

Министерство образования и науки РФ  
Новосибирский государственный технический университет

Кафедра ТПИ

Лабораторная работа № 5  
По Операционным системам и компьютерным сетям

Системы контроля версий

Факультет: ПМИ

Группа: ПМ-24

Бригада: 12

Студенты: Герасименко Вадим  
Параскун Иван

Преподаватель: Сивак М. А.

Новосибирск  
2024

## 1. Цель работы

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

## 2. Ход работы

### 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.

- 1.1 Подключаемся к серверу, указав его имя или IP-адрес и делаем доступным домашний каталог для всех пользователей командой **chmod -R 755 ~**

#### Подключение:

```
[vadimgerasimenko@MacBook-Air-Vadim-2 ~ % ssh pmi-b2412@217.71.130.133
```

```
Authorized users only. All activities may be monitored and reported.  
[pmi-b2412@217.71.130.133's password:
```

```
Authorized users only. All activities may be monitored and reported.  
Last login: Sat Apr 20 15:54:18 2024 from 94.180.110.29
```

#### Доступ:

```
[[pmi-b2412@pmi-srv-openscaler ~]$ chmod -R 755 ~  
[pmi-b2412@pmi-srv-openscaler ~]$ █
```

- 1.2 Создаем в домашнем каталоге сервера рабочий каталог с именем **public\_html**

```
[pmi-b2412@pmi-srv-openscaler ~]$ mkdir public_html  
[pmi-b2412@pmi-srv-openscaler ~]$ ls  
abc_12      history_12  online      reports      trash_12  
hard_abc1   monthly     public_html  soft_abc1
```

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

```
[vadimgerasimenko@MacBook-Air-Vadim-2 ~ % cat index.html  
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8">  
  <title> Тестовая HTML-страница </title>  
</head>  
<body>i  
  <h3> Пример  HTML-страницы</h3>  
  Группа ПМ-24  Бригада 12  
</body>  
vadimgerasimenko@MacBook-Air-Vadim-2 ~ % █
```

- 1.4 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: [http://имя\\_илиIP-адрес\\_сервера/~ваш\\_бригадный\\_логин](http://имя_илиIP-адрес_сервера/~ваш_бригадный_логин)

Убедились в работоспособности:

```
[vadimgerasimenko@MacBook-Air-Vadim-2 ~ % open index.html  
vadimgerasimenko@MacBook-Air-Vadim-2 ~ %
```



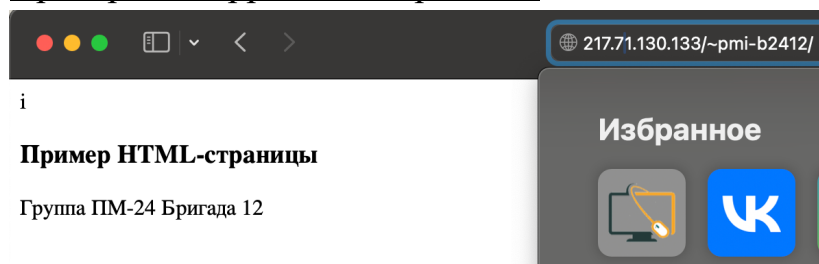
**Пример HTML-страницы**

Группа ПМ-24 Бригада 12

Загрузили на сервер:

```
[[pmi-b2412@pmi-srv-openscaler public_html]$ ls  
index.html  
[[pmi-b2412@pmi-srv-openscaler public_html]$ cat index.html  
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8">  
  <title> Тестовая HTML-страница </title>  
</head>  
<body>i  
  <h3> Пример HTML-страницы</h3>  
  Группа ПМ-24 Бригада 12  
</body>
```

Проверяем корректность работы:



