|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 5 | | |
| по дисциплине «Операционные системы, среды и оболочки» | | |
| **Системы контроля версий** | | |
|  | | |
|  | Группа ПМ-21 |  |
| Бригада 11 | Егупов Иван |
|  | Порсин Данил |
| Преподаватели | Кобылянский В.Г. Сивак М.А |
|  |  |
| Новосибирск, 2024 | | |

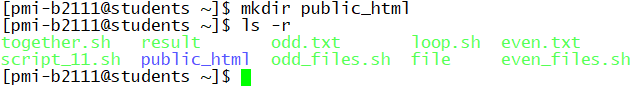
**Цель работы:** Целью выполнения работы является изучение систем контроля версий git и GitLab, а также подготовка репозитория для выполнения лабораторной работы № 6

**Часть 1:**

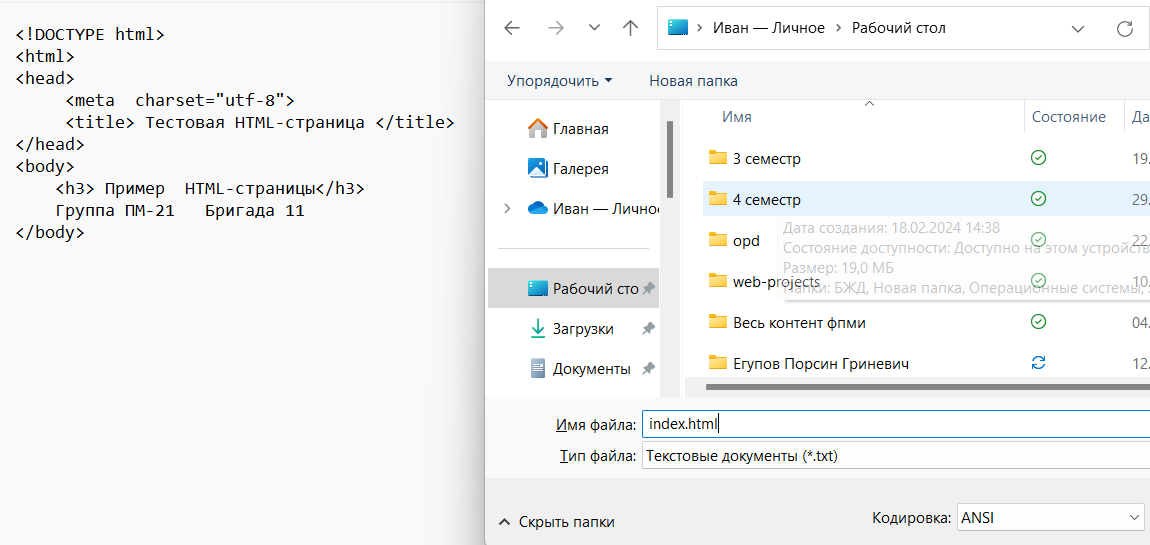
1. Выполнить все задания раздела 2.4 по созданию и редактированию файла **index.html.** После каждого внесения изменений необходимо после проверки корректности работы файла сохранить его обновленную версию в локальном репозитории с соответствующим комментарием. Результаты выполнения каждого задания занести в отчет.
2. Подключаемся через **putty** к серверу, указав его имя или IP-адрес и делаем доступным домашний каталог для всех пользователей командой **chmod -R 755 ~**



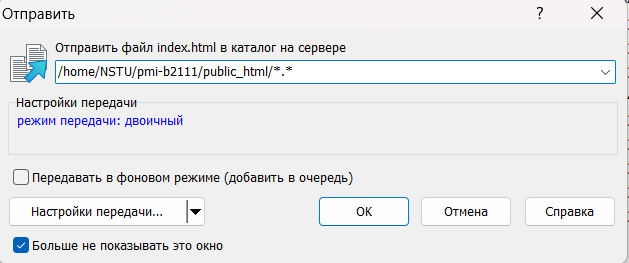
1. Создаем в домашнем каталоге сервера рабочий каталог с именем **public\_html**

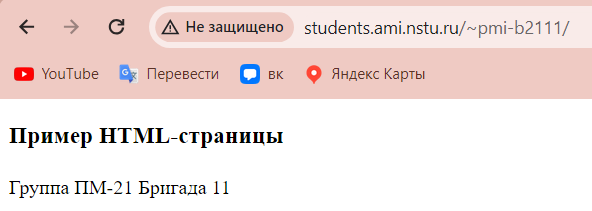


1. С помощью Блокнота создаем на Windows-компьютере файл **index.html**, содержащий сведения о наименовании группы и номере бригады, при сохранении файла указываем кодировку ANSI.

