

ضرورت استفاده از یک سیستم کنترل نسخه:

در طول روند تولید یک برنامه، چه به صورت تیمی و یا حتی انفرادی، بارها برای برنامه نویسان این نیاز پیش می‌آید که به نسخه‌های قدیمی‌تر فایل‌های خود دسترسی داشته باشند تا بتوانند آنچه را که در قبل نوشته‌اند مورد بازبینی قرار دهند. شاید کسانی که با سیستم‌های مدیریت نسخه آشنایی ندارند، این کار را با استفاده از copy و paste کردن فایل‌ها در پوشه‌های جداگانه انجام دهند؛ اما روند توسعه یک برنامه در محیط عملی، امکان استفاده از چنین روشی را به ما نمی‌دهد. زیرا مدیریت این فایل‌ها علی‌الخصوص در پروژه‌های تیمی، بعد از مدتی بسیار دشوار خواهد شد. بنابراین نیاز به سیستمی احساس می‌شود که بتواند این کار را به صورت خودکار انجام دهد.

وظیفه اصلی یک سیستم مدیریت کد، ایجاد یک رویه خودکار جهت دنبال کردن تغییرات فایل‌های ما است به طوری که بگویید هر فایل در چه زمانی، توسط چه کسی، به چه دلیل، چه تغییری کرده است.

آشنایی با Git:

Git توسط سازنده سیستم عامل لینوکس یعنی آقای Linus Torvalds و برای مدیریت کدهای آن ساخته شد که بعدها توسط Linux-BitKeeper ارتقا یافت. BitKeeper یک سیستم مدیریت کد توزیع شده است که البته رایگان نیست. تیم BitKeeper در ابتدا پروژه لینوکس را به صورت رایگان پشتیبانی می‌کرد اما در سال 2005 این حمایت را قطع کرد. در این هنگام تیم توسعه لینوکس تصمیم گرفت که خود یک سیستم مدیریت کد توزیع شده ایجاد کند. آن‌ها این سیستم را با Perl و C نوشتند و آن را برای اجرا شدن بر روی انواع سیستم عامل‌ها نظیر لینوکس و ویندوز و حتی مک آماده کردند اهداف اصلی Git عبارتند از:

- (1) سرعت بالا
- (2) سادگی
- (3) قدرت پشتیبانی بالا از Merge/Branching
- (4) یک سیستم کاملاً توزیع شده
- (5) قابلیت توسعه برای پروژه‌های بزرگ

تفاوت سیستم‌های متمرکز و توزیع شده:

سیستم‌های کنترل نسخه را می‌توان بر اساس خصوصیات مختلف در دسته‌های متفاوتی قرار داد اما از نظر معماری سیستم، به دو دسته‌ی زیر تقسیم می‌شوند:

- (۱) VCS (Version Control System) - سیستم‌های مدیریت نسخه متمرکز
 - (۲) DVCS (Distributed Version Control System) - سیستم‌های مدیریت نسخه توزیع شده
- در ادامه مقاله تفاوت این دو روش را بیان خواهیم نمود و به بررسی مزایا و معایب آن‌ها خواهیم پرداخت

تعریف Repository:

مخزن یا همان Repository محلی است که یک سیستم مدیریت نسخه از آن برای نگهداری تغییرات فایل‌ها استفاده می‌کند. در سیستم‌های VCS این مخزن به صورت متمرکز یا اصطلاحاً Centralized Repository می‌باشد. به این معنا که یک Repository بر روی یک ماشین، خواه سیستم خود برنامه نویسی (در پروژه‌های انفرادی) و خواه یک سرور قرار دارد (در پروژه‌های تیمی) و برنامه

نویسان تغییرات فایل‌های خود را به سمت این سرور می‌فرستند و این سرور وظیفه نگهداری تمامی نسخه‌ها و اطلاعات مربوطه از برنامه نویسان مختلف را به عهده دارد. اشکال این روش در این است که برنامه نویس تنها به نسخه جاری که بر روی سیستم خود است دسترسی دارد و اگر بنا به دلیلی بخواهد از نسخه‌های پیشین استفاده کند باید آن را از سرور بخواهد که این کار مشکل دیگری ایجاد می‌کند و آن این است که ممکن است برنامه نویس همیشه در موقعیتی نباشد که بتواند به سرور دسترسی داشته باشد. به همین دلیل این روش وابستگی زیادی برای برنامه نویس ایجاد می‌کند اما پیاده سازی این روش آسان‌تر از مدل توزیع شده است.