- 1.5 Переходим в каталог **public\_html** сервера и выводим список основных команд системы `git --help`

```

[pmi-b2412@pmi-srv-openscaler public_html]$ git --help
использование: git [--version] [--help] [-C <path>] [-c <name>=<value>]
               [--exec-path=<path>] [--html-path] [--man-path] [--info-path]
               [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
               [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
               [--super-prefix=<path>] [--config-env=<name>=<envvar>]
               <command> [<args>]

Стандартные команды Git используемые в различных ситуациях:

создание рабочей области (смотрите также: git help tutorial)
clone      Клонирование репозитория в новый каталог
init       Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: git help everyday)
add        Добавление содержимого файла в индекс
mv         Перемещение или переименование файла, каталога или символической ссылки
restore    Restore working tree files
rm         Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: git help revisions)
bisect     Выполнение двоичного поиска коммита, который вносит ошибку
diff       Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
grep       Вывод строк, соответствующих шаблону
log        Вывод истории коммитов
show       Вывод различных типов объектов
status     Вывод состояния рабочего каталога

выращивание, отметка и настройка вашей общей истории
branch     Вывод списка, создание или удаление веток
commit     Запись изменений в репозиторий
merge      Объединение одной или нескольких историй разработки вместе
rebase     Повторное применение коммитов над верхушкой другой ветки
reset      Сброс текущего состояния HEAD на указанное состояние
switch     Switch branches
tag        Создание, вывод списка, удаление или проверка метки, подписанной с помощью GPG

совместная работа (смотрите также: git help workflows)
fetch      Загрузка объектов и ссылок из другого репозитория
pull       Извлечение изменений и объединение с другим репозиторием или локальной веткой
push       Обновление внешних ссылок и связанных объектов

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
[pmi-b2412@pmi-srv-openscaler public_html]$

```

## 1.6 Создаем локальный репозиторий git. *git init*

```

[pmi-b2412@pmi-srv-openscaler public_html]$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/pmi-b2412/public_html/.git/
[pmi-b2412@pmi-srv-openscaler public_html]$

```

## 1.7 Устанавливаем имя и e-mail пользователя git по умолчанию.

```

[pmi-b2412@pmi-srv-openscaler public_html]$ git config --global user.name "Gerasimenko"
[pmi-b2412@pmi-srv-openscaler public_html]$ git config --global user.email v.gerasimenko.2022@stud.nstu.ru

```

## 1.8 Выводим сведения о состоянии файлов в репозитории. *git status*

```

[pmi-b2412@pmi-srv-openscaler public_html]$ git status
На ветке master

```

Еще нет коммитов

Неотслеживаемые файлы:

(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)

```

index.html

```

ничего не добавлено в коммит, но есть неотслеживаемые файлы (используйте «git add», чтобы отслеживать их)

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

```
[pmi-b2412@pmi-srv-openscaler public_html]$ git add .
[pmi-b2412@pmi-srv-openscaler public_html]$ git status
На ветке master
```

Еще нет коммитов

Изменения, которые будут включены в коммит:  
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)  
новый файл: index.html

## 1.10 Выполняем начальный коммит с добавлением комментария.

```
[pmi-b2412@pmi-srv-openscaler public_html]$ git commit -m "Пользователь pm-2412 добавил начальную версию страницы"
[master (корневой коммит) f2be184] Пользователь pm-2412 добавил начальную версию страницы
1 file changed, 10 insertions(+)
create mode 100644 index.html
```

## 1.11 Вносим изменения в файл **index.html**: группу и номер бригады оформляем в виде абзаца, добавляем сведения о студентах бригады и всплывающие под-сказки.

### Файл:

```
[pmi-b2412@pmi-srv-openscaler public_html]$ cat index.html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Тестовая HTML-страница </title>
</head>
<body>
  <h3 title="Это заголовок" > Пример HTML-страницы</h3>
  <p title="Это сведения о бригаде"> Группа ПМ-24 Бригада 12 </p>
  Студенты Герасименко В.Э., Параскун И. Г.
</body>
```

### Сайт:

#### Пример HTML-страницы

Группа ПМ-24 Бригада 12

Студенты Герасименко В.Э., Параскун И. Г.

## 1.12 После отладки страницы добавляем ее в репозиторий и выполняем коммит с соответствующим комментарием:

```
[pmi-b2412@pmi-srv-openscaler public_html]$ git add index.html
[pmi-b2412@pmi-srv-openscaler public_html]$ git commit -m "Добавление сведений о студентах"
[master 9217fd6] Добавление сведений о студентах
1 file changed, 5 insertions(+), 4 deletions(-)
```

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

### Файл:

```
[pmi-b2412@pmi-srv-openscaler public_html]$ cat index.html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Тестовая HTML-страница </title>
</head>
<body>
  <h3 title="Это заголовок" > Пример HTML-страницы</h3>
  <p title="Это сведения о бригаде"> Группа ПМ-24 Бригада 12 </p>
  Студенты Герасименко В.Э., Параскун И. Г.
  
</body>
```

