Git ile Sürüm Takibi

Ömer ÖZKAN

omer@ozkan.info
omer.ozkan@ozguryazilim.com.tr

Ulaşmak isterseniz?

- Kişisel web günlüğü:
  http://omerozkan.net
- Eposta:
  omer@ozkan.info
- Twitter:
  @omerozkan_
- Facebook
  facebook.com/omerozkan

Sürüm Takip Sistemi?

- Sürüm takip sistemi, bir bilgisayar yazılımının kaynak kodlarının bir depoya değişikliklerle birlikte kaydedilmesi ve erişilmesini sağlar.
- Sürüm takip sisteminde değişiklik yaparken, o değişikliğin kimin yaptığı, neden yapıldığını, hangi geliştirmelerin yapıldığını veya hataların çözüldüğü belirtilir.
- Yapılan bütün değişikliklerin tarihçesi saklanır.
- Merkezi bir sunucu ile çalışıldığında birden çok programcının ekip halinde geliştirme yapmasını kolaylaştırır
- VCS (Version Control Systems)

  RCS (Revision Control Systems)

Sürüm Takip Sistemi Olmazsa?

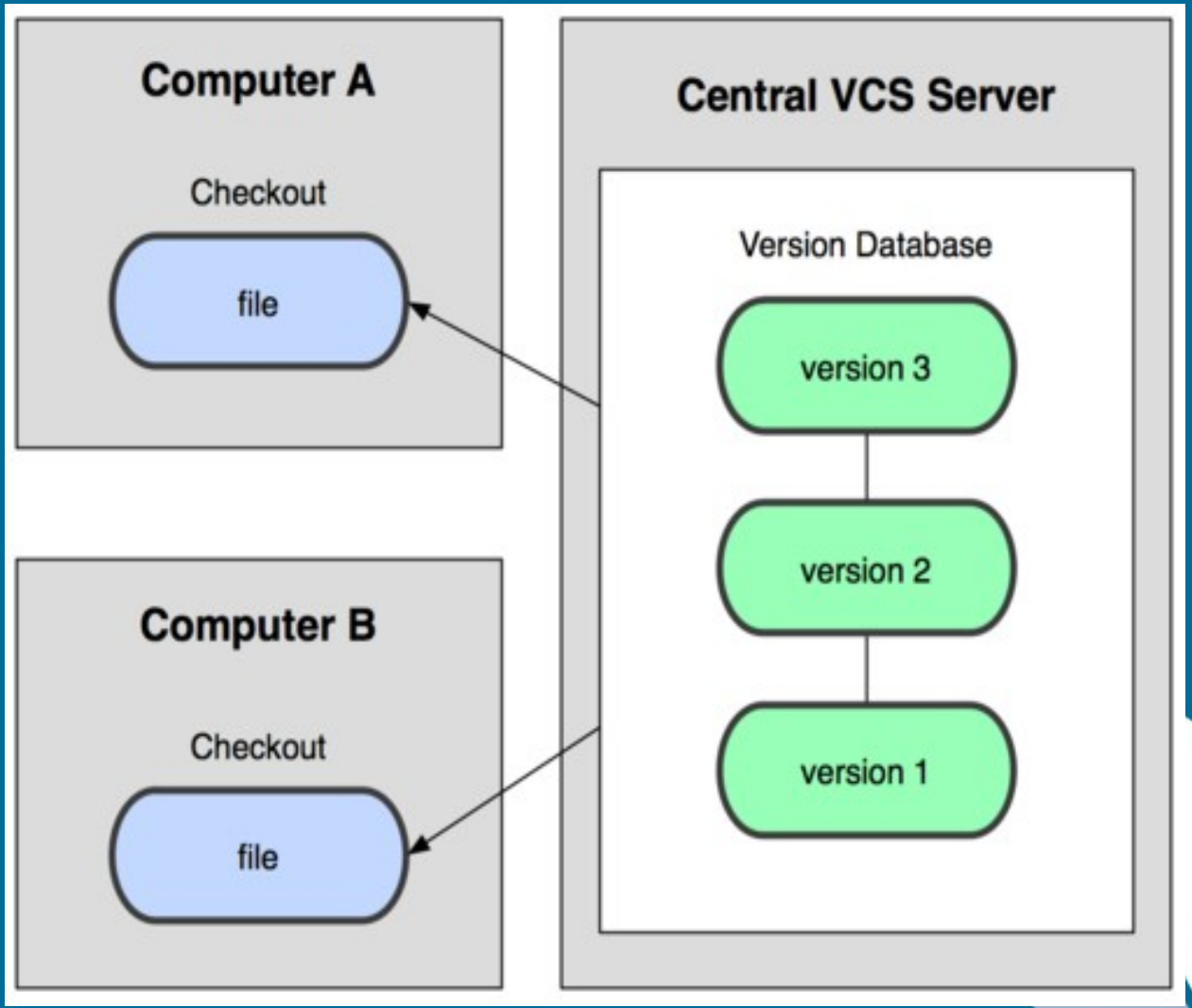
- Nasıl sürüm çıkarıyorsunuz?
- Ekip halinde nasıl çalışıyorsunuz?
- Geri almanız gereken değişiklikleriniz varsa değişiklikleri nasıl geri alıyorsunuz?
- Ekibinizden biri geliştirdiğiniz sistemi kullanılamaz hale getirdiğinde kimin yaptığını nasıl tespit ediyorsunuz? (Blaming)

Sürüm Takip Sistemleri

- Yerel Sürüm Takip Sistemleri
  - SCCS (1972), RCS (1982)
- Senkron Sürüm Takip Sistemleri
  - CVS, SVN
- Asenkron Sürüm Takip Sistemleri
  - SVN, Bazaar, Git, Mercurial

