

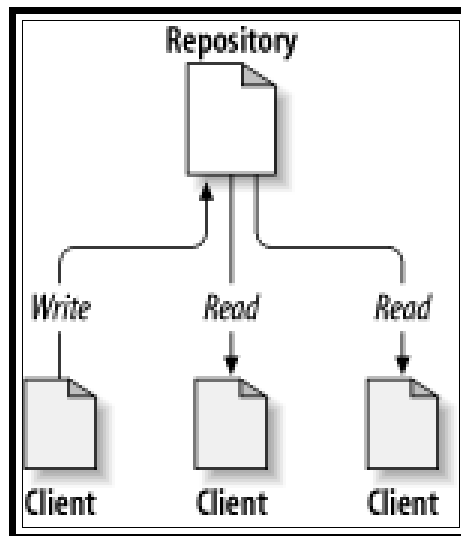
آشنایی با Subversion بخش دوم

مفاهیم اولیه

تمام سیستمهای کنترل ورژن به دنبال یک هدف ساده و اولیه هستند که همانا ساده سازی مشارکت کاربران مختلف در انجام پروژه و مدیریت فایلها به قسمی است که هیچ کاربری نتواند پا در کفش کاربر دیگری کرده و کار او را خواسته و یا ناخواسته خراب کند. مشکل کنترل ورژن و پیامدهای آن چیزی نیست که امروزه به آن پی برده باشند، بلکه از بدو ایجاد کامپیوترها و برنامه نویسی گسترده و گروهی این مشکل وجود داشته و همیشه سعی شده که راههایی برای مقابله با آن طراحی شود.

مخزن فایل (Repository)

در قلب subversion یک مخزن فایل وجود دارد که تمام فایلهای پروژه در آن ذخیره می شود. این سیستم دقیقاً یک سیستم خادم/مخدوم (Client/server) است، به این معنا که کل فایلهای یک پروژه درون مخزن فایل سرور مرکزی قرار میگیرد و تمام اعضاء پروژه در قالب دستگاههای کلاینت به سرور متصل شده و فایلها را مورد نیاز را دانلود می کنند. هر کاربر میتواند پس از انجام تغییرات لازم، فایلهای تغییر داده شده را دوباره به مخزن برگرداند و بدین صورت بقیه را از انجام تغییرات باخبر کند.

