

Балтийский государственный технический университет «ВОЕНМЕХ» им. Д. Ф.
Устинова

Кафедра И5
«Информационные системы и программная инженерия»

Лабораторная работа № 5
по дисциплине «Компьютерный практикум»
на тему «**Особенности использования систем контроля версий**»

Выполнил:
Студент Дубровский В.И.
Группа И582

Преподаватель:
Вальштейн К. В.

Санкт-Петербург
2019

Цель работы - изучить особенности различных систем контроля версий.

Часть 1. Работа с локальным репозиторием:

Для создания локального репозитория необходимо воспользоваться командой `git init`.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git init
Инициализирован пустой репозиторий Git в /home/vladislav/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL/.git/
```

Рисунок 5.1 – Результат работы команды `git init`

Далее создаём файл `.gitignore`, располагая его в папке с проектом, и указываем файлы, которые должен игнорировать `git`, в нашем случае это файл сборки.

```
.gitignore
1  # ignore bin folder
2  bin/
```

Рисунок 5.2 – Содержимое файла `.gitignore`

Чтобы добавить файлы в следующий коммит, нужно прописать команду `git add`, а для проверки состояния файлов воспользоваться командой `git status`.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git status
На ветке master

Еще нет коммитов

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

.gitignore
bin/
include/
lib/
source/

ничего не добавлено в коммит, но есть неотслеживаемые файлы (используйте «git add», чтобы отслеживать их)
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git add .
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ 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: .gitignore
новый файл: bin/main
новый файл: include/load.h
новый файл: include/matrix.h
новый файл: include/myarray.h
новый файл: lib/libload.so
новый файл: lib/libmatrix.so
новый файл: lib/libmyarray.so
новый файл: lib/libs.tar.gz
новый файл: source/load.cpp
новый файл: source/main.cpp
новый файл: source/matrix.cpp
новый файл: source/myarray.cpp

vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$
```

Рисунок 5.3 – Результат работы команд `git status` и `git add`

Снимок директории проекта происходит с помощью команды `git commit`.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git commit -m "First commit."
[master (корневой коммит) 89684b7] First commit.
12 files changed, 302 insertions(+)
create mode 100644 .gitignore
create mode 100644 include/load.h
create mode 100644 include/matrix.h
create mode 100644 include/myarray.h
create mode 100755 lib/libload.so
create mode 100755 lib/libmatrix.so
create mode 100755 lib/libmyarray.so
create mode 100644 lib/libs.tar.gz
create mode 100644 source/load.cpp
create mode 100644 source/main.cpp
create mode 100644 source/matrix.cpp
create mode 100644 source/myarray.cpp
```

Рисунок 5.4 – Результат работы команды git commit

Посмотреть историю коммитов можно с помощью команды git log.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git log
commit 89684b7f1bebf0c6c6e7ebca13d95c42427a31ef (HEAD -> master)
Author: alohaeee <ujmferro@yandex.ru>
Date: Sun Oct 27 14:39:36 2019 +0300

First commit.
```

Рисунок 5.5 – Результат работы команды git log

Создание ветки происходит набором команды git branch *имя ветки*.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git branch Win
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git branch Linux
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git log
commit 29c991399a524be3440d8acb60c7ad807b21d229 (HEAD -> master, Win, Linux)
```

Рисунок 5.6 – Результат работы команды git branch

Переход к ветке осуществляется командой git checkout. Перейдя на ветку Win удаляем поддержку ОС семейства Linux из файлов load.h и load.cpp

Сделаем коммит проделанной работы.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git commit -m "Pure Win version"
[Win 2964b55] Pure Win version
2 files changed, 1 insertion(+), 50 deletions(-)
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git log
commit 2964b55cd6d59316c8ca104769b39d1fb951f462 (HEAD -> Win)
Author: alohaeee <ujmferro@yandex.ru>
Date: Sun Oct 27 22:14:14 2019 +0300

Pure Win version
```