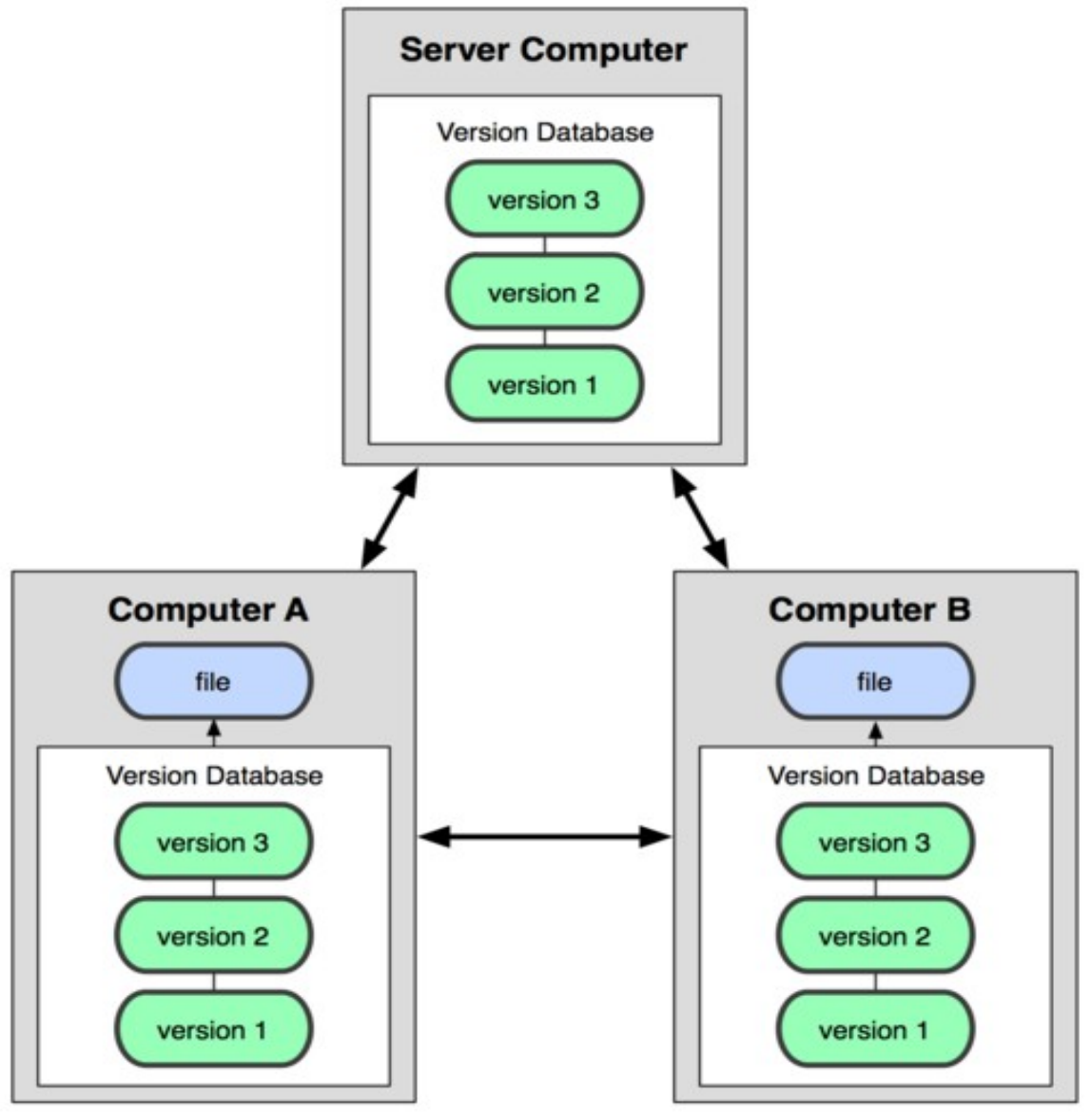
Merkezi Sürüm Takip Sistemleri

- CVS
- SVN
- Bazaar



Dağıtık Sürüm Takip Sistemi

- Git
- Mercurial



Git

- Git, küçük ve büyük ölçekli projelerin hızlı ve verimli olarak geliştirilmesini sağlayan ücretsiz, özgür bir sürüm takip sistemidir.
- Asenkron ve dağıtık bir sürüm takip sistemidir.
- GPL v2.0
- HTTP veya SSH protokolü üzerinden uzak depo ile çalışma

Git'in Ortaya Çıkışı

- Linux çekirdeği BitKeeper adlı dağıtık sürüm takip sistemi kullanılarak geliştirilmekteydi.
- BitKeeper, Linux çekirdeğini geliştiren topluluğun ücretsiz lisansını iptal etti.
- Linus Torvalds ve Linux geliştirme topluluğu bunun sonucunda GIT'i geliştirdi.

Git'in Özellikleri

- Hızlı
- Basit Tasarım
- Paralel Dallanma
- Dağıtık yapı
- Linux çekirdeği gibi büyük projelerde veya daha küçük projelerde etkin ve verimli kullanım

Kimler Git kullanıyor?

- Linux Kernel
- X.org
- Eclipse
- PostgreSQL
- Android
- Rails
- Gnome
- KDE
- Drupal

- Google
- Facebook
- Microsoft
- Twitter
- Linkedin
- ...

GIT vs SVN Benchmarking

Git Kurulumu

- OpenSuse

  zypper in git / zypper install git
- Ubuntu / Debian

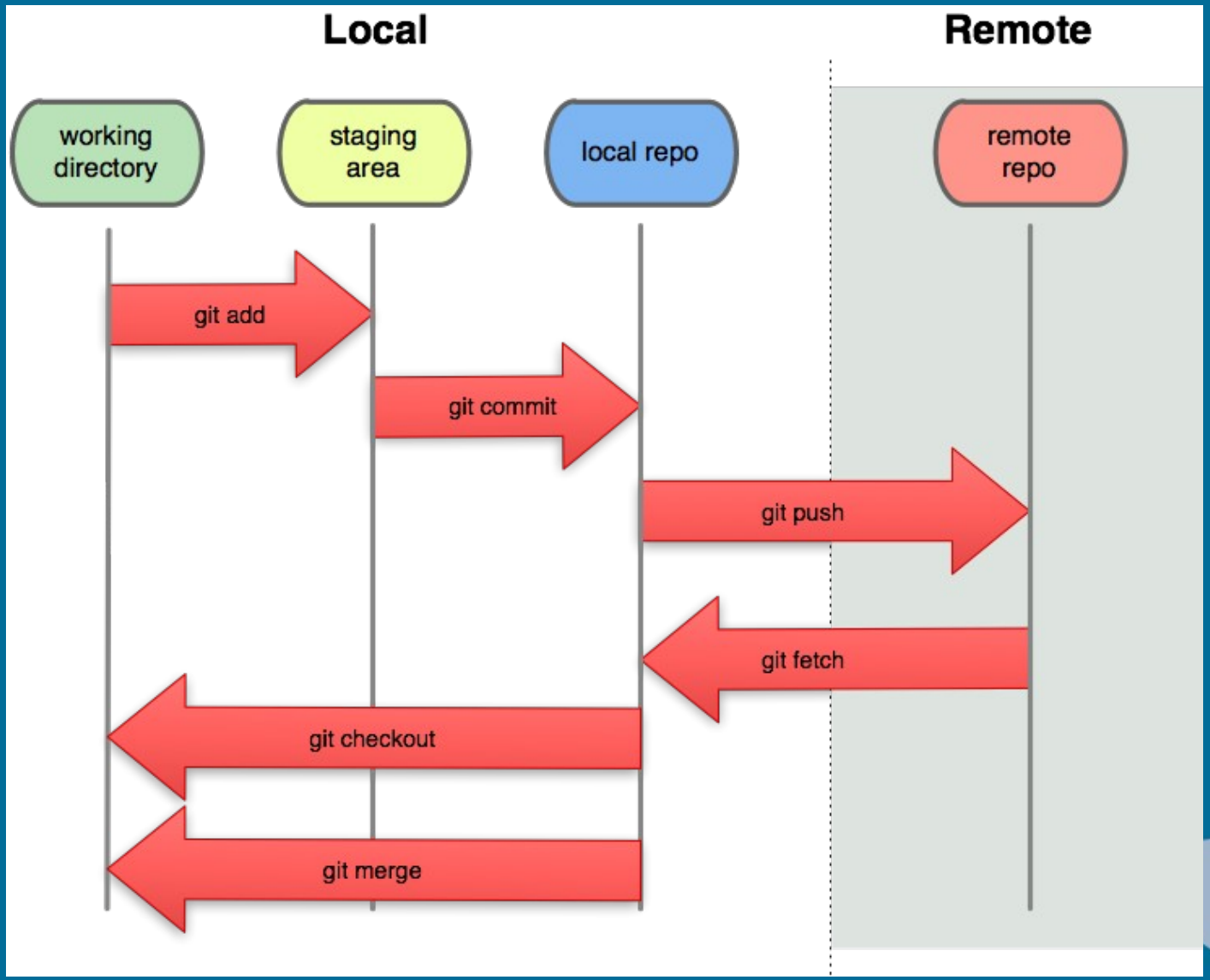
  apt-get install git
- Fedora / RedHat

  yum install git

Git Kullanımı

- Yerel depo oluşturma (init)
- Commit
- Dallanma (branch)
- Birleştirme (merge)
- Çakışma (conflict)
- Geri alma (revert)
- Sıfırlama (reset)
- Rebase
- Etiketleme (tag)
- Cherry-pick
- Uzak sunucu ile çalışma
- Stash
- Submodule

Git Çalışma Akışı



İlk Commit

- git init (Deponun ilklendirilmesi)
- echo “Hello git” >> first.file (içinde “Hello git” yazan yeni bir dosya)
- git add first.file (dosyayı stage'e ekleme)
- git commit -m “ilk commit” (commit)
- git status (deponuzun durumunu görüntüleme)

Staging

- git add dosya_yolu
- git mv kaynak_yol hedef_yol
- git rm dosya_yolu
- git commit -a
- git add -i

Git Checkout

- Bir dala geçmek için
  - Git checkout branchA
- Herhangi bir commit anına gitmek için
  - Git checkout 6c3dd0 (commit no)
- Herhangi etiketlediğiniz bir commit'e gitmek için
  - Git checkout tags/tag_adi