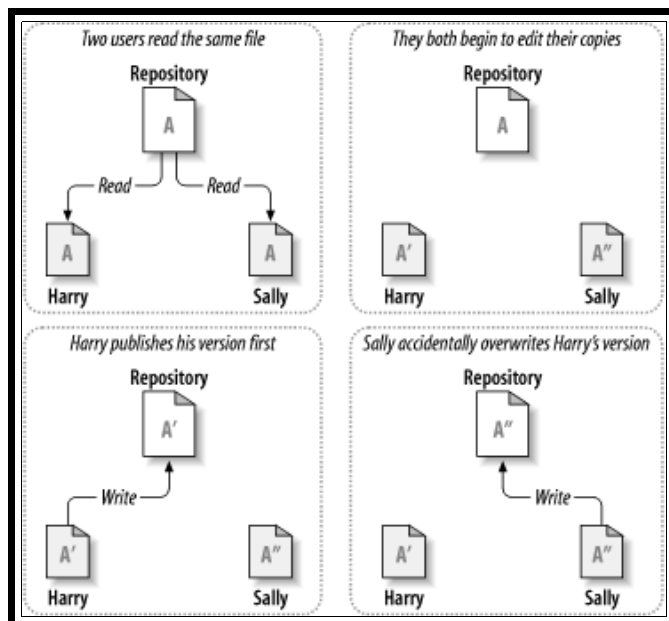


شکل ۱ یک سیستم خادم/مخدوم نمونه

با دیدن این مدل ممکن است این تصور در شما ایجاد شود که subversion تنها یک فایل سیستم است، و خوب این کاملاً تصور درستی است! منتها subversion یک فایل سیستم معمولی و مشابه چیزهایی که تا بحال با آنها آشنا بوده اید نیست، عاملی که subversion را در این قضیه متمایز میکند این است که subversion فایل سیستمی است که تمام تغییراتی که در آن نوشته شود را بخاطر می سپارد؛ تمام تغییراتی که روی تمام فایلها و یا حتی شاخه ها انجام شود، و این شامل عملیاتی چون پاک سازی، اضافه کردن و حتی عوض کردن ترتیب فایلها/شاخه ها می شود. وقتی که یک کاربر در حال دریافت فایلها/شاخه های پروژه از این فایل سیستم است، در واقع در حال مشاهده آخرین نسخه فایلهای موجود در فایل سیستم می باشد. منتها او امکان این را دارد که مثلاً نسخه مربوط به ۵ سال پیش را از فایل سیستم بخواهد و در آن صورت subversion مخزن را به حالت ۵ سال پیش که کاربر تقاضا کرده برای او نمایش می دهد. به بیان دیگر subversion میتواند در هر لحظه برای هر کاربر تصویر مورد نظر او از پروژه را در هر بازه زمانی که بخواهد ارائه دهد. به عنوان مثال کاربر میتواند سوالات مختلفی از تاریخچه سیستم را بپرسد، سوالاتی چون: "این شاخه که در حال حاضر در حال مشاهده آن هستیم در چهارشنبه دو هفته پیش شامل چه فایلهایی بوده" و یا "اسم آخرین کاربری که بر روی این فایل تغییرات ایجاد کرده چه بوده و چه تغییراتی را ایجاد کرده است؟"

مشکل به اشتراک گذاری فایلها

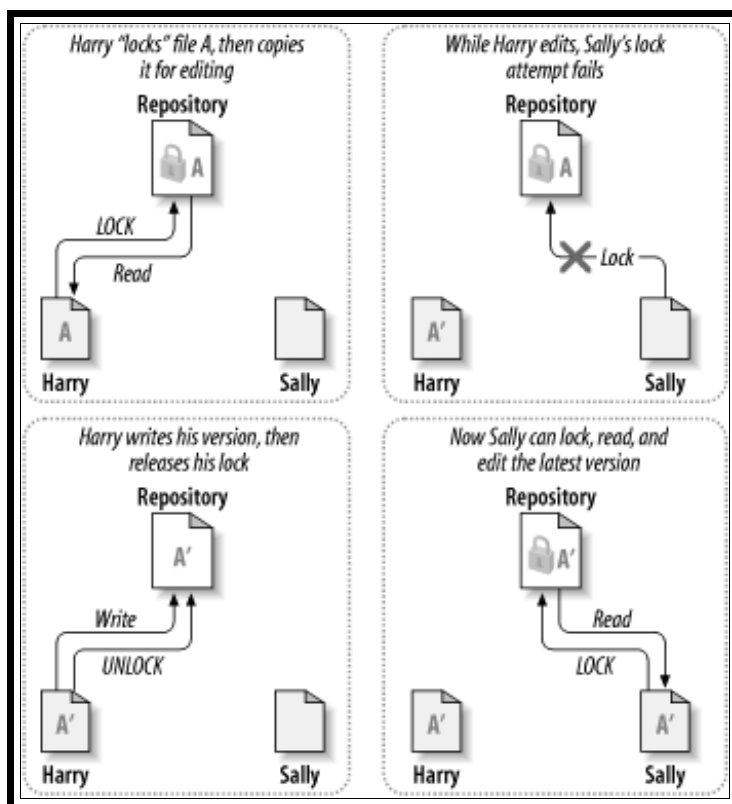
همانطور که در مقدمه ذکر شد، هنگامی که تعداد کاربران بر روی یک پروژه از یک نفر بیشتر باشد، همیشه امکان دوباره کاری و مشکلات عدم هماهنگی وجود دارد. با توجه به شکل ۲-۲ فرض کنید که هری و سکی همکار باشند. آنها هر دو در آن واحد تصمیم می گیرند که بر روی یک فایل واحد از مخزن کار کنند. اگر هری ابتدا تغییرات خود را در مخزن ذخیره کند، این امکان وجود دارد که چند لحظه بعد سکی اشتباهاً بخواهد که تغییرات خود را ذخیره کند و بدین صورت نسخه فایل سکی روی فایل هری نوشته شود. با اینکه تغییرات هری برای همیشه از بین نرفته اند (چون همانطور که ذکر شد سیستم تمام تغییرات را به خاطر می سپارد و تنها فایل سکی به عنوان جدیدترین نسخه از آن فایل در مخزن قرار گرفته است)، با اینحال تغییرات هری در فایل سکی وجود نخواهند داشت، چون سکی هیچوقت فرصت این را پیدا نکرده بود که قبل از ذخیره فایلش در مخزن از آخرین تغییرات هری باخبر شود. کار هری هم در سیستم گم شده است، حداقل اینکه دیگر در آخرین ورژن مخزن قرار ندارد. این دقیقاً شرایطی است که ما میخواهیم جلوی آنرا بگیریم!



