

# Распределённые системы контроля версий на примере mercurial

А. Худяков

11 июня 2013 г.

## Что это такое?

Инструмент для сохранения истории изменений и синхронизации изменений между несколькими людьми/компьютерами.

Обычно специализируется на текстовых данных (исходный код, LaTeX и т.д.)

## Бэкапы

- Все изменения записываются.
- Легко реплицировать на несколько компьютеров.

## История

- «Кто, когда и зачем это сделал?»
- У каждого коммита есть описание.

## Undo! Undo! Undo!

- Локальное: «Я сделал глупость»
- И глобальное: «надо использовать версию полугодовой давности»

## Песочница

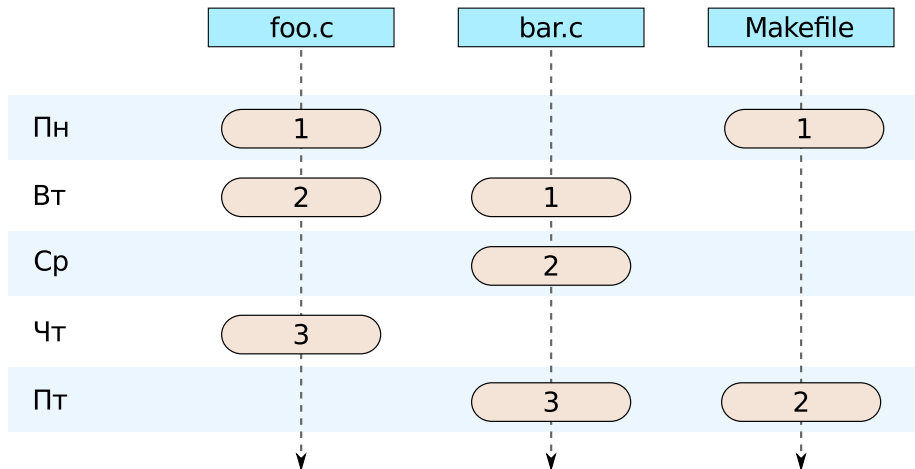
- Можно делать потенциально разрушительные изменения, не боясь что-либо сломать.

## Синхронизация

- Одновременная работа на нескольких компьютерах
- Несколько человек работают одновременно.

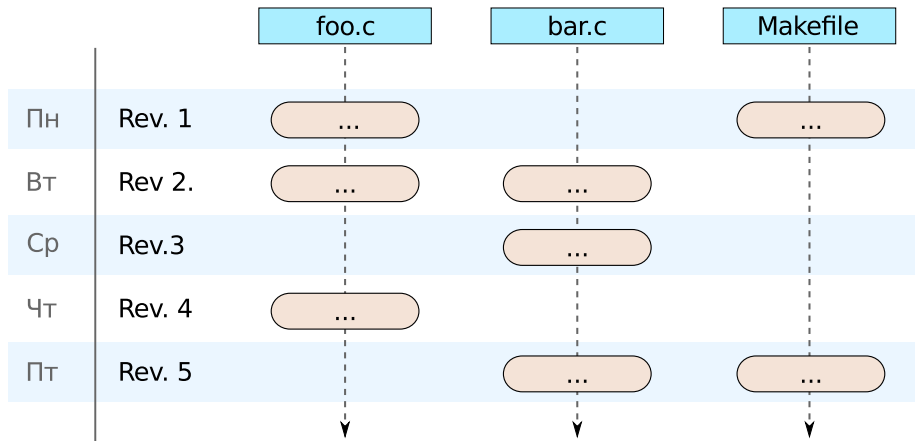
# Немного истории

Первое поколение (RCS,SCCS):



Файлы версионятся по отдельности.

## Второе поколение (CVS, SVN)



Версионирование состояния репозитория

## Repository

Хранилище для истории изменений

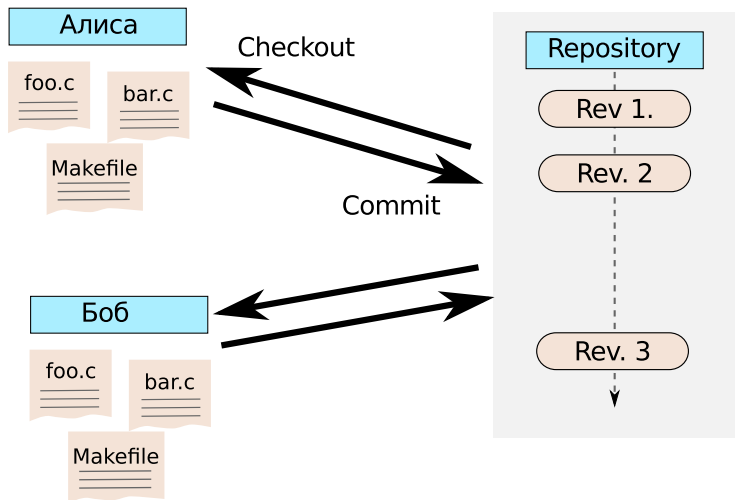
## Working copy

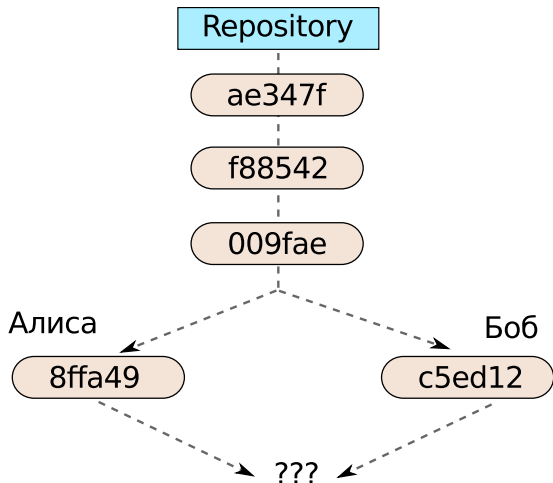
Копия какой-то ревизии с которой и ведётся работа.

