3-х файлов )

**2.24**

cmp - сравнивает два файла байт за байтом

comm - сравнивает построчно два файла на совпадение строк и на уникальные строки

**2.25**

split -l 5 list.txt

**Задание №3**

**3.1**

Сделал

**3.2**

git config --global user.name “Roman Kharchenko”

git config --global user.email [karlmarten2001@icloud.com](mailto:karlmarten2001@icloud.com)

**3.3**

Сделал

**3.4**

Сделал

**3.5**

Сделал

**3.6**

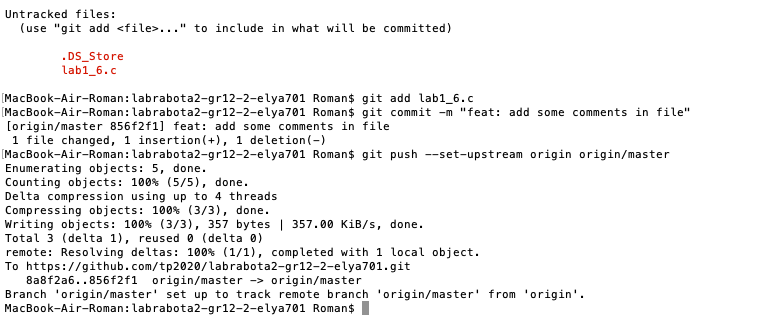
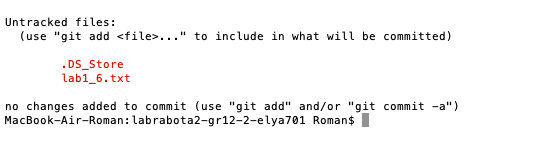
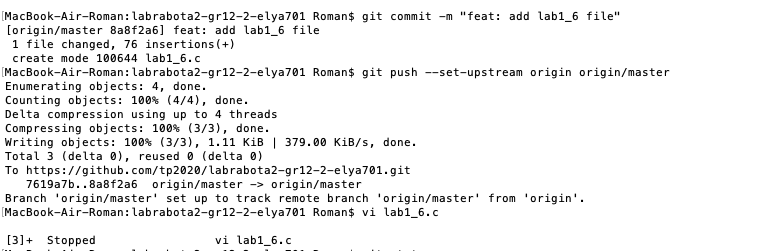
git clone ( link )

**3.7 - 3.12**

**Все используемые команды указаны в мануале,**

**а также визуализированы на**

**скринах слайдах ниже**

****

**3.13 - 3.19**

Использовал xcode с предустановленным git-клиентом.  
 **Более подробный манул по ссылке:**

[**https://www.youtube.com/watch?v=R0vhZIOZPzc&t=795s**](https://www.youtube.com/watch?v=R0vhZIOZPzc&t=795s)

**( на примере xcode 10.4, swift 4 )**

**Контрольные вопросы:**

1) Командная оболочка — это отдельный программный продукт,

который обеспечивает прямую связь между пользователем и

операционной системой.

2) Terminal предназначено для консольных задач и установка программ

3) Pwd

4) ls

5) /bin/bash /bin/zsh

6) Командй git init инициализация, в случаях когда мы только проект

создали новый и хотим его выложить в наш репозиторий.

7) Коммит - описание.

8) Для инициализации новой ветки.

9) Три состояния:

зафиксированный, измененный и подготовленный.

10) Ветка - полная копия проекта, изменяя которую вы никак не влияете

на сам проект.

11) --set-upstream origin master, set-upstream чтобы создать эту ветку в

нашем репозиторий

12) HEAD - это указатель на текущую ветку

13) Git --set-upstream origin our\_branch\_name ( для создания ветки )

14) git branch

15) git checkout [name]

16) git merge [name]

17) git remote set-url origin url-нового-репозитория

18) Git revert, reset

19) Rebase служит для перемещения.

20) Создание форка - это клонирование чужого репозитория.

Можно выполнить с помощью команд:

git clone [github repo]

cd [github repo]

git fork