



گیت (Git)

علی قاسم پور

www.j-22.ir

فهرست مطالب :

Version Control Systems	۱
What is Git ?	۲
Start with Git	۳
What is GitHub ?	۴
Thanks / Ask Qestion	۵



Git





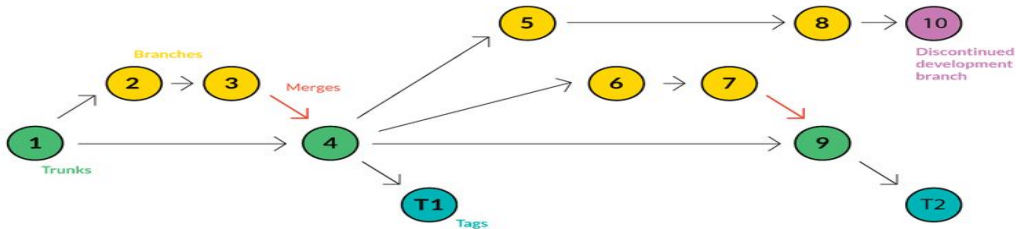
فصل اول

Version Control Systems



Version Control Systems

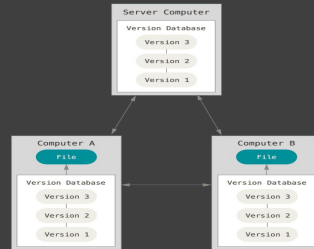
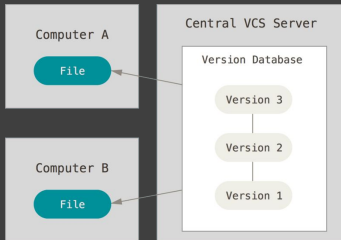
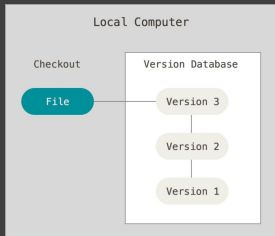
What is "version control"?



کنترل نسخه یک سیستم است که تغییرات بر روی یک فایل یا چند فایل را در طول بازه زمانی ثبت و هرگاه نیاز باشد امکان بازیابی آن است .
طبق صحبت Eric Sink سه هدف اصلی برای اینکه از کنترل نسخه استفاده کنیم است :

- ۱ - برای آنکه دولوپر ها با هم بر روی یک پروژه کار کنند .
- ۲ - برای جلوگیری از تکرار های احتمالی توسط دولوپر هایی که بر روی یک پروژه کار می کنند .
- ۳ - برای ثبت هر نسخه از هر چیزی که تابحال وجود داشته !

Different types of V.C.S



Local

بسیاری از مردم برای نسخه‌گذاری دایرکتوری و فایل‌های مختلفی بر روی کامپیوتر خود می‌ساختند و با نام و زمان آن‌ها را جدا می‌کردند. مهم‌ترین مشکلات این سیستم را میتوان به در دسترس نبودن برای بقیه و فراموشی و پاک کردن اشتباهی فایل‌ها نام برد. RCS یکی از ابزارهای این مدل از VCS است.

Centralized

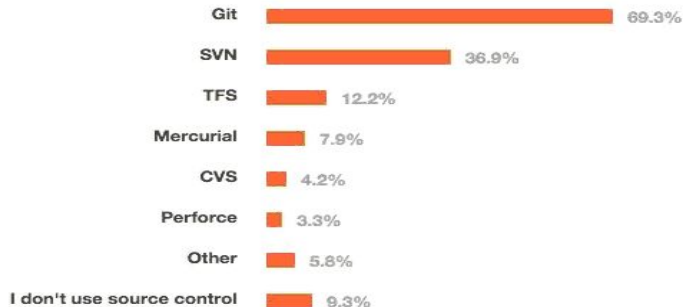
نسخه بعدی یک سرور برای قرار دادن نسخه‌ها و بقیه کلاینت‌ها به آن وصل می‌شوند. از مزایای آن می‌توان به دسترسی به آخرین نسخه‌ها و همچنین همکاری بین دولوپر‌ها اشاره کرد ولی با مشکل **Single point of failure** مواجه‌اند! از ابزارهای میتوان به CVS, Subversion و Perforce اشاره کرد.

Distributed

نسخه به‌نیه شده آنها به صورت بخشی در کل کلاینت‌ها است. به این صورت که هر کلاینت فقط آخرین نسخه فایل‌ها را چک نمی‌کند بلکه یک **mirror** از مخازن است. پس با افتادن یک کلاینت تمام کپی فایل‌ها در بقیه است! از ابزارهای آن میتوان به Git, Bazaar, Mercurial, و Darcs نام برد.

Quick Comparison Git,SVN,Mercurial,CVS

SOURCE CONTROL



16,694 responses

Chart from Stackoverflow's 2015 Developer Survey



Git
Δ





فصل دوم

What is Git ?

