

Технологии разработки программного обеспечения

Старший преподаватель Кафедры вычислительных систем
Елизавета Ивановна Токмашева

email: eliz_tokmasheva@sibgti.ru

2024, 1 курс, 2 семестр

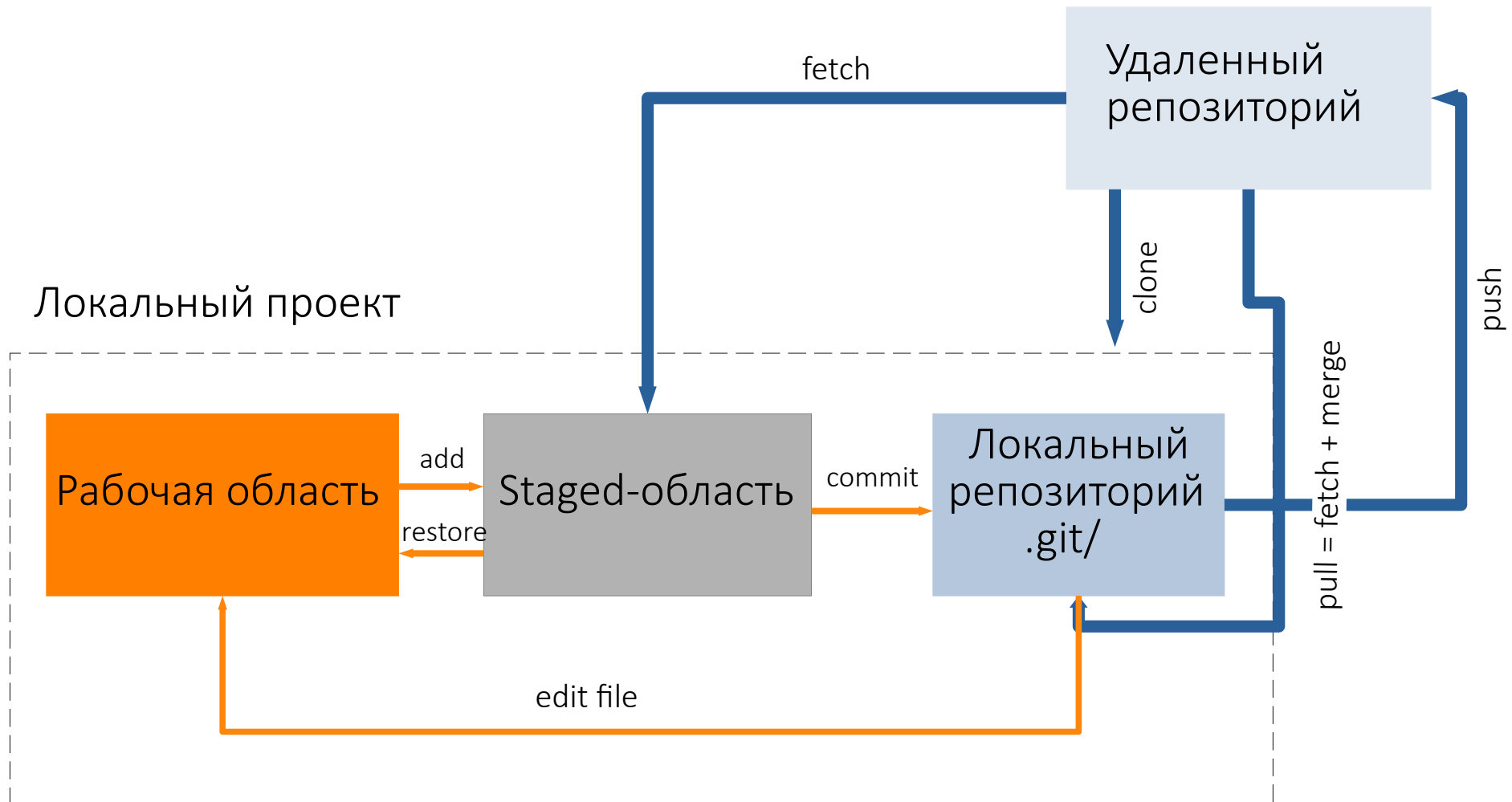
План:

- Повторение предыдущей лекции
1. Рабочие области: локальный проект, удаленный репозиторий
 2. Основные команды
 3. Состояния файлов в репозитории
 4. Коммит. Что это, как оформлять?