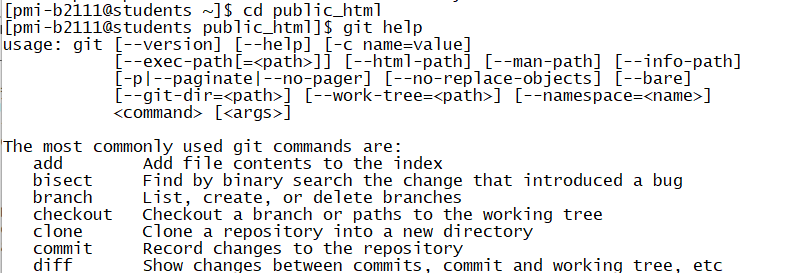


1. Выполняем отладку файла в среде Windows, после чего с помощью утилиты WinSCP копируем его в каталог **public\_html** сервера и проверяем корректность работы страницы в браузере по адресу:





5.Переходим в каталог **public\_html** сервераи выводим список основных команд системы git.



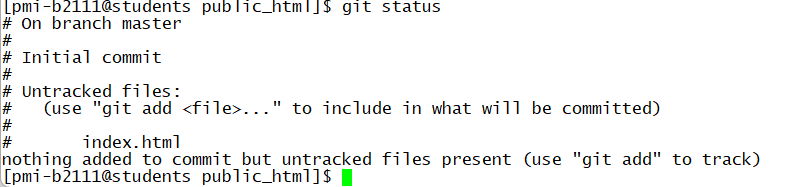
6.Создаем локальный репозиторий git.



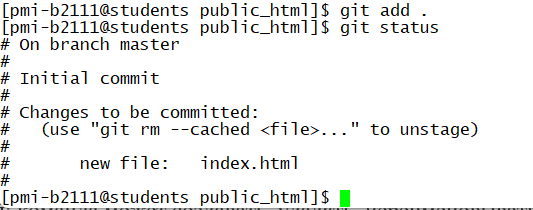
7. Устанавливаем имя и e-mail пользователя git по умолчанию.



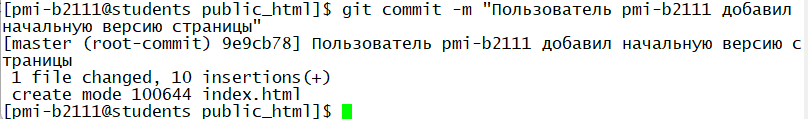
8. Выводим сведения о состоянии файлов в репозитории

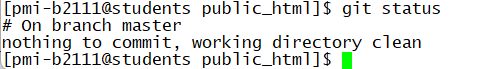


9. Добавляем файл **index.html** в репозиторий путем указания обобщенного имени рабочего каталога (индексируем файл) и повторно выводим сведения о состоянии файлов

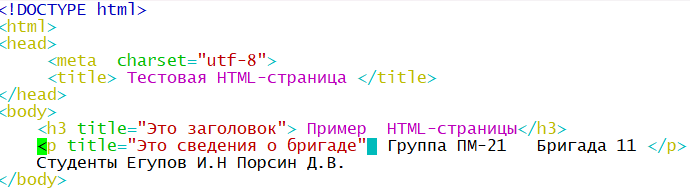
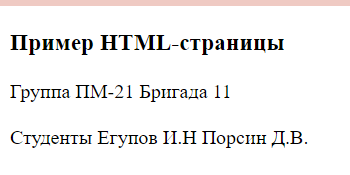


10. Выполняем начальный коммит с добавлением комментария.