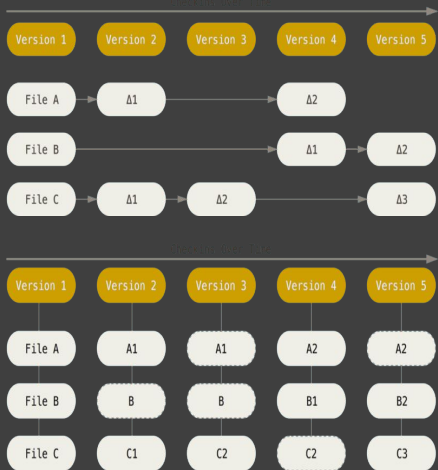


A Short History of Git

کرنل لینوکس به عنوان یک پروژه عظیم متن باز شروع به پیشرفت کرد و در کنار آن نرم افزار های مختلفی پیشرفت داده شد. این برنامه ها با یک مدل DVCS به نام Bitkeeper در سال ۲۰۰۲ شروع به نسخه گذاری کردند. در سال ۲۰۰۵ این رابطه بین برنامه نویسان شکست و شرکت BitKeeper بر شکست شد. در میان لینوس تروالدس تلاش برای ساخت یک ابزار کنترل نسخه به نام Git کرد که از ویژگی های زیر برخوردار باشد :

- سرعت
- طراحی ساده
- پشتیبانی از دولوپ غیر خطی (به صورت همروند باشد)
- کاملا DVCS
- توانایی مدیریت پروژه های بزرگ (داده های زیاد و سرعت بالا)

Why Git ?



Snapshot نه تغییرات

تفاوت اصلی Git با باقی VCS ها بر آن است که دیگران (مانند Subversion) فقط به لیست تغییرات فایل ها توجه می کنند ولی گیت به خود فایل ها توجه می کند !

گیت ترجیح بر آن دارد یک Snapshot از کلیه فایل ها و تغییرات به وجود آمده است پیش خود نگه دارد به طوری که میتوان آن را به mini filesystem نامید.

پس گیت فقط یک مدیریت روی نسخه ها نیست بلکه تمامی نسخه هارا همراه خود دارد .

Why Git ?

Git generally only add data

- تغییراتی که در گیت انجام می شود برگشت ناپذیر است .
- در بقیه VCS ها می توان فایل را تغییر داد و این باعث آسیب های احتمالی به پروژه است.
- این ویژگی باعث آن شده که بتوان تغییرات و تست های مورد نظر را بدون ترس از خرابی فایل ها انجام داد.

Git Has integrity

- تمام اجزای در گیت **Checksum** دارند و توسط آن کنترل می شوند. پس تقریباً غیر ممکن است محتوای این فایل ها یا دایرکتوری ها را تغییر داد.
- گیت از مکانیزم **Sha1sum** استفاده می کند .
- گیت علاوه بر فایل ها **Hash** آن ها را نیز در دیتابیس خودش نگه میدارد.

Every Operation is local

- بیشتر عملیات ها در **Git** فقط به فایل ها و منابع کامپیوتر مبداء نیاز دارد .
- بیشتر **VCS** سربرابر زیادی روی شبکه دارند اما گیت چون تمام سابقه را دارد نیازی به ترافیک در بستر شبکه را ندارد.



فصل سوم

Start with Git



Installing Git



گیت برای تمامی سیستم های عامل موجود است و میتوان آن را دانلود کرد.

<https://git-scm.com/download>

برای دانلود به سایت :

همچنین میتوانید آن را از طریق مخازن موجود برای لینوکس دانلود کرد :

• برای RedHat base :

```
$ sudo yum install git-all
```

• برای Debian base :

```
$ sudo apt-get install git-all
```

همچنین می توان آن را از سورس دانلود , کامپایل و نصب کرد .

First-Time Git Setup

برای تنظیم متغیر ها و موارد مورد نیاز از ابزاری به نام Git config استفاده می کنیم .

Your Identity :

```
$ git config --global user.name "Ali Ghasempour"
```

```
$ git config --global user.email ali.ghasempour@gmail.com
```

در این بخش فقط یکبار از آپشن --global استفاده می شود و از آن به بعد هرکس که بخواهد جدا از این اطلاعات کار کند باید این آپشن را استفاده نکند .

Your Editor :

```
$ git config --global core.editor vim
```

برای مثال در ویندوز نیز برای تعریف ویرایشگر به صورت زیر عمل می کنیم :

```
$ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe'  
-multinst -nosession"
```

First-Time Git Setup

Checking Your Setting :

```
$ git config --list
```

```
user.name=Ali Ghasempour
```

```
user.email=ali.ghasempor@gmail.com
```

```
color.status=auto
```

```
color.diff=auto
```

تمامی این تنظیمات در فایل هایی بسته به نسخه سیستم عامل شما در مکان های مختلف ذخیره می شوند. در بیشتر توزیع ها این فایل در **~/.gitconfig** یا **~/.config/git/config** هستند. همچنین پروفایل همه کاربران در سیستم در فایل **/etc/gitconfig** هست که با آپشن **--system** در دستور **git config** می توان مستقیماً این فایل را تغییر داد.

