

Командная строка. GIT

@pvavilin

1 апреля 2022 г.

Timestamp

Что такое timestamp?

```
import time  
  
print(time.time())
```

Timestamp

Что такое timestamp?

```
import time  
  
print(time.time())  
1648837018.2793298
```

Timestamp

Что такое timestamp?

```
import time
from datetime import datetime

t = time.time()
print(
    datetime.utcfromtimestamp(t)
)
2022-04-01 18:16:58.344531
```

Timestamp

Что такое timestamp?

А что будет если вызвать такой код?

```
from datetime import datetime  
  
print(  
    datetime.utcfromtimestamp(0)  
)
```

Timestamp

Что такое timestamp?

```
from datetime import datetime  
  
print(  
    datetime.datetime.utcnow()  
)  
1970-01-01 00:00:00
```

UNIX

1970.1.1

Дата отсчёта UNIX-эпохи



UNIX

Главные принципы UNIX:

- Всё есть файл (ну или почти всё)
- Каждая программа должна уметь делать **только одно** и делать это хорошо!

UNIX-way

Shell

- *sh*
- *bash*
- *zsh*
- *csh*
- *fish*
- *ksh*
- *tcsh*

Основные команды Bash (sh, zsh)

ls

```
|ls ~/workdir/generate-password/  
gen_pass  
README
```

Основные команды Bash (sh, zsh)

```
ls -l
```

```
|ls -l ~/workdir/generate-password/  
total 8  
drwxr-xr-x 3 pimiento pimiento 4096 Jun 12  2021  
gen_pass  
-rw-r--r-- 1 pimiento pimiento  420 Apr 13  2021  
README
```

Основные команды Bash (sh, zsh)

```
ls -a
```

```
|ls -a ~/workdir/generate-password/  
.  
..  
gen_pass  
.git  
README
```

Основные команды Bash (sh, zsh)

which

```
|which python3  
/usr/bin/python3
```

Основные команды Bash (sh, zsh)

```
mkdir
```

```
|mkdir /tmp/notexist
```

Основные команды Bash (sh, zsh)

```
rm
```

```
|rm -r /tmp/notexist
```

Основные команды Bash (sh, zsh)

```
grep
```

```
|grep pimiento /etc/passwd
```

```
pimiento:x:1000:1000:Pavel
```

```
Vavilin,,,:/home/pimiento:/bin/zsh
```


Основные команды Bash (sh, zsh)

cat

```
|cat /etc/lsb-release
```

```
DISTRIB_ID=LinuxMint
```

```
DISTRIB_RELEASE=20.3
```

```
DISTRIB_CODENAME=una
```

```
DISTRIB_DESCRIPTION="Linux Mint 20.3 Una"
```

Основные команды Bash (sh, zsh)

перенаправление

- 0 => stdin
- 1 => stdout
- 2 => stderr

Основные команды Bash (sh, zsh)

перенаправление

```
|rm -r /tmp/notexist 2> /tmp/rm.log  
|cat /tmp/rm.log  
rm: cannot remove '/tmp/notexist': No such file  
or directory
```

Основные команды Bash (sh, zsh)

pipe

Выход одной команды мы можем подать на вход другой команде!

```
| echo "Привет Яндекс Практикум" | wc -c  
45
```

```
| cat /etc/lsb-release | grep RELEASE \  
| cut -d '=' -f 2 | cut -d '.' -f 1  
20
```

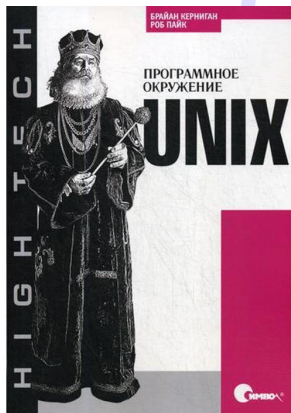
Основные команды Bash (sh, zsh)

```
| cat /dev/urandom | hexdump -c
```