### Сайт:

#### Пример HTML-страницы

Группа ПМ-24 Бригада 12

Студенты Герасименко В.Э., Параскун И. Г.



```
[pmi-b2412@pmi-srv-openscaler public_html]$ git add index.html
[pmi-b2412@pmi-srv-openscaler public_html]$ git commit -m "Добавление изображения"
[master 6c031ff] Добавление изображения
1 file changed, 1 insertion(+)
```

### 1.14 Просматриваем журнал и назначаем тэги каждому коммиту:

```
[pmi-b2412@pmi-srv-openscaler public_html]$ git tag v1 f2be18418fe881cb01e6db661a9db8f374658eb8
[pmi-b2412@pmi-srv-openscaler public_html]$ git tag v2 9217fd6365716a7693b424b05fede0f86bc594b2
[pmi-b2412@pmi-srv-openscaler public_html]$ git tag v3 6c031ff6237856bf10eb7898a5ec680f732366c1
[pmi-b2412@pmi-srv-openscaler public_html]$ git log
commit 6c031ff6237856bf10eb7898a5ec680f732366c1 (HEAD -> master, tag: v3)
Author: Gerasimenko <v.gerasimenko.2022@stud.nstu.ru>
Date: Sat Apr 20 18:22:46 2024 +0700
```

Добавление изображения

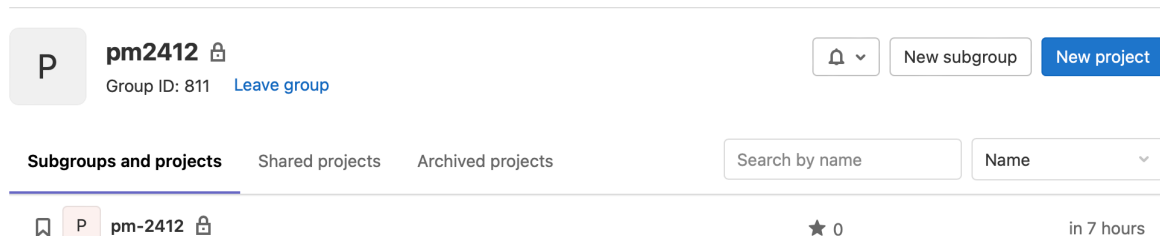
```
commit 9217fd6365716a7693b424b05fede0f86bc594b2 (tag: v2)
Author: Gerasimenko <v.gerasimenko.2022@stud.nstu.ru>
Date: Sat Apr 20 18:20:10 2024 +0700
```

Добавление сведений о студентах

```
commit f2be18418fe881cb01e6db661a9db8f374658eb8 (tag: v1)
Author: Gerasimenko <v.gerasimenko.2022@stud.nstu.ru>
Date: Sat Apr 20 18:18:12 2024 +0700
```

Пользователь pm-2412 добавил начальную версию страницы

- 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.



- Загрузить в свой проект все файлы из локального репозитория каталога **public\_html** `git push -u origin --all` Имена проектов имеют следующий вид (на примере бригады № 1 группы ПМИ-21): <https://gitlab.cloud.nstu.ru/fami/pmi-21/pmi-2101.git>

```
[pmi-b2412@pmi-srv-openscaler public_html]$ git remote add origin https://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm-2412
[pmi-b2412@pmi-srv-openscaler public_html]$ git push -u origin --all
Username for 'https://gitlab.cloud.nstu.ru': v.gerasimenko.2022@stud.nstu.ru
Password for 'https://v.gerasimenko.2022@stud.nstu.ru@gitlab.cloud.nstu.ru':
warning: переадресация на https://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm-2412.git/
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (9/9), 1.16 КиБ | 1.16 МиБ/с, готово.
Всего 9 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: To create a merge request for master, visit:
remote: http://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm-2412/-/merge_requests/new?merge_request%5Bsource_branch%5D=master
remote:
To https://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm-2412
* [new branch] master -> master
branch 'master' set up to track 'origin/master'.
[pmi-b2412@pmi-srv-openscaler public_html]$
```

### Вывод тэгов:

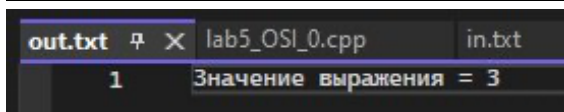
```
[pmi-b2412@pmi-srv-openscaler public_html]$ git tag
v1
v2
v3
```

## Часть 2.

4. 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.

