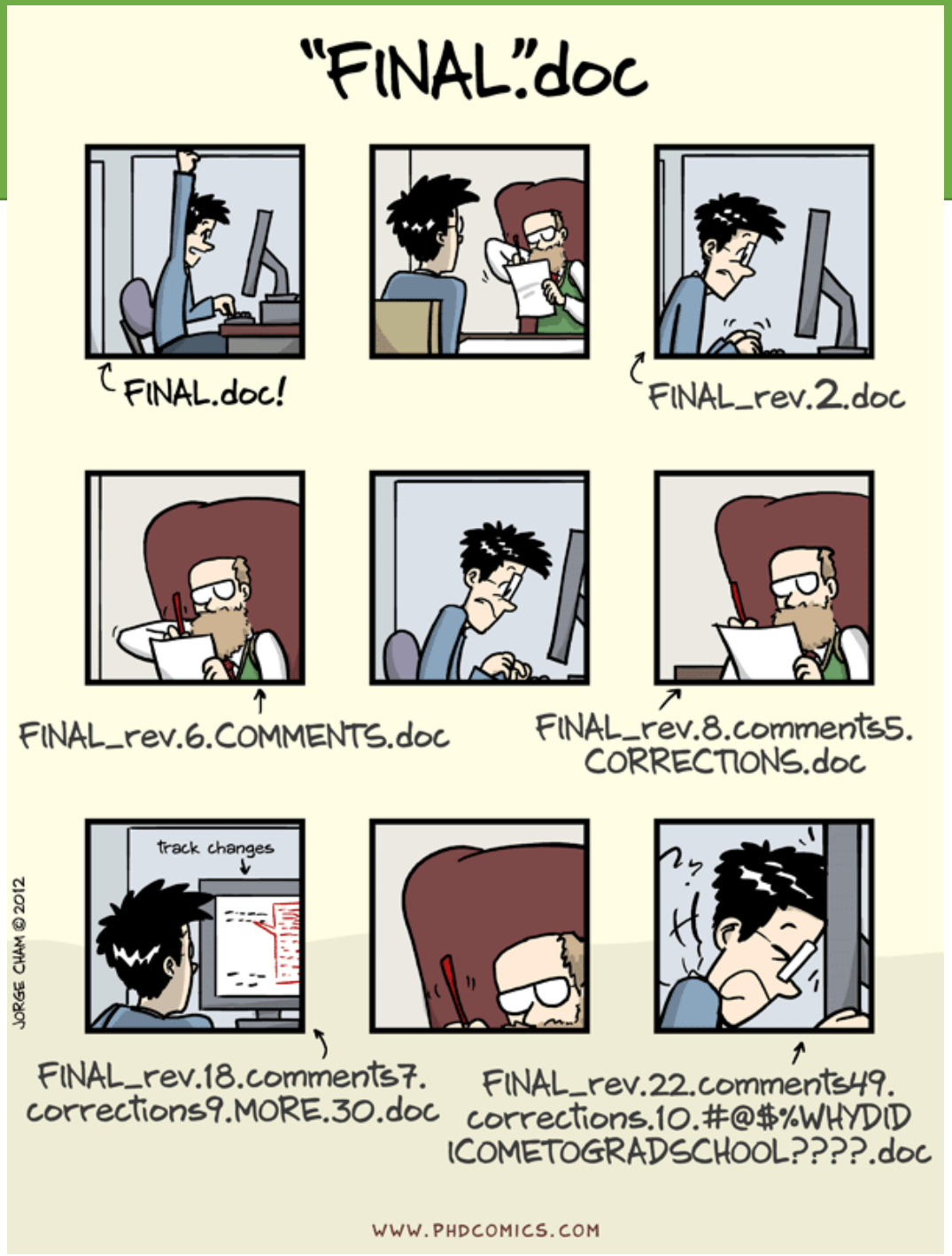
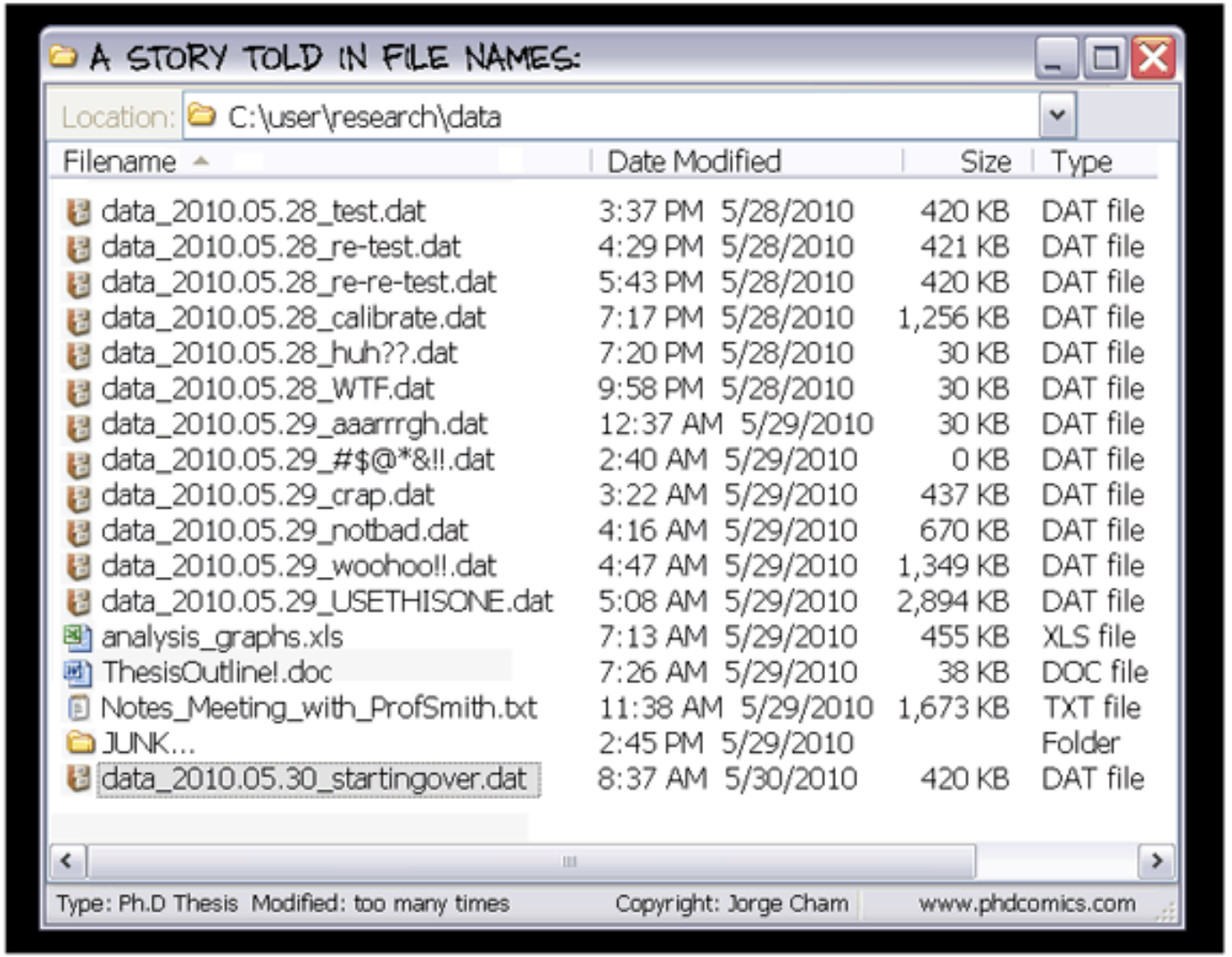
Git

