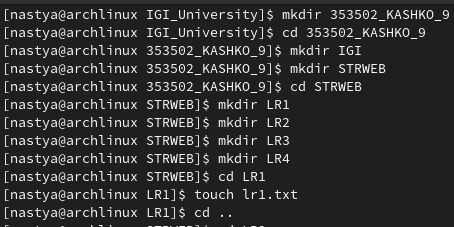
ИЗБРАННЫЕ ГЛАВЫ ИНФОРМАТИКИ

Лабораторная работа №1

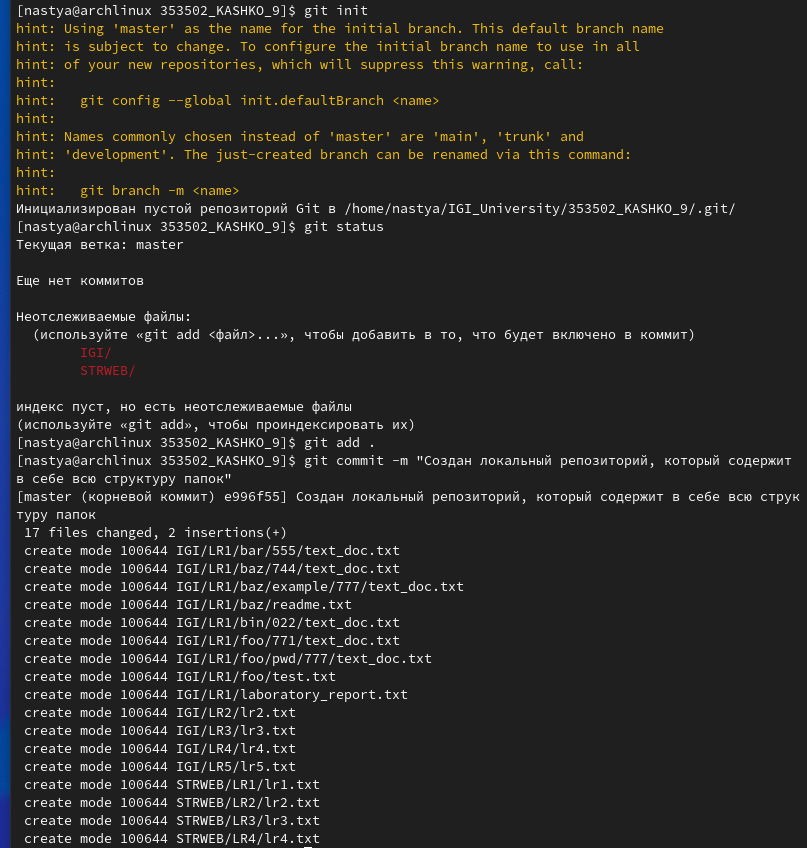
“Система контроля версий Git”

Кашко Анастасия Александровна, 353502

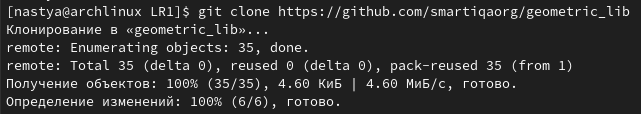
| Команда | Назначение |
| --- | --- |
| mkdir имя\_директории | Создать директорию (папку) |
| cd имя\_директории | Перейти в директорию (папку) |
| cd (..) | Перейти в основной каталог (выйти из текущей директории) |
| touch имя\_файла\_с\_расширением | Создать файл |
| echo “текст” > имя\_файла\_с\_расширением | Создать файл с текстом внутри |

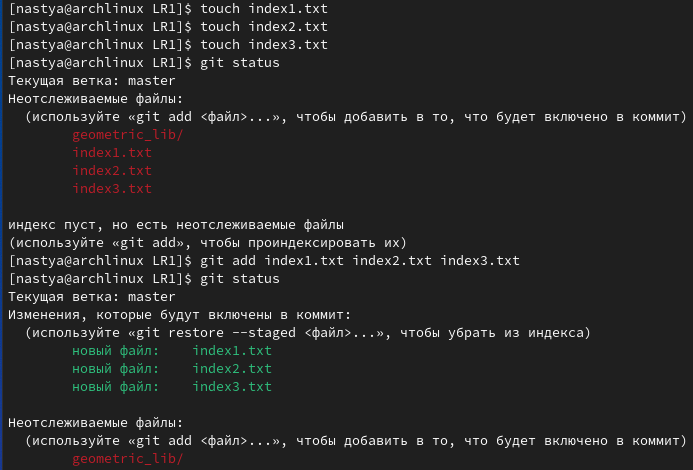


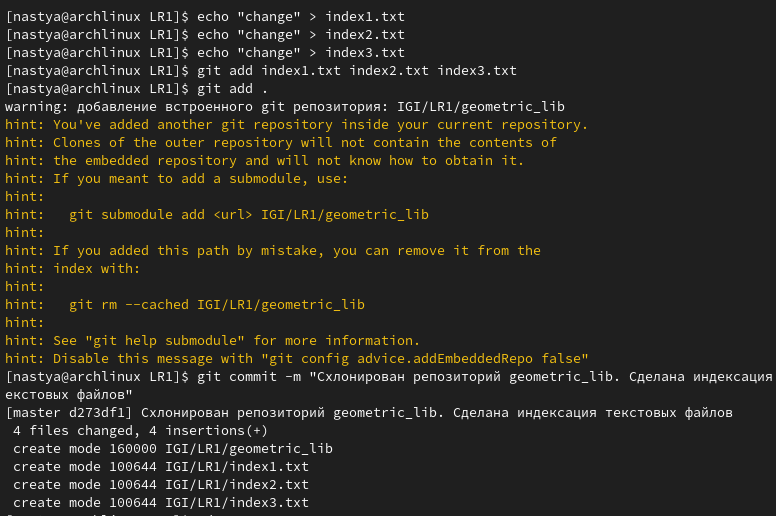
| Команда | Назначение |
| --- | --- |
| git init | Проинициализировать локальный репозиторий |
| git status | Отображение списка изменённых файлов |
| git add . | Проиндексировать файлы |
| git commit -m “текст” | Залить изменения в локальный репозиторий |