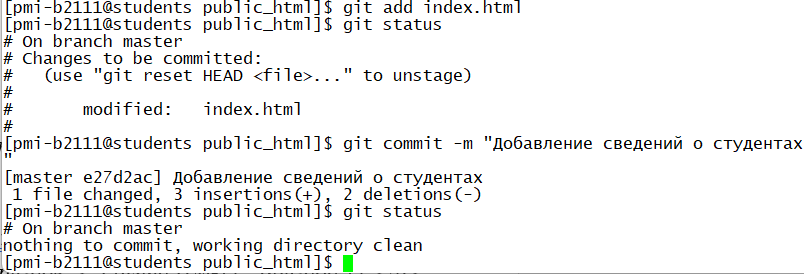


11. Вносим изменения в файл **index.html**: группу и номер бригады оформляем в виде абзаца, добавляем сведения о студентах бригады и всплывающие подсказки. Внесение изменений в исходный текст программы можно проводить с помощью редактора **vi,** основные сведения по которому приведены в приложении 1

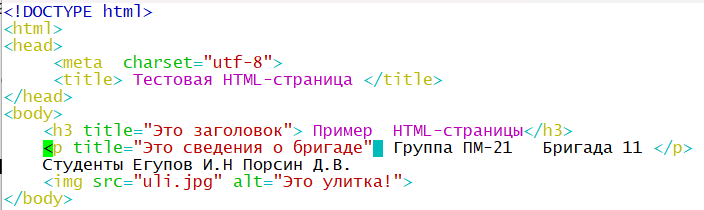
 

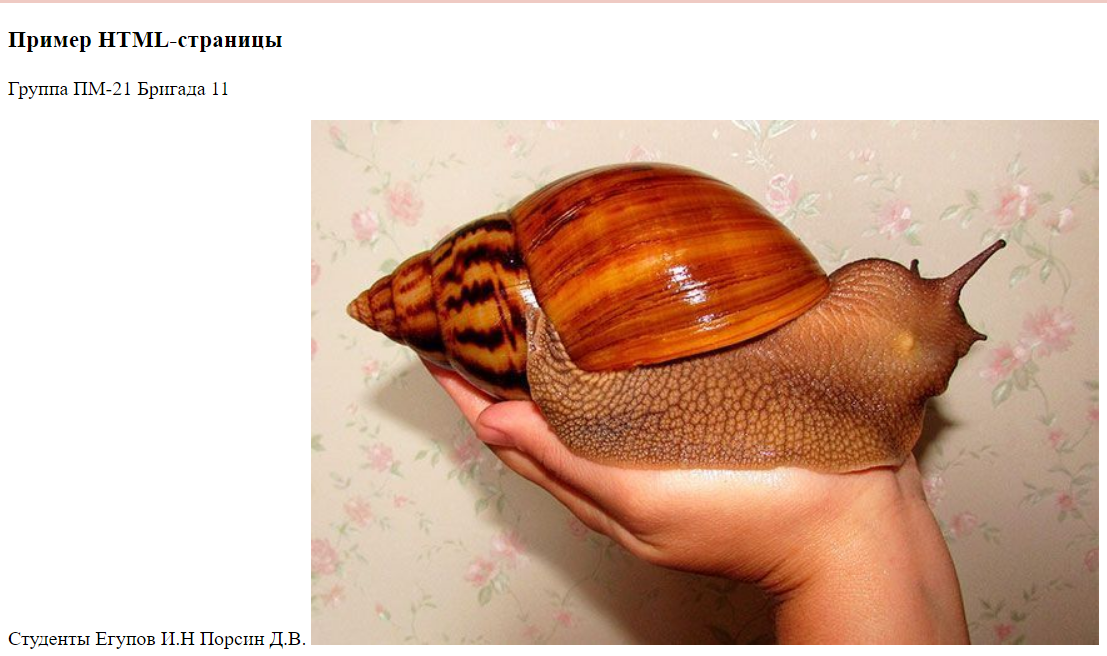
12. После отладки страницы добавляем ее в репозиторий и выполняем коммит с соответствующим комментарием:

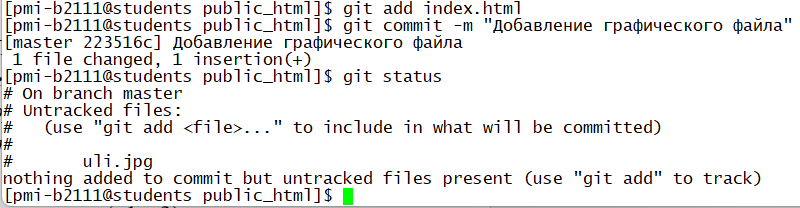
*git commit –m “Добавление сведений о студентах”*