Как перестать терять свои наработки
и начать жить

К чему такие сложности?





Почему это важно?

- Эффективно
- Удобно работать в команде
- Всегда можно откатиться к рабочей версии
- Ничего не пропадает (если было сохранено)
- Удобный путь к автоматизации процесса

Основные моменты

- Сохранять (почти) все, как только оно создано
- Изменения должны быть небольшими
- Сохранять часто
- Хранить каждый проект в отдельной папке

https://swcarpentry.github.io/good-enough-practices-in-scientific-computing/

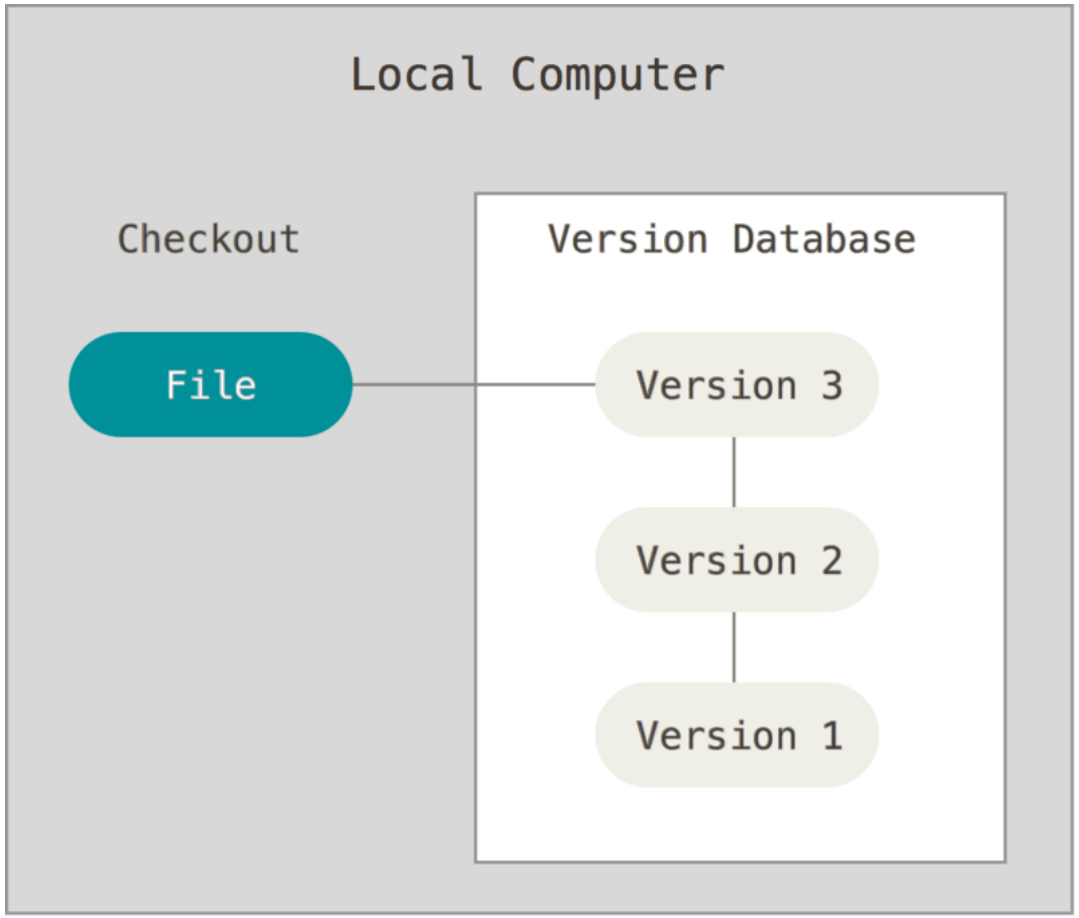
Пример. Хранение версий вручную

- Как хранить различные версии?
- Как их различать?
- Где держать список изменений для каждого файла?

Работа в команде:

- Как обрабатывать изменения от каждого члена команды?
- Как разрешать конфликты?