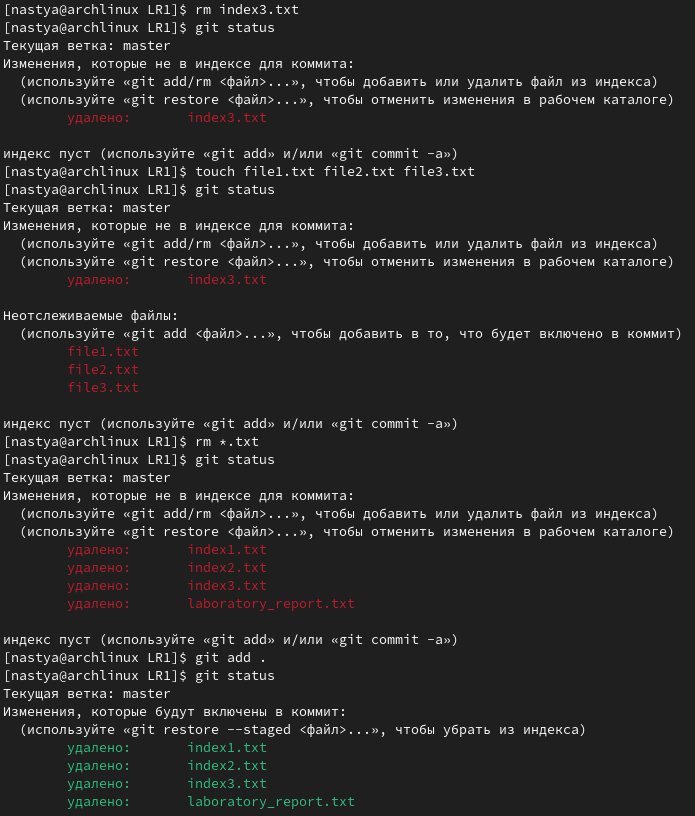
| Команда | Назначение |
| --- | --- |
| git clone | Клонировать репозиторий |



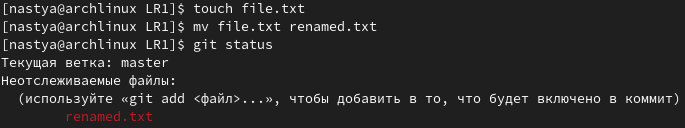




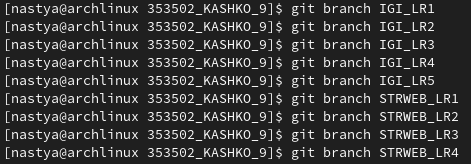
| Команда | Назначение |
| --- | --- |
| rm имя\_файла (git rm) | Удалить файл из системы (из лок. репозитория) |
| rm \*.”расширение” | Удалить группу файлов с одинаковым расширением |
| rm -r имя\_папки | Удалить директорию |
| touch имя\_файла\_с\_расширением1 имя\_файла\_с\_расширением2 | Создать группу файлов с одинаковым расширением |

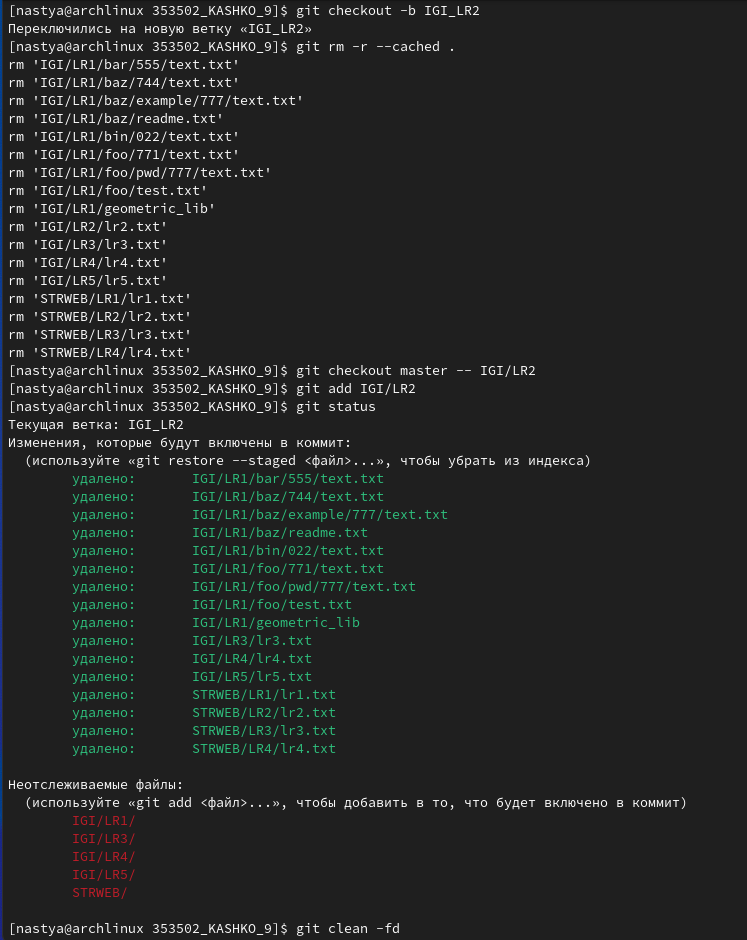


| Команда | Назначение |
| --- | --- |
| mv старое\_имя новое\_имя | Переименовать файл |