Dallanma

- Bir dal oluşturmak için

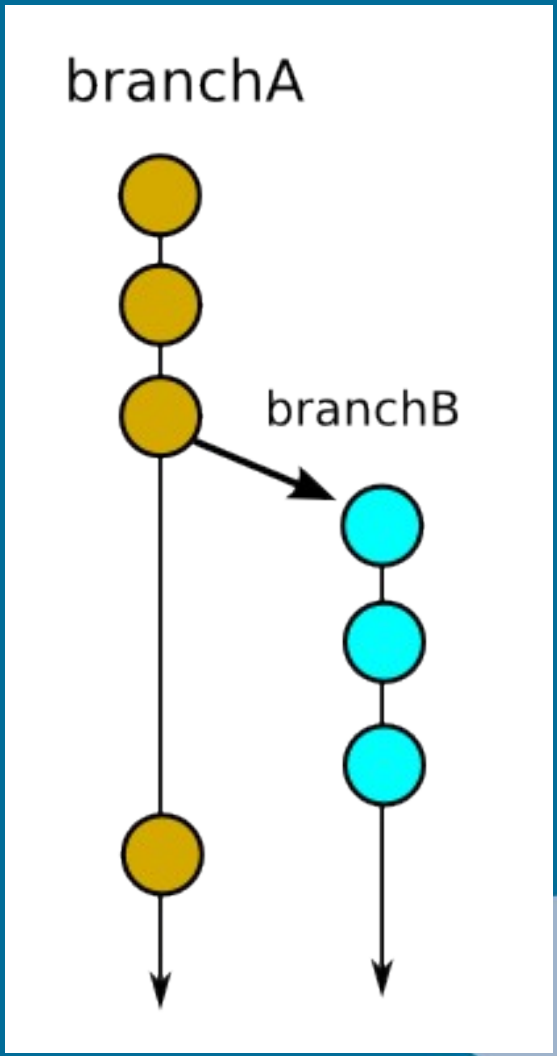
  git branch branchB

- Yeni dalda çalışmak için

  git checkout branchB

- Her ikisini tek komutta yapmak için

  git checkout -b branchB



Dallanma

- Depodaki dalları ve hangi dalda olduğunuzu görüntüleyebilmek için:

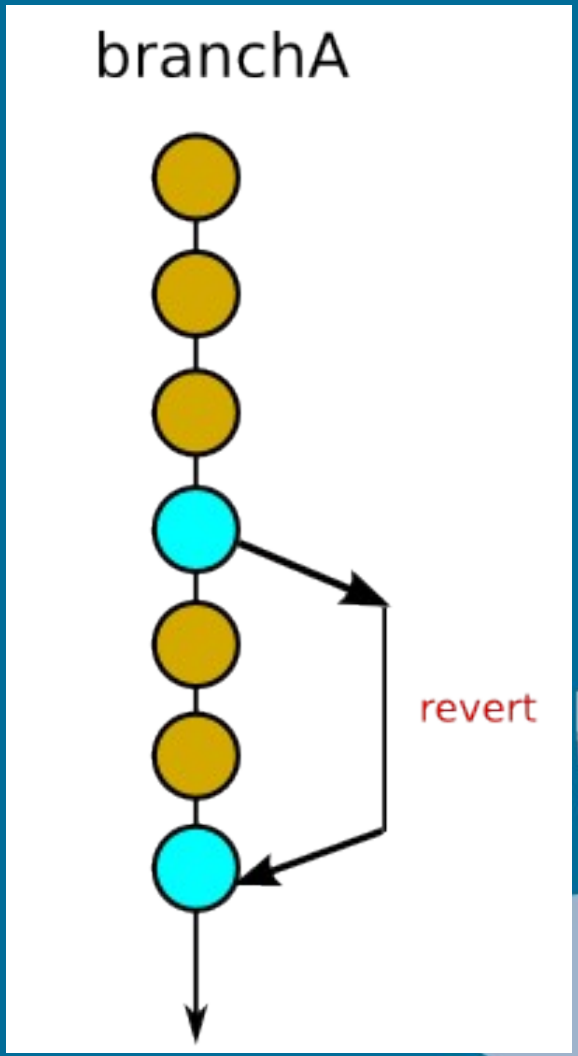
  git branch

- İki dal arasındaki farkı görebilmek için:
  - git diff branchA branchB

Geri Alma

- Git revert commit_id komutu ile yapılır ve commit_id'deki konuma geri dönülür
- Revert işlemi ayrı bir commit'dir.

  git revert a23b46



Sıfırlama

- Üç tür kullanımı vardır.
  - git reset --soft
  - git reset --mixed
  - git reset --hard
- Diyelimki depomuzda 3 commit yapılmış olsun:

  A – B – C

Git reset -- soft

A – B – C

- Git reset -- soft B komutu verildiğinde:
- HEAD, B'yi işaret eder,
- Workspace de yapılan son değişiklikler silinmez.
- C commiti ile yapılan değişiklikler de stage'de yer alır.

Git reset -- mixed

A – B – C

- Git reset -- mixed B komutu verildiğinde:
- HEAD, B'yi işaret eder,
- Workspace de yapılan son değişiklikler silinmez,
- C commiti ile yapılan değişiklikler çalışma alanında (workspace) yer alır.

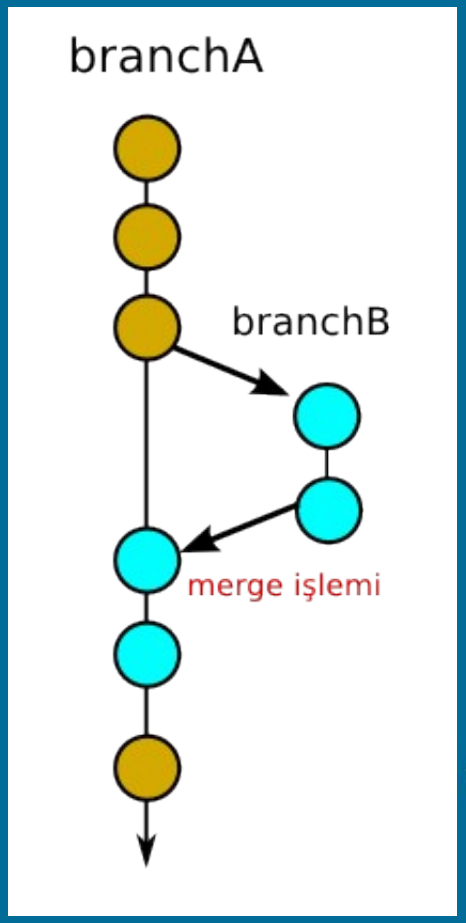
Git reset -- hard

A – B – C

- Git reset -- hard B komutu verildiğinde:
- HEAD, B'yi işaret eder,
- Workspace de yapılan bütün değişiklikler silinir,
- C commiti ile yapılan değişiklikler de silinir.
- Git reset işlemlerinden sonra mutlaka
  git status ile kontrol edilmelidir.