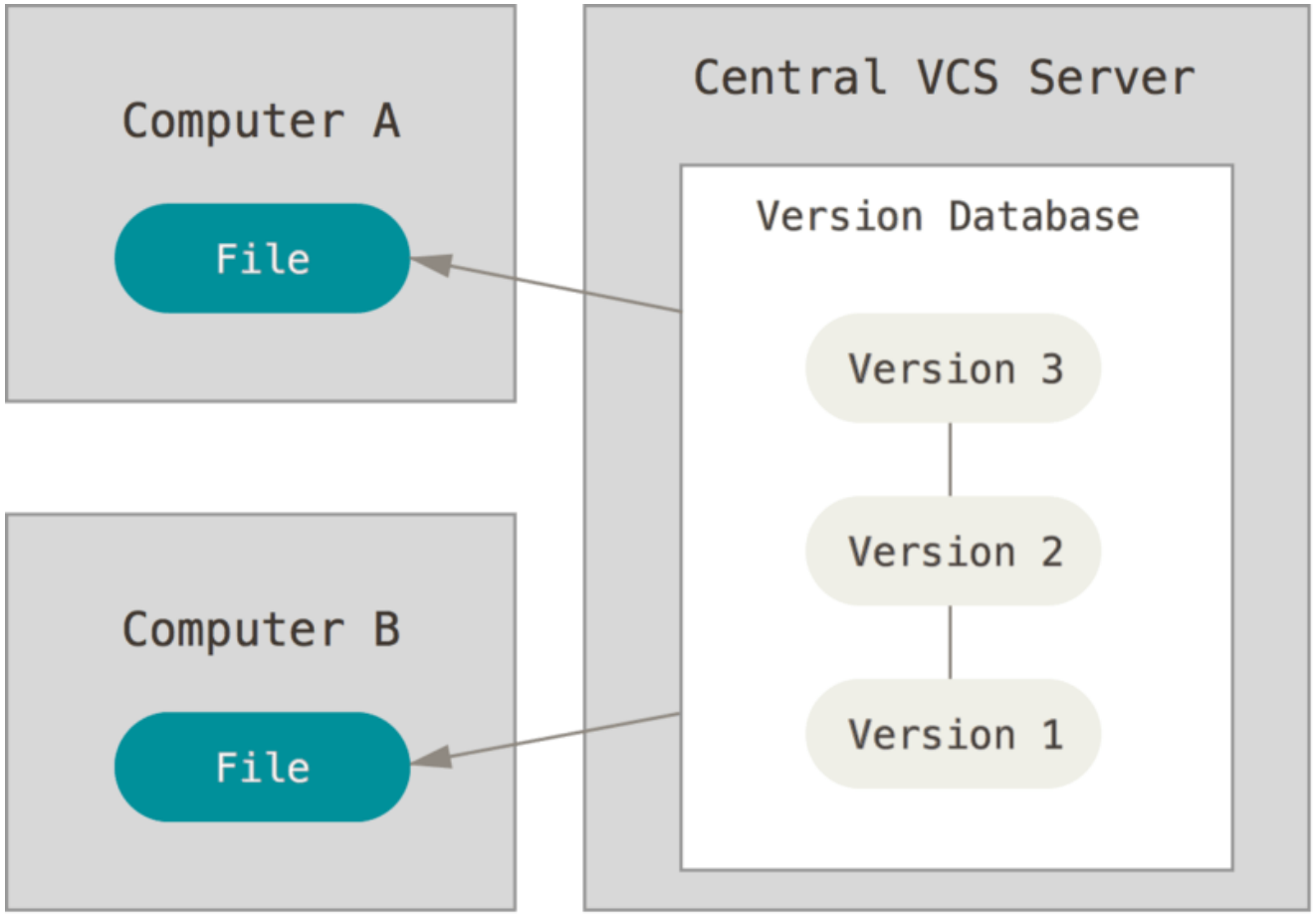
Разные подходы к VCS



Централизованный:

локально

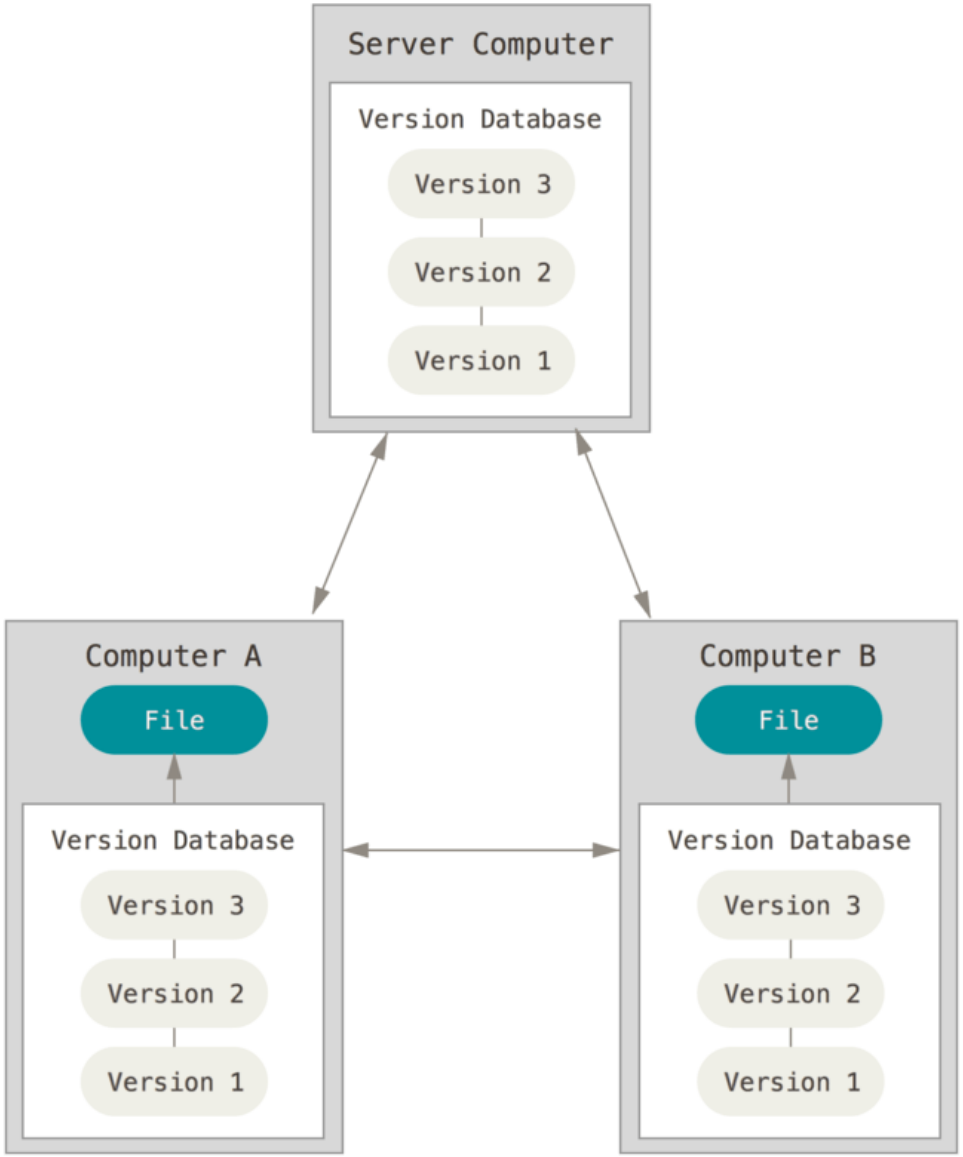
Разные подходы к VCS



Централизованный:

«главный» сервер
хранит всю историю

Разные подходы к VCS



Распределённый:

каждый компьютер хранит свою копию истории

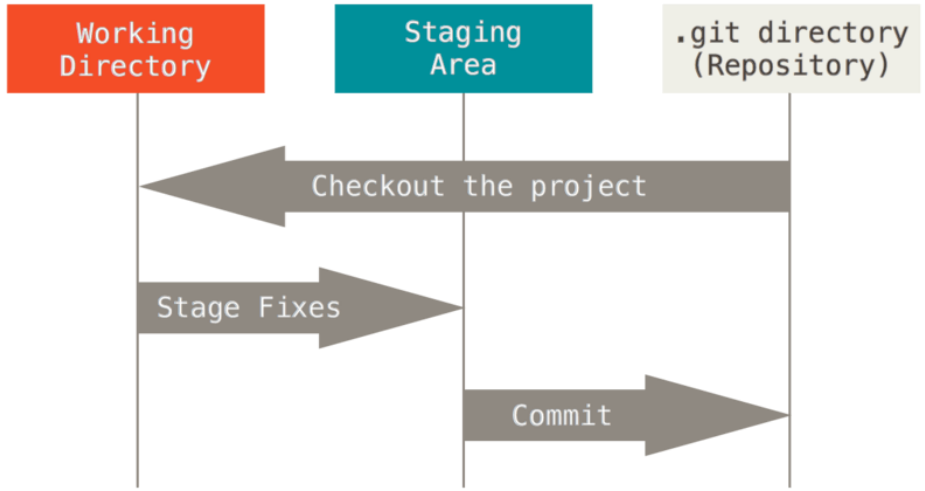
Git

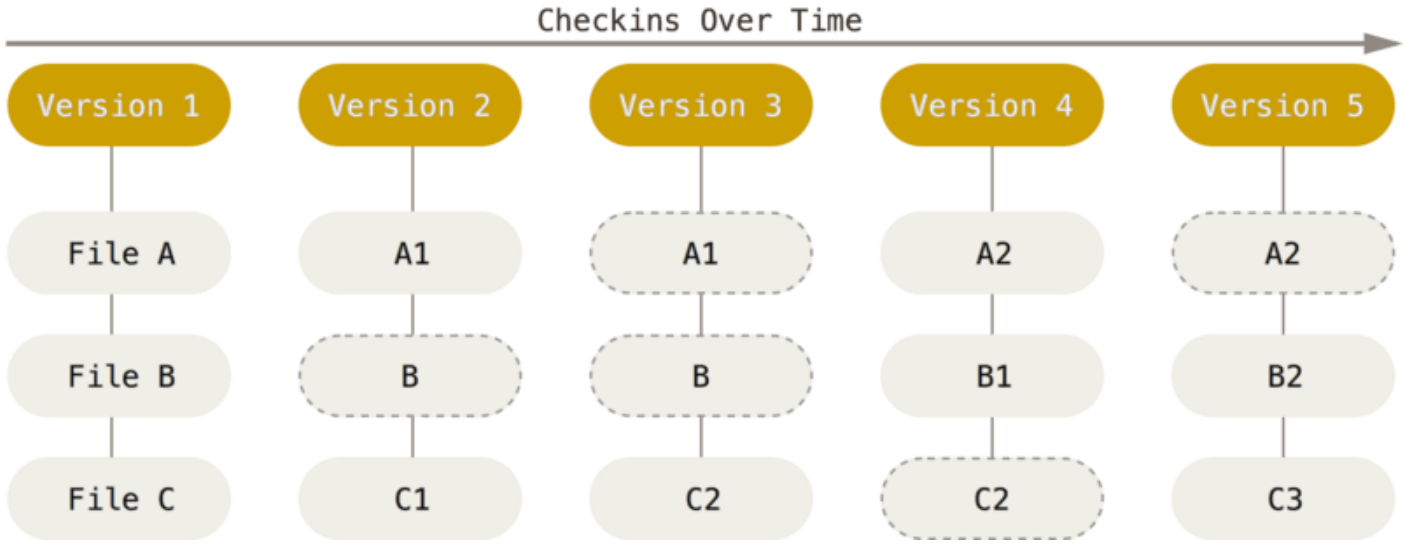
Сделан для работы с ядром Linux в 2005 году

- Скорость
- Простая архитектура
- Хорошая поддержка нелинейной разработки (тысячи параллельных веток)
- Полная децентрализация
- Возможность эффективного управления большими проектами, такими как ядро Linux (скорость работы и разумное использование дискового пространства)

Философия

- Снимки вместо различий
- Локальность (почти) всех операций
- Целостность
- Не знаешь что делать – добавляй данные





Установка / настройка

Linux:

```
sudo apt install git
```

Windows/Mac:

https://git-scm.com/download/win

- Имя пользователя и адрес электронной почты

Общий синтаксис

```
git <subcommand> [switches] arguments
```

Например

```
git clone --depth 10 https://github.com/lammps/lammps.git
git log --graph
```