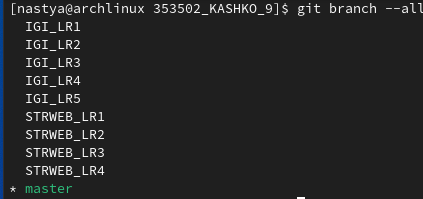


| Команда | Назначение |
| --- | --- |
| git branch имя\_ветки | Создать ветку |
| git checkout -b имя\_ветки | Создать ветку |
| git checkout имя\_ветки | Переключиться на ветку |
| git checkout -f имя\_ветки | Переключиться принудительно на ветку |
| git rm -r --cached . | Удалить всё содержимое ветки |
| git checkout имя\_ветки\_с\_которой\_перенести\_файла -- папки которые\_ надо\_перенести | Перенести на текущую ветку необходимы папки |
| git clean -fd | Очистить все неотслеживаемые файлы |
| git ls-tree -r HEAD | Просмотреть все папки, файлы текущей ветки |
| git branch --all | Просмотр всех существующих веток |

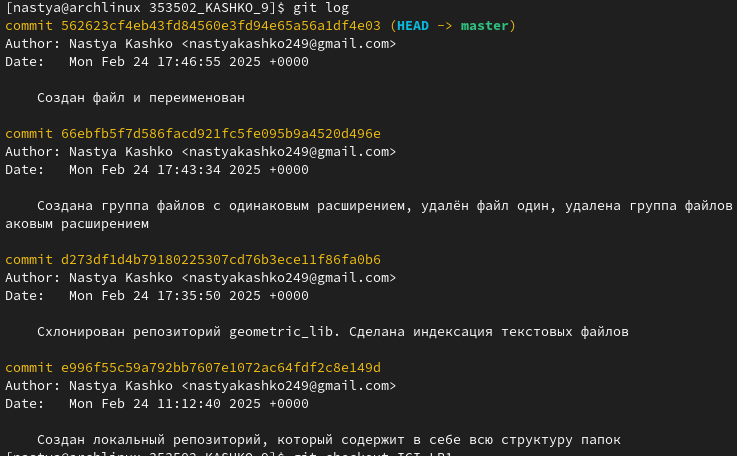


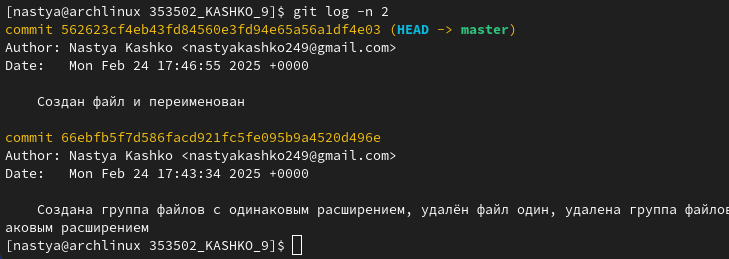