Время стать крутыми хакерами!

```
013=40  006 202 0 3 217 0 031 0 222 231 0 630 025 220
01459=0  203 202 0 217 0 031 0 222 231 0 630 025 220
0151130  202 0 201 236 202 225 0 210 0 230 0 0 0
015=40  0 0 227 003 215 4 214 0 211 0 217 0
0167100 203 0 021 +220 226 0 001 224 210 0 0 0
016=340 0 0 0 221 201 0 4 0 237 217 027 210
016=40 031 006 0 0 203 0 0 222 201 001 0 221 0
01725=0 036 033 0 225 0 025 021 0 0 032 0 0 0
0177860 034 215 207 0 0 202 0 0 206 0 201 216 0
0177=0 0 0 0 224 0 0 223 0 206 0 200 0 035 213
0180090 231 0 177 5 220 213 0 0 0 1 237 3 0 227
0196280 205 -033 7 237 0 0 0 225 224 0 0 0 0
0197=0 0 0 026 016 0 0 025 224 0 0 0 0
0105420 206 024 0 0 0 0 0 005 224 0 022 0 0 0
0107430 0 215 1 9 0 020 030 9 0 3 1 017 0
01159=0 010 004 210 0 4 0 0 0 0 0 0 0 0
011=800 032 0 0 -033 6 213 0 215 0 204 0 220 0 230 0
011=100 0 0 235 0 225 0 227 0 030 214 0 237 213 004 0
011=100 0 0 224 0 002 0 202 0 0 0 0 0 0 0
011=950 4 0 0 0 0 020 0 0 0 202 022 0 0 0
011=360 226 020 032 212 0 0 0 237 235 0 0 002 0 223
011=534=0 021 0 005 0 0 204 017 0 0 0 0 0
011=890 0 5 210 235 0 0 0 207 230 216 223 0 0 0
020=540 0 030 206 0 0 0 0 4 7 6 215 0 003 0
020=020 0 6 2 017 214 0 0 0 0 0 003 0 0
021=206 237 0 0 034 231 0 227 0 020 0 222 0
02230=0 0 0 005 0 200 0 6 7 0 0 0 0 0
0235370 0 0 025 0 0 0 0 0 0 235 221 0 0 017
0237=50 037 0 4 037 0 8 0 020 234 0 0 004 0 215
023=440 0 222 0 0 0 030 0 2 2 0 235 0
0245120 0 215 0 203 030 0 221 0 033 236 0 0
0256030 2 3 200 0 024 037 0 3 036 0 0 224
026370 0 0 0 0 036 0 0 0 7 200 017 0 0
026=190 0 0 021 0 0 0 0 0 017 0 0 0 0
027170 0 1 005 0 0 230 0 0 0 0 025 0 0 3
027710 223 210 221 022 025 206 033 020 033 0 212 023 0 211 4
027=0 217 021 0 236 233 5 0 0 003 220 223 0 0 027
028=140 0 005 0 0 0 7 021 0 0 035 0 0 0 0
028=100 0 0 0 035 216 0 0 0 0 023 0 0 0
02870 0 0 023 030 225 023 222 0 0 223 0 030
02=9070 227 217 0 0 207 026 7 206 0 214 003 4 0 236 222 4
```

UNIX это интересно!



`#!/usr/bin/env python`

Шебэ́нг? Шебанг? Shebang.

hello.py

```
#!/usr/bin/env python3  
print("Hello World!")
```

\$>

```
chmod +x hello.py
```

Что почитать?

Про UNIX идеологию в целом

Брайн Керниган, Роб Пайк «Программное окружение UNIX»

Bash

«Advanced Bash-Scripting Guide». На OpenNet можно найти перевод

SSH

(В MS Windows чаще всего используют *PuTTY*)

```
|ssh user@ya.ru  
|scp user@ya.ru:/answers.txt ./
```

VI

Редактор де-факто на всех UNIX-like системах

как выйти из

как выйти из **vim**

как выйти из **запоя**

как выйти из **зоны комфорта**

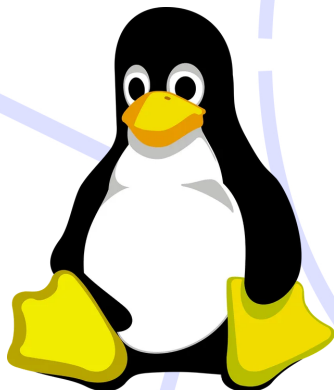
как выйти из **депрессии**

VI

МЕМАСИКИ

шпаргалку по Vi/Vim пришлю после вебинара

Linux. При чём тут GIT?