## Commit

Акт записи изменений в репозиторий.

# Workflow





Если хранить историю в каждой копии и делать коммиты локально, история становится нелинейной!



## Централизованные

- История хранится на сервере. У клиентов есть только копия одной ревизии.
- История линейна.
- Почти любые операции требуют наличия связи с сервером.

## Распределённые

- Все репозитории равноправны и всех есть полная история.
- История — ациклический граф.
- Связь нужна только для приёма/передачи изменений из других репозиториях.

Сейчас есть две основные DVCS:

① Git

<http://git-scm.com/>

② Mercurial

<http://mercurial.selenic.com/>

Также заслуживает упоминания

- Darcs, он использует другой подход к хранению истории

<http://darcs.net/>

- Bazaar

<http://bazaar.canonical.com/en/>

## Linux

```
$ aptitude install mercurial
```

## Windows

TortoiseHG

## Настройка ~/.hgrc

```
[ui]
username = Random Joe <random.joe@example.com>
verbose  = True
editor   = nano
merge    = kdiff3
```

```
[extensions]
hgext.record =
color       =
```

Общая структура команд mercurial:

```
$ hg command parameters
```

Команду можно сокращать до однозначного префикса:

```
$ hg status → hg st
```

Самая полезная команда:

```
$ hg help [command]
```

# Создание репозиториев

```
$ hg init [path]
```

Инициализирует пустой репозиторий

```
$ hg clone source [dest]
```

Делает полную копию репозитория со всей историей.

```
$ hg clone ~/HG/some-repo test
```

```
$ hg clone ssh://example.com/some-repo
```

```
$ hg update [-C] [-r revision]
```

Обновляет рабочую директорию до указанной ревизии, текущей если не указано.

**-C** отбрасывает все изменения в рабочей копии

**-r rev** обновляется до ревизии *rev*

# Коммит и добавление/удаление файлов

```
$ hg (ci|commit) [file list]
```

Записывает изменения в «file list» в историю.

```
$ hg add [files]
```

Добавляет файл в очередь на коммит. Пока не сделан коммит он не записан в историю, и закоммитить не добавленный файл.

```
$ hg rm [file list]
```

Удаляет файл. Аналогично, пока не сделан коммит, факт удаления не фиксируется в истории.

```
$ hg st[tatus]
```

Список изменённых, поставленных на добавление/удаление файлов:

```
$ hg st
M mercurial.tex          - [M] Modified
R somefile               - [R] Removed
A figs/centralized.svg  - [A] Added
? figs/centralized.pdf  - [?] Unknown
? figs/history-DVCS.pdf
```



# Состояние репозитория

```
$ hg diff [files]
```

Изменения в файлах:

```
$ hg diff
--- a/mercurial.tex      Tue Jun 11 00:06:48 2013 +0400
+++ b/mercurial.tex      Tue Jun 11 13:32:04 2013 +0400
@@ -21,22 +21,20 @@
     \frametitle{Создание репозитория}

     \begin{block}{Создание}
-    {\texttt {$ hg init [path]}
+    \begin{block}{}
+    {\texttt {$ hg init [path]}
+    \end{block}

     Инициализирует пустой репозиторий
\end{block}
```

```
$ hg log
```

История изменений:

```
$ hg log
changeset: 154:f56df41ce110
user:      Aleksey Khudyakov <alexey.skladnoy@gmail.com>
date:      Sat Jun 01 14:51:25 2013 +0400
files:     Data/Vector/Fixed/Cont.hs
description:
Inline fromListM
...
```

```
$ hg revert [-a] [file list]
```

Откатывает изменения в рабочей копии до состояния, записанного в репозитории для файлов «file list» или для всех с «-a»

```
$ hg rollback
```

Откатывает последний коммит.

```
$ hg pull [-u] [remote repo]
```

Затягивает изменения из удалённого репозитория. Pull не обновляет рабочую копию! Для этого надо сделать либо «hg up», либо добавить ключ «-u».

```
$ hg push [remote repo]
```

Записывает изменения в удалённый репозиторий.

```
.hg/hgrc
```

```
[paths]
```

```
default = ssh://meson2.inr.ac.ru/HG/misc/2013-mercurial
```

Репозиторий default используется если путь не указан явно.

```
$ hg heads
```

Список головных ревизий.

```
$ hg merge
```

Сливает две головные ревизии. Если слияние не может быть сделано автоматически, то используется mergetool, указанный в конфиге.



```
'Alexey Khudyakov <alexey.skladnoy@gmail.com>  
'Alexey Khudyakov <alexey.skladnoy@gmail.com>  
'Alexey Khudyakov <alexey.skladnoy@gmail.com>  
'Alexey Khudyakov <alexey.skladnoy@gmail.com>
```

```
2013-06-11 14:34:03  
2013-06-11 14:31:50  
2013-06-11 14:32:36  
2013-06-11 14:31:23
```

История после merge

Что я не упомянул:

- Ветки (branches).
- Расширения (extensions).

## Mercurial

- Mercurial: The definitive guide  
<http://hgbook.red-bean.com/read/>
- Mercurial wiki  
<http://mercurial.selenic.com/wiki/>
- hg-git: interoperability between mercurial and git  
<http://hg-git.github.io/>