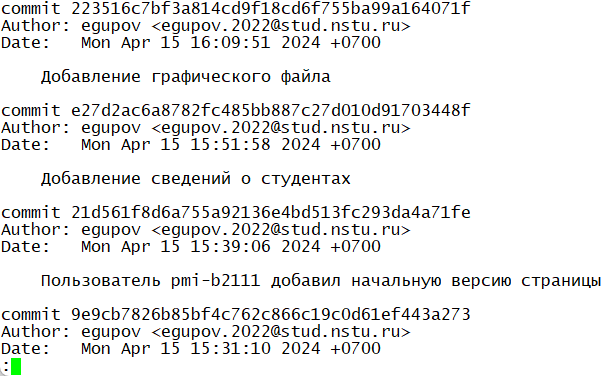
13. Повторно изменяем файл **index.html ,** добавляя вывод любого графического файла, и выполняем коммит с комментарием аналогично п.12:





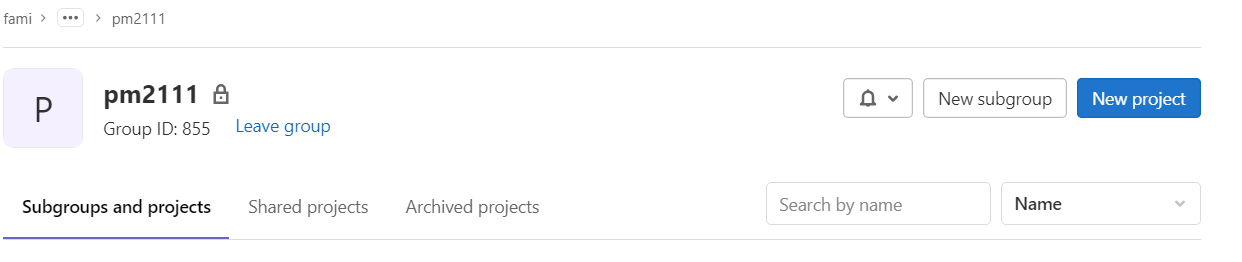


14. Просматриваем журнал и назначаем тэги (v1..v3) каждому коммиту:



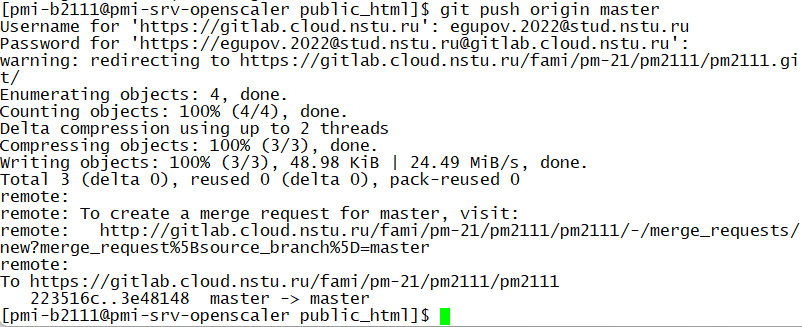
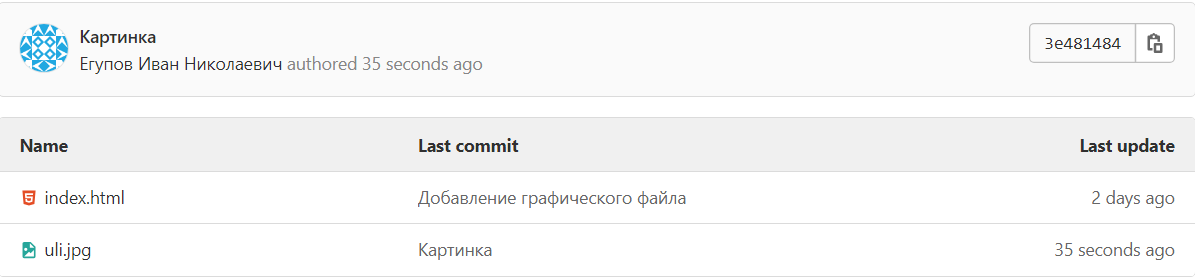


1. Зарегистрироваться в GitLab НГТУ и создать в своей подгруппе (например, pmi-21) проект с именем, содержащим номер вашей учебной группы и номер бригады (например, pmi-2101). Также проект может создаваться преподавателем.



