

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №1

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Ваньянц И.М. « » _____ 20__ г.

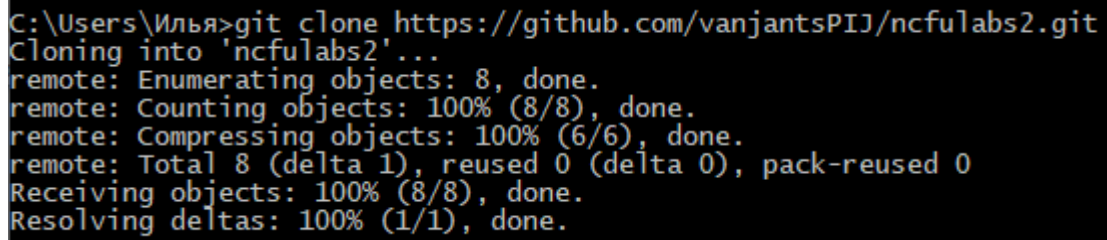
Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

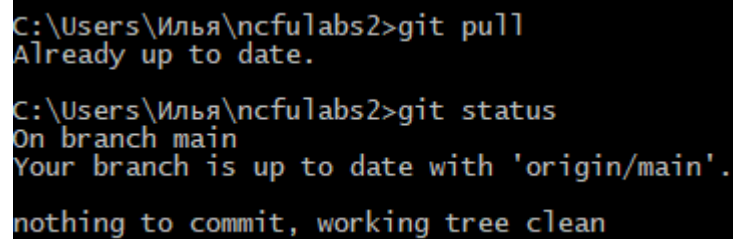
Ставрополь 2021

ХОД РАБОТЫ



```
C:\Users\Илья>git clone https://github.com/vanjantsPIJ/ncfulabs2.git
Cloning into 'ncfulabs2'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1 – локальный репозиторий создан



```
C:\Users\Илья\ncfulabs2>git pull
Already up to date.

C:\Users\Илья\ncfulabs2>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Рисунок 2 – сбор всех новшеств из удаленного репозитория в локальный и его дальнейший статус

```

C:\Users\Илья\ncfulabs2>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\Илья\ncfulabs2>git add .
C:\Users\Илья\ncfulabs2>git commit -m third commit
error: pathspec 'commit' did not match any file(s) known to git
C:\Users\Илья\ncfulabs2>git commit -m "third commit"
[main f5d75c6] third commit
 1 file changed, 2 insertions(+), 1 deletion(-)

```

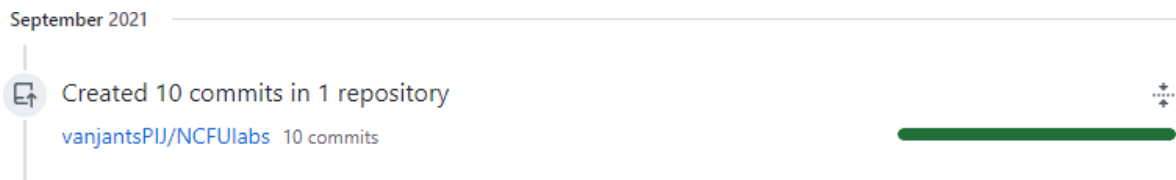
Рисунок 3 – добавление всех изменений в контроль версий и первый коммит

```

C:\Users\Илья\ncfulabs2>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/vanjantsPIJ/ncfulabs2.git
 2dcf759..f5d75c6  main -> main

```

Рисунок 4 – загрузка всех изменений на удаленный репозиторий



КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Система контроля версий — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.
2. Локальные СКВ – сильно подвержены появлению ошибок; Центральные СКВ - единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками. Если жёсткий диск, на котором хранится центральная БД, повреждён, а своевременные бэкапы отсутствуют, вы потеряете всё — всю историю проекта, не считая единичных снимков репозитория, которые сохранились на локальных машинах разработчиков.
3. Git – распределённая СКВ
4. Подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы.
5. В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме.
6. У Git есть три основных состояния, в которых могут находиться ваши файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged).
Зафиксированный значит, что файл уже сохранён в вашей локальной базе. К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы. Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит. Связь: из рабочей директории в область подготовленных файлов, из ОПФ в git – директорию, из git – директории в рабочую директорию.
7. Учетная запись пользователя, в которой он может создавать и изменять репозитории и то, что в них находится.

8. Персональные и публичные.
9. Создание репозитория на github, его клонирование в локальное хранилище устройства, работа с файлами репозитория на его локальном аналоге, отправка этих изменений в удаленный репозиторий.
10.
 - 1.git version – проверка версии
 2. git config --global user.name – установка вашего имени
 3. git config --global user.email – установка вашего email
 4. git clone – клонирование вашего репозитория в локальное хранилище
11. GNU, BSD, MIT, CCZ, BSL, MPL
12. Скопировать адрес репозитория из github, и при поместить его после команды git clone. Клонирование репозитория нужно для того, чтобы вы могли работать с ним на локальном уровне, а также иметь резерв.
13. Командой git status.
14. Изменение локального репозитория, добавление в него новых файлов, идентификация изменений в файлах, фиксация этих изменений, применение всех изменений на удалённом репозитории.
15. При добавлении или изменении файла в локальном репозитории он ожидает от пользователя сохранения определённых или всех изменений. После команды git add все изменения сохраняются и становятся доступны для коммита. После команды git commit все изменения фиксируются. При команде git push все изменения применяются в удалённом репозитории.
16. git clone //ссылка на репозиторий
Git init
Git pull
//внос изменений
Git add .
Git commit -m “комментарий”
Git push
17. Gitlab, BitBucket, SourceForge и т.д
В отличие от GitHub, GitLab предлагает бесплатные частные репозитории для проектов с открытым исходным кодом.
18. При установке git пользователь получает два графических инструмента gitk и git-gui. Чтобы вызвать gitk нужно перейти в

каталог с репозиторием и набрать команду `gitk`, а `git-gui` вызывается при помощи команды `git gui`.