Getting a Git Repository

برای ساخت Repository یا مخزن باید ابتدا :

```
$ git init
```

در این حالت یک دایرکتوری به نام `.git` در پروژه شما می سازد. برای اضافه کردن فایل های مورد نظر از دستور زیر کمک میگیریم :

```
$ git add *.c
```

```
$ git commit -m 'initial project version'
```

در این مثال تمام فایل هایی که با `C` تمام میشوند توسط گیت ترک میشوند و همچنین کامنت نیز به آن اضافه کردیم .

Getting a Git Repository

اگر می خواهید در یک پروژه شرکت داشته باشید می توانید تمامی فایل ها و تمامی نسخه های قبل را با دستور Clone دریافت کنید :

```
$ git clone https://github.com/libgit2/libgit2 mylibgit
```

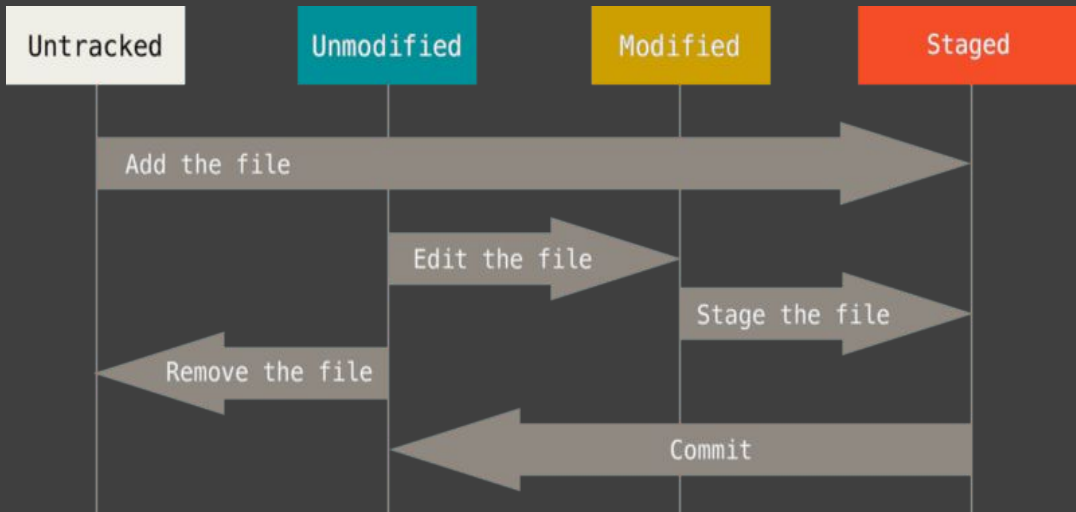
در این مثال یک دایرکتوری به نام mylibgit می سازد و فایل های مورد به ادرس فوق را در آن قرار می دهد.

Git از پروتکل های مختلفی استفاده می کند :

```
https://  
git://  
user@server://path/to/repo.git
```

که دو روش آخر از ssh استفاده می کنند.

Recording Changes to the Repository



Working With Remote

نسخه های پخشى از VCS ها مانند گیت گاهی از یک سرور اصلی (چه در اینترنت چه شبکه داخلی) برای تبادل اطلاعات استفاده می کنند. به این صورت که می توانید آخرین تغییرات را PUSH یا PULL بکنید :

PUSH :

```
$ git push origin master
```

PULL :

```
$ git pull origin master
```

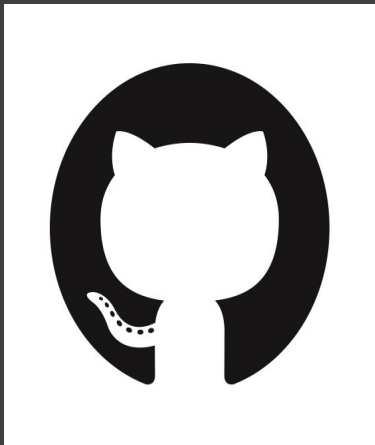


فصل چهارم

What is GitHub ?



What is GitHub ?



GitHub

گیت هاب یک شبکه اجتماعی است که بر پایه VCS گیت کار می کند. GitHub از دو بخش Git و Hub تشکیل شده است که Hub به معنی اجتماعی و تفریح است. در این شبکه بر پایه وب می توانید تمامی اعمال گیت را انجام داده و با توجه به قابلیت های تعاملی آن با بقیه افراد ارتباطات نزدیک تر داشته باشید. نحوه کار به گیت هاب دقیقا مشابه با گیت است با این تفاوت که بسیار در دسترس تر , زیبا تر و تعاملی تر است .




سوال , انتقاد , نظر



 WWW.j-22.ir

 ali.ghasempor@gmail.com

 0933-439-6492

1. <https://git-scm.com>
2. <https://webinerds.com/version-control-systems-keep-your-code-in-order/>
3. <http://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>



با تشکر از توجه شما