### Программа работоспособна:

```
"lab5_OSI.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll".  
"lab5_OSI.exe" (Win32). Загружено "C:\Windows\System32\msvcrt.dll".  
"lab5_OSI.exe" (Win32). Загружено "C:\Windows\System32\rpcrt4.dll".  
Поток 0x1278 завершился с кодом 0 (0x0).  
Поток 0x2650 завершился с кодом 0 (0x0).  
Программа "[4720] lab5_OSI.exe" завершилась с кодом 0 (0x0).
```



5. 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.

### Создание каталога:

```
[[pmi-b2412@pmi-srv-openscaler ~]$ mkdir workdir  
[[pmi-b2412@pmi-srv-openscaler ~]$ ls  
abc_12      history_12  online      reports     workdir  
hard_abc1   monthly    public_html trash_12
```

### Перенос файла:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ ls  
PosfixExpression_v1.cpp  
[[pmi-b2412@pmi-srv-openscaler workdir]$ head -10 PosfixExpression_v1.cpp  
#include <stdio.h>  
#include <ctype.h>  
#include <stdlib.h>  
#include <locale.h>  
  
struct list  
{  
    char elem;  
    list *next;  
};  
  
[[pmi-b2412@pmi-srv-openscaler workdir]$
```

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

### Создаем git-репозиторий:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ git init  
подсказка: Using 'master' as the name for the initial branch. This default branch name  
подсказка: is subject to change. To configure the initial branch name to use in all  
подсказка: of your new repositories, which will suppress this warning, call:  
подсказка: git config --global init.defaultBranch <name>  
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
подсказка: 'development'. The just-created branch can be renamed via this command:  
подсказка: git branch -m <name>  
Инициализирован пустой репозиторий Git в /home/pmi-b2412/workdir/.git/  
[[pmi-b2412@pmi-srv-openscaler workdir]$ git config --global user.name "Gerasimenko"  
[[pmi-b2412@pmi-srv-openscaler workdir]$ git config --global user.email v.gerasimenko.2022@stud.nstu.ru  
[[pmi-b2412@pmi-srv-openscaler workdir]$ git status  
На ветке master
```



### Добавляем начальную версию программы:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ git add PostfixExpression.cpp
[[pmi-b2412@pmi-srv-openscaler workdir]$ git commit -m "Создана начальная версия программы."
[master (корневой коммит) 763ecbf] Создана начальная версия программы.
 1 file changed, 84 insertions(+)
 create mode 100644 PostfixExpression.cpp
```

---

### Добавляем входные данные:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ git add in.txt
[[pmi-b2412@pmi-srv-openscaler workdir]$ git commit -m "Загружены входные данные."
[master 0e0b915] Загружены входные данные.
 1 file changed, 1 insertion(+)
 create mode 100644 in.txt
```

---

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.

### Программа доработана с учетом всех требований.

### Справка оформлена.

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

### Сохранение доработанной программы в ветке master:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ git add PostfixExpression.cpp
[[pmi-b2412@pmi-srv-openscaler workdir]$ git commit -m "Программа исправлена в соответствии с требованиями."
[master ca2afb5] Программа исправлена в соответствии с требованиями.
 1 file changed, 218 insertions(+), 84 deletions(-)
 rewrite PostfixExpression.cpp (61%)
[[pmi-b2412@pmi-srv-openscaler workdir]$ █
```

---

### Создание ветки docum и сохранение справки:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ git checkout -b docum
Переключено на новую ветку «docum»
[[pmi-b2412@pmi-srv-openscaler workdir]$ git add справка.txt
[[pmi-b2412@pmi-srv-openscaler workdir]$ git commit -m "Добавлена справка."
[docum b7b3572] Добавлена справка.
 1 file changed, 19 insertions(+)
 create mode 100644 справка.txt
```

---

9. Сформировать исполняемый файл программы с помощью компилятора **gcc/g++**:

```
[[pmi-b2412@pmi-srv-openscaler workdir]$ g++ -o PostfixExpression PostfixExpression.cpp
[[pmi-b2412@pmi-srv-openscaler workdir]$ ./PostfixExpression
```

---

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

### Программа работает исправно, ошибок не обнаружено.



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.

```
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test1.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test2.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test3.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test4.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test5.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test6.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git add test7.txt
[pmi-b2412@pmi-srv-openscaler workdir]$ git commit -m "Добавлены входные данные."
[docum 81be176] Добавлены входные данные.
7 files changed, 6 insertions(+)
create mode 100644 test1.txt
create mode 100644 test2.txt
create mode 100644 test3.txt
create mode 100644 test4.txt
create mode 100644 test5.txt
create mode 100644 test6.txt
create mode 100644 test7.txt
```

---

#### Test 1

Цель: проверка на отсутствие файла "in.txt".

