git

http://git-scm.com/

Пару слов о git

- Популярная распределенная система контроля версий, спроектированная для хранения больших проектов
- Создана Линусом Торвальдсом
- Изначально написана для использования в проекте Linux Kernel

Почему git?

- Децентрализирован
- Удобные для использования ветки(branches)
- Быстр даже на таких объемных проектах, как Linux Kernel
- Удобен в совместной распределенной работе
- Большое community

Немного о внутренней структуре

The Object Database

- Преймущества SHA-1
  - git может быстро определить идентичность объектов, просто сравнив хэши
  - Т.к. механизм вычисления хэша идентичен – тот же метод работает для репозитариев
  - git может обнаруживать ошибки, сличая прочитанные данные с хэшем.

Типы объектов

- Blob – хранение файлов
- Tree - объединяет один или несколько blob'ов в директорию. Также может включать другие tree-объекты, формируя иерархию директорий.
- Commit – объект объединяющий структуры tree в направленный ацикличный граф.
- Tag – символическая ссылка на объект

Commit

```
$ git show -s --pretty=raw 2be7fcb476
commit 2be7fcb4764f2dbcee52635b91fedb1b3dcf7ab4
tree fb3a8bdd0ceddd019615af4d57a53f43d8cee2bf
parent 257a84d9d02e90447b149af58b271c19405edb6a
author Dave Watson <dwatson@mimvista.com> 1187576872 -0400
committer Junio C Hamano <gitster@pobox.com> 1187591163 -0700

    Fix misspelling of 'suppress' in docs

    Signed-off-by: Junio C Hamano <gitster@pobox.com>
```

tree

```
$ git ls-tree fb3a8bdd0ce
100644 blob 63c918c667fa005ff12ad89437f2fdc80926e21c    .gitignore
100644 blob 5529b198e8d14decbe4ad99db3f7fb632de0439d    .mailmap
100644 blob 6ff87c4664981e4397625791c8ea3bbb5f2279a3    COPYING
040000 tree 2fb783e477100ce076f6bf57e4a6f026013dc745    Documentation
100755 blob 3c0032cec592a765692234f1cba47dfdcc3a9200    GIT-VERSION-GEN
100644 blob 289b046a443c0647624607d471289b2c7dcd470b    INSTALL
100644 blob 4eb463797adc693dc168b926b6932ff53f17d0b1    Makefile
100644 blob 548142c327a6790ff8821d67c2ee1eff7a656b52    README
```

Blob

```
$ git show 6ff87c4664

 Note that the only valid version of the GPL as far as this project
 is concerned is _this_ particular version of the license (ie v2, not
 v2.2 or v3.x or whatever), unless explicitly otherwise stated.
...
```

Tag

- Не только красивое имя

```
$ git cat-file tag v1.5.0
object 437b1b20df4b356c9342dac8d38849f24ef44f27
type commit
tag v1.5.0
tagger Junio C Hamano <junkio@cox.net> 1171411200 +0000
GIT 1.5.0
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.6 (GNU/Linux)
iD8DBQBF0lGqwMbZpPMRm5oRAuRiAJ9ohBLd7s2kqjkKlq1qqC57SbnmzQCdG4ui
nLE/L9aUXdWeTFPron96DLA=
=2E+0
-----END PGP SIGNATURE-----
```

Pack

- Для сохранения места и убыстрения работы git умеет объединять объекты в pack-file'ы.

```
$ git repack
Generating pack...
Done counting 6020 objects.
Deltifying 6020 objects.
 100% (6020/6020) done
Writing 6020 objects.
 100% (6020/6020) done
Total 6020, written 6020 (delta 4070), reused 0 (delta 0)
Pack pack-3e54ad29d5b2e05838c75df582c65257b8d08e1c created.
```

index

- Индекс – это бинарный файл, содержащий отсортированный список файлов с путями, права, и SHA-1 blob объектов.
- Индекс содержит всю необходимую информацию для генерации tree объекта для нового коммита
- Индекс обеспечивает быстрое сравнение между объектами описанными в индексе и рабочей копией
- Индекс позволяет эффективно обнаруживать конфликты

Централизованное хранилище против распределенного?