GIT

- GitHub
- BitBucket
- GitLab
- ...

GIT

Время приключений!



.gitignore

glob

```
mkdir /tmp/yap
cd /tmp/yap
python3 -m venv venv
cd /tmp/yap/venv
ls lib/*/*/*/*.py | head -n 5

lib/python3.8/site-packages/pip/__init__.py
lib/python3.8/site-packages/pip/__main__.py
lib/python3.8/site-packages/pkg_resources/__init__.py
lib/python3.8/site-packages/pkg_resources/py31compat.py
lib/python3.8/site-packages/setuptools/archive_util.py
```

.gitignore

```
bash>
```

```
|man 7 glob |head -n5
```

```
GLOB(7)
```

```
Linux Programmer's Manual
```

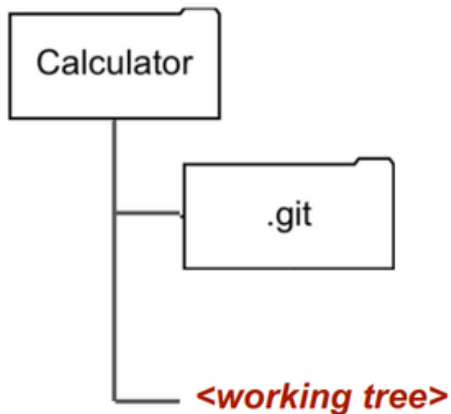
```
NAME
```

```
glob - globbing pathnames
```