شکل ۲ مشکل کار همزمان هری و سلی

راه حل اول: روش قفل گذاری-ایجاد تغییرات-آزاد سازی

خیلی از سیستمهای کنترل ورژن از این مدل استفاده می کنند. در این روش مخزن فایل تنها اجازه کار یک کاربر را بر روی یک فایل می دهد. به عنوان مثال اگر هری بخواهد که بر روی فایل A کار کند، ابتدا باید آنرا قفل کرده، سپس تغییرات لازم را روی آن اعمال نماید و پس از آن دوباره فایل را آزاد کند. بدین صورت اگر بعد از قفل گذاری هری و هنگامی که او در حال اعمال تغییرات روی فایل است، سلی بخواهد بر روی فایل کار کند، سیستم اجازه اینکار را به او نمی دهد. تنها کاری که او می تواند انجام دهد این است که می تواند فایل را بخواند و منتظر شود تا تغییرات هری تمام شود. وقتی که هری فایل را آزاد کند، دیگر نوبت او تمام شده است و سلی می تواند فایل را مجدد قفل کرده و شروع به کار کند که در آن صورت کاربران دیگر حق اعمال تغییرات بر روی فایل را نخواهند داشت. شکل ۲-۳ روش کار این مدل را نشان می دهد.



شکل ۳ روش قفل گذاری- اعمال تغییرات- آزاد سازی

با اینکه در نگاه اول ممکن است این روش خیلی کارآمد به نظر برسد، ولی اگر شما هم نگاهی به مشکلات آن بیندازید با من هم عقیده خواهید شد که این روش احمقانه ترین روشی است که برای کار گروهی ممکن است پیشنهاد شود!

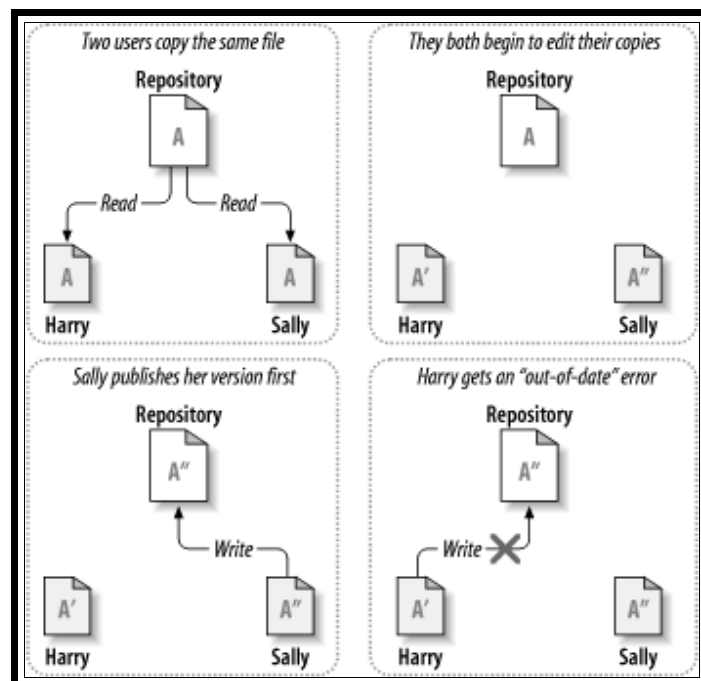
• این روش مشکلات مدیر سیستم را چند برابر میکند ممکن است بعد از اینکه هری فایل را قفل کرد و تغییرات خود

را انجام داد، به کل فراموش کند که فایل را دوباره آزاد کند. حتی ممکن است او به یک مسافرت دو هفته ای برود، در این صورت سکی بیچاره دست از پا دراز تر باید منتظر آزاد سازی قفل هری بماند. در اینحالت او به مدیر سیستم مراجعه می کند و از او می خواهد که قفل هری را آزاد کند، که این خود دردسری خواهد بود برای مدیر بیچاره سیستم، بعلاوه اینکه هیچوقت تضمین صد در صدی وجود ندارد که در هنگام آزاد سازی قفل توسط مدیر کل سیستم، هری واقعا در حال کار بر روی فایل نبوده باشد!