3)Загрузить в свой проект все файлы из локального репозитория каталога **public\_html**

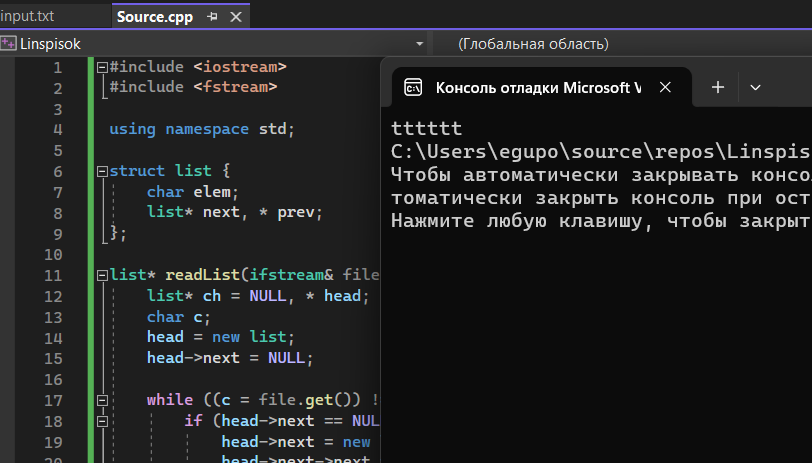
*git push –u origin --all*

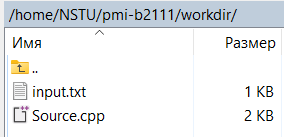
**Часть 2**

1. Проверить работоспособность программ, разработанных по курсам «Структуры данных и алгоритмы» (для групп ПМ) или « Информационные технологии и основы программирования» (для групп ПМИ) в среде Windows. Для каждой бригады достаточно иметь одну программу.

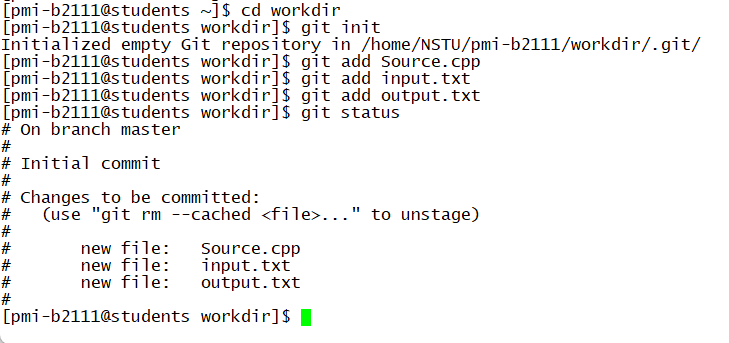
Задание: Дана последовательность символов, оканчивающаяся точкой. Написать подпрограмму, которая в двусвязном кольцевом списке с фиктивным звеном удаляет все символы, у которых равные соседи (первый и последний элемент считать соседями). Используя эту подпрограмму, удалить символы, у которых равные соседи.



1. В домашнем каталоге сервера *pmi-srv-openscaler* создать рабочий каталог **workdir** и скопировать файлы с исходным текстом программы в созданный рабочий каталог с помощью утилиты WinSCP.



1. Перейти в рабочий каталог, создать в нем новый git-репозиторий и добавить в главную ветку (*master)* исходные файлы программы.