Лекция

Продолжение знакомства с git

Git на локальной машине

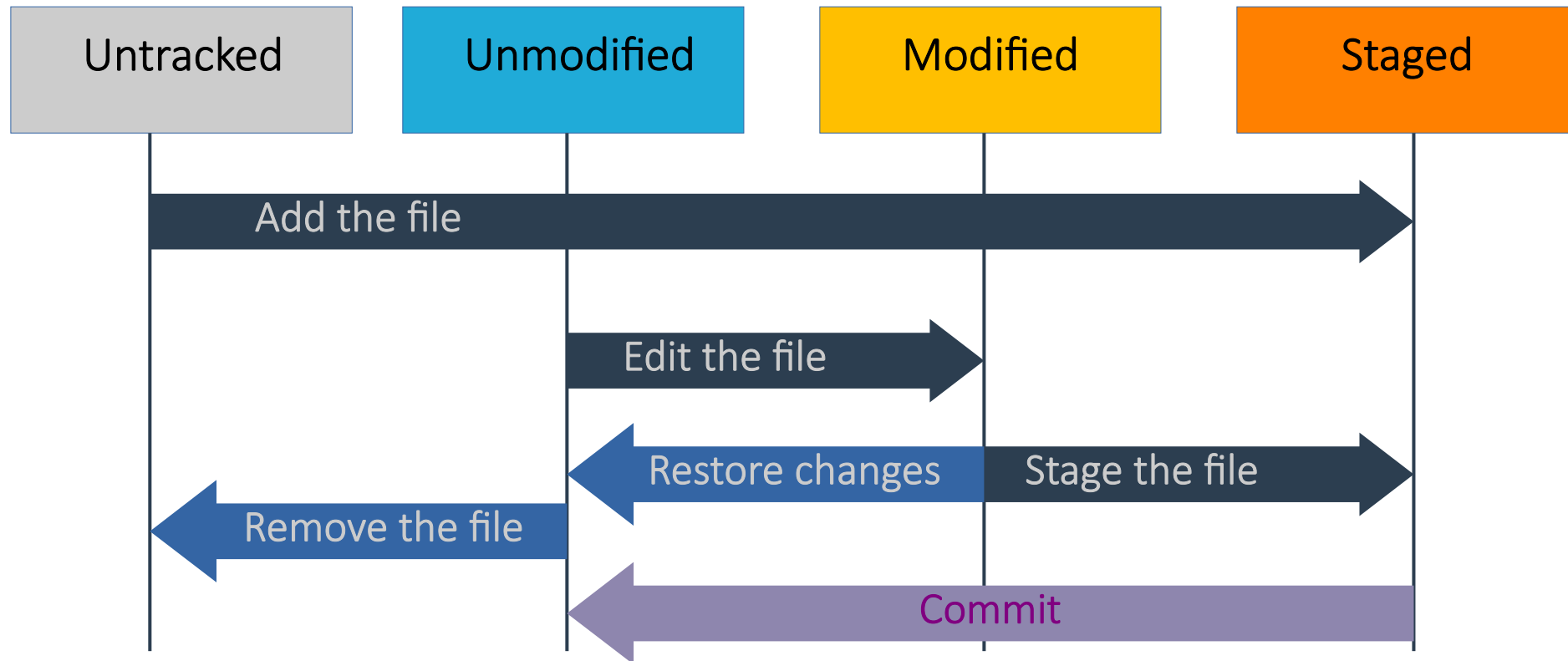


Git на локальной машине. Основные команды

- \$ git add // добавление файла под версионный контроль (индексирование файла)
- \$ git commit // фиксация изменений, создание коммита
- \$ git status // просмотр состояния файлов в репозитории
- \$ git log // просмотр истории изменений (список коммитов)
- \$ git push // обновление удаленного репозитория (загрузка сохраненных изменений на сервер)

Git на локальной машине.

Состояния файлов в репозитории



Git на локальной машине

Commit — объект или команда?

Коммит, как объект СКВ git, -- снимок изменений во времени.

Для фиксации изменений используется команда

```
$ git commit
```

Основной принцип создания коммита:

Один коммит включает в себя одно изменение;

ВАЖНО! Изменения в каждом коммите самостоятельны и независимы;

Комментарий отражает внесенное изменение;

Комментарий пишется не для себя, а для третьих лиц.

