

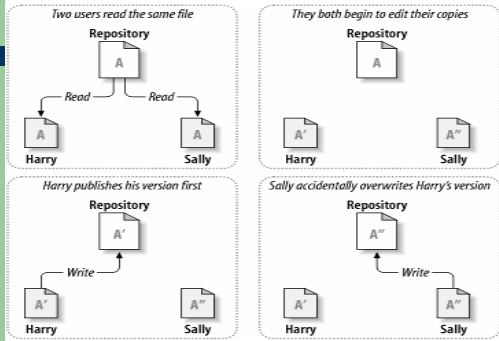
Subversion

Control de Versiones

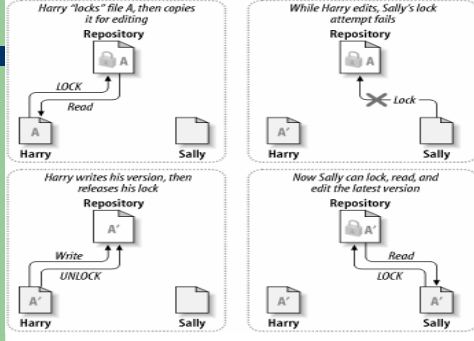
Qué es Subversion?

- Sistema de control de versiones
 - Gestiona archivos y directorios a lo largo del tiempo
 - Recuerda todos los cambios hechos en el repositorio
 - Permite que varios desarrolladores puedan trabajar sobre el mismo repositorio simultáneamente evitando conflictos

El problema a evitar



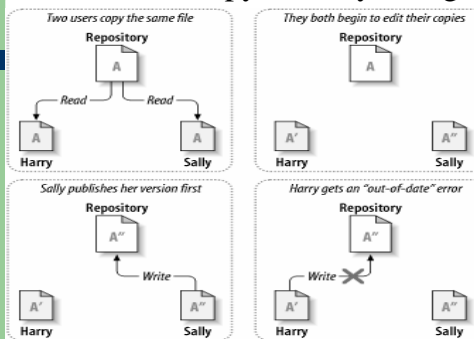
Solución 1: Lock-Modify-Unlock



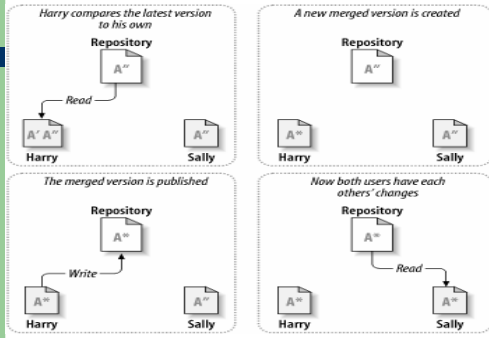
Solución 1: Lock-Modify-Unlock

- Imposibilidad de edición del mismo archivo simultáneamente
- Puede causar problemas administrativos
 - Olvido de un lock

Solución 2: Copy-Modify-Merge



Solución 2: Copy-Modify-Merge



Solución 2: Copy-Modify-Merge

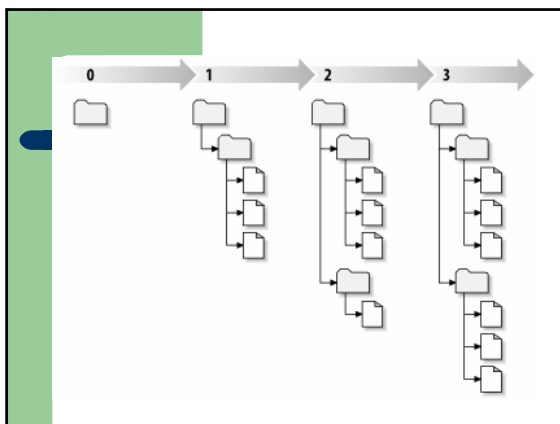
- Usuarios pueden trabajar en paralelo, sin necesidad de esperarse entre ellos
- Cuando se trabaja en los mismos archivos los conflictos son infrecuentes
- El tiempo de resolución de un conflicto es menor que el perdido por un sistema de bloqueo

Copia de Trabajo (Working Copy)

- Imagen local de los directorios y archivos del repositorio obtenida al hacer un *checkout*
- Sobre ella se trabaja localmente y luego se publican los cambios (*commit*)
- Contiene directorios administrativos (.svn) para el correcto funcionamiento del sistema