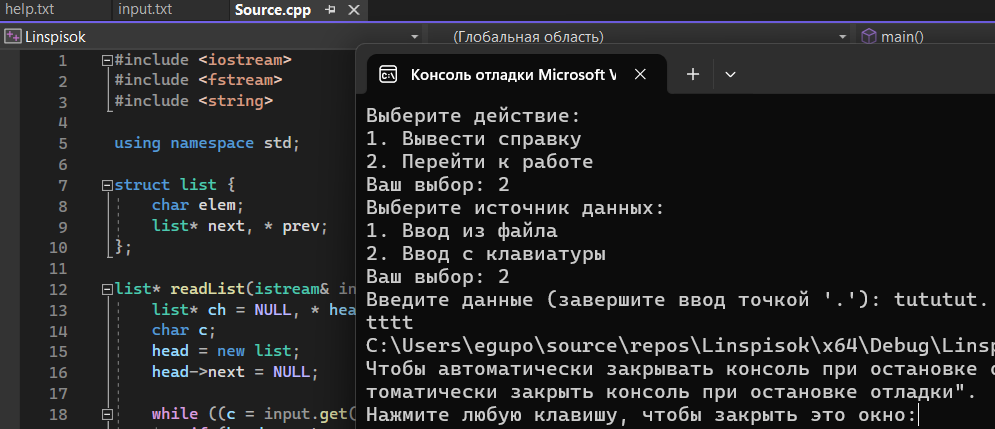


7)Доработать текст программы с учетом следующих требований:

возможность исполнения в среде Linux;

ввод данных с клавиатуры и из файла;

наличие контроля входных данных;  
наличие встроенной справки, содержащей краткое руководство пользователя и сведения о разработчиках; справку оформить в виде отдельного текстового файла.



Текст доработанной программы:

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

char elem;

list\* next, \* prev;

};

list\* readList(istream& input) {

list\* ch = NULL, \* head;

char c;

head = new list;

head->next = NULL;

while ((c = input.get()) != '.') {

if (head->next == NULL) {

head->next = new list;

head->next->next = head->next;

head->next->prev = head->next;

head->next->elem = c;

ch = head->next;

}

else {

ch->next = new list;

ch->next->prev = ch;

ch->next->elem = c;

ch->next->next = head->next;

head->next->prev = ch->next;

ch = ch->next;

}

}

return head;

}

void writeList(list\* head) {

list\* p = head->next;

if (p->next == NULL)

cout << " ";

else {

if (p->next == p)

cout << " ";

else {

do {

cout << p->elem;

p = p->next;

} while (p != head->next);

}

}

}

void deleteElems(list\* head) {

list\* first = head->next, \* p = NULL, \* q = first;

if (first->next == first)

return;

else {

do {

if (q->prev->elem == q->next->elem) {

if (q == first)

first = q->next;

p = q;

q->next->prev = q->prev;

q->prev->next = q->next;

q = q->prev;

delete p;

p = NULL;

}

q = q->next;

} while (q->next != first);

}

head->next = first;

}

void showHelp() {

ifstream helpFile("help.txt");

if (!helpFile.is\_open()) {

cout << "Error open file with help\n";

return;

}

string line;

while (getline(helpFile, line)) {

cout << line << endl;

}

helpFile.close();

}

int main() {

setlocale(0, "");

int choice;

cout << "Select an action:" << endl;

cout << "1. Issue a certificate" << endl;

cout << "2. Go to work" << endl;

cout << "Your select: ";

cin >> choice;

if (choice == 1) {

showHelp();

return 0;

}

else if (choice != 2) {

cout << "Incorrect select\n";

return 1;

}

cout << "Select source of date:" << endl;

cout << "1. Input from file" << endl;

cout << "2. Input from keyboard" << endl;

cout << "Your select: ";

cin >> choice;

list\* qList;

if (choice == 1) {

ifstream read("input.txt");

if (!read.is\_open()) {

cout << "Error open input file\n";

return 1;

}

qList = readList(read);

read.close();

}

else if (choice == 2) {

cout << "Input data (point in the end '.'): ";

qList = readList(cin);

}

else {

cout << "Incorrect input\n";

return 1;

}

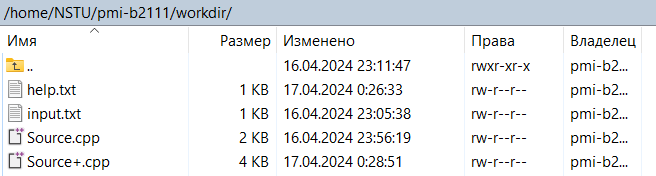
deleteElems(qList);