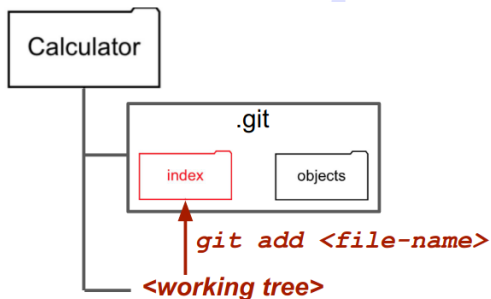
```
python>
```

```
|import glob
```

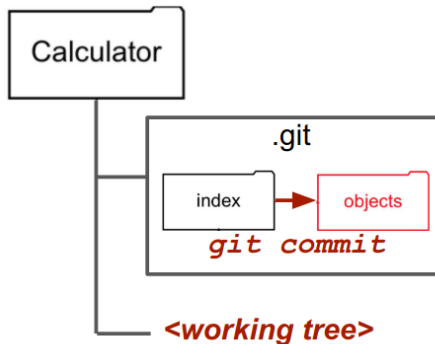

Что внутри .git



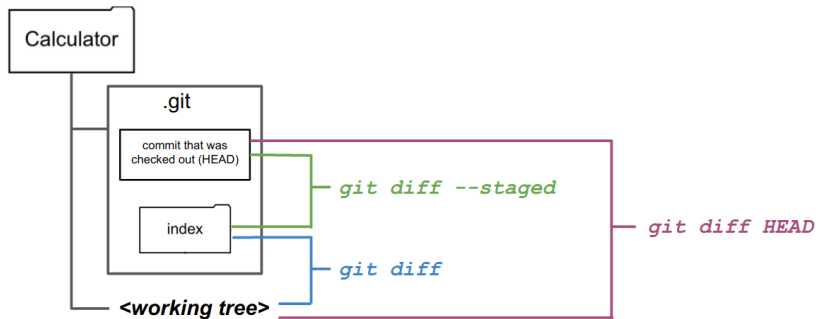
Что внутри .git



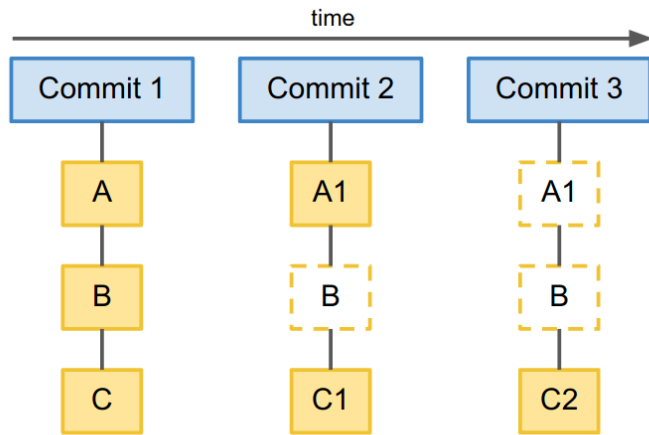
Что внутри .git



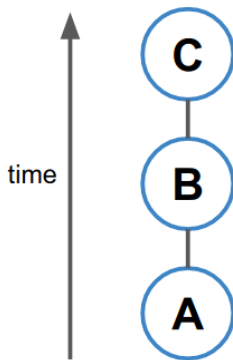
diff



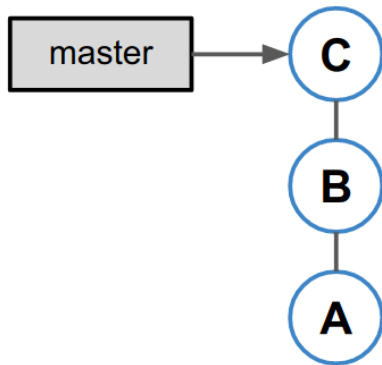
Commits



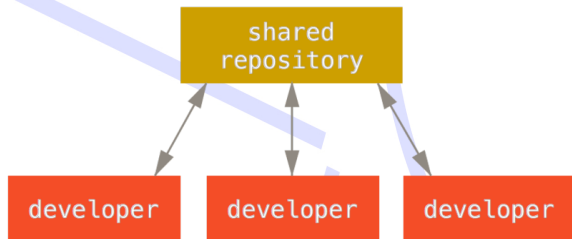
Commits



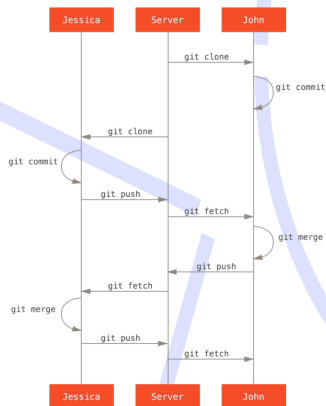
Commits



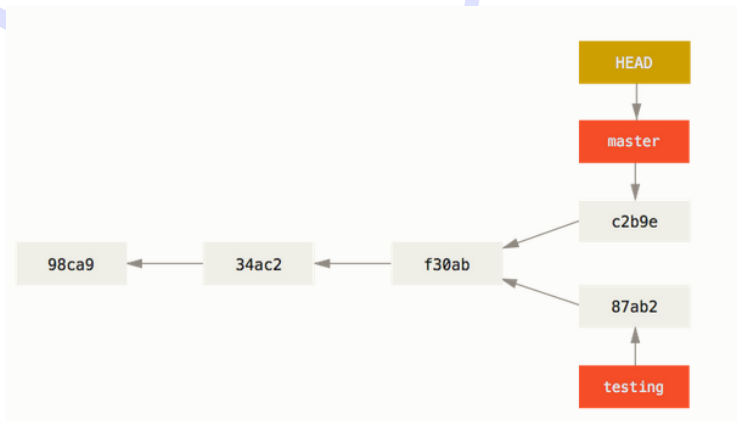
Совместная разработка



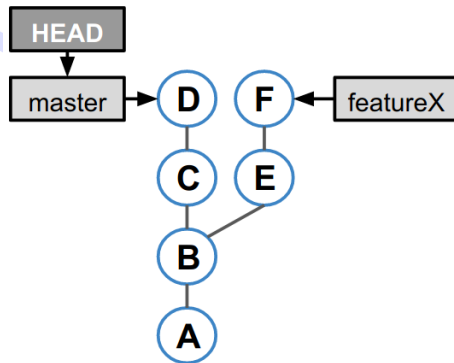
Совместная разработка



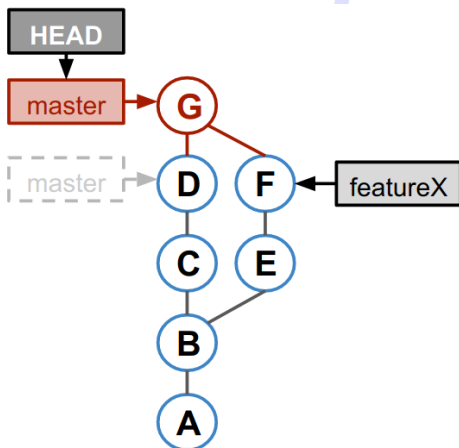
Branches



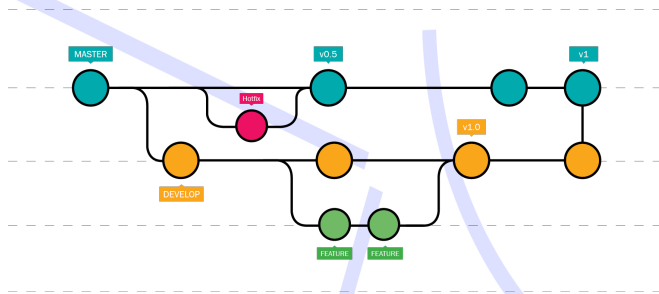
Merge



Merge

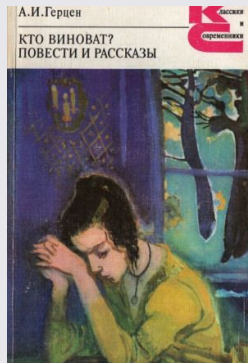


Merge



Кто виноват?

git blame



Кто виноват?

```
git blame
```

```
|git blame