Birleştirme

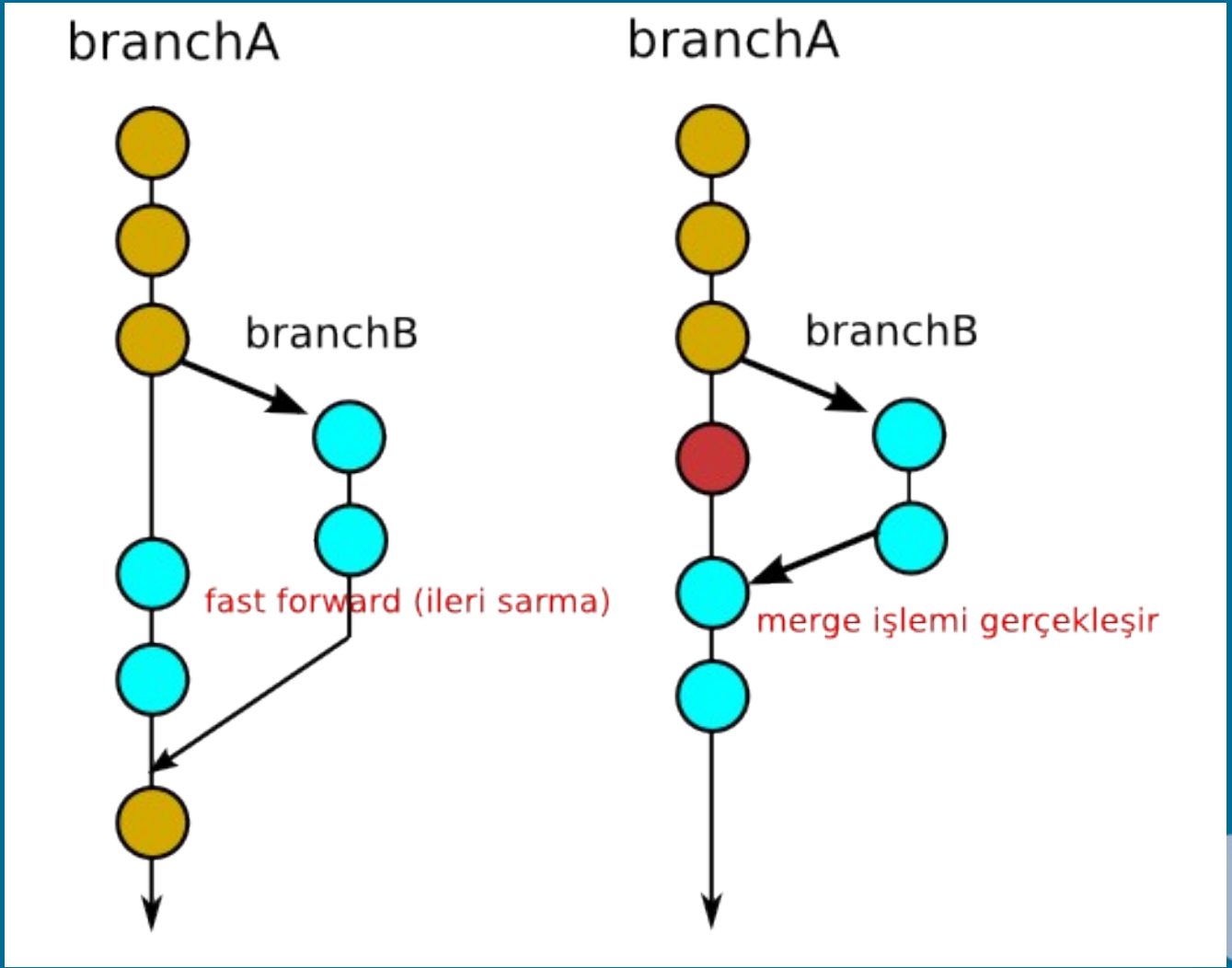
- Bir dalı üzerinde çalıştığınız dala birleştirmek için
  - git merge branchB



Çakışma (Conflict)

- Git merge komutu ile eğer otomatik birleştirme yapılamaz ise çakışma (conflict) olur.
- İlgili çakışmaları düzeltilip bir commit daha yapılır.
- Çakışma çözümleri için araçlar kullanılabildiği gibi basit bir editor ile de gerçekleştirilebilir.

İleri Sarma ve Birleştirme

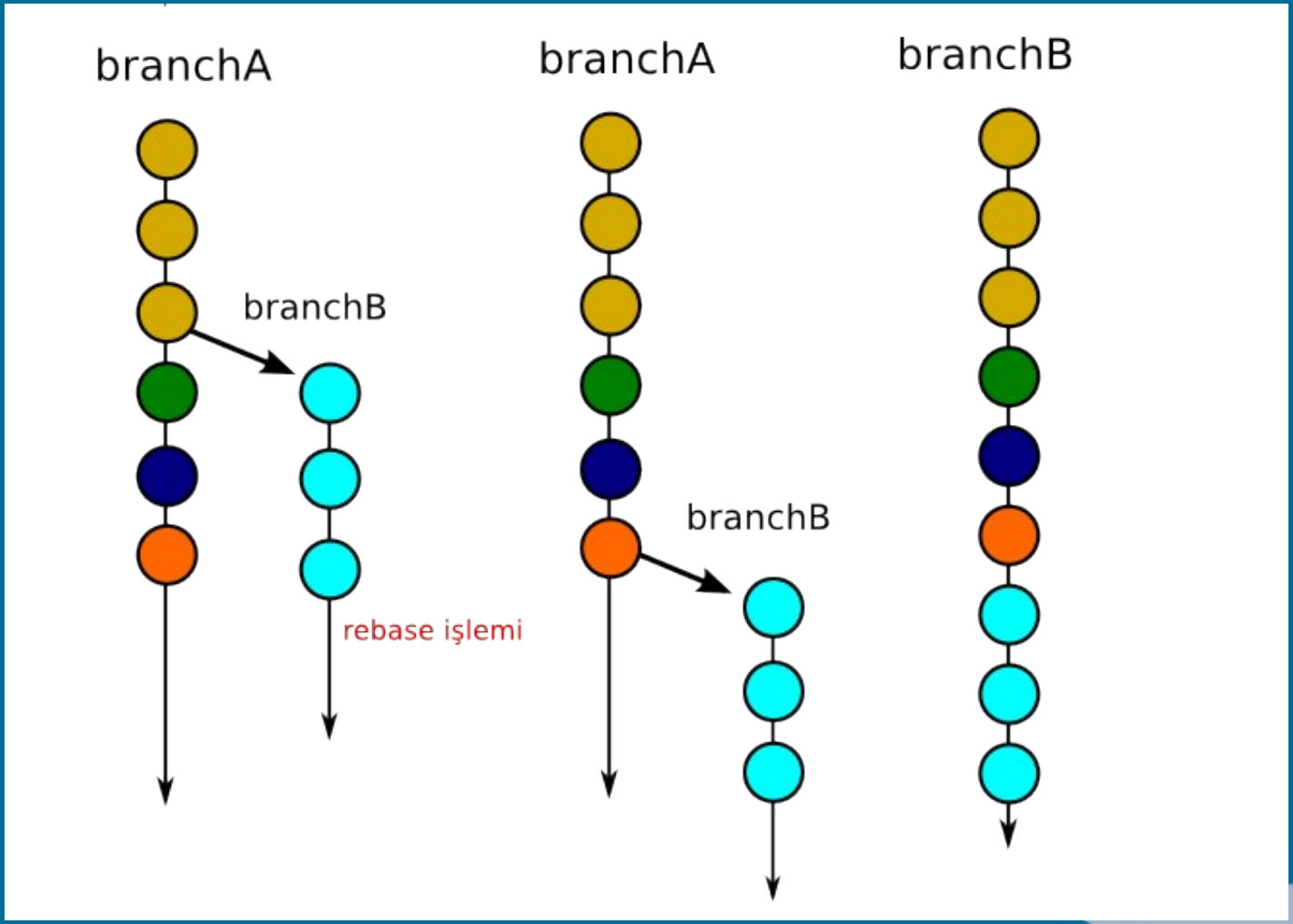


Rebase (Yeniden temellendirme)

- Bir dalı temel alarak diğerine dallandınız.
- Temel aldığınız dalda commit'ler yapıldı ve bu commit'leri yeni oluşturduğunuz dala eklemek ve sanki henüz yeni dallanmış gibi etkilenmesini isterseniz yeniden temellendirebilirsiniz:

```
git rebase branch_adi
```

Rebase (Yeniden Temellendirme)



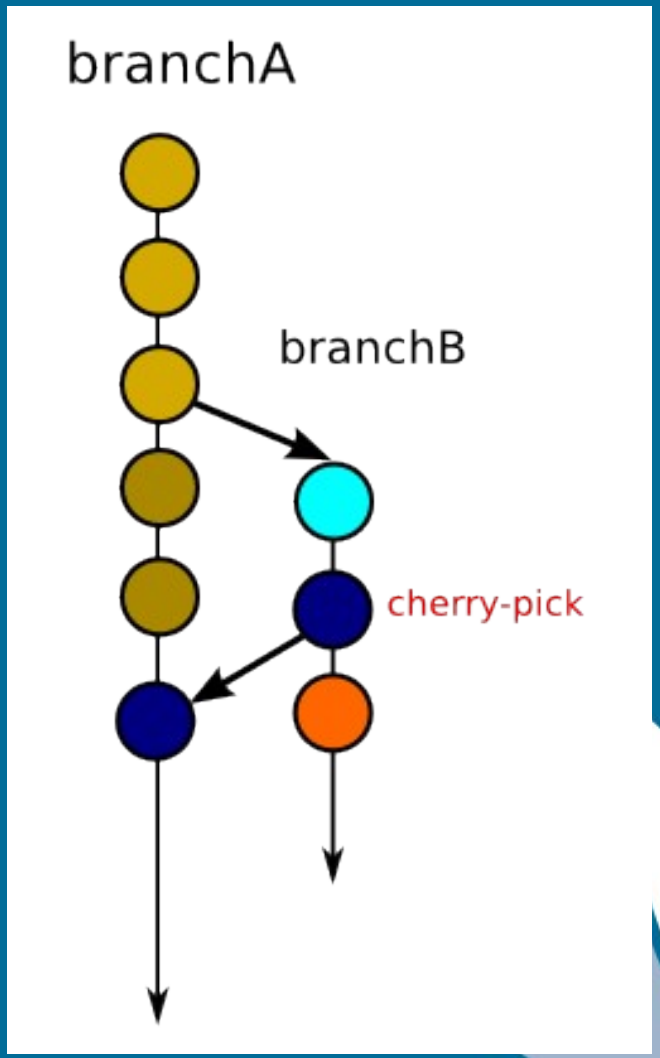
Etiketleme

- Etiketleme yapmak için
- Git tag -a v1.0.0 -m “version 1.0.0”
- Etiketleri listelemek için
- Git tag -l veya git tag -l “v1.0.*”

