Subversion

Nikola Brežnjak
OSL

ToC

- Intro
- Basic commands
- Basic concept of version control programms
- Commands in every day usage

Sadržaj

- **Intro**
- Basic commands
- Basic concept of version control programms
- Commands in every day usage

Where to use Subversion?

- Problems:
  - During the development of some program you make a few versions of the same file, and you yourself don't know any more what is realy changed in which version
  - You want to see what changes you made in the current version opposite to lets say for example version 3.
  - You work in a group on some project and someone erases your changes by mistake
- Subversion i s the tool that solves all this problems and more...

What is Subversion?

- Free open source program for version control and file exchanging
- It works with files and folder as well
- File directory tree is put in so called repository, which is very similar to ftp
- Allowed access to repository through network, which is good if more people work on the same project
- Mostly it's used for managing text file documents

History

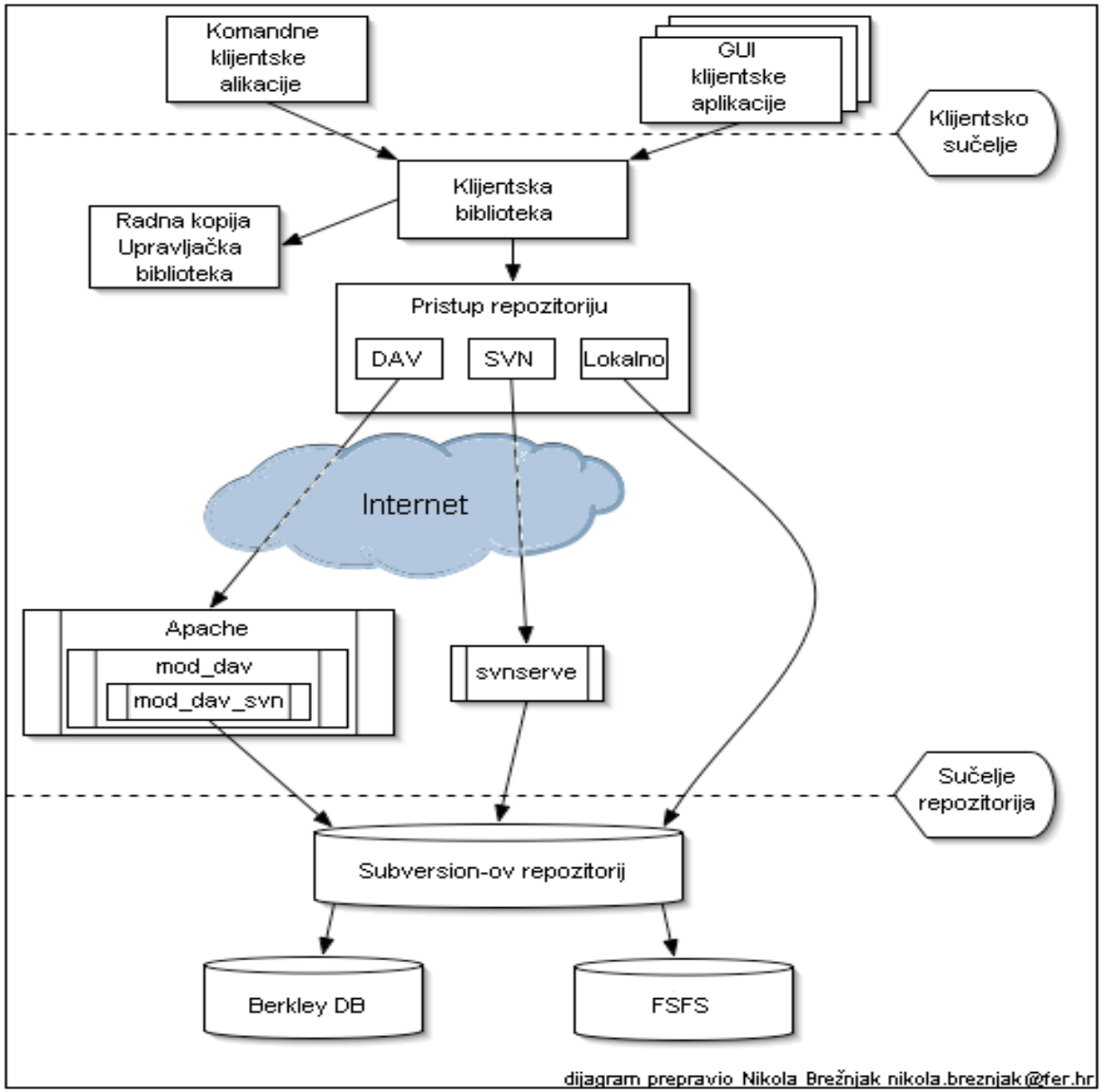
- Developed by CollabNet
- http://www.collab.net/
- Substitute for CVS (Concurrent Versions Systems)