Централизованная VCS

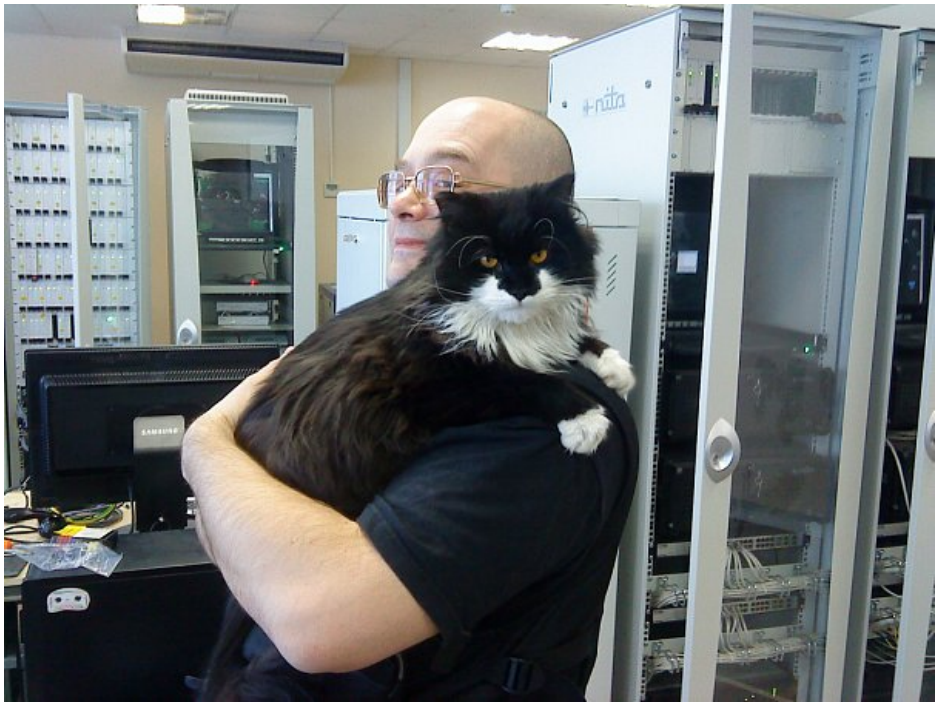
- Центральный репозиторий
  - История версия хранится на сервере
- Checkout
  - Забираем “рабочую версию”

Распределенный git

- Каждая рабочая директория git является полноценным репозитарием, с полной историей, не зависимым от доступа к сети либо доступности центрального сервера
- Коммиты происходят автономно
- В дальнейшем, коммиты могут заливаться и скачиваться между репозиториями вместе со всей историей