Рисунок 5.7 – Результат работы команды git commit на ветке Win

Проделаем аналогичную работу на ветке Linux.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git status
На ветке Linux
Изменения, которые не в индексе для коммита:
(используйте «git add <файл>...», чтобы добавить файл в индекс)
(используйте «git checkout -- <файл>...», чтобы отменить изменения
в рабочем каталоге)

        изменено:      include/load.h
        изменено:      source/load.cpp

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git add .
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git commit -m "Pure Linux version"
[Linux 81cc60b] Pure Linux version
2 files changed, 3 insertions(+), 50 deletions(-)
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git log
commit 81cc60b37ba63240b1564a7010ad225d5c9c3382 (HEAD -> Linux)
Author: alohaeee <ujmferro@yandex.ru>
Date: Sun Oct 27 22:25:14 2019 +0300

Pure Linux version
```

Рисунок 5.8 – Результат работы команды git commit на ветке Linux

Сравнивать различия на ветках будем с помощью команды git diff.

```
void* loadDLL(const char* libraryName);
void closeDLL(void *library);

+
template<class T>
T loadFunction(void* library, const char* functionName)
{
    T function = nullptr;
    if(library != nullptr)
    {
-       function = reinterpret cast<T>(GetProcAddress((HINSTANCE)library, functionName));
+       function = reinterpret cast<T>(dlsym(library,functionName));
        if(function != nullptr)
        {
            return function;
        }
        else
        {
-           std::cerr << "Cannot open function, name: " << functionName << std::endl;
+           std::cerr << "Cannot open function, name: " << functionName << std::endl;
+           << dlerror() << std::endl;
            return nullptr;
        }
    }
}

diff --git a/source/load.cpp b/source/load.cpp
index 7d5e0b1..ac285df 100644
--- a/source/load.cpp
+++ b/source/load.cpp
@@ -1,13 +1,13 @@
#include "load.h"

-
void* loadDLL(const char* libraryName)
{
-    void* library = LoadLibrary(libraryName);
+    void* library = dlopen(libraryName, RTLD LAZY);
    if(library == nullptr)
    {
-        std::cerr << "Can't open library, name: " << libraryName << std::endl;
+        std::cerr << "Can't open library, name: " << libraryName << std::endl;
+        << dlerror() << std::endl;
        return nullptr;
    }
    return library;
@@ -15,5 +15,5 @@ void* loadDLL(const char* libraryName)

void closeDLL(void *library)
{
-    FreeLibrary((HINSTANCE)library);
-}
+    dlclose(library);
+}
```

Рисунок 5.9 – Результат работы команды git diff Win Linux

Для восстановления изначальной версии программы достаточно перейти на ветку master и удалить созданные ветки.

```
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git branch -D Linux
Ветка Linux удалена (была 81cc60b).
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git branch -D Win
Ветка Win удалена (была 2964b55).
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git status
На ветке master
ничего коммитить, нет изменений в рабочем каталоге
vladislav@vladislav-HP-ZBook-14-G2:~/Рабочий стол/Computer Workshop/Lab4-MakeFile/LAB3-DLL$ git log
commit 29c991399a524be3440d8acb60c7ad807b21d229 (HEAD -> master)
Author: alohaeee <ujmferro@yandex.ru>
Date:   Sun Oct 27 21:57:23 2019 +0300

    fill gitignore

commit 89684b7f1bebf0c6c6e7ebca13d95c42427a31ef
Author: alohaeee <ujmferro@yandex.ru>
Date:   Sun Oct 27 14:39:36 2019 +0300