Differences between SVN and CVS

Subversion's features :

- Version control of files and directories, while CVS just directories
- Adding, deleting , copying and renaming files and directories, while CVS has bad support for this
- Network repositry access
- Attomic comits (all or nothing goes in the repository), and CVS hasn't got this
- Consitensy while showing differences between files using binary algorithm

Arhitektura Subversion-a



- Repozitorij - smještene sve verzionirane datoteke
- Klijentski program - lokalno upravljanje tzv. radnim kopijama
- Pristup repozitoriju, direktan ili preko mreže

Komponente Subversion-a

Naziv komponente	Opis rada komponente
svn	komandni klijentski program
svnversion	program za ispis stanja radne kopije
svnlook	alat za pregledavanje repozitorija
svnadmin	alat za stvaranje, podešavanje ili popravljanje repozitorija
svndumpfilter	program za filtriranje repozitorija
mod_dav_svn	modul za Apache HTTP server
svnserve	samostalan server, za pristup repozitoriju preko mreže

Ponovimo...

- Što je i za što se koristi Subversion
- Razlike Subversion-a i CVS-a
- Arhitektura Subversiona - repozitorij, klijentski program i dio za pristup repozitoriju
- Komponente - svn i svnadmin

Sadržaj

- Uvod
- **Osnovne naredbe**
- Osnovni koncepti programa za kontrolu verzije
- Naredbe

Stvaranje repozitorija (1)

- Provjera verzije Subversion-a:

```
$ svn --version
svn, version 1.3.2 (r19776)
   compiled Jul 13 2006, 04:22:38
Copyright (C) 2000-2006 CollabNet.
Subversion is open source software, see http://subversion.tigris.org/
This product includes software developed by CollabNet (http://www.Collab.Net/).
```

- Stvaranje repozitorija:

```
$ svnadmin create /put/do/repozitorija

$ ls /put/do/repozitorija
conf/  dav/  db/  format  hooks/  locks/  README.txt
```

Stvaranje repozitorija (2)

- Repozitorij upravlja datotekama i direktorijima pa je na korisniku da interpretira pojedine direktorije kao projekte

- Zadaci:
  - Provjeriti instaliranu verziju Subversion-a
  - Napraviti repozitorij naziva repos u radnom direktoriju /home/korisnik/repos, gdje je korisnik vaše korisničko ime
  - Izlistati sadržaj novostvorenog repozitorija

Uvođenje projekta u repozitorij (1)

- Treba imati već neki projekt
- Poddirektoriji branches, tags i trunk nisu nužni, ali se često koriste u praksi
- Za uvođenje projekta u repozitorij treba ukucati:

```
$ svn import /home/hitman/projekt1
file:///put/do/repozitorija/projekt1 -m "inicijalno umetanje"

Adding       /tmp/myproject/branches
Adding       /tmp/myproject/tags
Adding       /tmp/myproject/trunk
Adding       /tmp/myproject/trunk/hello.c
Adding       /tmp/myproject/trunk/proba.c
…
Committed revision 1.
```

Uvođenje projekta u repozitorij (2)

- Datoteke unutar repozitorija nije moguće izlistati naredbom ls
- Direktorij /home/hitman/projekt1 se može obrisati.
- Zadaci:
  - Napraviti direktorij proba sa poddirektorijima tags, branches i trunk.
  - U direktoriju trunk napraviti datoteku hello.c
  - Uvesti direktorij proba u repozitorij repos pod imenom projekt1

Preuzimanje projekta iz repozitorija (1)