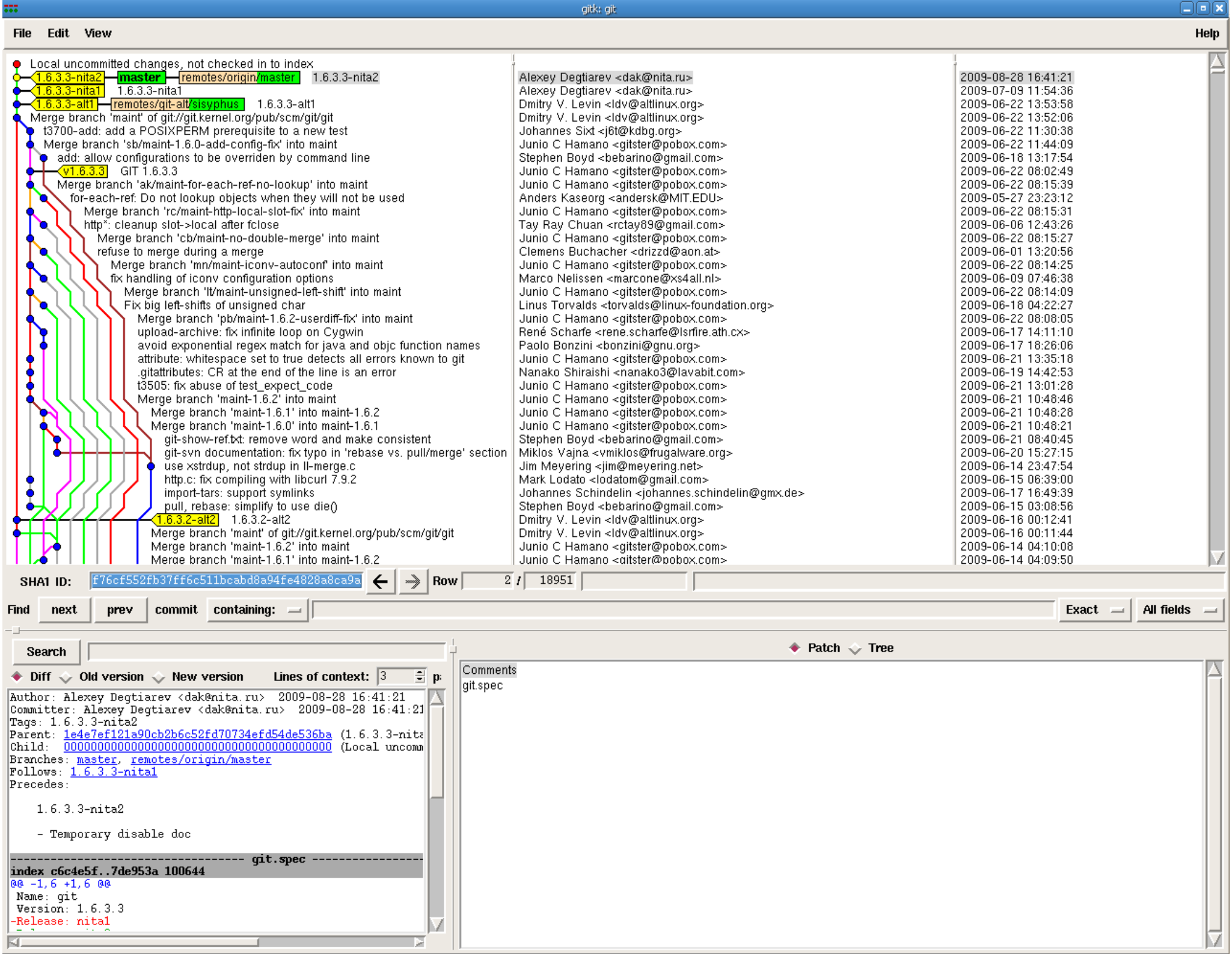
Подождите-ка, автономные коммиты??!!!!!! 11111пыщьпыщь

Теперь можно спокойно изменять код



- Прямо на ноутбуке в аэропорту
- В поезде
- Дома
- Да где угодно!

Когда ты последний раз создавал branch?