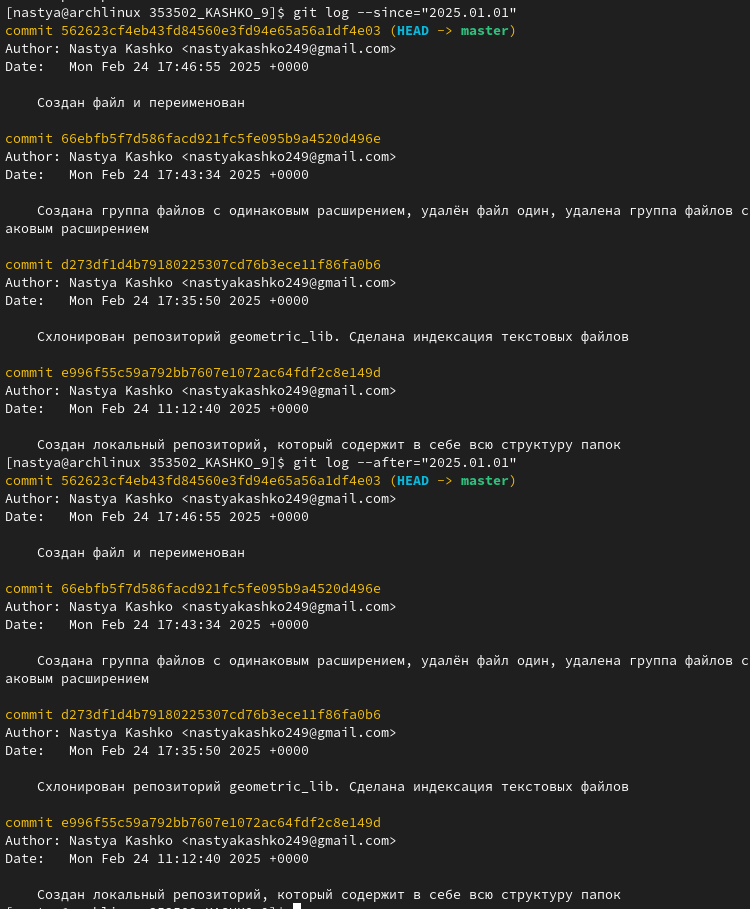


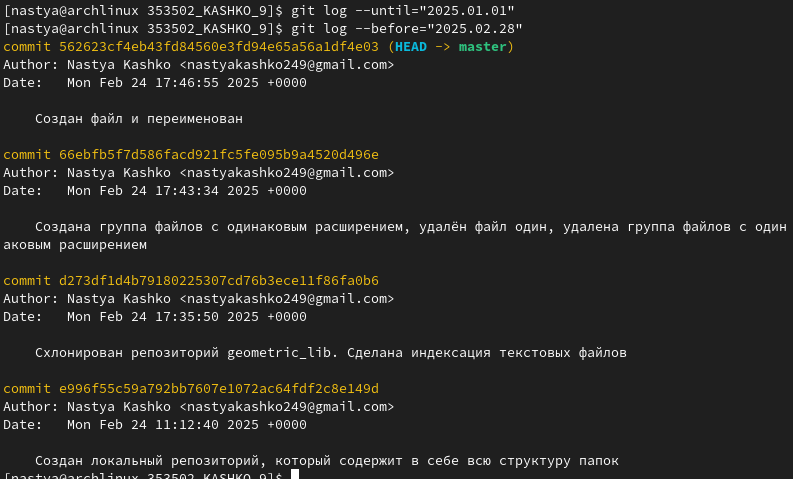


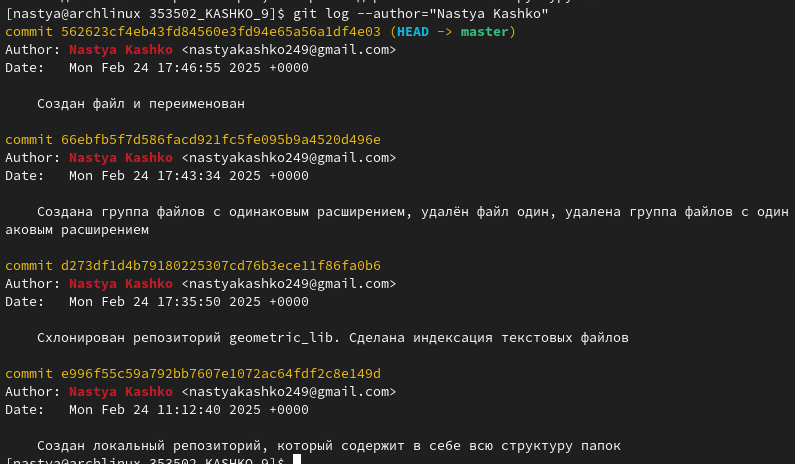
| Команда | Назначение |
| --- | --- |
| git log | История всех коммитов на текущей ветке |
| git log -n “число” | История последних “число” коммитов |
| git log --since=“дата” | История коммитов после указанной даты |
| git log --after=“дата” | История коммитов после указанной даты |
| git log --until=“дата” | История коммитов до указанной даты |
| git log --before=“дата” | История коммитов до указанной даты |
| git log --author=“имя” | История коммитов по автору |
| git log --committer=“имя” | История коммитов по коммитеру |
| git log --grep=“слово” | История коммитов, которые содержат “слово” |