Ожидаемый результат: "Возникла ошибка при открытии файла."

Результат: "Возникла ошибка при открытии файла."

#### Test 2

Цель: проверка на пустоту файла "in.txt".

Ожидаемый результат: "Введены некорректные данные."

Результат: "Введены некорректные данные."

#### Test 3

Содержимое: +12

Цель: попытка ввода выражения в префиксной форме.

Ожидаемый результат: "Введены некорректные данные."

Результат: "Введены некорректные данные."

#### Test 4

Содержимое: 12+34

Цель: количество операндов превышает допустимое значение, относительно количества операторов.

Ожидаемый результат: "Введены некорректные данные."

Результат: "Введены некорректные данные."

#### Test 5

Содержимое: 12\*-+

Цель: количество операторов превышает допустимое значение, относительно количества операндов.

Ожидаемый результат: "Данные введены некорректно."

Результат: "Данные введены некорректно."

## Test 6

Содержимое: pm-2412

Цель:

На вход поступают символы, не являющиеся операндами и операторами.

Ожидаемый результат: "Введены некорректные данные."

Результат: "Введены некорректные данные."

## Test 7

Содержимое: 123\*45+-+

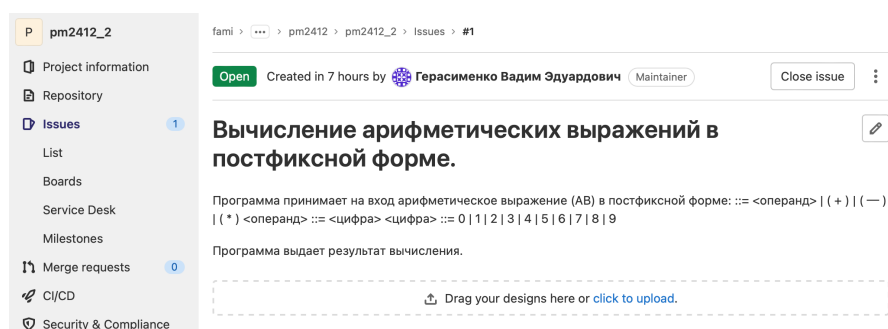
Цель: Проверка работоспособности программы: постфиксная запись выражения "1 + 2 \* 3 - (4 + 5)".

Ожидаемый результат: "Значение выражения = -2".

Результат: "Значение выражения = -2".

## 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».

```
[pmi-b2412@pmi-srv-openscaler workdir]$ git remote add origin http://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm2412_2.git
[pmi-b2412@pmi-srv-openscaler workdir]$ git push origin master docum
Username for 'https://gitlab.cloud.nstu.ru': v.gerasimenko.2022@stud.nstu.ru
Password for 'https://v.gerasimenko.2022@stud.nstu.ru@gitlab.cloud.nstu.ru':
warning: переадресация на https://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm2412_2.git/
Перечисление объектов: 21, готово.
Подсчет объектов: 100% (21/21), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (12/12), готово.
Запись объектов: 100% (21/21), 3.95 КиБ | 1.97 МБ/с, готово.
Всего 21 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: To create a merge request for docum, visit:
remote: http://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm2412_2/-/merge_requests/new?merge_request%5Bsource_branch%5D=docum
remote:
To http://gitlab.cloud.nstu.ru/fami/pm-24/pm2412/pm2412_2.git
 * [new branch]      master -> master
 * [new branch]      docum -> docum
```



## 3. Вывод:

В ходе лабораторной работы были изучены системы контроля версий git и GitLab, а также подготовлен репозиторий для выполнения лабораторной работы № 6. Контрольные вопросы проработаны.

## Приложение:

### Условие задачи:

В текстовом файле записано без ошибок арифметическое выражение (AB) в постфиксной форме:

$\langle AB \rangle ::= \langle \text{операнд} \rangle \mid (\langle AB \rangle + \langle AB \rangle) \mid (\langle AB \rangle - \langle AB \rangle) \mid (\langle AB \rangle * \langle AB \rangle)$

$\langle \text{операнд} \rangle ::= \langle \text{цифра} \rangle$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Вычислить как целое число значение выражения.