Первоначальная настройка

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

Встроенная справка

```
git help [subcommand]
```

Например

```
git help commit
git help –a

git help tutorial
```

Кратко

```
git <subcommand> -h
```

Начало работы

```
git init
```

```
git clone <URL>
```

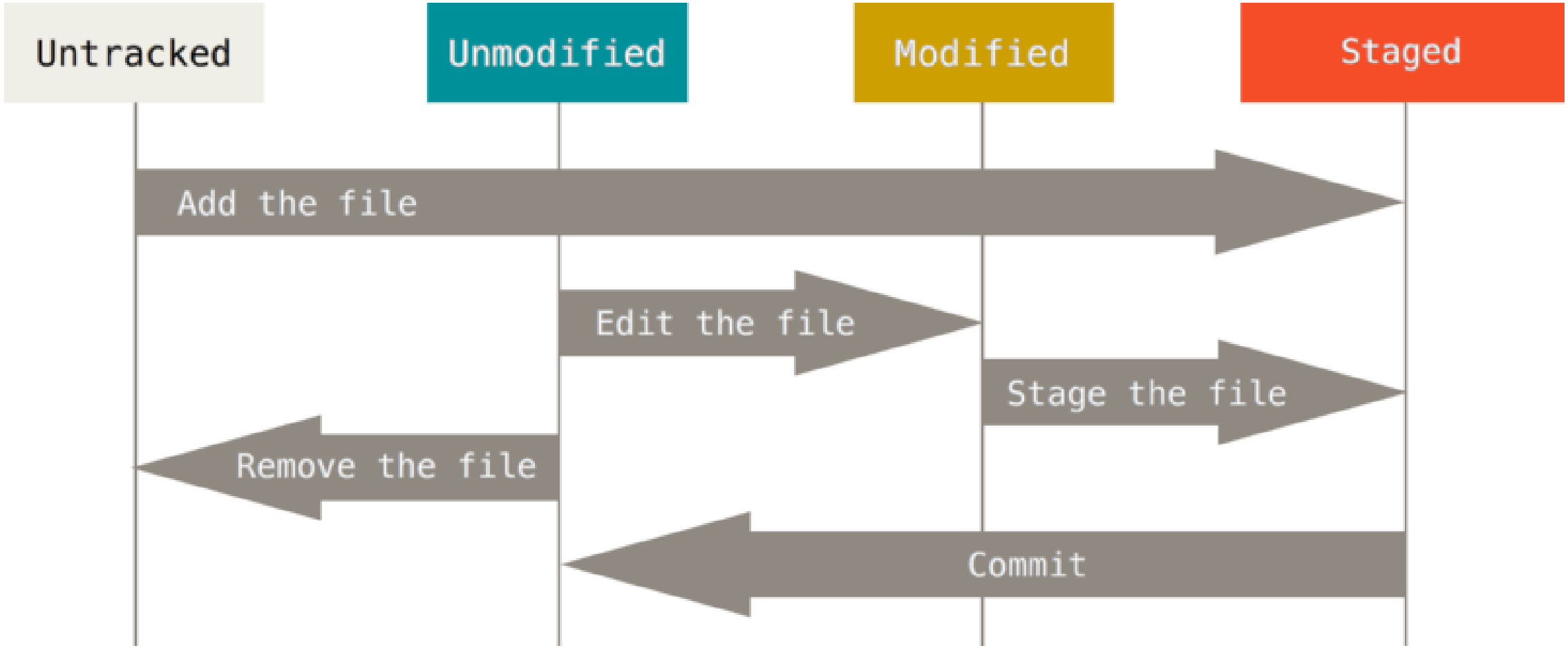
4 состояния файлов в репозитории

неотслеживаемый

неизмененный

измененный

проиндексированный



Проверка состояния

```
git status
```

```
git diff
```

Сохранение изменений

Добавление в индекс

```
git add <file>
```

Коммит

```
git commit
```

Для уже проиндексированных файлов это проще:

```
git add -u
```

Задание

- Установить, настроить git
- Создать репозиторий
- Создать в нем новый файл
- Проиндексировать его
- Записать изменения в историю (сделать коммит)
- Между каждым шагом смотреть статус репозитория

Игнорирование файлов

Не все файлы одинаково полезны.

Специальный файл `.gitignore` хранит,
какие файлы сохранять не надо.

1. Пустые строки или строки, начинающиеся с #, игнорируются.
2. Работают обычные glob-шаблоны, причем они применяются рекурсивно ко всему дереву каталогов внутри репозитория.
3. Если начать шаблон с /, то рекурсии не будет.
4. Если шаблон закончить /, то это явное указание на папку.
5. Если начать шаблон с !, то его смысл будет обратным.

```
libs-sources/
build/

# result files
*.kmc4.txt
*.kmc4.yml
*.kmc4.c.txt
/results/*
/brown/*
/steps-deposition-times/*
*.parquet
*.log

# renders
/renders/
/*.max
/*.ms
/plane-texture

# archives of them
*.tar.bz2
*.zip
*.tar
*.7z

# executables
main
/main-talk
/main-debug
/main-steps
main.exe
/main-talk.exe
/main-debug.exe
/main-steps.exe
command.sh
*.stackdump
a.out
*.exe

# produced by ipynb
*.pdf
*.html
*.eps
.ipynb_checkpoints
/kMC4.html
cachedir/
```

Частные случаи

Удаление файла

```
git rm <file>
```

Переименование файла

```
git mv <file>
```

Чтение истории

`git log`

```
$ git log
commit 2777984355534d1c402965576538c8295f29da17 (HEAD)
Merge: 87dae19019 3fb8857be5
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Tue Feb 4 09:51:39 2025 -0500

    Merge pull request #4461 from akohlmey/last-minute-fixes

    Last minute fixes for next feature release

commit 3fb8857be50980934ee6cbf4be9417204430200c
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Mon Feb 3 22:19:12 2025 -0500

    relax epsilon for tests on ARM64

commit 113b2e47f078a6923d753ce04ce0431328f2d6a0
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Mon Feb 3 22:18:45 2025 -0500

    must set val.iarg before processing any arguments
```

```
$ git log --oneline --graph
*   35214120ad (HEAD -> develop, origin/develop, origin/HEAD) Merge pull request #4025 from oywg11/ilp-
|\
| * e98df7018b update the example files
| * 40f27eb7cf update doc file for ilp/tmd
| * 20183ac9cc update potential file for ilp/tmd
| * 8dcf980d0a update doc of ilp/tmd
| * ba4ac991b6 small fix for ilp/tmd and KC/full
* |   6a636d5491 Merge pull request #4023 from ndtrung81/unittest-debug
|\ \
| * | 6c798412b4 fix typo and remove unnecessary quotes
| * | af222711ae Added text to explain the output of ctest -V a bit more
| * | 188e1090e9 add some corrections and clarifications
| * | 58ed034d7a Updated Developer unittest for debugging failed unit tests individually
| |/
* |   6489e475b9 Merge pull request #4022 from akohlmey/cmap-fixes-for-charmm-gui
|\ \
| |/
|/|
| * 695a81ef70 avoid uninitialized data access
| * 1ab406ee1a read CMAP data blocks one at a time and catch EOF exception to stop reading
| * 4bf1b1d9c0 some refactoring and modernization
| * fb6a5843b9 handle comments in CMAP coeff assignments
| * 96ef731f06 remove unused items
| * e100a42087 (re)throw EOF exception when next_dvector() has not yet read any items
|/
*   df7f3b8dea Merge pull request #3938 from rbberger/compute_reaxff_bonds_local
|\
| * 1df91f21a1 workaround hack for macOS
| * 7dab2b7eee add new package files to .gitignore
| * 603837c96c add versionadded tag
| *   a98ea8c3b2 Merge branch 'develop' into compute_reaxff_bonds_local
| |\
| * \   37cb1ce30f Merge branch 'develop' of github.com:lammps/lammps into compute_reaxff_bonds_local
| |\ \
| * | | 0cf4c9e7a3 Whitespace
| * | | 930fbe8c5d reaxff/atom: First attempt of filtering by group
| * | | e241f08cfe Finish first version of compute reaxff/atom docs
| * | | b72c34d497 compute reaxff/atom: return tag[i] instead of i
| * | | a5cc181358 Start with compute reaxff/atom documentation
| * | | fd83ed4004 compute reaxff/atom: add support for pair hybrid
| * | | 16f0806da0 Rename compute to reaxff/atom
```

Задание

- Добавить несколько новых файлов
- Потренироваться с игнорированием
- Сделать несколько новых коммитов
- Посмотреть историю репозитория

