

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра «Информационная безопасность систем и технологий»

Отчет

о практической работе №1

на тему ««Системы контроля версий и совместной разработки» и «Управление  
сборкой программных проектов »

Дисциплина: ТиМП

Группа: 22ПТ2

Выполнил: Авдонин А.С.

Количество баллов:

Дата сдачи: 25 сентября

Принял: к.т.н., доцент Лупанов М.Ю.

2023

- 1 Цель работы: изучить основы git и создание Makefile.
- 2 Задания к лабораторной работе:
  - 2.1 Создать свой аккаунт на github.com
  - 2.2 Клонировать учебный репозиторий <https://github.com/MikeLupanov/StudyMake>. git в свой аккаунт на github.com
  - 2.3 Сделать локальную копию репозитория на своем компьютере
  - 2.4 Внести изменения в Readme.md на локальном компьютере, сделать коммит, и синхронизировать локальный репозиторий с сетевым
  - 2.5 Ознакомиться с примером проекта example-1 и исправить цель clean в make-файле, чтобы она корректно работала, т. е. удаляла все файлы, созданные в процессе сборки. Проверьте корректность работы цели clean.
  - 2.6 Создать в каталоге файл .gitignore для игнорирования в коммитах исполняемых и прочих файлов, создаваемых процессе сборки.
  - 2.7 Сделать все необходимые для выполнения п.п. 5 и 6 коммиты, и синхронизировать локальный репозиторий с сетевым.
  - 2.8 Ознакомиться с примером example-2 и добавить в make-файле переменную для имени файла с исходным кодом. Проверьте корректность его работы для всех целей.
  - 2.9 Создать в каталоге файл .gitignore для игнорирования в коммитах исполняемых и прочих файлов, создаваемых процессе сборки.
  - 2.10 Сделать все необходимые для выполнения п.п. 8 и 9 коммиты, и синхронизировать локальный репозиторий с сетевым.
  - 2.11 Ознакомиться с примером example-3 и создать для него make-файл. За основу можно взять make-файл из примера example-2. Проверьте корректность его работы для всех целей.
  - 2.12 Создать в каталоге файл .gitignore для игнорирования в коммитах исполняемых и прочих файлов, создаваемых процессе сборки.

2.13 Сделать все необходимые для выполнения п.п. 11 и 12 коммиты, и синхронизировать локальный репозиторий с сетевым.

2.14 Сделать отчет о выполненной работе, включив в него ссылку на свой репозиторий на сайте github.com

3 Результат работы:

3.1 Создан свой аккаунт на github.com.

3.2 Учебный репозиторий <https://github.com/MikeLupanov/StudyMake>. git клонирован в свой аккаунт на github.com. Результат представлен на рисунке 1.