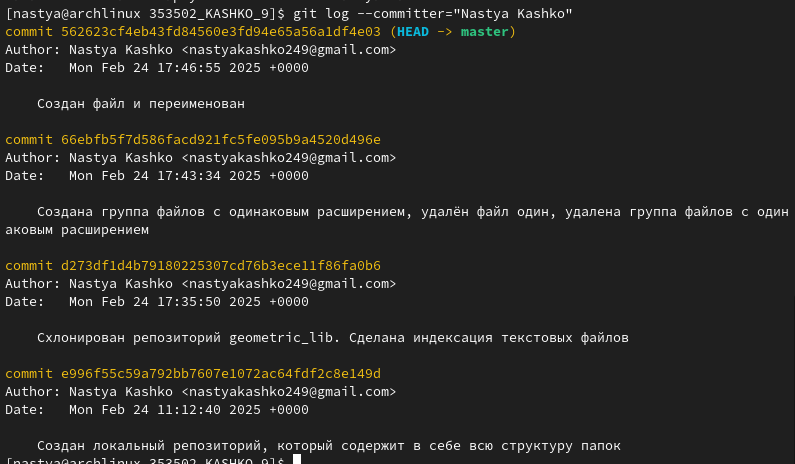


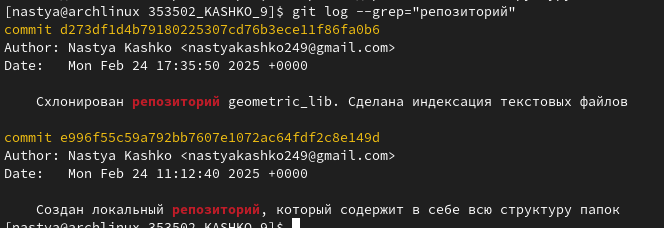




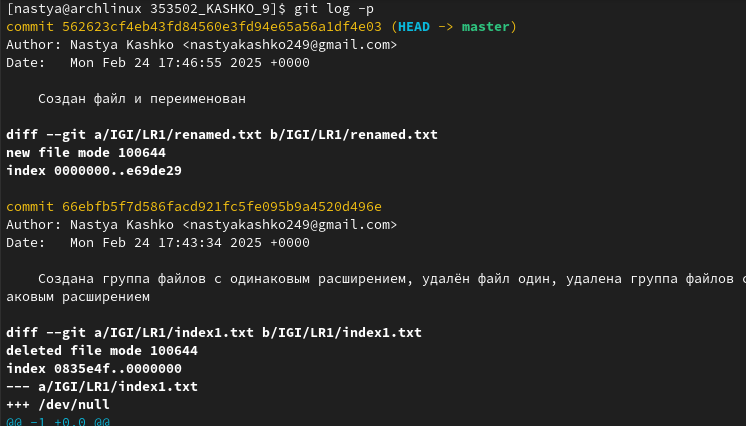


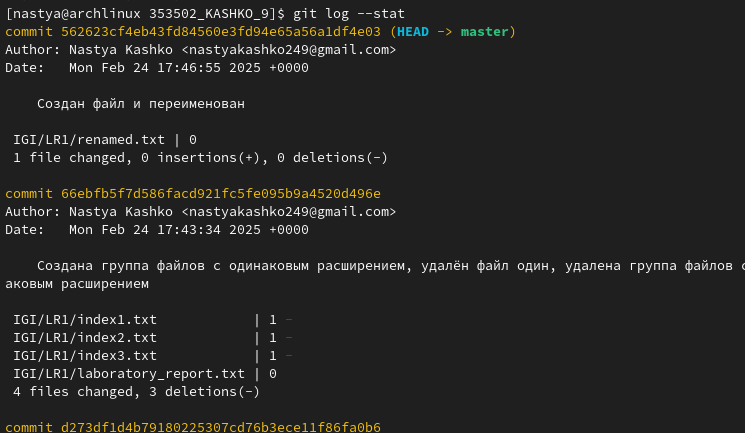


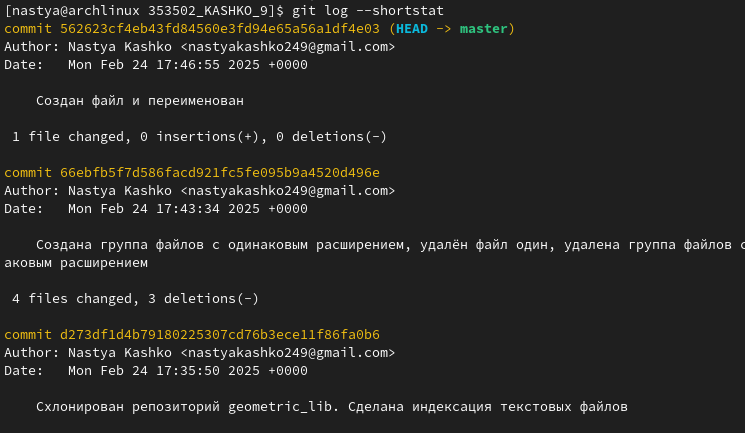


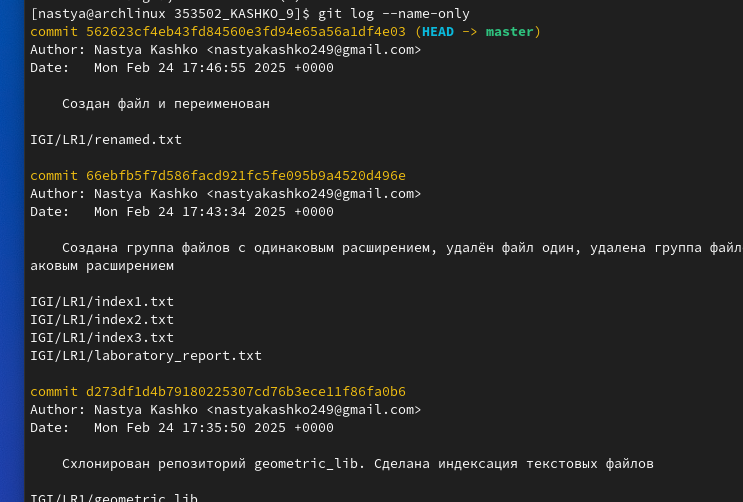


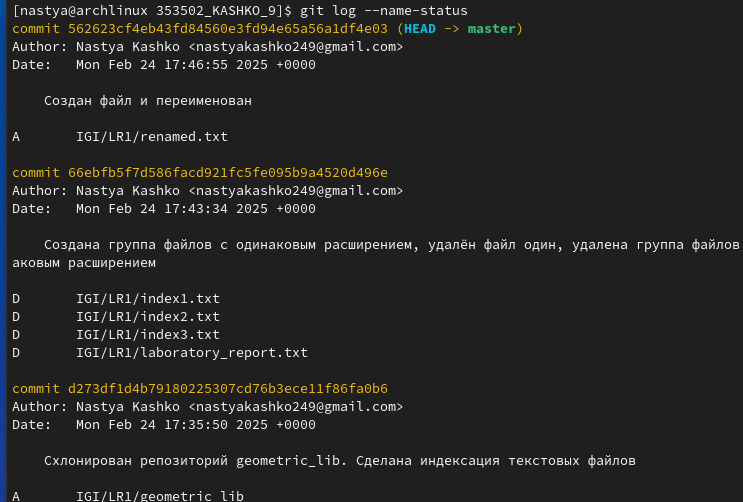
| Команда | Назначение |
| --- | --- |
| git log -p | Показывает все изменения коммитов (удаление / создание / изменение строк и т.д.) |
| git log --stat | Показывает какие файлы изменились и сколько строк добавлено / удалено |
| git log --shortstat | Аналогично git log - -stat, но показывает только конечную строку с количеством изменений |
| git log --name-only | Выводит список изменённых файлов |
| git log --name-status | Выводит список изменённых файлов + статус изменения (А - добавление, М - модификация, D - удаление) |
| git log --relative-date | Вывод времени сделанных коммитов |
| git log --pretty | Позволяет настраивать формат вывода (например, git log - -pretty=short - краткое, но читаемое) |
| git log --graph | Визуализация истории коммитов (граф коммитов) |