- Za mijenjanje podataka iz repozitorija treba stvoriti radnu kopiju podataka iz repozitorija
- Radna kopija
  - stablo direktorija, sadrži datoteke
  - privatna, promjene dostupne drugima tek nakon stavljanja u repozitorij
  - moguće imati veći broj radnih kopija

Preuzimanje projekta iz repozitorija (2)

- Za preuzimanje projekta iz repozitorija ukucati:

```
$svn checkout file:///put/do/repozitorija/projekt1/trunk radna_kopija
A  projekt/hello.c
A  projekt/proba.c
…
Checked out revision 1.
```

- Nastaje radna kopija direktorija trunk i sprema se u drektorij radna_kopija
- Slovo A označava koje se datoteke dodaju u radnu kopiju

Sadržaj direktorija radne kopije

```
$ ls -all -C radna_kopija
.  ..  hello.c  opcenito.c  .svn/
```

- **.svn** - skriveni direktorij u kojem su zapisani:
  - podaci o datotekama koje sadrže neobjavljene (*eng. unpublished*) *promjene ili*
  - *zastarjele* (eng. out of date) promjene u odnosu na one u repozitoriju
- Zadaci:
  - Preuzeti projekt proba sa repozitorija i spremiti ga u direktorij proba_radna_kopija
  - Ispisati sadržaj direktorija trunk u direktoriju proba_radna_kopija

Ponovimo...

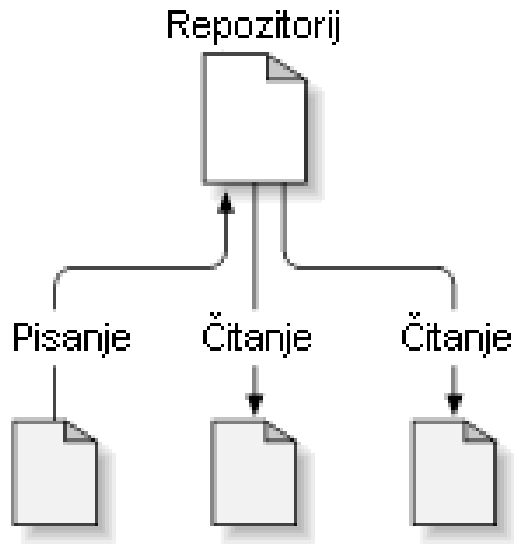
- Stvaranje repozitorija
  - svnadmin create /put
- Uvođenje projekta u repozitorij
  - svn import /home/hitman/projekt1 file:///put/projekt1 -m "inicijalno umetanje"
- Preuzimanje projekta iz repozitorija
  - svn checkout file:///put/projekt1/trunk radna_kopija

Sadržaj

- Uvod
- Osnovne naredbe
- **Osnovni koncepti programa za kontrolu verzije**
- Naredbe

Repozitorij

- Jezgra Subversion-a
- Sadrži sve podatke
- Klijenti se spajaju na repozitorij i onda pišu ili čitaju podatke sa njega
- Sličan datotečnom poslužitelju
- Klijent ima mogućnost uvida u ranije verzije datotečnog sistema