    First commit.
```

Рисунок 5.10 – Восстановление изначальной версии

Часть 2. Работа с глобальным репозиторием.

Основные различия CVS и SVN:

- 1) SVN создавалась как альтернатива CVS с целью исправить недостатки CVS;
- 2) Как и CVS, SVN это бесплатная система контроля версий с открытым исходным кодом. С той лишь разницей, что распространяется под лицензией Apache, а не под Открытым лицензионным соглашением GNU.
- 3) Для сохранения целостности базы данных SVN использует атомарные операции. При появлении новой версии к финальному продукту применяются либо все исправления, либо ни одно из них. Таким образом, код защищают от хаотичных частичных правок, которые не согласуются между собой и вызывают ошибки.
- 4) В то время как в CVS операции с ветками кода дорогостоящие и не предусмотрены архитектурой системы, SVN создана как раз для этого. То есть, для более крупных проектов с ветвлением кода и многими направлениями разработки.

Основные различия SVN и Git:

- 1) Git разрабатывалась главным образом под Linux, то именно в этой ОС она работает быстрее всего
- 2) В Git каждая копия рабочей папки .git содержит всю историю разработки
- 3) Значительное увеличение быстродействия
- 4) Отсутствует потребность в центральном сервере. Каждый пользователь, который имеет копию проекта является носителем всех версий проекта, за счет чего мала вероятность потери проекта
- 5) Гибкая работа с файлами проекта и возможность не добавлять их в версию проекта
- 6) Не копирует файл в новую версию, если он не был изменен, а ссылается на предыдущую версию с последним изменением файла

Описание процесса регистрации и работы с одним из рассмотренных глобальных репозиториев (GitHub):

- 1) Если мы хотим иметь возможность сохранения проекта в интернете, создадим репозиторий на Github. Для начала нужно зарегистрироваться на сайте github.com.
- 2) После регистрации нажимаем кнопку "+" и вводим название репозитория.
- 3) Выбираем тип Public (репозиторий всегда Public для бесплатной версии) и нажимаем Create.
- 4) В результате мы создали репозиторий на сайте Github. На экране мы увидим инструкцию, как соединить наш локальный репозиторий со вновь созданным.

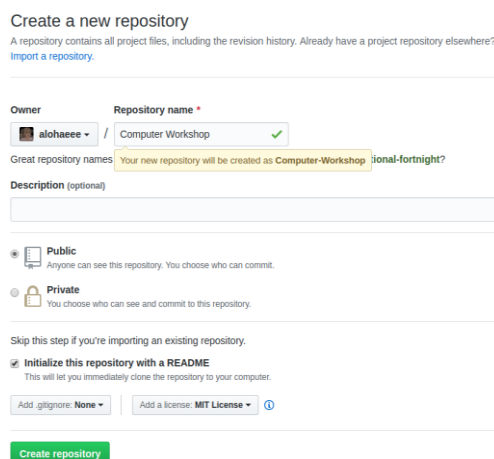


Рисунок 5.11 – Создание удаленного репозитория

- 5) Связать локальный репозиторий с удалённым можно с помощью команды
`git remote add origin https://github.com/alohaeeee/Computer-Workshop`
- 6) Добавляем файлы в созданный репозиторий с помощью `git push origin`
- 7) Для получения обновлений с удаленного репозитория воспользуйтесь командой:
`git pull`
- 8) Если вы изменили ваши локальные файлы, то команда *git pull* выдаст ошибку. Если вы уверены, что хотите перезаписать локальные файлы, файлами из удаленного репозитория то выполните команды:
`git fetch --all`
`git reset --hard github/master`
- 9) В случае, если в удаленном репозитории лежат файлы с версией более новой, чем у вас в локальном, то команда `git push` выдаст ошибку. Если вы уверены, что хотите перезаписать файлы в удаленном репозитории несмотря на конфликт версий, то воспользуйтесь командой:
`git push -f`
- 10) Иногда возникает необходимость отложить ваши текущие изменения и поработать над файлами, которые находятся в удаленном репозитории. Для этого отложите текущие изменения командой:
`git stash`
- 11) После выполнения этой команды ваша локальная директория будет содержать файлы такие же, как и при последнем коммите. Вы можете загрузить новые файлы из удаленного репозитория командой *git pull* и после этого вернуть ваши изменения которые вы отложили командой: `git stash pop`
- 12) Клонирование существующего удалённого репозитория происходит с помощью команды `git clone`.