gitk: git
File Edit View Help
Local uncommitted changes, not checked in to index
1.6.3.3-nita2 master remotes/origin/master 1.6.3.3-nita2
1.6.3.3-nita1 1.6.3.3-nita1
1.6.3.3-alt1 remotes/git-alt/sisyphus 1.6.3.3-alt1
Merge branch 'maint' of git://git.kernel.org/pub/scm/git/git
t3700-add: add a POSIXPERM prerequisite to a new test
Merge branch 'sb/maint-1.6.0-add-config-fix' into maint
add: allow configurations to be overriden by command line
v1.6.3.3 GIT 1.6.3.3
Merge branch 'ak/maint-for-each-ref-no-lookup' into maint
for-each-ref: Do not lookup objects when they will not be used
Merge branch 'rc/maint-http-local-slot-fix' into maint
http*: cleanup slot->local after fclose
Merge branch 'cb/maint-no-double-merge' into maint
refuse to merge during a merge
Merge branch 'mn/maint-iconv-autoconf' into maint
fix handling of iconv configuration options
Merge branch 'lt/maint-unsigned-left-shift' into maint
Fix big left-shifts of unsigned char
Merge branch 'pb/maint-1.6.2-userdiff-fix' into maint
upload-archive: fix infinite loop on Cygwin
avoid exponential regex match for java and objc function names
attribute: whitespace set to true detects all errors known to git
.gitattributes: CR at the end of the line is an error
t3505: fix abuse of test_expect_code
Merge branch 'maint-1.6.2' into maint
Merge branch 'maint-1.6.1' into maint-1.6.2
Merge branch 'maint-1.6.0' into maint-1.6.1
git-show-ref.txt: remove word and make consistent
git-svn documentation: fix typo in 'rebase vs. pull/merge' section
use xstrdup, not strdup in ll-merge.c
http.c: fix compiling with libcurl 7.9.2
import-tars: support symlinks
pull, rebase: simplify to use die()
1.6.3.2-alt2 1.6.3.2-alt2
Merge branch 'maint' of git://git.kernel.org/pub/scm/git/git
Merge branch 'maint-1.6.2' into maint
Merge branch 'maint-1.6.1' into maint-1.6.2
Alexey Degtiarev <dak@nita.ru> 2009-08-28 16:41:21
Alexey Degtiarev <dak@nita.ru> 2009-07-09 11:54:36
Dmitry V. Levin <ldv@altlinux.org> 2009-06-22 13:53:58
Dmitry V. Levin <ldv@altlinux.org> 2009-06-22 13:52:06
Johannes Sixt <j6t@kdbg.org> 2009-06-22 11:30:38
Junio C Hamano <gitster@pobox.com> 2009-06-22 11:44:09
Stephen Boyd <bebarino@gmail.com> 2009-06-18 13:17:54
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:02:49
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:15:39
Anders Kaseorg <andersk@MIT.EDU> 2009-05-27 23:23:12
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:15:31
Tay Ray Chuan <rctay89@gmail.com> 2009-06-06 12:43:26
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:15:27
Clemens Buchacher <drizzd@aon.at> 2009-06-01 13:20:56
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:14:25
Marco Nelissen <marcone@xs4all.nl> 2009-06-09 07:46:38
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:14:09
Linus Torvalds <torvalds@linux-foundation.org> 2009-06-18 04:22:27
Junio C Hamano <gitster@pobox.com> 2009-06-22 08:08:05
René Scharfe <rene.scharfe@lsrfire.ath.cx> 2009-06-17 14:11:10
Paolo Bonzini <bonzini@gnu.org> 2009-06-17 18:26:06
Junio C Hamano <gitster@pobox.com> 2009-06-21 13:35:18
Nanako Shiraishi <nanako3@lavabit.com> 2009-06-19 14:42:53
Junio C Hamano <gitster@pobox.com> 2009-06-21 13:01:28
Junio C Hamano <gitster@pobox.com> 2009-06-21 10:48:46
Junio C Hamano <gitster@pobox.com> 2009-06-21 10:48:28
Junio C Hamano <gitster@pobox.com> 2009-06-21 10:48:21
Stephen Boyd <bebarino@gmail.com> 2009-06-21 08:40:45
Miklos Vajna <vmiklos@frugalware.org> 2009-06-20 15:27:15
Jim Meyering <jim@meyering.net> 2009-06-14 23:47:54
Mark Lodato <lodatom@gmail.com> 2009-06-15 06:39:00
Johannes Schindelin <johannes.schindelin@gmx.de> 2009-06-17 16:49:39
Stephen Boyd <bebarino@gmail.com> 2009-06-15 03:08:56
Dmitry V. Levin <ldv@altlinux.org> 2009-06-16 00:12:41
Dmitry V. Levin <ldv@altlinux.org> 2009-06-16 00:11:44
Junio C Hamano <gitster@pobox.com> 2009-06-14 04:10:08
Junio C Hamano <gitster@pobox.com> 2009-06-14 04:09:50
SHA1 ID: f76cf552fb37ff6c511bcabd8a94fe4828a8ca9a
Row 2 / 18951
Find next prev commit containing: Exact All fields
Search
Diff Old version New version Lines of context: 3
Author: Alexey Degtiarev <dak@nita.ru> 2009-08-28 16:41:21
Committer: Alexey Degtiarev <dak@nita.ru> 2009-08-28 16:41:21
Tags: 1.6.3.3-nita2
Parent: 1e4e7ef121a90cb2b6c52fd70734efd54de536ba (1.6.3.3-nita
Child: 0000000000000000000000000000000000000000 (Local uncom
Branches: master, remotes/origin/master
Follows: 1.6.3.3-nita1
Precedes:
1.6.3.3-nita2
- Temporary disable doc
git.spec
index c6c4e5f..7de953a 100644
@@ -1,6 +1,6 @@
Name: git
Version: 1.6.3.3
-Release: nita1
Patch Tree
Comments
git.spec

Почему пользоваться бранчами с git удобно?

- Это быстро:

  $ time git checkout -b newbranch

  Switched to a new branch 'newbranch'

  real 0m0.291s
- Персонально
- Слияние не сосёт

Зачем вообще столько бранчей?

- Каждая отдельная фича/бага – свой бранч
- Разработка хорошо проработанных изменений
- Коммиты сразу и часто
- Анализ и пересмотр кода перед слиянием

Почему git быстр?