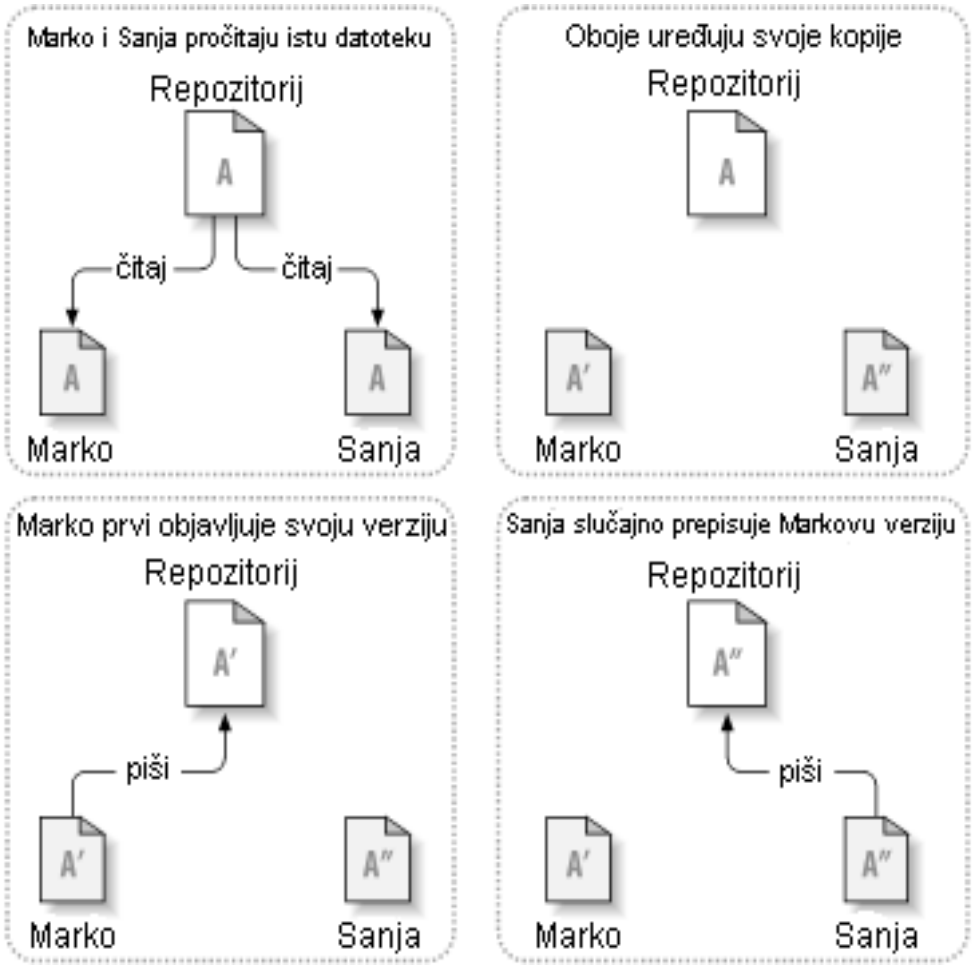


Modeli kontrole verzija

- Svi programi za kontrolu verzije moraju omogućiti korisnicima da dijele informacije
- Ali isto tako moraju spriječiti slučajne izmjene tuđih informacija
- Dakle, potrebno je onemogućiti situaciju u kojoj netko stavi svoju verziju datoteke u repozitorij, a da je onda netko drugi kasnije prebriše sa svojom verzijom datoteke