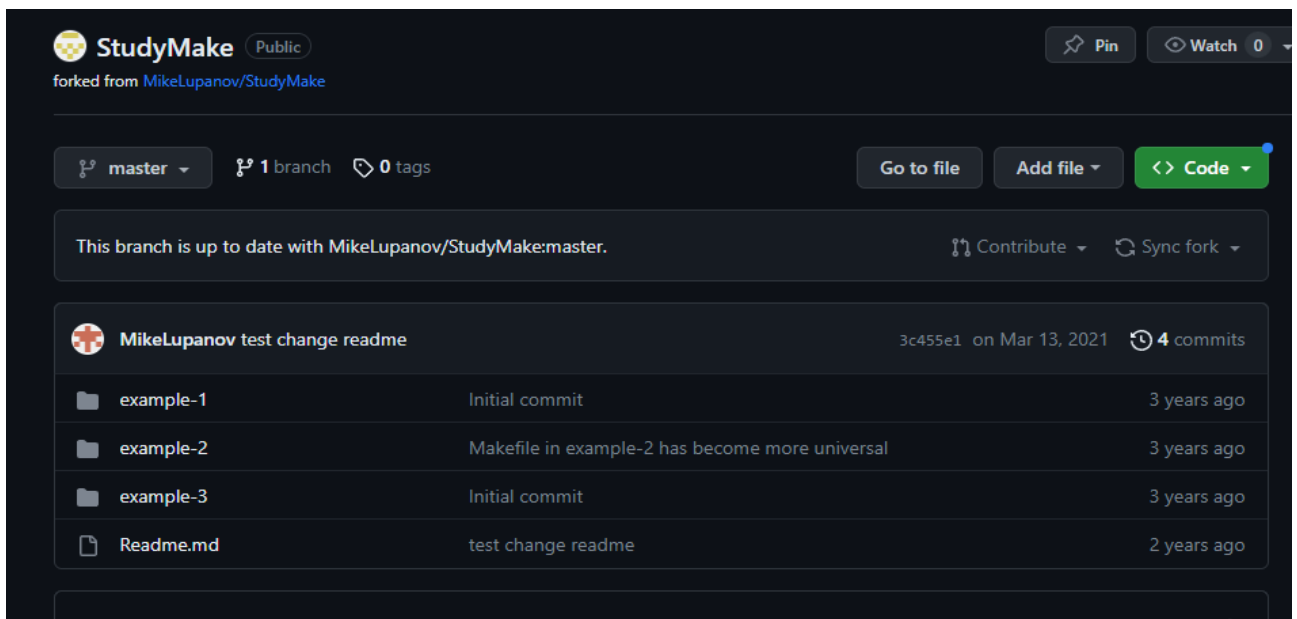


Рисунок 1 - Клонированный репозиторий

3.3 Сделана локальная копия репозитория на компьютер. Результат представлен на рисунке 2.

```
stud@virtdeb:~/Desktop/Git_Lab$ git clone git@github.com:agent-gabena/StudyMake.git
Клонирование в «StudyMake»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 20 (delta 1), reused 1 (delta 1), pack-reused 16
Получение объектов: 100% (20/20), готово.
Определение изменений: 100% (2/2), готово.
```

Рисунок 2 - Сделана локальная копия репозитория на компьютер

3.4 Внесены изменения в Readme.md на локальном компьютере, сделан коммит, и синхронизирован локальный репозиторий с сетевым. Результат представлен на рисунках 3,4.

```
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (use "git restore <file>..." to discard changes in working directory)
        изменено:   Readme.md

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git add .
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые будут включены в коммит:
  (use "git restore --staged <file>..." to unstage)
        изменено:   Readme.md

stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git commit -m"new readme"
[master c7e91d2] new readme
1 file changed, 1 insertion(+), 1 deletion(-)
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git push -u origin master
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 343 bytes | 343.00 KiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:agent-gabena/StudyMake.git
   ab7f9e1..c7e91d2  master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
```

Рисунок 3 - Синхронизация репозитория



Рисунок 4 - Внесенные изменения

3.5 example-1 и исправлен. Цель clean в make-файле, корректно работает, т. е. удаляет все файлы, созданные в процессе сборки. Проверена корректность работы цели clean. Результат представлен на рисунках 5,6.

```

stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-1$ ls
hello.cpp  Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-1$ make
g++ -g -O0 hello.cpp -o hello-dbg -Wall
g++ -O2 hello.cpp -o hello -DNDEBUG -Wall
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-1$ ls
hello  hello.cpp  hello-dbg  Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-1$ make clean
rm hello
rm hello-dbg
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-1$ ls
hello.cpp  Makefile

```

Рисунок 5 - Проверка clean

```

clean:
    rm hello
    rm hello-dbg

```

Рисунок 6 - Внесенные изменения

3.6 В example-2 изменен make-файл, добавлена переменная для имени файла с исходным кодом. Проверена корректность его работы. Результат представлен на рисунках 7,8.

```

stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-2$ ls
hello.cpp  Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-2$ make
g++ -g -O0 -Wall hello.cpp -o hello-dbg
g++ -O2 -Wall -DNDEBUG hello.cpp -o hello
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-2$ ls
hello  hello.cpp  hello-dbg  Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-2$ make clean
rm -f hello hello-dbg
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-2$ ls
hello.cpp  Makefile