### Код программы:

```
#include <stdio>
#include <ctype.h>
#include <locale.h>

struct list
{
    char elem;
    list *next;

    list(char _elem = 0, list *_next = NULL) : elem(_elem), next(_next) {}
};
typedef list stack;

stack *push(char c = 0, stack *s = NULL)
{
    stack *p = new stack(c, s);
    return p;
}

stack *pop(stack *s = NULL)
{
    stack *p = s;
    s = s->next;
    delete p;
    return s;
}

stack *expressionCalculation (stack *s, char c)
{
    if (!s || !s->next)
    {
        printf("Данные введены некорректно.");
        return NULL;
    }
}
```

```

else
{
    char op2 = s->elem;
    s = pop(s);
    char op1 = s->elem;
    s = pop(s);

    switch (c)
    {
        case '-':
            s = push(op1 - op2, s); return s;
        case '+':
            s = push(op1 + op2, s); return s;
        case '*':
            s = push(op1 * op2, s); return s;
        default:
            {
                printf("Выражение содержит недопустимые символы.");
                return NULL;
            }
    }
}
}

stack *inputConsole(stack *s = NULL)
{
    char c = 0;
    bool isPostfix = false;
    int operandsCount = 0;
    int operatorsCount = 0;
    printf("Введите постфиксное выражение: ");
    getchar();

    for (char c = getchar(); c != '\n' && c != EOF; c = getchar())
    {
        if (isdigit(c))
        {
            s = push(c - '0', s);
            isPostfix = true;
            operandsCount++;
        }
        else
        {
            if (isPostfix)
            {

```

```

        s = expressionCalculation(s, c);
        if (!s)
            return NULL;
        operatorsCount++;
    }
    else
    {
        printf("Введены некорректные данные.");
        return NULL;
    }
}

if (operandsCount != operatorsCount + 1)
{
    printf("Введены некорректные данные.\n");
    return NULL;
}
return s;
}

stack *inputFile(stack *s = NULL)
{
    FILE *f = NULL;
    bool isPostfix = false;
    int operandsCount = 0;
    int operatorsCount = 0;

    if ((f = fopen("in.txt", "r")))
    {
        char c = 0;
        for ( ; !feof(f) && (c = fgetc(f)) != EOF; )
        {
            if (isdigit(c))
            {
                s = push(c - '0', s);
                isPostfix = true;
                operandsCount++;
            }
            else
            {
                if (isPostfix)
                {
                    s = expressionCalculation(s, c);
                    if (!s)

```

```

        return NULL;
        operatorsCount++;
    }
    else
    {
        printf("Введены некорректные данные.");
        return NULL;
    }
}

if (operandsCount != operatorsCount + 1)
{
    printf("Введены некорректные данные.\n");
    return NULL;
}
return s;
}
else
{
    printf("Возникла ошибка при открытии файла.");
    return NULL;
}
fclose(f);
return s;
}

void outputFile(stack *s = NULL)
{
    FILE *f = NULL;

    if ((f = fopen("out.txt", "w")))
    {
        fprintf(f, "Значение выражения = %d", s->elem);
        fclose(f);
    }
}

void outputConsole(stack *s = NULL) { printf("Значение выражения = %d",
    s->elem); }

int main()
{
    setlocale(0, "");

    stack *s = NULL;

```



```

int inputMethod = 0;

do
{
    printf("Выберите способ ввода постфиксного выражения: \n");
    printf("1. Данные из файла \"in.txt\". \n");
    printf("2. Ввести вручную. ");

    if (scanf("%d", &inputMethod) != 1 || (inputMethod != 1 &&
        inputMethod != 2))
    {
        printf("Некорректный выбор способа ввода данных.\n\n");
        while (getchar() != '\n');
    }
} while (inputMethod != 1 && inputMethod != 2);

if (inputMethod == 1)
    s = inputFile(s);
else if (inputMethod == 2)
    s = inputConsole(s);

if (s)
{
    int outputMethod = 0;

    do
    {
        printf("Выберите способ вывода результата вычисления: \n");
        printf("1. Вывести в файл \"out.txt\". \n");
        printf("2. Вывести в консоль. ");

        if (scanf("%d", &outputMethod) != 1 || (outputMethod != 1 &&
            outputMethod != 2))
        {
            printf("Некорректный выбор способа вывода данных.\n\n");
            while (getchar() != '\n');
        }
    } while (outputMethod != 1 && outputMethod != 2);

    if (outputMethod == 1)
        outputFile(s);
    else if (outputMethod == 2)
        outputConsole(s);
}
}

```