در مدل توزیع شده علاوه بر یک مخزن که بر روی یک سرور قرار داد و تمامی نسخه‌ها در آن جا نگهداری می‌شود، هر برنامه نویس یک نسخه محلی مخزن را نیز در اختیار دارد. به این ترتیب وابستگی برنامه نویس به سرور کاهش می‌یابد؛ همچنین می‌توان با ایجاد SubRepositoryها یک ساختار درختی ایجاد نمود که هر کدام از این زیر سیستم‌ها در نهایت اطلاعات را در سرور اصلی قرار می‌دهند. علاوه بر این به دلیل ساختار توزیع شده، امکان بک آپ گیری در این روش مطمئن‌تر است. زیرا تنها وابسته به یک سرور نیست و می‌تواند بر روی سیستم‌های مختلف توزیع شده باشد. البته از اشکالات این روش پیچیدگی پیاده سازی بیشتر آن نسبت به سیستم‌های متمرکز است.

اما سوال این جا است که ما حقیقتا چه چیزی را باید ذخیره کنیم ؟

پاسخ به این سوال بسیار ساده است: هر آنچه برای ما مهم است که این شامل فایل‌های کد، فایل‌های پیکربندی، خروجی‌های نظیر dll و غیره است. البته در این بین استثنائاتی نظیر فایل‌های EXE و یا پکیج‌های نصب شده وجود دارد که در بسیاری از موارد نیازی به پیگیری نسخه‌های آن‌ها نیست اما تمامی این‌ها وابسته به نظر برنامه نویس است.

در ادامه مقالات ما به تعاریف مورد نیاز در سیستم‌های مدیریت کد، ساختار Git و چگونگی نصب و استفاده آن خواهیم پرداخت.

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۱۱ ۳:۱۵

سلام

من هم به صورت تجربی باهاش کار کردم برای همین با بعضی از اصطلاحات مشکل دارم و معنی شون رو نفهمیدم مثل push یا pull و ... بسیار خوبه که مرجعی به زبان فارسی این مفاهیم رو توضیح بده .

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۰ ۲۱:۴۴

سلام

تشکر از مقالات مفیدتون - بنده تا بحال از سیستم مدیریت کد استفاده نکردم . به نظر شما برای شروع ، بهتر هست که از چه سیستمی شروع کنم ؟ تعریف SVN و Git رو شنیدم ، اما نیاز به راهنمایی دقیق تری دارم . با تشکر

نویسنده: AliReza
تاریخ: ۱۳۹۱/۰۸/۲۳ ۹:۵۱

سلام

مقاله خیلی خوبی بود من قبلا SVN را کمی کار کردم ولی خوب نتونستم از آن استفاده کنم

ولی چطور Git را تهیه کنیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۳ ۹:۵۸

[برچسب Git](#) را در سایت دنبال کنید. در قسمت سوم آن به نحوه تهیه و نصب اشاره شده.

نویسنده: مهرداد
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۳۳

عالی بود

نویسنده: علی پناهی
تاریخ: ۱۳۹۲/۱۰/۲۰ ۲۲:۴۸

کسی در مورد نصب svn یا git بر روی کامپیوتر خونه و اتصال کاربران از طریق اینترنت می تونه راهنمایی بکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۰ ۲۳:۴

- نصب آن را در [قسمت 4](#) این سری پیگیری کنید. کار با یک سرور ریموت را در [قسمت 9](#) آن مطالعه کنید. البته در این بین تمام قسمت ها را باید مطالعه کنید تا [نظم منطقی آن](#) برقرار شود.

- اینترنت خانه شما اگر IP ثابت دارد، به همین ترتیب از بیرون قابل استفاده خواهد بود (البته [پورت 9418](#) را باید در فایروال سیستم باز کنید). اگر ندارد یک [VPS](#) ارزان بخرید و Git را روی آن نصب کنید یا با ISP خودتان برای گرفتن IP ثابت مذاکره کنید (می فروشند). یا اینکه مثلا سایتی مانند [BitBucket](#) ، مخزن Git خصوصی رایگان تا 5 نفر عضو گروه را نیز به شما ارائه می دهد.