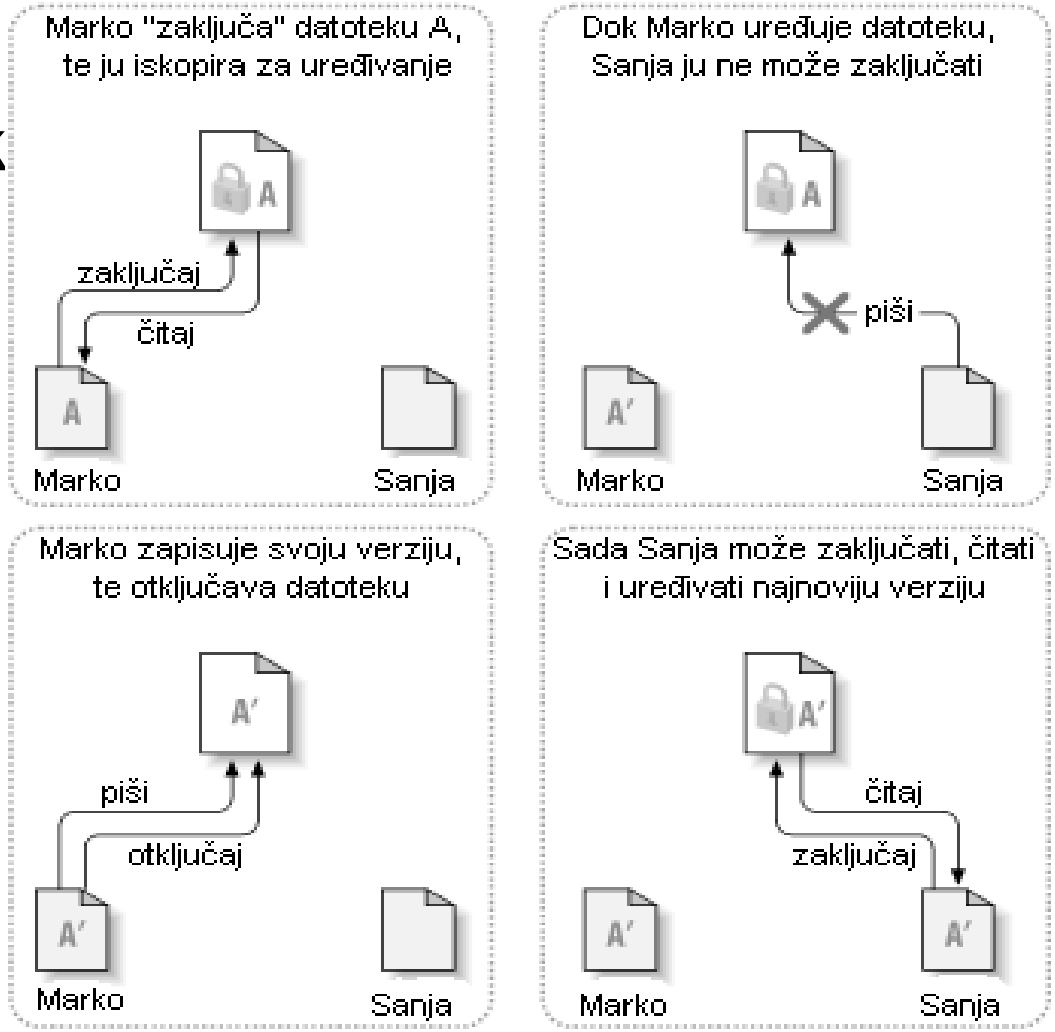
Problem koji treba izbjeći

- Markova verzija datoteke neće biti izgubljena, jer sistem pamti sve promjene
- Markove promjene izostaju iz najnovije verzije datoteke - greška!



Zaključaj-modificiraj-otključaj (1)

- Samo jedna osoba može mijenjati određenu datotek
- Marko mora “zaključati” datoteku prije nego što je može mijenjati
- Sanja može datoteku samo čitati i čekati da je Marko otključa