Revisiones

- Instantánea del repositorio en un determinado momento
- Se crean cada vez que se realiza un *commit*
- Se les asigna un número natural único, siempre uno mayor que la revisión anterior
- Son globales al repositorio, no a archivos individuales



Trabajando con Subversion

- Ayuda
 - `svn help <subcommand>`

Configuración cliente

- Se encuentra en el directorio `$HOME/.subversion`
 - config
 - `store-passwords = yes/no`
 - Servers
 - `http-proxy-host = proxy.itba.edu.ar`
 - `http-proxy-port = 8080`

Checkout

- Crea una copia de la última revisión (*HEAD*) del repositorio en la máquina local
- `svn checkout URL [PATH]`
 - `svn checkout https://pampero.it.itba.edu.ar/svn/prot_grupo0`

Checkout

Schema	Método de acceso
<code>file:///</code>	Acceso directo (en el disco local)
<code>http://</code>	Acceso via protocolo WebDAV a servidor Apache con soporte SVN
<code>https://</code>	Idem <code>http://</code> , pero con encriptación SSL
<code>svn://</code>	Acceso via protocolo propietario a un servidor <code>svnserve</code>
<code>svn+ssh://</code>	Idem <code>svn://</code> , pero a través de un túnel <code>ssh</code>

Ciclo básico de trabajo

- Actualizar la copia local
 - `svn update`
- Hacer cambios
 - `svn add`
 - `svn delete`
 - `svn copy`
 - `svn move`
 - `svn mkdir`

Ciclo básico de trabajo

- Examinar los cambios
 - `svn status`
 - `svn diff`
 - `svn revert`
- Resolver conflictos
 - `svn resolved`
- Publicar los cambios
 - `svn commit`

Update

- Sincroniza la copia local con la última revisión del repositorio
- Códigos de salida:

U foo	Archivo foo actualizado
A foo	Archivo o directorio foo agregado a copia local
D foo	Archivo o directorio foo borrado de copia local
G foo	Archivo foo "mergeado" satisfactoriamente
C foo	Archivo foo en conflicto

Hacer cambios a la copia local

- Cambios a archivos
 - Se hacen los cambios directamente
 - No es necesario informar a Subversion
 - Los cambios se detectan automáticamente
 - Ej: edición de un archivo

Hacer cambios a la copia local

- Cambios en el árbol de directorios
 - Son inmediatos en la copia local
 - Se efectivizan en el repositorio luego de un *commit*

Cambios en el árbol de directorios

- `svn add PATH`
 - Pone archivos o directorios bajo control de versiones, agendando su adición al repositorio
- `svn delete PATH`
 - Borra archivos o directorios del control de versiones
- `svn copy SRC DST`
 - Duplica un archivo o directorio en la copia local o el repositorio, recordando la historia
- `svn move SRC DST`
 - Mueve y/o renombra un archivo o directorio en la copia local o repositorio, recordando la historia

Examinar los cambios

- `svn status [PATH]`
 - Informa el estado de los archivos y directorios de la copia local
 - Opción `-u` muestra archivos o directorios desactualizados

A item	Item agendado para adición
C item	Item en conflicto
D item	Item agendado para borrado
M item	Contenido de item modificado
? Item	Item no está bajo control de versiones

Examinar los cambios

- `svn diff`
 - Muestra los cambios producidos en los archivos
 - La salida se muestra en *unified diff format*. Util para crear parches (patches).
- `svn revert PATH [--recursive]`
 - Deshace las modificaciones locales de un archivo
 - También puede deshacer operaciones agendadas como add y delete.

Resolver conflictos

- Al aparecer un conflicto
 - Se imprime una **C** durante el *update* indicando el conflicto
 - Si es un archivo al que se le puede hacer un merge (ej. archivo de texto) coloca marcas de conflicto (*conflict markers*)

Resolver conflictos (cont.)

- Para cada archivo en conflicto se generan hasta tres archivos sin versionar en la copia local:
 - filename.mine
 - Archivo en conflicto tal cual se encontraba antes de hacer el update (sin las marcas de conflicto)
 - filename.OLDREV
 - Archivo que fue la revisión BASE antes de empezar a editarlo
 - filename.NEWREV
 - Archivo que corresponde a la revisión HEAD del repositorio
- OLDREV es el número de revisión del archivo en el directorio .svn y NEWREV el número de la revisión HEAD.