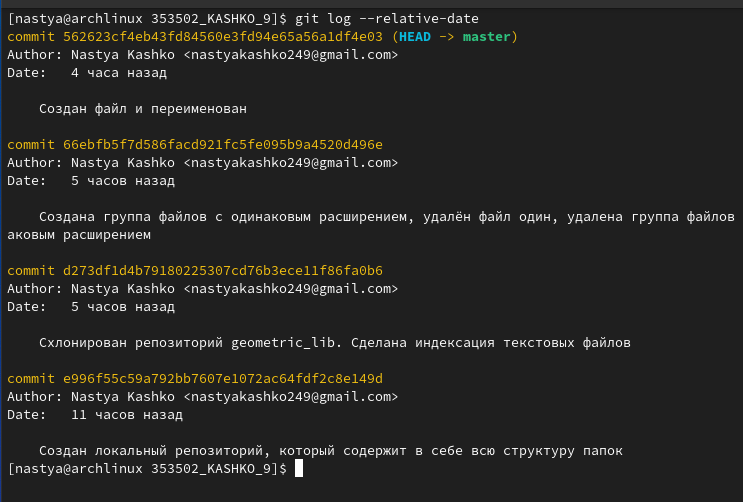


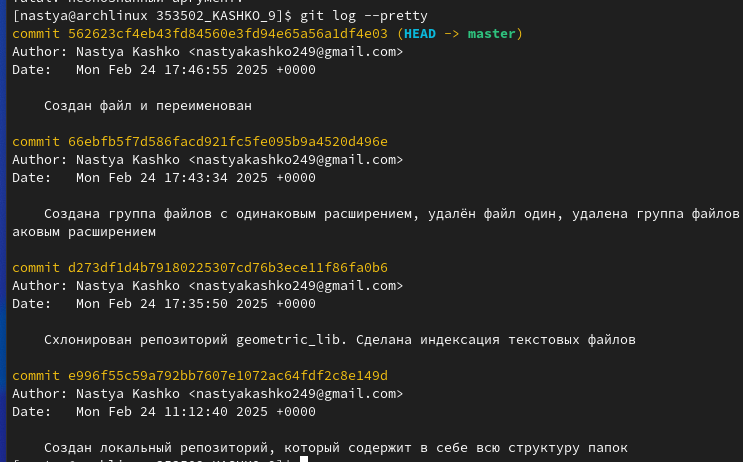


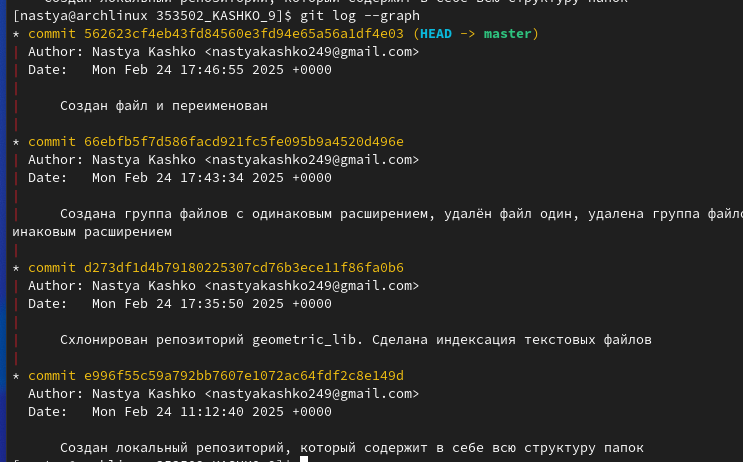




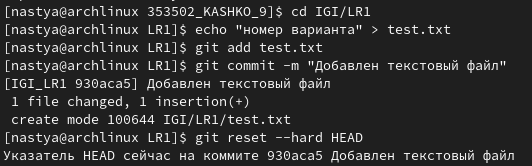


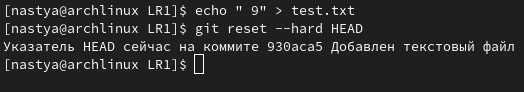


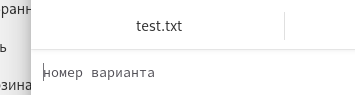


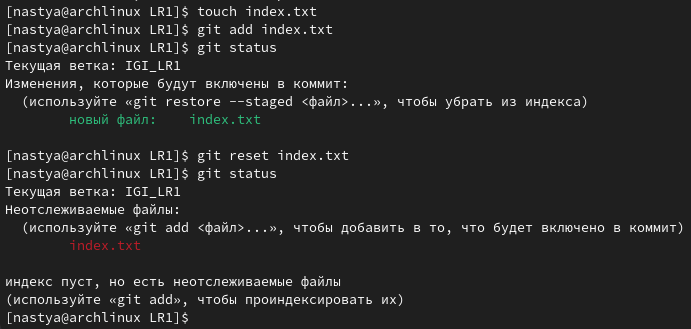


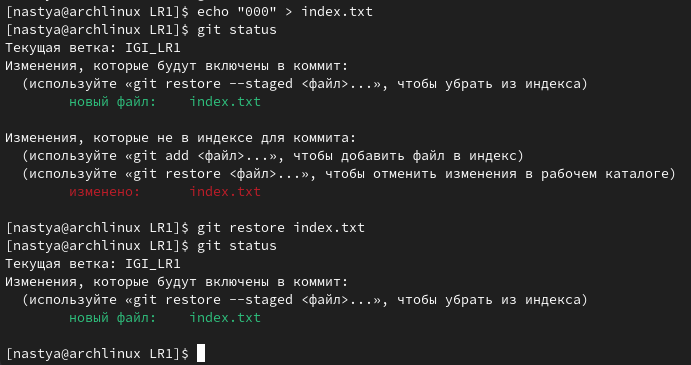
| Команда | Назначение |
| --- | --- |
| git reset --hard HEAD | Откатывает состояние ветки до последнего коммита |
| git reset имя\_файла | Удаляет файл из индекса |
| git restore имя\_файла | Отменить изменения файла, если он не проиндексирован |