```

Откат изменений файла

HEAD

Снимок последнего коммита,
родитель следующего

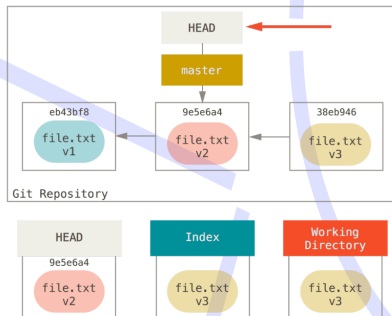
Индекс (Index)

Снимок следующего намеченного
коммита

Рабочий каталог (Working tree)

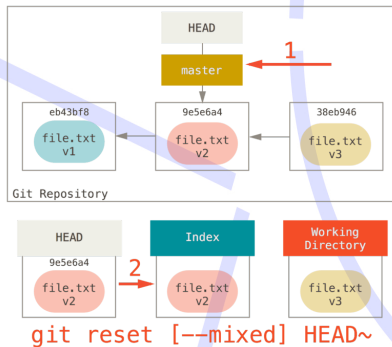
Песочница

git reset

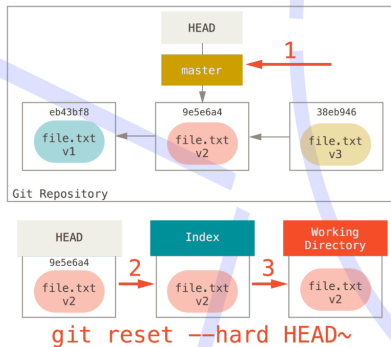


`git reset --soft HEAD~`

git reset



git reset



git checkout

git reset

```
git reset --hard [branch] file
```

Так не может быть, потому что *git reset* перемещает указатель HEAD, а он не может указывать на несколько мест одновременно.

git checkout

git checkout

```
git checkout [ --patch ] <hash> -- file
```

Замещает содержимое файла в текущем HEAD на содержимое из коммита <hash>.

GIT. что почитать?

Классика

GIT

Советую (english)

GIT Concepts and Workflows

Спасибо!

Задавайте ваши вопросы