Problemi modela zaključaj-modificiraj-otključaj

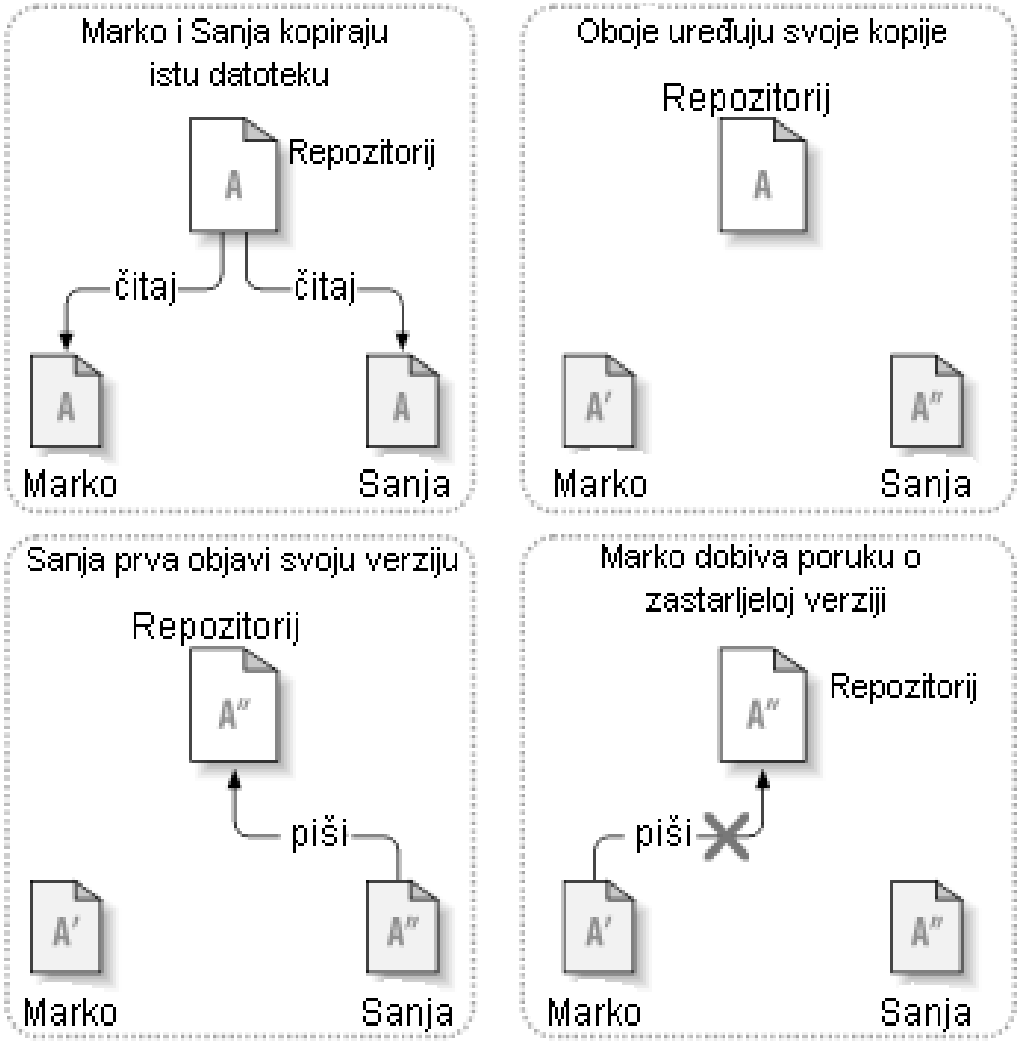
- Marko može zaboraviti otključati datoteku
- Nemogućnost istovremenog obrađivanja datoteke u slučaju da se promjene ne preklapaju
- U slučaju da Marko i Sanja uređuju dvije različite datoteke A i B koje su međusobno povezane, mogućnost je da one nakon mijenjanja više neće moći raditi zajedno.
- Ipak, model zaključavanja se koristi kod binarnih datoteka (glazba, grafika, itd.)