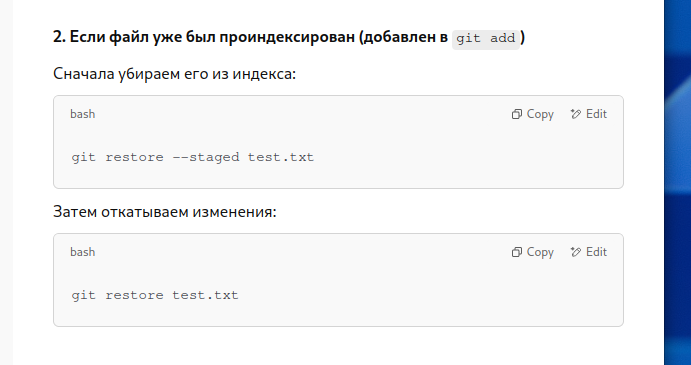




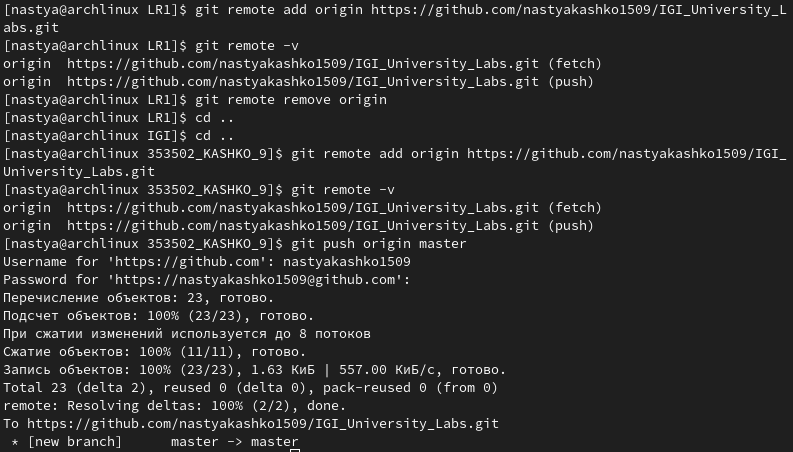


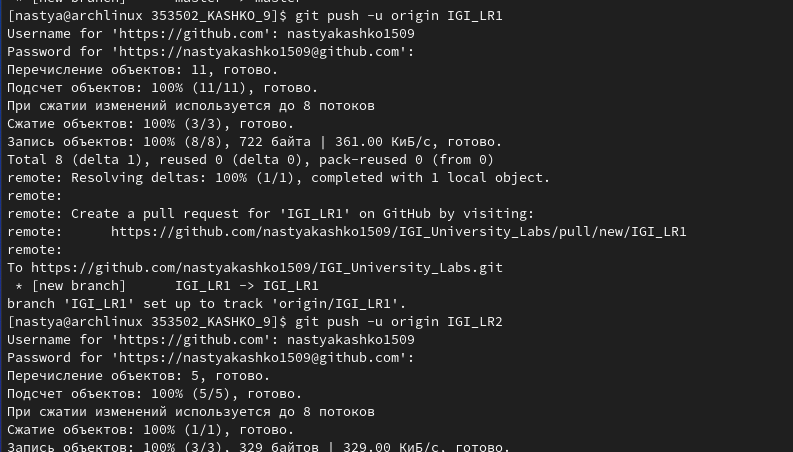


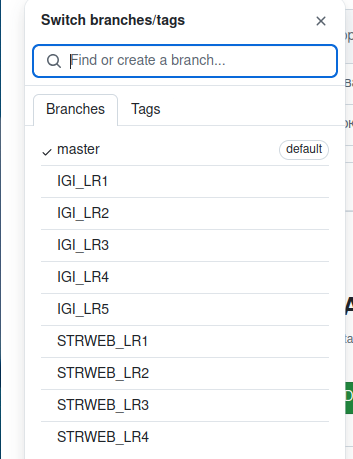




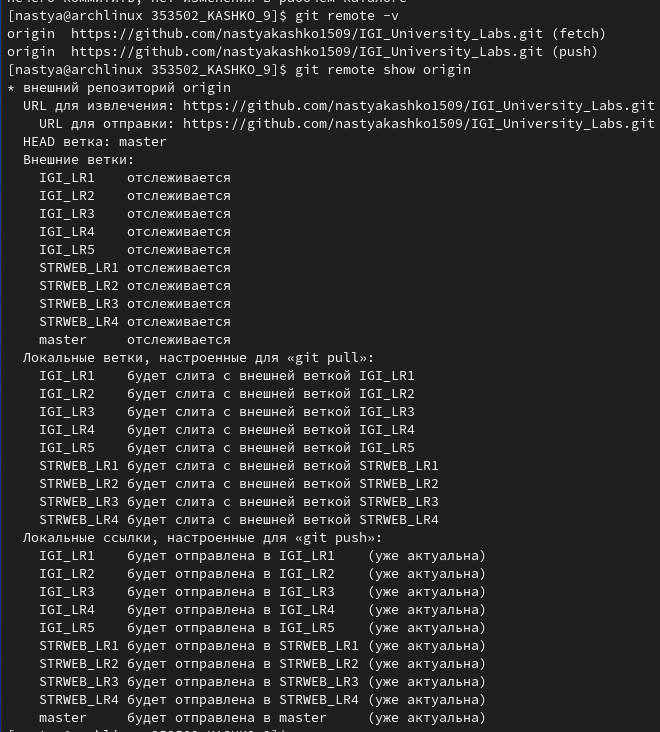
| Команда | Назначение |
| --- | --- |
| git remote add origin ссылка (https) git remote -v git push origin master | Связываем локальный репозиторий с удалённым |
| git remote remove origin | Удалить связь локального с удалённым репозиторием |
| git push -u origin имя\_ветки | Запушить остальные ветки на удалённый репозиторий |