```

Рисунок 7 - Проверка работоспособности Makefile

```

RELEASE_FLAGS = -O2 -Wall -DNDEBUG
DEBUG_FLAGS   = -g -O0 -Wall
RELEASE_EXEC   = hello
DEBUG_EXEC     = hello-dbg
TARGET         = hello.cpp

all: debug release

debug: $(DEBUG_EXEC)

hello-dbg: $(TARGET)
    g++ $(DEBUG_FLAGS) $(TARGET) -o $(DEBUG_EXEC)

release: $(RELEASE_EXEC)

hello: $(TARGET)
    g++ $(RELEASE_FLAGS) $(TARGET) -o $(RELEASE_EXEC)

```

Рисунок 8 - Внесенные изменения

3.7 В example-3 создан make-файл. За основу взят make-файл из примера example-2. Проверена корректность его работы. Результат представлен на рисунках 9, 10.

```

.PHONY: all debug release clean

RELEASE_FLAGS = -O2 -Wall -DNDEBUG
DEBUG_FLAGS   = -g -O0 -Wall
RELEASE_EXEC   = hello
DEBUG_EXEC     = hello-dbg
TARGET         = Kazino-Royal.cpp

all: debug release

debug: $(DEBUG_EXEC)

hello-dbg: $(TARGET)
    g++ $(DEBUG_FLAGS) $(TARGET) -o $(DEBUG_EXEC)

release: $(RELEASE_EXEC)

hello: $(TARGET)
    g++ $(RELEASE_FLAGS) $(TARGET) -o $(RELEASE_EXEC)

```

Рисунок 9 - Созданный Makefile

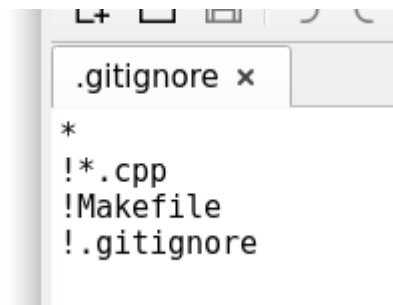
```

stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ ls
Kazino-Royal.cpp Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ make
g++ -g -O0 -Wall Kazino-Royal.cpp -o hello-dbg
g++ -O2 -Wall -DNDEBUG Kazino-Royal.cpp -o hello
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ ls
hello hello-dbg Kazino-Royal.cpp Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ make clean
rm -f hello hello-dbg
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ ls
Kazino-Royal.cpp Makefile
stud@virtdeb:~/Desktop/Git_Lab/StudyMake/example-3$ █

```

Рисунок 10 - Проверка работоспособности Makefile

3.8 Создан в каталоге файл .gitignore для игнорирования в коммитах файлов, создаваемых процессе сборки. Результат представлен на рисунке 11.



```

.gitignore x
*
!*.cpp
!Makefile
!.gitignore

```

Рисунок 11 - Содержимое gitignore

3.9 Сделаны все необходимые для выполнения п.п. 5, 6, 8, 9, 11 и 12 коммиты, и синхронизирован локальный репозиторий с сетевым. Результат представлен на рисунке 12.

```

stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые будут включены в коммит:
  (use "git restore --staged <file>..." to unstage)
    новый файл:      .gitignore
    изменено:        example-1/Makefile
    изменено:        example-2/Makefile
    новый файл:      example-3/Makefile

stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git add .
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git commit -m"make files, gitignore"
[master f0f832e] make files, gitignore
 4 files changed, 37 insertions(+), 7 deletions(-)
 create mode 100644 .gitignore
 create mode 100644 example-3/Makefile

stud@virtdeb:~/Desktop/Git_Lab/StudyMake$ git push -u origin master
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (15/15), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (9/9), 1.03 KiB | 1.03 MiB/s, готово.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:agent-gabena/StudyMake.git
   c7e91d2..f0f832e  master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
stud@virtdeb:~/Desktop/Git_Lab/StudyMake$

```

Рисунок 12 - Синхронизация изменений

3.10 Сделан отчет о выполненной работе, ссылка на гитхаб:

<https://github.com/agent-gabena/StudyMake>

4 Вывод: изучены принципы работы с Git.