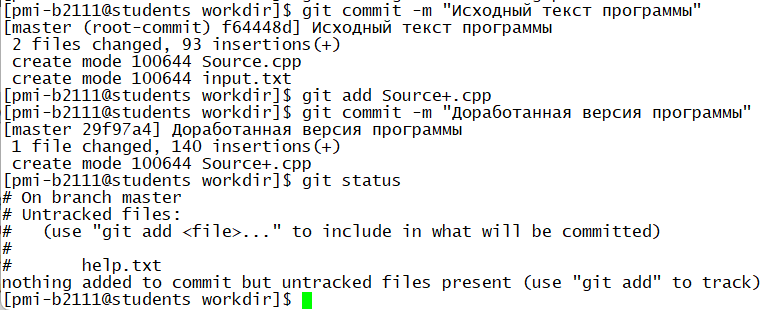
writeList(qList);

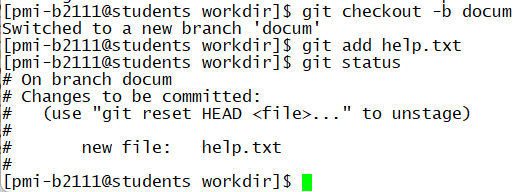
return 0;

}

1. Создать вторую ветку репозитория (*docum*) и сохранить доработанную программу в ветке *master*, а справку – в ветке *docum*.



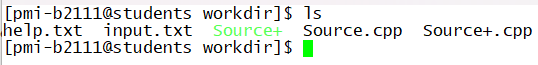




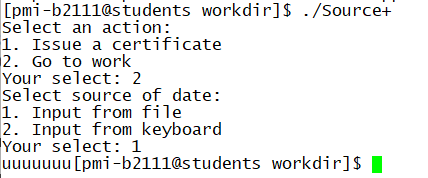
1. Cформировать исполняемый файл программы с помощью компилятора **gcc**:

*gcc -o prog1 prog1.c*



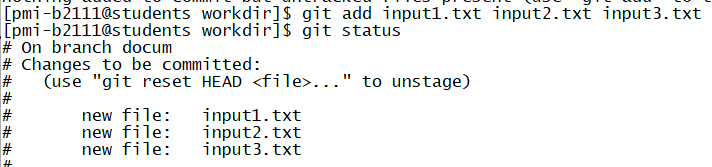


10)При необходимости исправить ошибки, обнаруженные компилятором. Каждую исправленную версию программы или справки сохранять в соответствующей ветке репозитория.



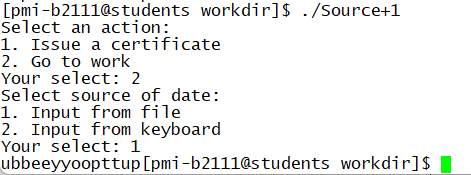
Ошибок в компиляции не обнаружено, программа работает исправно

11)Для проверки работоспособности программ разработать не менее трех файлов с тестовыми наборами входных данных, которые разместить в ветке *docum*. Результаты работы программ на всех тестовых наборах входных данных занести в отчет.

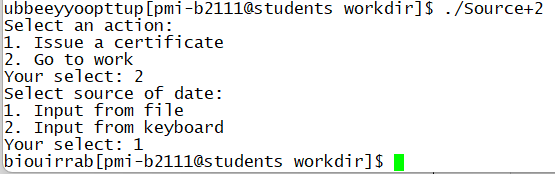


*Ввиду особенности программы, при которой нельзя в автоматическом режиме изменить имя файла с входными данными, мы создали 3 исполняемых файла для каждого набора входных данных*

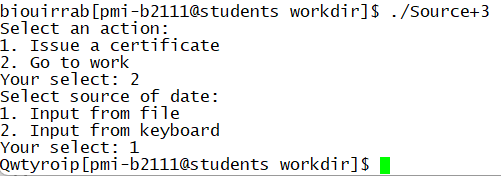
*1 файл:* ubbbereyuyoooptutup.



*2 файл:* abiouirtrab.



1. файл: Qwtyroip.



12) Загрузить все файлы из локального репозитория в облачный репозиторий НГТУ с сохранением всех веток. В разделе Issues проекта GitLab добавить описание задания, например, «поиск строки максимальной длины».

Произведем слияние веток master на локальном и удаленном репозитории



И загрузим обновленную версию на облачный репозиторий