Resolver conflictos (cont.)

- Para resolver el conflicto se debe hacer algo de lo siguiente
 - Mergear el archivo en conflicto "a mano" quitando las marcas de conflicto
 - Copiar uno de los archivos temporales generados sobre el archivo original de la copia local
 - Ejecutar `svn revert <archivo>` para descartar los cambios locales
- Una vez hecho esto hay que ejecutar
 - `svn resolved <archivo>` para marcar el conflicto como resuelto (borra además los 3 archivos generados).

Publicar los cambios

- `svn commit`
 - Envía los cambios de la copia local al repositorio
 - Los commits son atómicos: si falla en alguna etapa se deshace todo el commit
 - Se debe ingresar un mensaje de log para documentar el commit
 - Setear previamente variable de entorno `SVN_EDITOR` o usar opción `-m` para pasar el mensaje.
 - Al finalizar informa el nuevo número de revisión

Examinando la historia

- `svn log [-r startrev[:endrev]] [-v]`
 - Muestra información acerca de la historia de un archivo o directorio
- `svn diff`
 - Muestra diferencias entre archivos
 - Usos:
 - `svn diff`
 - Muestra cambios locales
 - `svn diff -r rev`
 - Se compara la copia local con la revisión especificada del repositorio
 - `svn diff -r rev1:rev2`
 - Compara dos revisiones del repositorio

Examinando la historia

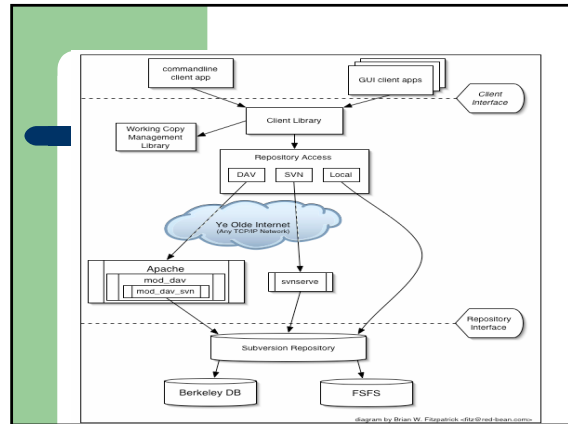
- `svn cat [-r rev] URL`
 - Muestra el contenido de un archivo del repositorio en la salida estándar
- `svn list URL`
 - Muestra el contenido del directorio de un repositorio sin necesidad de bajar los archivos a la máquina local
- `svn blame URL`
 - Muestra el contenido de un archivo mostrando quién modificó cada línea y en qué revisión

Export

- Crea una copia de la última revisión (*HEAD*) del repositorio en la máquina local sin información de versionado (sin directorios .svn)
- Util para generar una release del proyecto
- `svn export URL [PATH]`
 - `svn export`
`https://pampero.it.itba.edu.ar/svn/prot_grupo0`

Administración

- Creación de repositorios
 - `svnadmin create [-fs-type fsfs] /path/to/repos`



Configuración del servidor (svnserve)

- Configuración del repositorio
 - `/path/to/repos/conf`
 - `svnserve.conf`
 - [general]
 - `password-db = passwd`
 - `realm = example realm`
 - `passwd`
 - [users]
 - `user = password`

Configuración del servidor (svnserve)

- Agregar en `/etc/services`
 - `svn 3690/tcp # Subversion`
 - `svn 3690/udp # Subversion`
- Si se usa `inetd`: agregar en `/etc/inetd.conf`
 - `svn stream tcp nowait svnowner /usr/bin/svnserve`
 - `svnserve -i -r /path/to/repos`

Configuración del servidor (svnserve)

- Si se usa `xinetd`, agregar el archivo `svnserve` en `/etc/xinetd.d`:
 - `service svn`
 - {
 - `flags = REUSE`
 - `socket_type = stream`
 - `wait = no`
 - `user = svnowner`
 - `server = /usr/bin/svnserve`
 - `server_args = -i -r /path/to/repos`
 - `log_on_failure += USERID`
 - `disable = no`
 - }

Configuración del servidor (svnserve)

- En ambos casos (`inetd` y `xinetd`) el usuario `svnowner` debe tener los permisos necesarios para leer y escribir en el path al repositorio