- Значительная часть операций не требует связи с другими репозитариями:
  - Diff's
  - Просмотр истории
  - Коммиты изменений
  - Слияния веток
  - Получение любой версии файла
  - Переключение между ветками

Создание репозитария

```
$ pwd
/somepath
$ git init
$ git add .
$ git commit
```

Клонирование из другого репозитария

```
$ git clone REPO_URL
```

REPO_URL's:

```
ssh://git.nita/people/dak/packages/dbinterface.git
http://git.nita/gears/lib/libtransport.git
git://git.kernel.org/pub/scm/git/git.git
```

git status

- Untracked files
  - Обычно это либо промежуточные файлы сборки(можно добавлять в игнорлист), либо новые файлы
- Changed but not updated
  - Локально измененные файлы, не в индексе
- Changes to be commited
  - Изменения, которые будут включены в коммит

Индекс

- Место для файлов вашего будущего коммита
- Позволяет иметь дополнительное, временное хранилище для файлов, пока коммит еще не созрел

Коммиты

- git commit
  - Создает коммит, в который попадают файлы из индекса
- git commit -a
  - Создает коммит, в который также попадают все измененные(но не новые!) файлы, не содержащиеся в индексе

Diff's

- git diff
  - Показывает разницу между индексом и рабочим деревом
- git diff –cached
  - Показывает разницу между индексом и последним коммитом
- git diff HEAD
  - Показывает разницу между рабочим деревом и последним коммитом(головой)

Лог

- git log
  - История коммитов
- git log -S”void dumbMethod(“
  - Найдет коммит, который добавил эту строчку
- git log -p
  - Покажет историю в виде патчей на каждый коммит

Управление ветками

Создание новой ветки:

```
$ git branch new_branch
```

Переключится в ветку:

```
$ git checkout new_branch
```

Совместить и то и то:

```
$ git checkout -b new_branch
```

Еще немного веток

Посмотреть все ветки

```
$ git branch
*new_branch
master
```

Удалить ветку

```
$ git branch -d ALREADY_MERGED
$ git branch -D FAIL_BRANCH
```

Просмотр изменений с ветками

Показать лог между ветками

```
$ git log new_branch..master
```

И изменения:

```
$ git diff new_branch..master
```

И патчики:

```
$ git format-patch new_branch..master
```

Слияние, rebase, исправление конфликтов

Для слияния с некой веткой достатчно выполнить:

$ git merge somebranch

Возможен также вариант:

$git rebase somebranch

В случае неоднозначного слияния, после исправления конфликтов достаточно сделать commit.

Совместная работа с git

Хостинг git-репозитариев

- git-daemon
- gitosis
- github
- git.nita

Что такое git.nita

- Хостинг git-репозитариев команды разработчиков с интерфейсом поиска
- Управление Access Control Lists для сборки
- Управление сборкой в различные репозитарии.

Хостинг репозитариев

- Доступ по RSA ключу и ssh
- Gpg-подпись
- packages/public/private репозитарии
- Gears

Оповещения по e-mail

- Email-subscription
  - $USER $PACKAGE $REFTYPE $REFNAME
- Email-distribution
  - $PACKAGE $REFTYPE $REFNAME $MAILTO

Команды git.nita

```
$ ssh git.nita help

Available commands:

help

charset <path to git repository> <charset>

clone <path to git repository> <path to directory>

default-branch <path to git repository> <branch>

find-package <pattern>

init-db <path to directory>

ls <path to directory>

mv-db <path to source directory> <path to destination directory>

quota

repack <path to git repository> <value>

rm-db <path to git repository>
```

ACL

- ACL — список пользователей, которые могут запустить сборку с помощью git.nita, и список пользователей, которые могут залить/собрать пакет как NMU. Для NMU дополнительно указывается интервал действия разрешения.

- NMU – Non-maintaner update

Управление сборкой

- src-репозитарий: сборка исходников в архивы tar.gz
- nita-ULL-RC-RH53-sid: сборка в пакеты Red Hat Enterprise Linux 5.3 (проект Planeta Oracle, Pulkovo).