Git на локальной машине

История изменений: git log

\$ git log

Выводит историю изменений в обратном хронологическом порядке

```
eliz@LAPTOP-F6OV9FH5:~/ctest$ git log
commit cd46041baf3f131fe74032967566c7321adcaa01 (HEAD -> master, origin/master, origin/HEAD)
Author: Bas van den Berg <b.van.den.berg.nl@gmail.com>
Date:   Wed Jan 13 06:21:52 2021 +0100

    copyright: update to 2021

commit cf30ad66e99ce802b36db168d5a484f620db5131
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Sep 16 22:07:13 2020 -0400

    Remove unused uname variable

commit 78a37432bd3c5a972872d2e9239598a751f1f5c6
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Sep 16 22:06:53 2020 -0400

    Compile the example tests with both a C and C++ compiler

commit f8a83d694b2d19d5d2b2945cca37dadad944c99b
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Aug 19 23:09:41 2020 -0400

    Add basic C++ support
```

хэш-суммма коммита

Лекция 2

Игнорирование файлов (.gitignore)

Автоматическая сборка (make) однофайлового приложения

Новые полезные команды git

Git на локальной машине

Игнорирование файлов: .gitignore

```
$ cat .gitignore
```

```
# no *.o or *.a files
*.o
*.a

# no temporary files
*~

*swp

# ignore all files in the bin/ directory
bin/

# only ignore the root TODO file, not
# subbdir/TOD
TODO

# no *.txt files
*.txt

# but track input.txt
!input.txt

# ignore all *.txt files in the doc/ directory
doc/**/*.txt
```

```
eliz@LAPTOP-F60V9FH5:~/ipcalc$ git status --ignored
On branch main
Your branch is up to date with 'origin/main'.

Ignored files:
(use "git add -f <file>..." to include in what will be committed)
    bin/
    build/
    test_start

nothing to commit, working tree clean
eliz@LAPTOP-F60V9FH5:~/ipcalc$ git add bin
The following paths are ignored by one of your .gitignore files:
bin
Use -f if you really want to add them.
```

.gitignore — текстовый файл, содержащий список файлов и/или директорий исключенных из-под версионного контроля.

Название файла обязательно начинается с «ТОЧКИ».

Система автоматизации сборки make

make — утилита для преобразования файлов из одной формы в другую. Чаще всего применяется для сборки приложений.

Makefile — файл с инструкциями для исполнения утилитой make

Состоит из правил вида:

```
<цель>: [зависимость] ...  
        [команда]  
        ...
```

Команда для запуска утилиты:

\$ make

Если в локальной директории присутствуют несколько мейкфайлов или название файла отличается от названия «по умолчанию» (makefile, Makefile), необходимо явно указать make из какого файла выполнять инструкции:

\$ make -f MakeFile

Как написать простейший мейкфайл?

Задача: скомпилировать приложение из одного файла (исходника)

Команда компиляции:

```
$ gcc -Wall -Werror -o main main.c
```

Система автоматизации сборки Однофайловые приложения

\$ cat Makefile

```
all: main

main: main.c
    gcc -Wall -Werror -o main main.c

clean:
    rm main

run:
    ./main
```

Система автоматизации сборки

Однофайловые приложения

Достоинства:

- Удобно использовать для написания небольших программ
- Код программы располагается в одном файле, что упрощает навигацию по коду

Недостатки однофайлового приложения:

- Неэффективная работа несколькими разработчиками над приложением/проектом
- Затруднено ориентирование по коду в большом проекте
- Пересборка всего проекта даже при незначительных изменениях

Полезные команды git

Git на локальной машине

История изменений: log

\$ git log

Выводит историю изменений в обратном хронологическом порядке

```
eliz@LAPTOP-F6OV9FH5:~/ctest$ git log
commit cd46041baf3f131fe74032967566c7321adcaa01 (HEAD -> master, origin/master, origin/HEAD)
Author: Bas van den Berg <b.van.den.berg.nl@gmail.com>
Date:   Wed Jan 13 06:21:52 2021 +0100

    copyright: update to 2021

commit cf30ad66e99ce802b36db168d5a484f620db5131
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Sep 16 22:07:13 2020 -0400

    Remove unused uname variable

commit 78a37432bd3c5a972872d2e9239598a751f1f5c6
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Sep 16 22:06:53 2020 -0400

    Compile the example tests with both a C and C++ compiler

commit f8a83d694b2d19d5d2b2945cca37dadad944c99b
Author: Andrew Eckel <aceckel@gmail.com>
Date:   Wed Aug 19 23:09:41 2020 -0400

    Add basic C++ support
```

Git на локальной машине

История изменений: log

Хеш-сумма коммита представляет собой строку из 40 шестнадцатеричных символов (a-f, 0-9) вычисляется с помощью алгоритма SHA-1 и имеет вид:

Cf30ad66e99ce802b36db168d5a484f620db5131

Полезные опции для git log:

--graph	отображает ASCII граф с ветвлениями и историей слияний
--oneline	печатает каждый коммит в одну строку
--since --until	опции для ограничения вывода по времени
-p	Показывает патч для каждого коммита
--stat	Выводит статистику об измененных файлах

Git на локальной машине

Внесение изменений в последний коммит: amend

Для изменения заголовка коммита или добавления забытого файла в коммит используется опция `--amend`

Примеры использования:

1. Добавить в последний коммит забытый файл

```
$ git commit -m «Delete unused function»
```

```
$ git add <forgotten_file>
```

```
$ git commit --amend
```

2. Изменить заголовок последнего коммита

```
$ git commit -m «Delete unused function»
```

```
$ git commit --amend -m «Delete unused function in header file»
```

Git на локальной машине

Внесение изменений в последний коммит: amend

```
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   ipcllc.c
        modified:   ipcllc.h

no changes added to commit (use "git add" and/or "git commit -a")
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git add ipcllc.c
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git commit -m "Delete unused function"
[main b64d04c] Delete unused function
 1 file changed, 1 deletion(-)
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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:   ipcllc.h

no changes added to commit (use "git add" and/or "git commit -a")
```

Git на локальной машине

Внесение изменений в последний коммит: amend

```
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git add ipcalc.h
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   ipcalc.h

eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git commit --amend
[main 55bf148] Delete unused function
Date: Thu Feb 10 00:40:58 2022 +0700
2 files changed, 3 deletions(-)
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Git на локальной машине

Внесение изменений в последний коммит: amend

```
eliz@LAPTOP-F6OV9FH5:~/ipcalc/src$ git log --stat
commit 55bf148cfa9ea7e72b7d70401ae03591757782e4 (HEAD -> main)
Author: name <>
Date: Thu Feb 10 00:40:58 2022 +0700

    Delete unused function

src/ipcalc.c | 1 -
src/ipcalc.h | 2 --
2 files changed, 3 deletions(-)

commit e6a3f9ffd7c5c2d8cf258eeb3418b7c811e43f44
```

Git на локальной машине

Отмена изменений: reset

Откат изменений без сохранения внесенных изменений

```
$ git reset--hard HEAD~1
```

или

```
$ git reset--hard e6a3f9f
```

```
eliz@LAPTOP-F6OV9FH5:~/ipcalc$ git reset --hard HEAD~1
HEAD is now at e6a3f9f Rename makefile
eliz@LAPTOP-F6OV9FH5:~/ipcalc$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
eliz@LAPTOP-F6OV9FH5:~/ipcalc$ git hist
* e6a3f9f (HEAD -> main) Rename makefile
```

Git на локальной машине.

git diff

Просмотреть внесенные изменения, но которые еще не проиндексированы

```
$ git diff
```

Команда сравнивает содержимое рабочего каталога с содержимым индекса.

```
eliz@LAPTOP-F6OV9FH5:~$ cat main.c
#include <stdio.h>

int main()
{
    printf("Hello, World!");
    return 0;
}
```

```
eliz@LAPTOP-F6OV9FH5:~$ git diff
diff --git a/main.c b/main.c
index 8ccd62a..7d38eaf 100644
--- a/main.c
+++ b/main.c
@@ -2,6 +2,6 @@

int main()
{
-    printf("Hello, World!");
+    printf("Hello, World!\n");
    return 0;
}
```


Git на локальной машине

git diff

Если требуется посмотреть что уже проиндексировано и что попадёт в коммит выполните

```
$ git diff --staged
```

Команда сравнивает ваши индексированные изменения с

ПОСЛЕДНИМ КОММИТОМ

```
eliz@LAPTOP-F60V9FH5:~$ git diff
diff --git a/main.c b/main.c
index 7d38eaf..cccdef2 100644
--- a/main.c
+++ b/main.c
@@ -3,5 +3,6 @@
 int main()
 {
     printf("Hello, World!\n");
+    printf("What a beautiful day!");
     return 0;
 }
```

```
eliz@LAPTOP-F60V9FH5:~$ git add main.c
eliz@LAPTOP-F60V9FH5:~$ git diff
eliz@LAPTOP-F60V9FH5:~$ git diff --staged
diff --git a/main.c b/main.c
index 7d38eaf..cccdef2 100644
--- a/main.c
+++ b/main.c
@@ -3,5 +3,6 @@
 int main()
 {
     printf("Hello, World!\n");
+    printf("What a beautiful day!");
     return 0;
 }
```