Cherry-pick

- Bir dala herhangi bir daldan sadece belirli bir commit'i almak istiyorsanız:

  git cherry-pick commit_id



Uzak Depo ile Çalışma

- Yerelinizde var olan bir depoya uzak depo eklemek için:
  - git remote add *remote_name url*
- Uzak depoyu yerelinize klonlamak için:
  - git clone *url*

Uzak Depo ile Çalışma

- Uzak depoya commit'leri göndermek için

  git push origin branchA

- Yerel deponuzu, yereldeki dallarınızda herhangi bir değişiklik olmadan güncellemek için:

  git fetch

- Yerel deponuzu güncellemek, uzak sunucuda yapılan commitleri yerel dallara almak için

  git pull origin branchA

  - Git pull komutu fetch ve merge işlemini gerçekleştirir.

Uzak Depo ile Çalışma

- Uzak depoda yeni bir dal oluşturmak için:
  - git push -u origin develmaster
- Uzak depoya etiketleri göndermek için
  - git push origin tag_name
  - git push origin --tags

Stash

git stash save mesajiniz

git stash

- Listelemek için

  git stash list

- Stash ile kaydettiğiniz değişiklikleri tekrar çalışma alanınıza aktarmak için

  git stash apply

  git stash pop (stack; son değişikliği uygular ve stash listesinden siler)

- Stash ile kaydettiğiniz değişikliği uygulamadan silmek için
  - git stash drop

Submodule

- Git ile bir veya birden fazla depoyu, ana deponuza ekleyerek hiyerarşik olarak geliştirme yapabilirsiniz.

  cd anadepo/altdepolar

  git clone *altdepo1_url*

  cd anadepo/

  git submodule add url ./altdepolar/altdepo1

  git commit -m “altdepo1 submodule olarak eklendi”

  git push

- Alt modüllerle ilgili veriler deponuzun kök dizininde .gitmodules dosyasında yer alır.

Git Help

- Git ile ilgili bütün komutlara git help komutu ile ulaşabilirsiniz.
- Bir komutun detaylı dökümanına ulaşmak için

  git help komut_adi

Git'i Deneyin

- Herhangi bir yazılım, araç kurmadan web tarayıcısı üzerinden git deneyin, öğrenin:

  http://try.github.io

Git – SVN

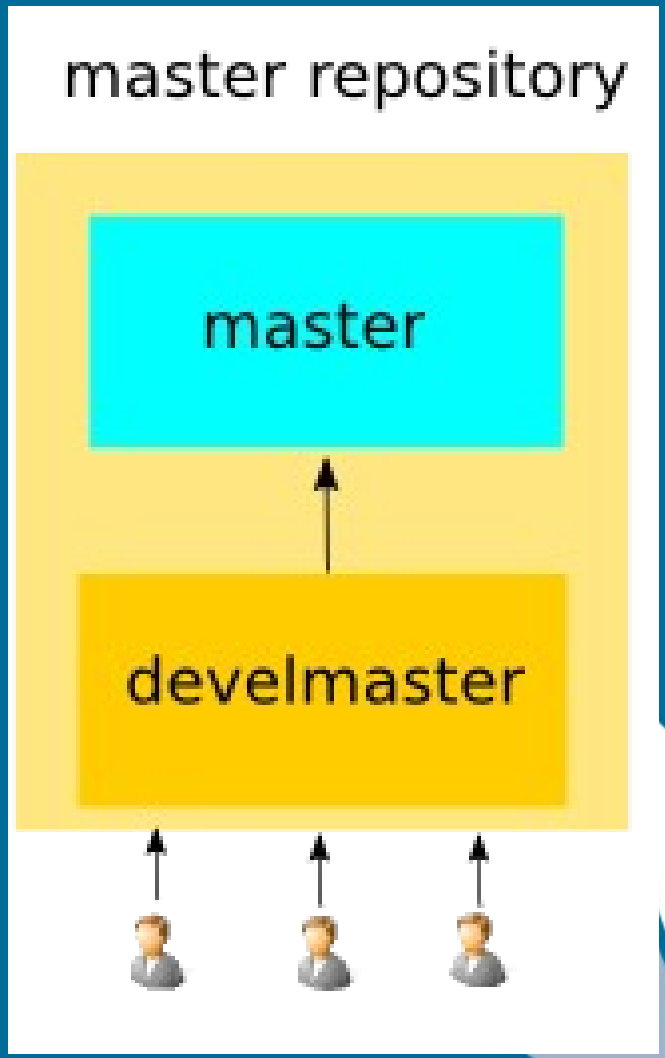
- SVN depoları için Git istemcisi kullanabilirsiniz!
  - git svn clone url -T trunk -b branches -t tags
  - git svn clone url -s (standart)
  - git show-ref
  - git commit -m “commit mesaji”
  - git svn dcommit (git push)
  - git svn fetch (git svn rebase ile commit'ler yerele alınır, git pull)
  - git svn rebase
  - git svn branch yeni_dal
- Detaylı doküman: http://git-scm.com/book/tr/Git-and-Other-Systems-Git-and-Subversion

Git ile Yazılım Geliştirme

- Git ile yazılım geliştirmek için genel olarak kullanılan üç yöntem vardır:
  - Geliştirme Dalı
  - Özellik Dalları
  - Geliştirici Depoları

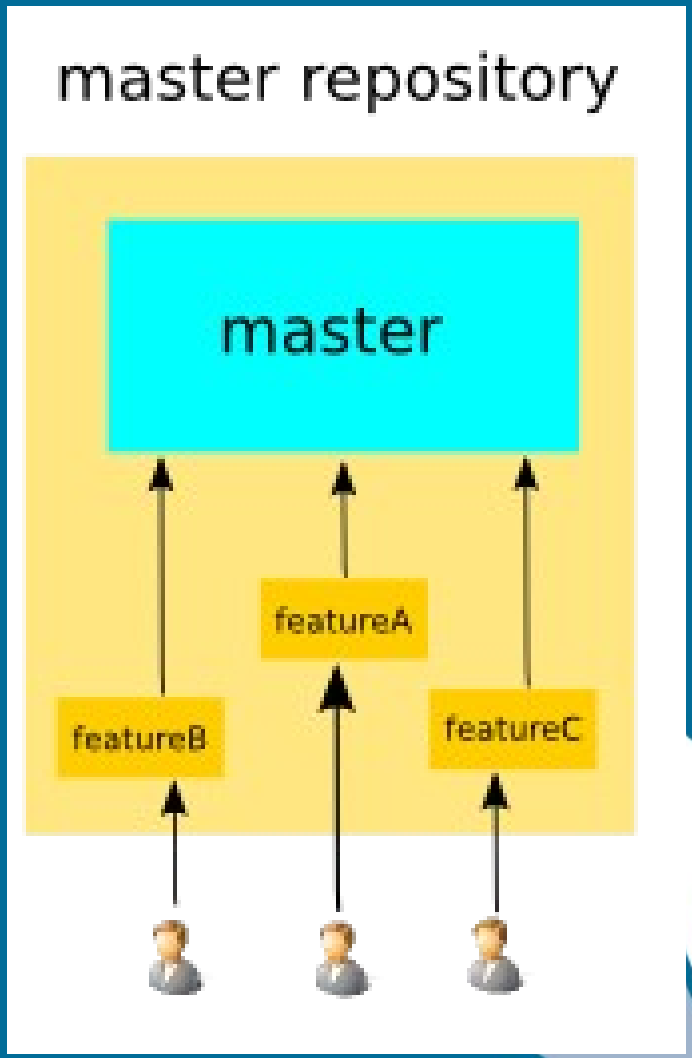
Geliştirme Dalı

- Sürüm çıkartılan anadal dışında (genellikle master) bir geliştirme dalı bulunur. (Örnek: develmaster)