Теги

```
git tag
```

```
$ git tag --column
lammps-gui-v1.5          patch_22Dec2022          patch_4Feb2020
patch_10Aug2017          patch_22Feb2018          patch_4Feb2025
patch_10Feb2021          patch_22Jun2018          patch_4Jan2019
patch_10Mar2017          patch_22Nov2016          patch_4May2017
patch_10Mar2021          patch_22Oct2020          patch_4May2022
patch_10Oct2018          patch_22Sep2017          patch_4Nov2016
patch_11Apr2017          patch_23Jun2017          patch_5Feb2018
patch_11Aug2017          patch_23Jun2022          patch_5Jun2019
patch_11May2018          patch_23Jun2022_update1  patch_5May2020
patch_11Oct2016          patch_23Jun2022_update2  patch_5Nov2016
patch_12Dec2018          patch_23Jun2022_update3  patch_5Oct2016
patch_12Oct2016          patch_23Jun2022_update4  patch_5Sep2018
patch_13Apr2017          patch_23Oct2017          patch_6Aug2019
patch_13Dec2016          patch_24Aug2020          patch_6Jan2017
patch_13Feb2017          patch_24Dec2020          patch_6Jul2017
patch_13Oct2016          patch_24Jan2020          patch_6Oct2016
```

Откат изменений

Откат коммита

```
git commit --amend
```

Раз-индексирование файла

```
git restore --staged <file>
```

Отмена изменений до прошлого коммита

```
git restore <file>
```

Задание

- Добавить несколько новых тегов
- Потренироваться с откатом изменений

Удаленные источники

```
git clone
```

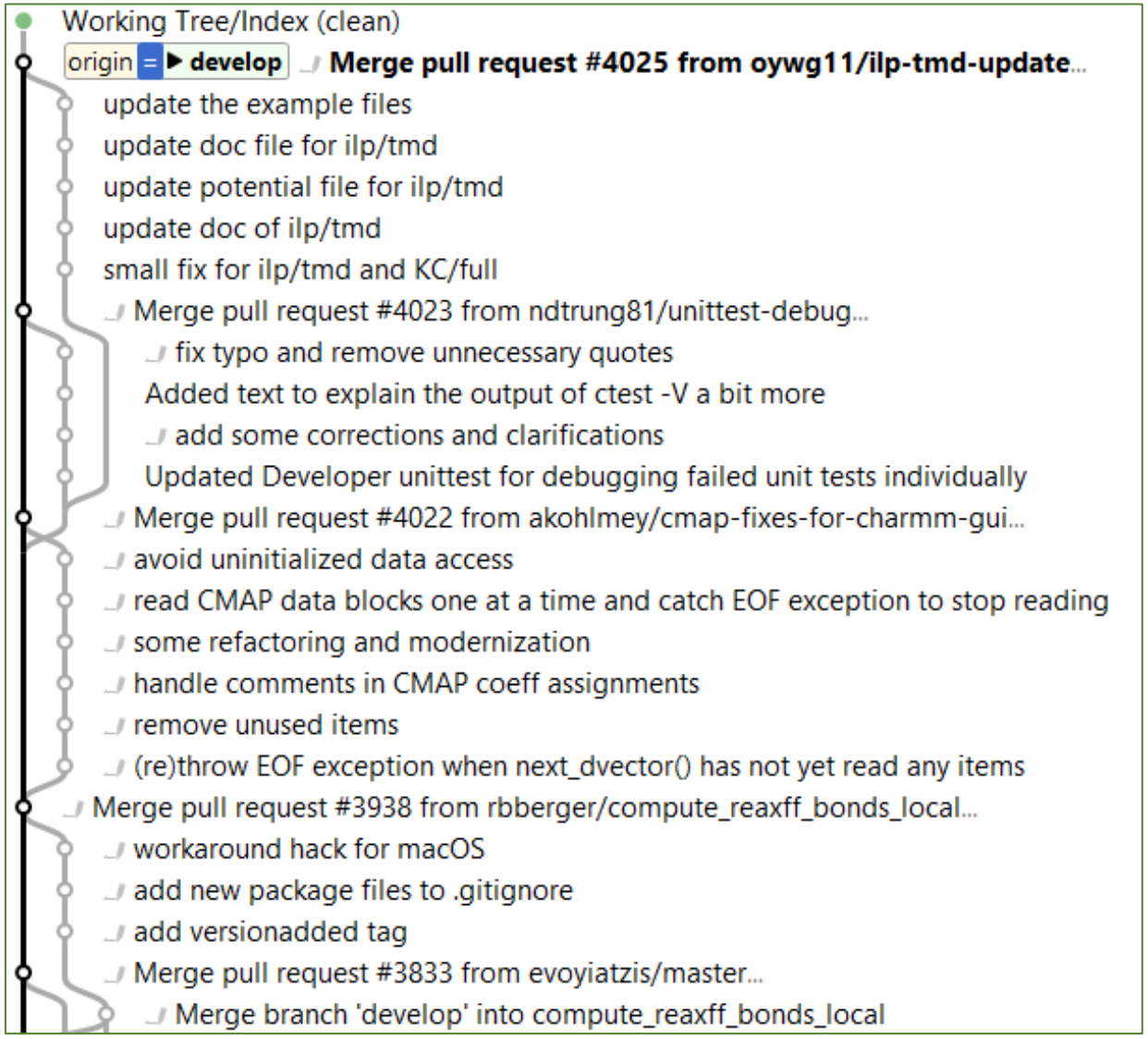
```
git remote
```

```
git fetch
```

```
git pull
```

```
git push
```