- قفل کردن دست و پا گیر است از کجا معلوم که هری تنها در حال تغییر بر روی خطوط اولیه فایل نباشد، در حالی که سکی فقط بخواهد خطوط انتهایی را تغییر دهد؟ در اینصورت می بینید که تغییرات این دو در واقع هیچ تاثیری روی یکدیگر نداشته و ما با اعمال سیاست قفل گذاری تنها کار را برای آنها مشکل کرده ایم.
- قفل گذاری حس امنیت غلطی را ایجاد می کند فرض کنید که هری بخواهد روی فایل A کار کند، او این فایل را قفل کرده و کار بر روی آنرا شروع می کند. حال اگر سکی بخواهد که روی فایل B کار کند او نیز این فایل را قفل کرده و کار بر روی آنرا شروع می کند. در این حالت هر دو بسیار احساس خوبی داشته و مطمئنند که فایلی که بر روی آن کار می کنند متعلق به آنهاست و هیچ بنی بشر دیگری نمی تواند تغییرات آنها را از بین ببرد و اخلالی در کار فایلیشان ایجاد کند، در صورتی که اگر فایل A به گونه ای نوشته شده باشد که برای کار نیاز به متغیرها و توابع فایل B داشته باشد، آنگاه ممکن است سکی فایل B را جوری تغییر دهد که فایل A هری بیچاره دیگر هیچگاه کار نکند! بدتر از همه اینکه هری در حین تغییر فایل A متوجه می شود که فایل کار نمی کند و این کار نکردن این فکر را در او ایجاد می کند که تغییرات او باعث از کار افتادن فایل شده است!

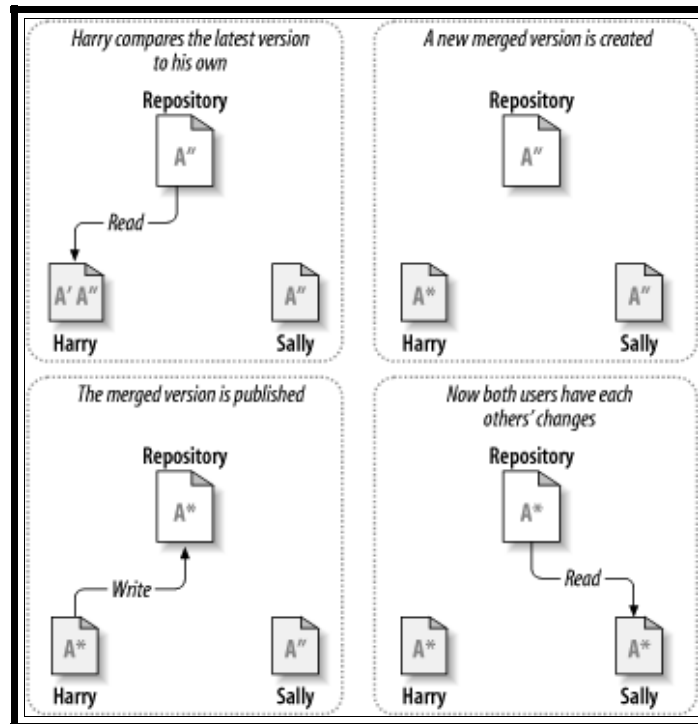
راه حل دوم: روش کپی- ایجاد تغییرات- ترکیب

subversion، CVS و بعضی دیگر از سیستمهای کنترل ورژن از این روش استفاده می کنند. در این روش هر کاربر پس از برقراری ارتباط با مخزن فایل که بر روی سرور قرار گرفته است، نسخه ای از آخرین ورژن موجود بر روی مخزن فایل را بر روی کامپیوتر خود (کلاینت) کپی می کند. به این نسخه کپی شده که عینا مشابه با آخرین نسخه فایلهای موجود بر روی مخزن فایل در لحظه کپی برداری است، نسخه کاری (Working Copy) می گویند. بدین صورت هر کاربری بر روی کامپیوتر خود نسخه ای از فایلهای موجود در مخزن فایل را ذخیره می کند و تغییرات خود را بجای اینکه مستقیما بر روی مخزن فایل موجود در سرور اعمال نماید، بر روی فایلهای موجود در کامپیوتر خود اعمال می کند. در نهایت او بعد از اینکه از تغییرات انجام شده راضی بود، با دادن دستوری به کامپیوتر خود، تمام فایلهای موجود در دستگاه را دوباره به مخزن فایل منتقل می کند و بدین صورت ورژن جدیدتری از فایلهایی که او تغییرات در آنها اعمال کرده است بر روی سرور مرکزی قرار می گیرد. اگر در هنگامی که او در حال انجام تغییرات بر روی فایلهای موجود بر روی کامپیوتر خود بوده، کاربر دیگری زودتر از او بر روی فایلهای موجود در نسخه کاری خود کار کند و آنها را در سرور قرار دهد، دیگر مخزن فایل به کاربر اول ما اجازه قرار دادن آن فایلهای را بر روی سرور نمی دهد و پیغامی مناسب مبنی بر اینکه این فایلهای در هنگامی که او غایب بوده توسط شخص/اشخاص دیگری تغییر یافته اند صادر می نماید. در این حالت کاربر ما باید تغییراتی که در هنگام غیبت او اتفاق افتاده است را مجددا از مخزن فایل دریافت کرده و آن تغییرات را با تغییرات خود ترکیب نماید و سپس حاصل کار را که مخلوطی از تغییرات خود بعلاوه تغییرات کاربران دیگر است را مجددا در مخزن فایل قرار دهد. بدین صورت دیگر کاربر ما از تغییرات انجام شده مطلع می شود و خطر اینکه تغییرات اعمال شده توسط کاربران دیگر را نبیند از بین می رود. در بیشتر موارد subversion کاربر را در عمل ترکیب یاری می کند، بدین صورت که تفاوت فایل موجود در نسخه کاری او را با آخرین نسخه از آن فایل موجود در مخزن فایل نمایش می دهد، ولی در نهایت برای یک ترکیب درست در بیشتر مواقع به هوش یک انسان نیاز است.