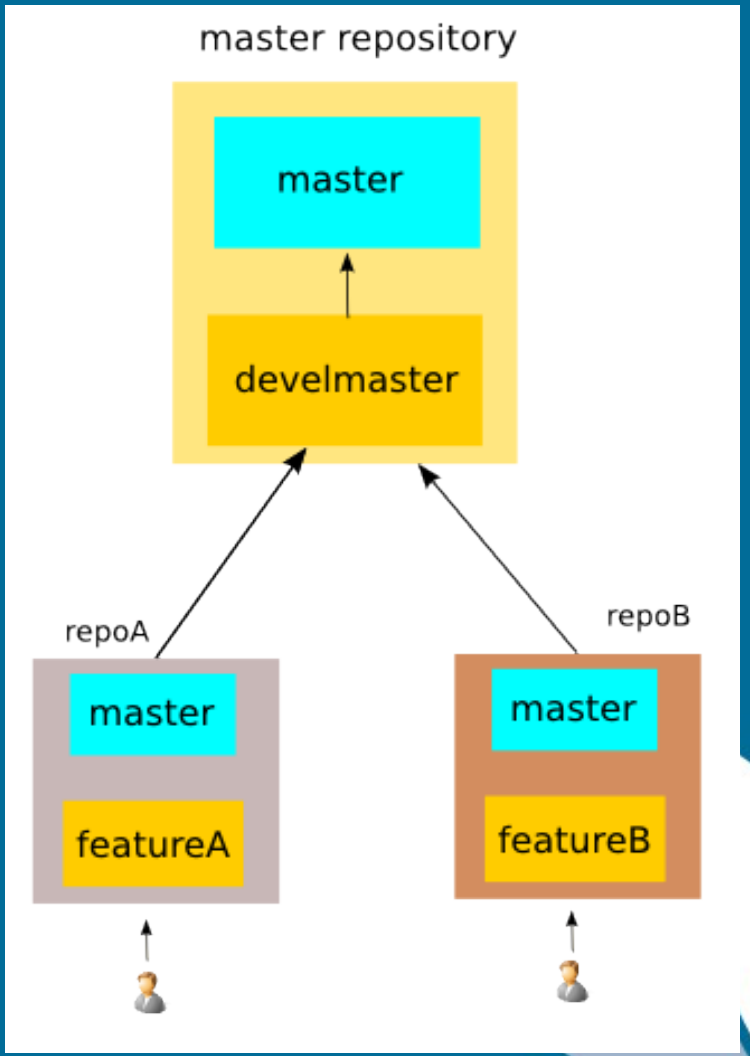
Özellik Dalları

- Yazılıma her eklenen özellik için ayrı bir dal oluşturulur.



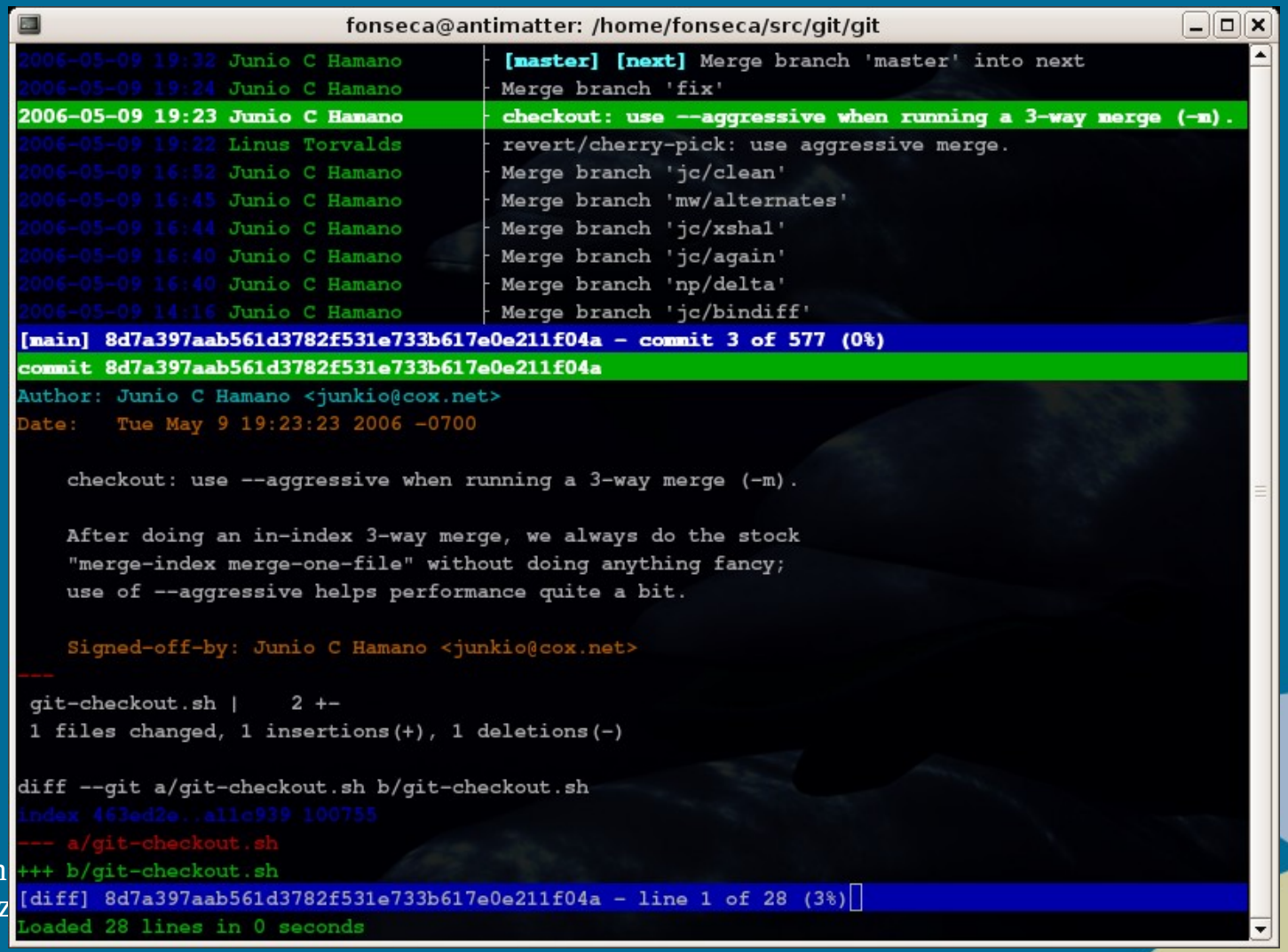
Geliştirici Depoları

- Her geliştiricinin kendine özel bir klonu vardır.