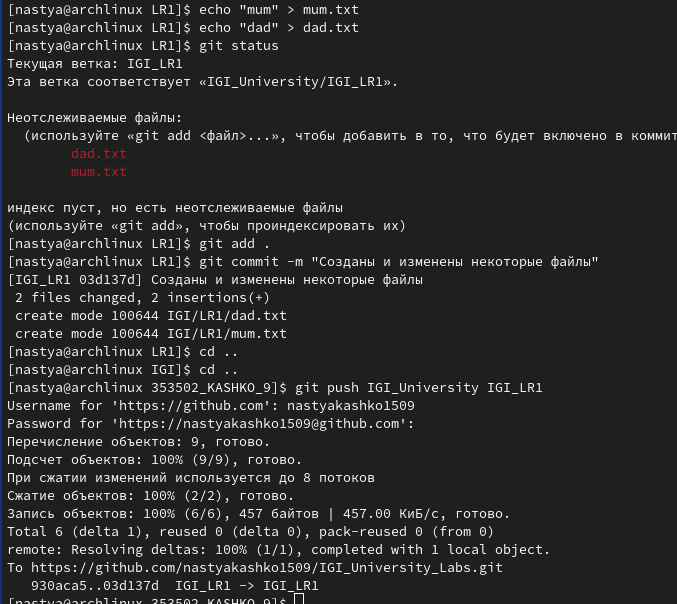


| Команда | Назначение |
| --- | --- |
| git remote -v | Покажет все удалённые репозитории |
| git remote show origin | Подробная информация про удалённые репозитории |
| git remote rename старое\_имя новое\_имя | Переименовать удалённый репозиторий |
| git fetch имя\_удалённого\_репозитория | Получить удалённые изменения |
| git push имя\_удаленного\_репозитория имя\_ветки | Запушить изменения ветки на удалённый репозиторий |



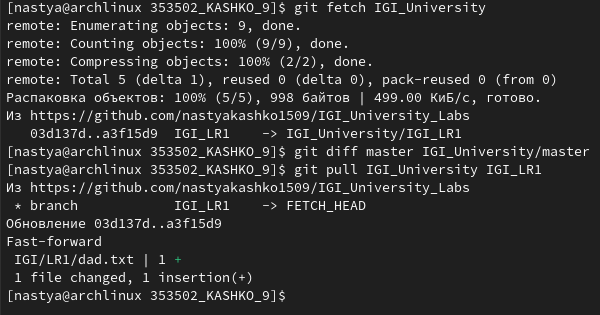


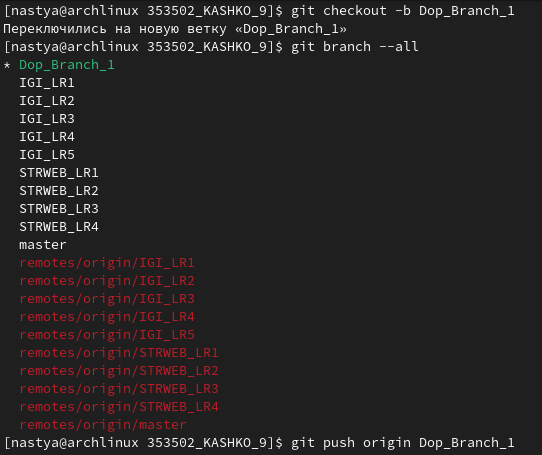


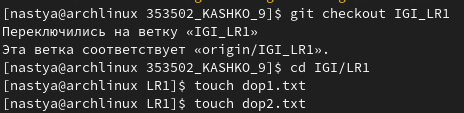


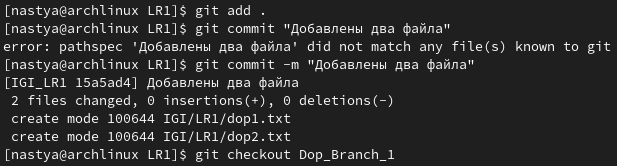
Зашла на github.com, на ветке IGI\_LR1 в файл внесла изменения, переношу изменения на локальный репозиторий:

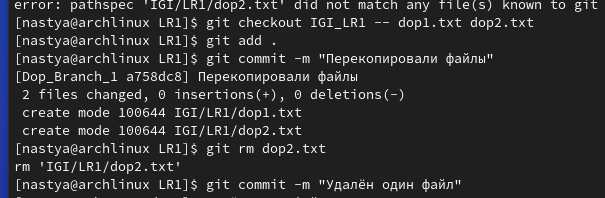
git fetch IGI\_University -> git pull IGI\_University IGI\_LR1











| Команда | Назначение |
| --- | --- |
| git push --delete origin имя\_ветки | Удалить ветку с удаленного репозитория |
| git merge имя\_ветки | Сливаем изменения в текущую ветку с указанной ветки |
| git config --system user.name имя | Установить имя пользователя на системном уровне |
| git config --system user.email почта | Установить почту на системном уровне |

--global - на глобальном уровне

--local - на локальном уровне

Доп. параметры: git config --global core.editor "vim" (редактор коммитов), git config --global fetch.prune true (автоматическое удаление ссылок на ветки, которые были удалены).