Предлагаемая схема разработки

- Контрибьютор
  - Вносит изменения в проект
  - Отсылает патчи с изменениями
- Майнтейнер
  - Просматривает и анализирует патчи(Code review)
  - Применяет патчи
  - Собирает версии, обновляя репозитарий

Контрибьютор

Fork, then clone

- Копируем себе репозитарий в хранилище(если еще не делали)
  - `$ ssh git.nita clone /gears/lib/libtransport.git public`
- Забираем себе
  - `$ git clone ssh://git.nita/people/dak/public/libtransport.git`

Вносим изменения

- `$ git checkout -b newbranch`
- `$ echo “some text” >> README`
- `$ git commit -a`

Заливаем изменения на сервер

- `$ git checkout master`
- `$ git merge newbranch`
- `$ git push origin master`

Извещаем майнтернера о изменениях

- Для регулярных оповещений настраиваем e-mail нотификатор

Майнтейнер

Нелегкая жизнь майнтейнера

- Узнает об изменениях
- Добавляет источник
  - `$ git remote add ....`
- Вытягивает изменения
  - `$ git fetch dak/master`
- Анализирует их с помощью `log/diff/gitk/etc`
- Применяет `merge`

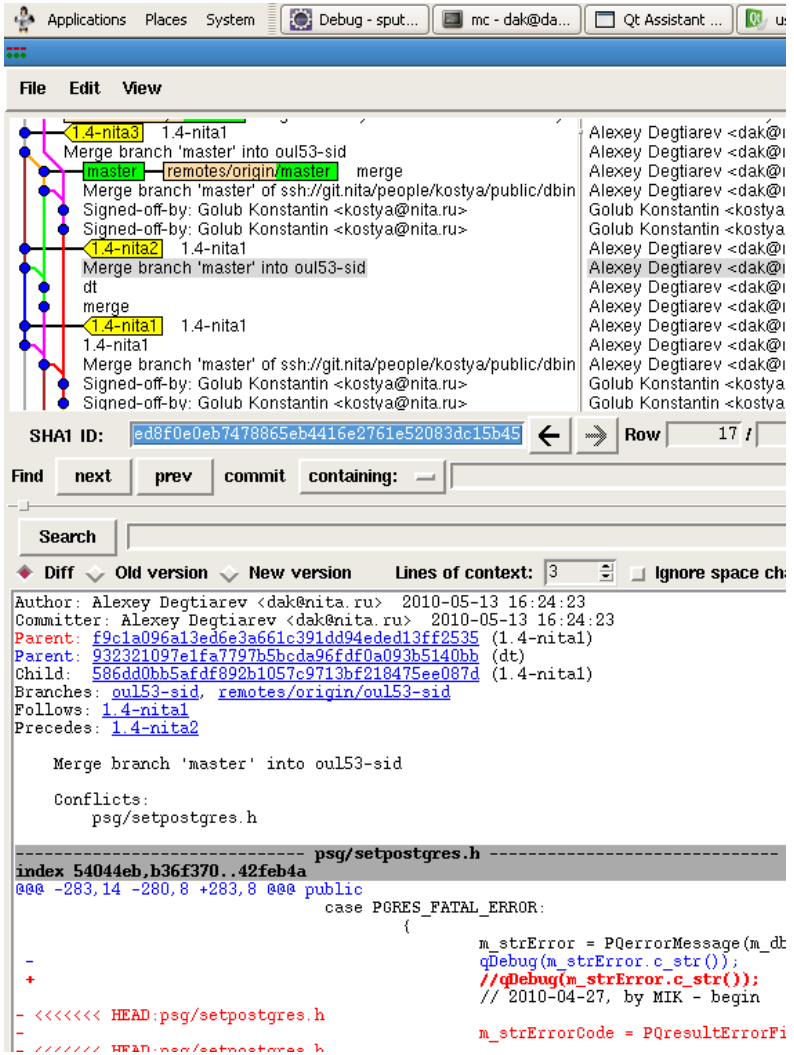
Осталось собрать проект

- Создаем и подписываем метку с версией
  - `$ git tag -s (gear-create-tag)`
- Пушим изменения на сервер
  - `$ git push --all ; git push --tags`
- Даем команду на сборку
  - `$ ssh git.nita build -b nita-ULL-RC-RH53-sid libtransport 1.4.0-nita1`