شکل ۴ روش کپی- ایجاد تغییرات- ترکیب، قسمت اول

پاراگراف بالا به اندازه کافی واضح بود، ولی برای کامل تر شدن مطلب مثال معروف هری و سلی خود را برای این قسمت نیز تکرار می‌کنیم! فرض کنید هری و سلی هر دو نسخه های کاری خود را با کپی کردن فایل های موجود در مخزن فایل بر روی کامپیوترهای خود ایجاد نمایند. آنها هر دو با هم بصورت موازی کار می‌کنند و تغییراتی در فایل A می‌دهند. سلی ابتدا تغییرات خود را بر روی مخزن فایل اصلی ذخیره می‌کند (فایل موجود در نسخه کاری خود را بروی سرور منتقل می‌کند). حال اگر چند لحظه بعد هری بخواهد که تغییرات خود را در مخزن فایل ذخیره نماید، پیغامی دریافت می‌کند مبنی بر اینکه فایل موجود در نسخه کاری او از آخرین باری که نسخه کاری خود را از روی مخزن فایل بروز کرده است تغییر کرده است. بنابر این هری از کامپیوتر خود می‌خواهد که آخرین تغییراتی که بر روی فایل موجود در مخزن فایل اعمال شده است را با فایل موجود در نسخه کاری او ترکیب نماید، بدین معنا که خطوط تغییر یافته و یا اضافه شده را در فایل موجود در نسخه کاری او کپی نماید. اگر این تغییرات بگونه ای باشد که خود کامپیوتر بتواند آنرا انجام دهد، مانند اینکه سلی تنها تغییراتی را در خطوط اولیه فایل داده باشد، در حالی که هری تنها خطوط انتهایی فایل را عوض کرده باشد، آنگاه خود کامپیوتر می‌تواند این ترکیب را انجام دهد، در غیر اینصورت همانطور که ذکر شد هری ناچار است که خود شخصا هر ۲ فایل را با هم مقایسه کرده و ترکیبات لازم را انجام دهد. بعد از اینکه عمل ترکیب انجام شد، هری نسخه خود را که هم اکنون جدیدترین نسخه از آن فایل حساب می‌شود (چون هم شامل تغییرات خودش می‌باشد و هم تغییرات سلی) درون مخزن فایل قرار می‌دهد و بدین صورت همه چیز ختم به خیر می‌شود!



شکل ۵ روش کپی- ایجاد تغییرات- ترکیب، قسمت دوم

ممکن است در نگاه اول این روش به نظر کمی نامنظم و سرسری برسد، و لیکن تا کنون بهترین روشی است که برای اینکار شناخته شده و دست کاربران و مدیر/مدیران سیستم را به اندازه کافی باز می‌گذارد، در عین حالی که جلوی اشتباه و تصادم (Conflict) را تا حد بسیار بالای می‌گیرد. همانطور که مشخص است اصل هر سیستم کنترل ورژنی به تعامل کاربران با یکدیگر و در تماس بودن آنها با هم دیگر پایه گذاری شده است. حال اگر این سیستم بصورت دستی و ایرانی پیاده شود، همانطور که گفته شد مشکلات عدیده خود را دارد. با پیاده سازی یک سیستم اتوماتیک و آنلاین، هدف ما تنها حذف روابط فیزیکی کاربران با یکدیگر و بوجود آوردن سیستمی قوی جهت نظم بخشیدن به این فرآیند می‌باشد؛ هیچگاه فراموش نکنید که در انتها این کاربران هستند که باید پروژه خود را به انجام برسانند و هیچ سیستم مکانیزه ای تاکنون چنین وظیفه ای را به عهده نگرفته است!

بیزن هومند