Ветвление



`git branch`

`git checkout`

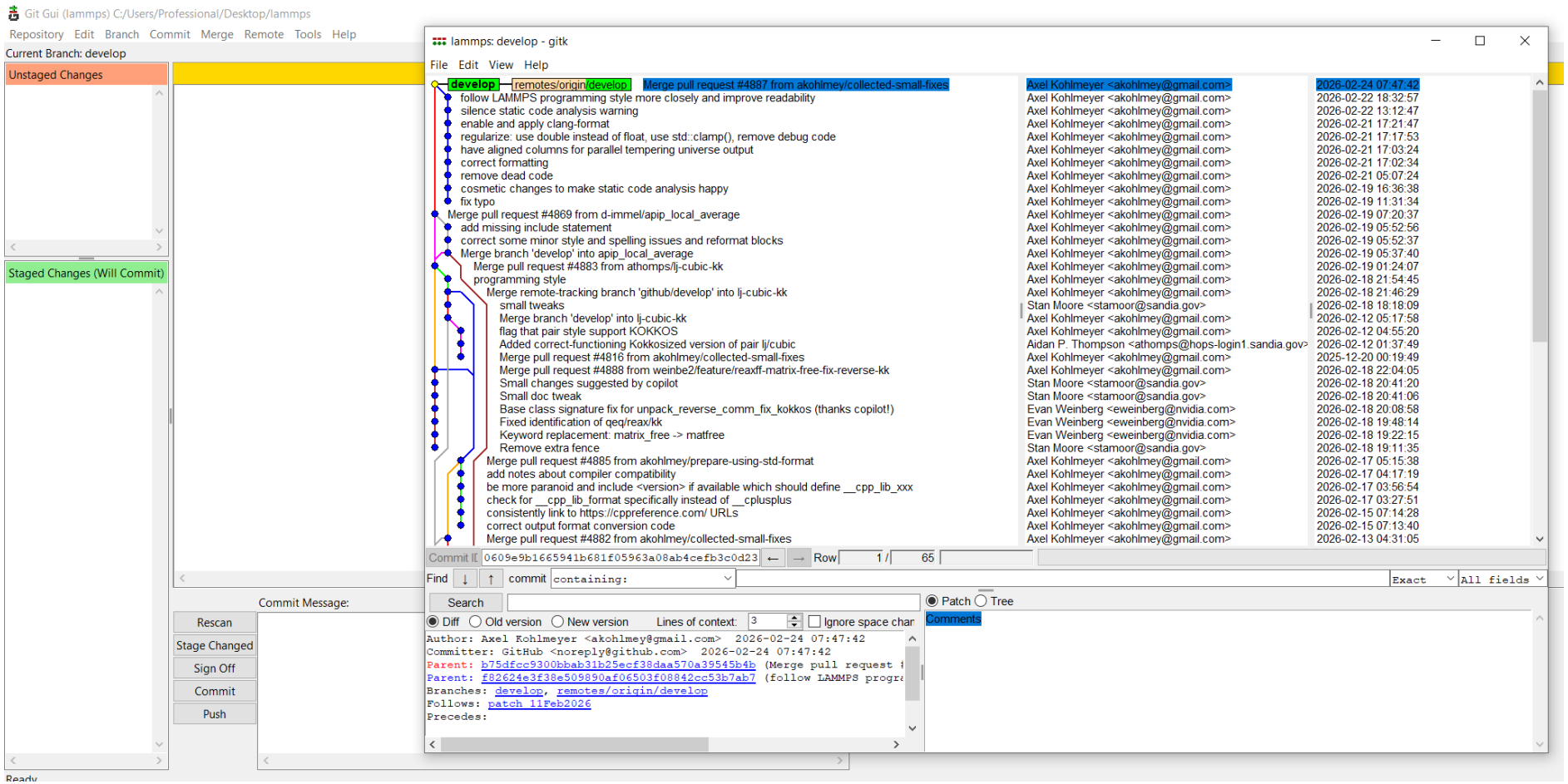
`git merge`

Временное хранилище

```
git stash
```

Графические клиенты

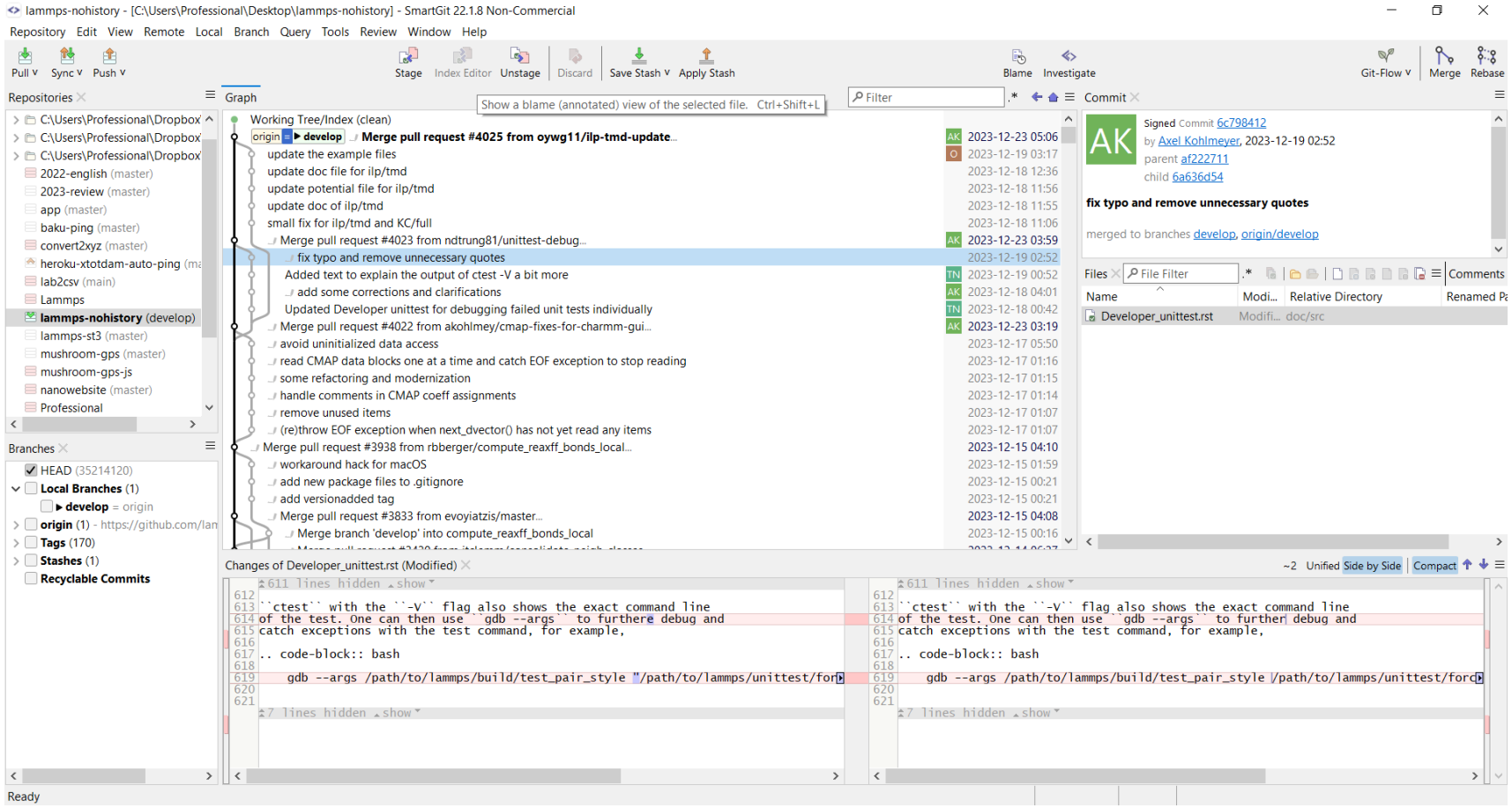
- **Git Gui**
- SmartGit
- SourceTree
- Sublime Merge
- …