Гуевые красивости

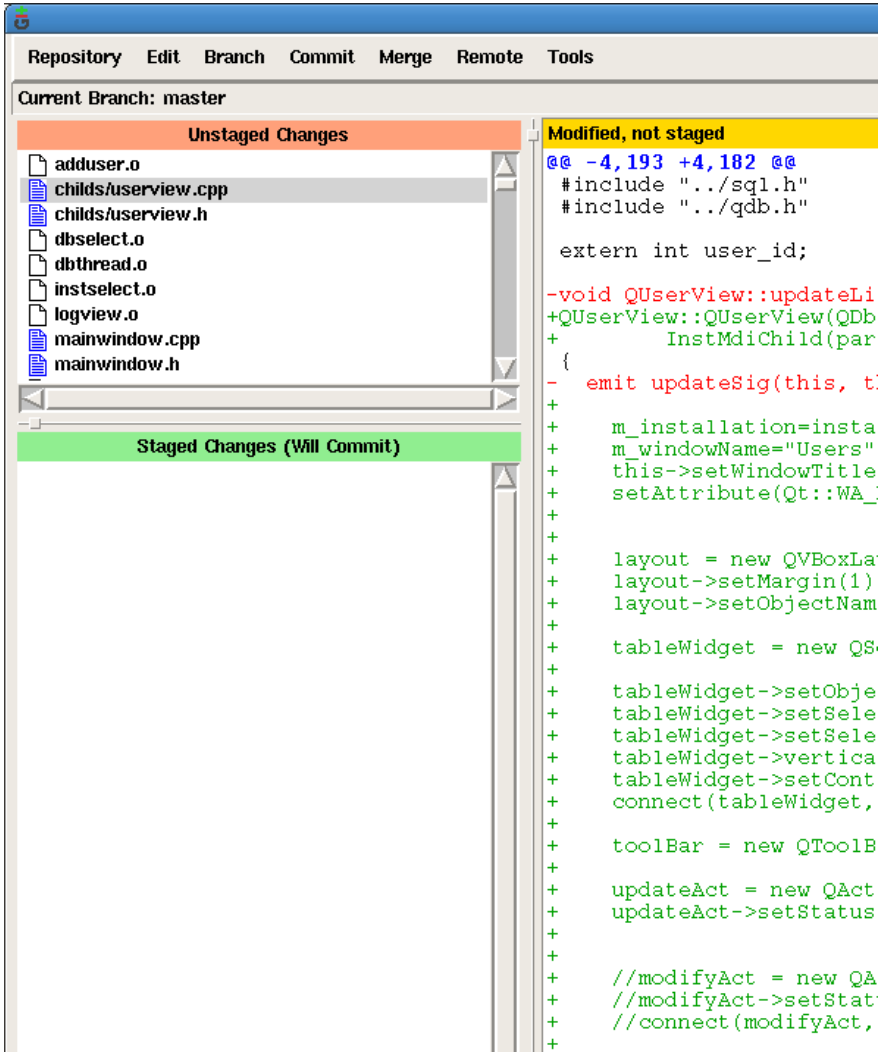
Gitk

- Поставляется с git
- Удобно отображает ветвление



git-gui

- Поставляется вместе с git
- Визуализирует работу с индексами и локальными изменениями



Используется в

* Amarok[54][55]
* Android[56]
* Arch Linux
* Aquamacs Emacs
* BlueZ[57]
* Btrfs[58]
* Clojure[59]
* CakePHP[60]
* Debian[61]
* Digg[62]
* DragonFly BSD[63]
* Elinks[64]
* Fedora
* FFmpeg [65]
* Freenet[66]
* FreeSWITCH[67]
* git[68]
* GIMP
* GNOME[69][70]
* GPM[71]
* GStreamer[72]
* gThumb[73]
* GTK+[74]
* Hurd[75]
* jQuery[76]
* Laika (EHR testing framework)[77]
* LilyPond (music typesetting)[78]
* Linux kernel
* Linux Mint[79][80]
* LMMS[81]
* Maemo[82]
* MeeGo[83]
* Merb[84]
* MicroEMACS
* MooTools[85]
* One Laptop Per Child (OLPC)[86]
* OpenFOAM[87]
* openSUSE[88]
* Penumbra: Overture [89][90]
* Perl[91]
* phpBB[92]
* Prototype.js[93]
* Qt[94]
* Reddit[95]
* rsync[96]
* Ruby on Rails[97]
* Samba[98]
* SproutCore[99]
* Starlink[100]
* Sugar[101]
* SWI-Prolog[102]
* VLC[103]
* VTK[104]
* Wine[105]
* Xfce[106]
* Xiph[107]
* X.org Server[108]
* x264[103]
* YUI[109]

FLAME NOW!