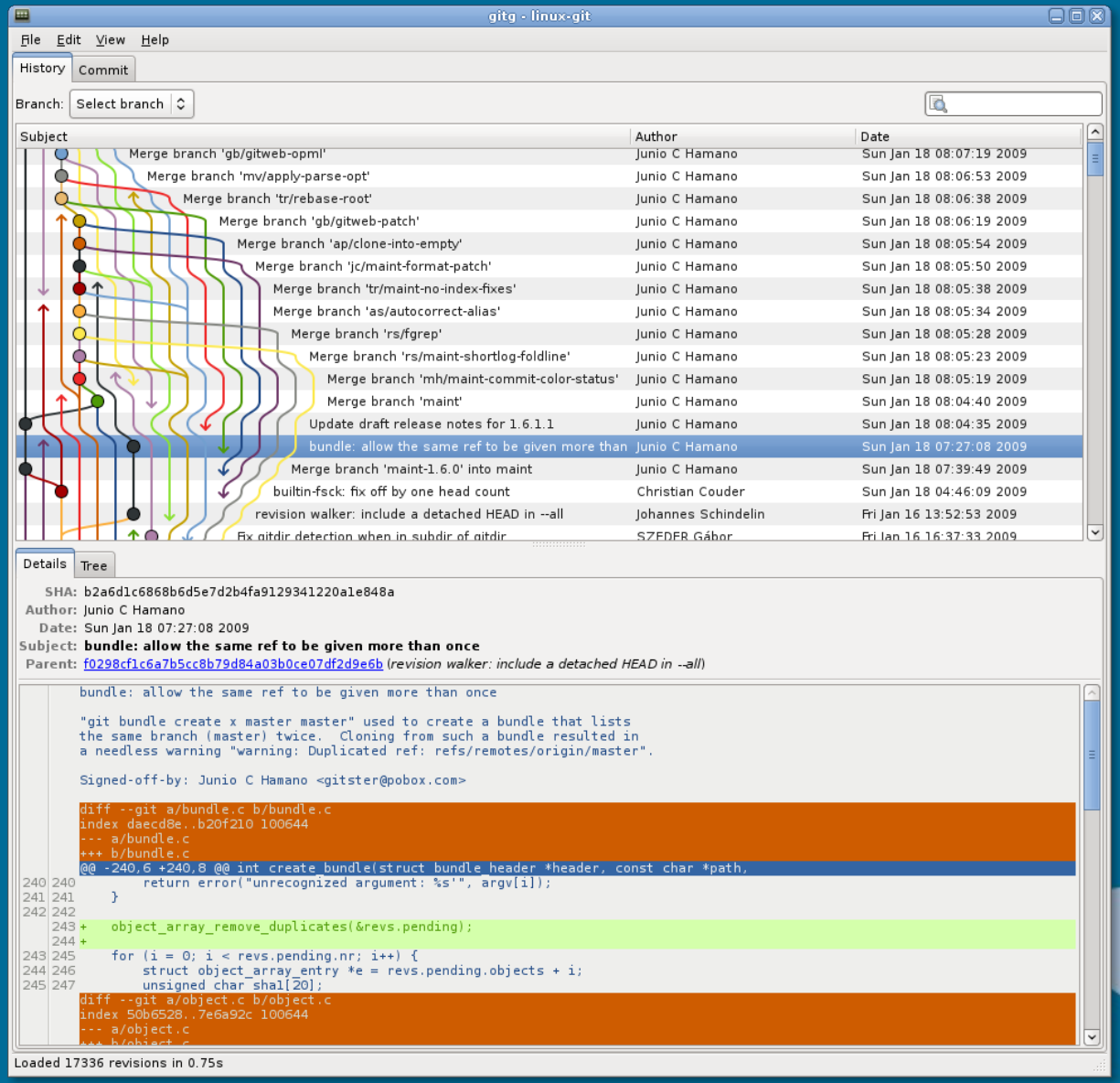


Git Araçları

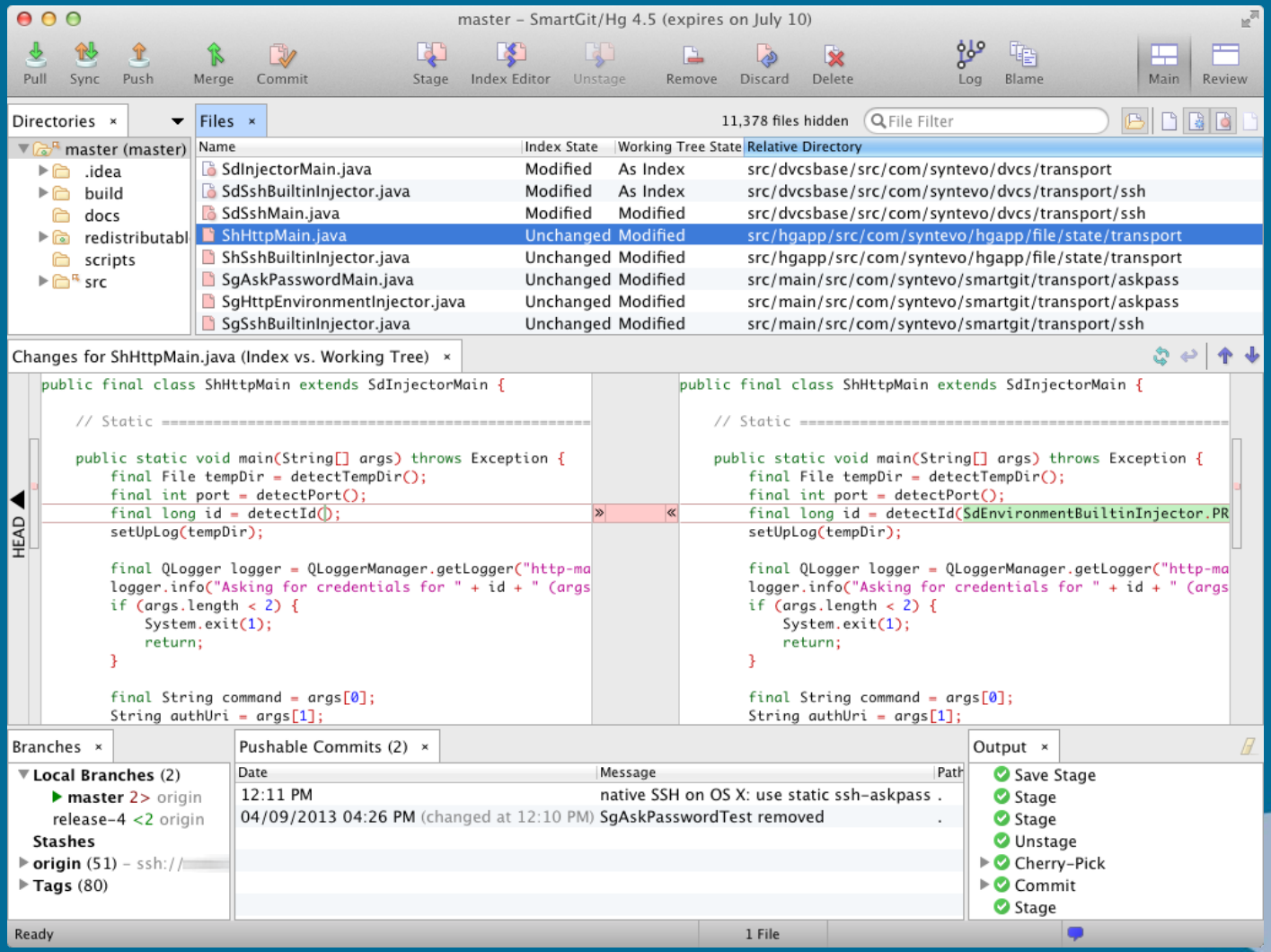
- Tig (konsol aracı)