https://git-scm.com/downloads/guis

Графические клиенты

- Git Gui
- **SmartGit**
- SourceTree
- Sublime Merge
- …



https://git-scm.com/downloads/guis

Задание

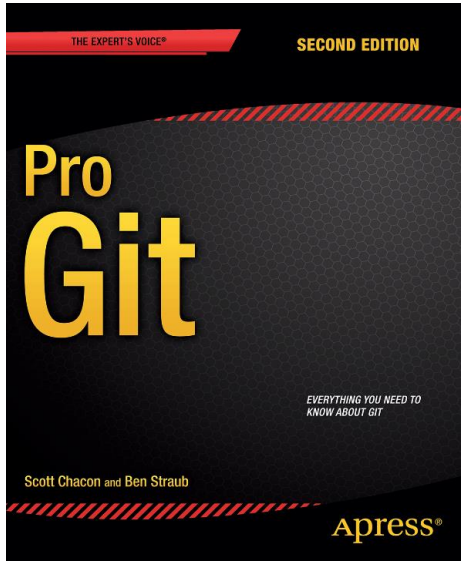
- Посмотреть на свой репозиторий в любом графическом клиенте

Дальнейшее изучение

```
git help
```

https://git-scm.com/doc

https://git-scm.com/book/ru/v2



Глава 7!

Облачные хранилища

GitHub

GitLab

Bitbucket

Задание*

- Загрузить свой репозиторий в облачное хранилище на своё усмотрение

Поиск виноватого

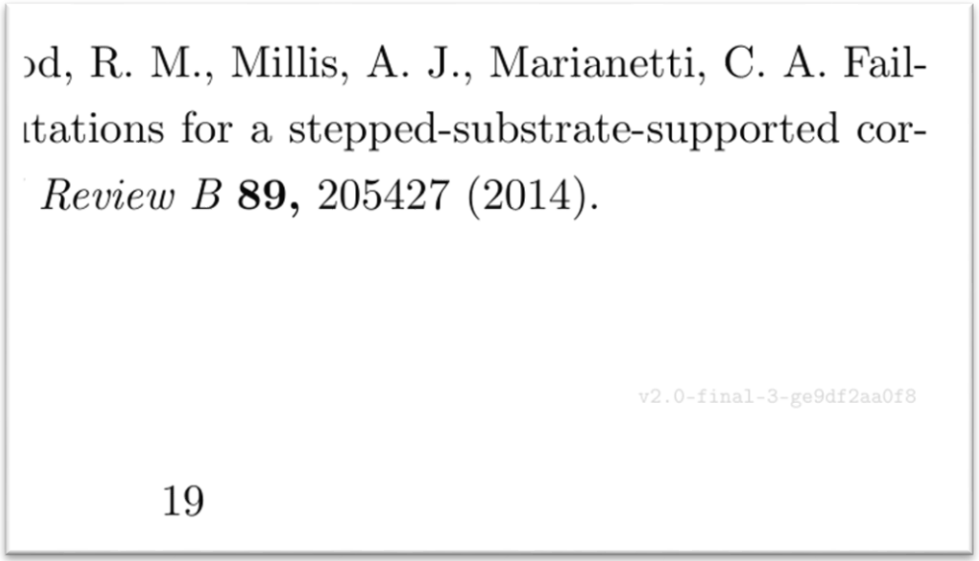
```
$ git blame verlet.h
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  1) /* -*- c++ -*- ----------------------------------------------------------
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  2)    LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator
09c19a936b4 (Axel Kohlmeyer 2021-05-24 14:17:12 -0400  3)    https://www.lammps.org/, Sandia National Laboratories
2132b1d904e (Axel Kohlmeyer 2022-10-24 11:00:27 -0400  4)    LAMMPS development team: developers@lammps.org
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  5)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  6)    Copyright (2003) Sandia Corporation.  Under the terms of Contract
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  7)    DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  8)    certain rights in this software.  This software is distributed under
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000  9)    the GNU General Public License.
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 10)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 11)    See the README file in the top-level LAMMPS directory.
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 12) ------------------------------------------------------------------------- */
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 13)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 14) #ifdef INTEGRATE_CLASS
980244dd0c4 (Axel Kohlmeyer 2021-04-27 22:14:00 -0400 15) // clang-format off
a1665fddc8e (Axel Kohlmeyer 2021-04-30 13:51:14 -0400 16) IntegrateStyle(verlet,Verlet);
980244dd0c4 (Axel Kohlmeyer 2021-04-27 22:14:00 -0400 17) // clang-format on
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 18) #else
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 19)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 20) #ifndef LMP_VERLET_H
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 21) #define LMP_VERLET_H
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 22)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 23) #include "integrate.h"
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 24)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 25) namespace LAMMPS_NS {
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 26)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 27) class Verlet : public Integrate {
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 28)  public:
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 29)   Verlet(class LAMMPS *, int, char **);
0099d2584b8 (Richard Berger 2022-01-18 15:10:35 -0500 30)   void init() override;
0099d2584b8 (Richard Berger 2022-01-18 15:10:35 -0500 31)   void setup(int flag) override;
0099d2584b8 (Richard Berger 2022-01-18 15:10:35 -0500 32)   void setup_minimal(int) override;
0099d2584b8 (Richard Berger 2022-01-18 15:10:35 -0500 33)   void run(int) override;
87b99306ba9 (Axel Kohlmeyer 2022-05-04 12:30:17 -0400 34)   void force_clear() override;
0099d2584b8 (Richard Berger 2022-01-18 15:10:35 -0500 35)   void cleanup() override;
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 36)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 37)  protected:
7fcd4498646 (Axel Kohlmeyer 2021-05-14 19:16:07 -0400 38)   int triclinic;    // 0 if domain is orthog, 1 if triclinic
7fcd4498646 (Axel Kohlmeyer 2021-05-14 19:16:07 -0400 39)   int torqueflag, extraflag;
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 40) };
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 41)
7fcd4498646 (Axel Kohlmeyer 2021-05-14 19:16:07 -0400 42) }    // namespace LAMMPS_NS
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 43)
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 44) #endif
^8e9f616a2e (sjplimp        2016-02-03 20:55:24 +0000 45) #endif
```

git blame

Ссылка на текущую версию

```
git describe --always --abbrev=10 --tags --dirty
```

```
$ git describe --always --abbrev=10 --tags --dirty
patch_19Nov2024-1200-g2777984355-dirty
```



od, R. M., Millis, A. J., Marianetti, C. A. Fail-
itations for a stepped-substrate-supported cor-
Review B **89,** 205427 (2014).

v2.0-final-3-ge9df2aa0f8

19