Git Cola

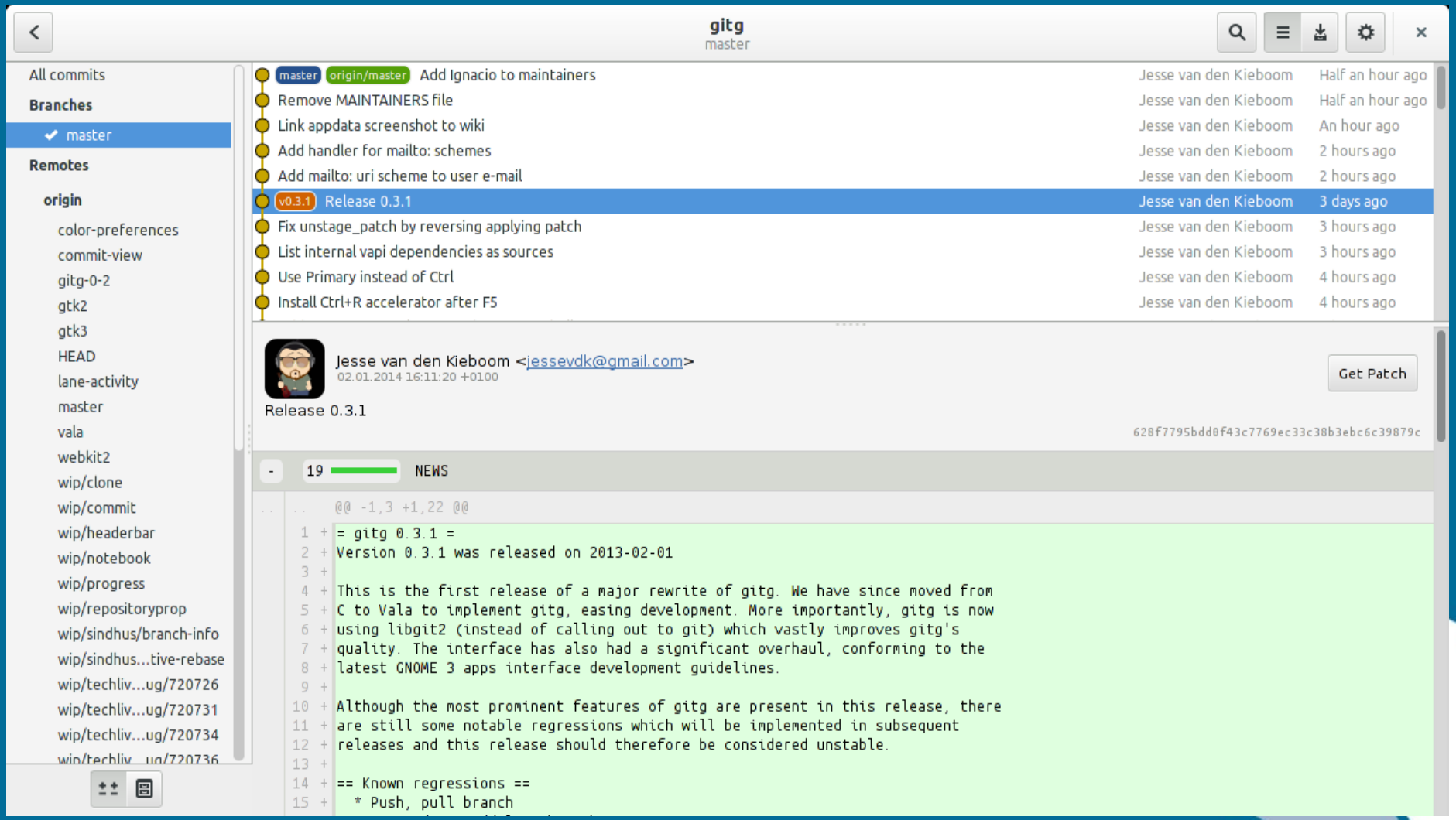

gitg - linux-git
File Edit View Help
History Commit
Branch: Select branch
Subject Author Date
Merge branch 'gb/gitweb-opml' Junio C Hamano Sun Jan 18 08:07:19 2009
Merge branch 'mv/apply-parse-opt' Junio C Hamano Sun Jan 18 08:06:53 2009
Merge branch 'tr/rebase-root' Junio C Hamano Sun Jan 18 08:06:38 2009
Merge branch 'gb/gitweb-patch' Junio C Hamano Sun Jan 18 08:06:19 2009
Merge branch 'ap/clone-into-empty' Junio C Hamano Sun Jan 18 08:05:54 2009
Merge branch 'jc/maint-format-patch' Junio C Hamano Sun Jan 18 08:05:50 2009
Merge branch 'tr/maint-no-index-fixes' Junio C Hamano Sun Jan 18 08:05:38 2009
Merge branch 'as/autocorrect-alias' Junio C Hamano Sun Jan 18 08:05:34 2009
Merge branch 'rs/fgrep' Junio C Hamano Sun Jan 18 08:05:28 2009
Merge branch 'rs/maint-shortlog-foldline' Junio C Hamano Sun Jan 18 08:05:23 2009
Merge branch 'mh/maint-commit-color-status' Junio C Hamano Sun Jan 18 08:05:19 2009
Merge branch 'maint' Junio C Hamano Sun Jan 18 08:04:40 2009
Update draft release notes for 1.6.1.1 Junio C Hamano Sun Jan 18 08:04:35 2009
bundle: allow the same ref to be given more than Junio C Hamano Sun Jan 18 07:27:08 2009
Merge branch 'maint-1.6.0' into maint Junio C Hamano Sun Jan 18 07:39:49 2009
builtin-fsck: fix off by one head count Christian Couder Sun Jan 18 04:46:09 2009
revision walker: include a detached HEAD in --all Johannes Schindelin Fri Jan 16 13:52:53 2009
Fix gitdir detection when in subdir of gitdir SZEDER Gábor Fri Jan 16 16:37:33 2009
Details Tree
SHA: b2a6d1c6868b6d5e7d2b4fa9129341220a1e848a
Author: Junio C Hamano
Date: Sun Jan 18 07:27:08 2009
Subject: bundle: allow the same ref to be given more than once
Parent: f0298cf1c6a7b5cc8b79d84a03b0ce07df2d9e6b (revision walker: include a detached HEAD in --all)
bundle: allow the same ref to be given more than once
"git bundle create x master master" used to create a bundle that lists
the same branch (master) twice. Cloning from such a bundle resulted in
a needless warning "warning: Duplicated ref: refs/remotes/origin/master".
Signed-off-by: Junio C Hamano <gitster@pobox.com>
diff --git a/bundle.c b/bundle.c
index daecd8e..b20f210 100644
--- a/bundle.c
+++ b/bundle.c
@@ -240,6 +240,8 @@ int create_bundle(struct bundle_header *header, const char *path,
return error("unrecognized argument: %s'", argv[i]);
}
+ object_array_remove_duplicates(&revs.pending);
+
for (i = 0; i < revs.pending.nr; i++) {
struct object_array_entry *e = revs.pending.objects + i;
unsigned char sha1[20];
diff --git a/object.c b/object.c
index 50b6528..7e6a92c 100644
--- a/object.c
Loaded 17336 revisions in 0.75s

Smart Git



Özgür Yazılım A.Ş.
www.ozguryazilim.com.tr

Gitg


gitg
master
All commits
Branches
master
Remotes
origin
color-preferences
commit-view
gitg-0-2
gtk2
gtk3
HEAD
lane-activity
master
vala
webkit2
wip/clone
wip/commit
wip/headerbar
wip/notebook
wip/progress
wip/repositoryprop
wip/sindhus/branch-info
wip/sindhus...tive-rebase
wip/techliv...ug/720726
wip/techliv...ug/720731
wip/techliv...ug/720734
master origin/master Add Ignacio to maintainers Jesse van den Kieboom Half an hour ago
Remove MAINTAINERS file Jesse van den Kieboom Half an hour ago
Link appdata screenshot to wiki Jesse van den Kieboom An hour ago
Add handler for mailto: schemes Jesse van den Kieboom 2 hours ago
Add mailto: uri scheme to user e-mail Jesse van den Kieboom 2 hours ago
v0.3.1 Release 0.3.1 Jesse van den Kieboom 3 days ago
Fix unstage_patch by reversing applying patch Jesse van den Kieboom 3 hours ago
List internal vapi dependencies as sources Jesse van den Kieboom 3 hours ago
Use Primary instead of Ctrl Jesse van den Kieboom 4 hours ago
Install Ctrl+R accelerator after F5 Jesse van den Kieboom 4 hours ago
Jesse van den Kieboom <jessevdk@gmail.com>
02.01.2014 16:11:20 +0100
Get Patch
Release 0.3.1
628f7795bdd0f43c7769ec33c38b3ebc6c39879c
19 NEWS
@@ -1,3 +1,22 @@
= gitg 0.3.1 =
Version 0.3.1 was released on 2013-02-01
This is the first release of a major rewrite of gitg. We have since moved from
C to Vala to implement gitg, easing development. More importantly, gitg is now
using libgit2 (instead of calling out to git) which vastly improves gitg's
quality. The interface has also had a significant overhaul, conforming to the
latest GNOME 3 apps interface development guidelines.
Although the most prominent features of gitg are present in this release, there
are still some notable regressions which will be implemented in subsequent
releases and this release should therefore be considered unstable.
== Known regressions ==
* Push, pull branch

Gitlab

Git Hosting

- Gitorius : gitorius.org
- Gitlab : gitlab.com/cloud
- Github: github.com
- Bitbucket: bitbucket.org
- Google Code: code.google.com

SORULAR?

Ulaşmak isterseniz?

- Kişisel web günlüğü:
  http://omerozkan.net
- Eposta:
  omer@ozkan.info
- Twitter:
  @omerozkan_
- Facebook
  facebook.com/omerozkan

TEŞEKKÜRLER

Özgür Yazılım A.Ş.
www.ozguryazilim.com.tr