Kopiraj-modificiraj-spoji

- Svaki klijent dobiva svoju radnu kopiju datoteka i direktorija iz repozitorija
- Moguć istovremeni rad na vlastitim kopijama
- Privatne kopije se spajaju (*eng. merge*) u novu konačnu verziju (*eng. final version*)
- Sistemi za kontrolu verzija pomažu kod spajanja, ali je korisnik odgovoran da se to ispravno obavi
- Subversion koristi ovaj model rješenja

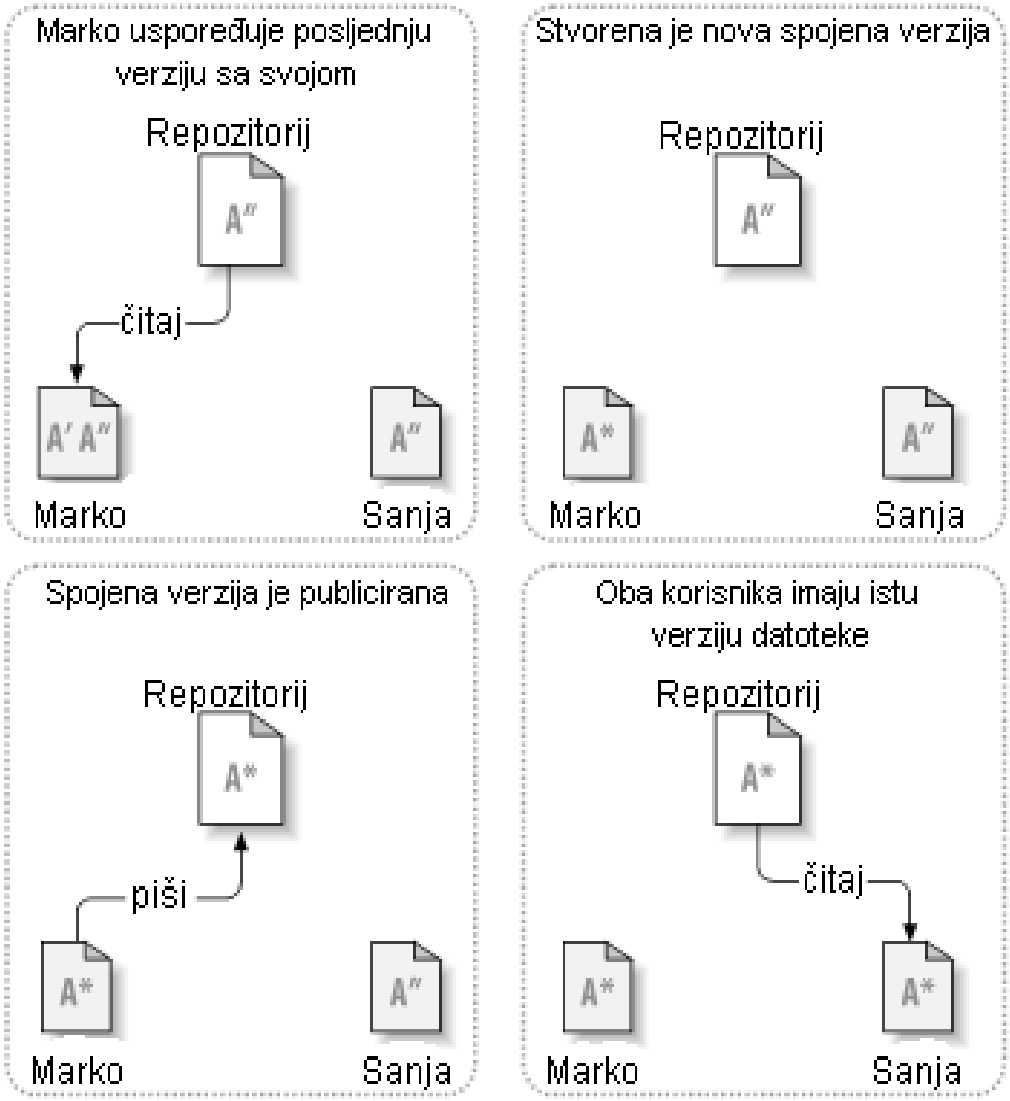
Poruka o zastarjeloj verziji

- Markov pokušaj zapisivanja u repozitorij se odbija, uz poruku o zastarjeloj verziji (*eng. out of date*)



Spajanje promjena

- Nakon poruke o zastarjeloj verziji Marko traži od svoj klijenta da spoji promjene
- Ako se Sanjine promjene ne preklapaju sa Markovim, promjene se spajaju u jednu datoteku
- Marko može tu spojenu datoteku spremiti natrag u repozitorij



Konflikt

- Konflikt - Sanjine promjene se preklapaju sa Markovim
- Markova datoteka je označena i on može vidjeti promjene koje su uzrokovale konflikt
- Subversion automatski otkriva stanje konflikta
- Konflikt mora rješiti korisnik sam
- U praksi se konflikti rijetko događaju

Ponovimo...

- Repozitorij - mjesto gdje se spremaju svi podaci
- Problem koji treba izbjeći - ne prebrisati tuđe promjene
- Zaključaj-modificiraj-otključaj - datoteke se “zaključavaju”
- Kopiraj-modificiraj-spoji -svatko dobije svoju kopiju datoteke

Pristup repozitoriju

Shema	Pristupna metoda
file:///	direktan pristup repozitoriju (na lokalnom disku)
http://	pristup preko WebDAV protokola na Apache poslužitelj
https://	isto kao i http://, ali sa SSL enkripcijom (eng. encryption)
svn://	pristup preko posebnog protokola na svnserve poslužitelj
svn+ssh://	isto kao i svn://, ali preko SSH tunela

- Zadatak: Preuzeti neki projekt sa https://svn.sourceforge/svnroot/ikev2 i pogledati sadržaj

Pisanje u repozitorij (1)

- Evidencija promjene sadržaja neke datoteku u .svn direktoriju
- Objava (eng. publish) promjena samo na eksplicitni zahtjev
- Pretpostavimo da je sadržaj datoteke hello.c u radnoj kopiji promijenjen
- Za objavljivanje promjena u repozitoriju ukucati:

```
$ svn commit hello.c -m "dodana linija printf"
Sending        hello.c
Transmitting file data .
Committed revision 2.
```

Pisanje u repozitorij (2)

- Prekidač `-m` (eng. switch) - dodavanje poruke u zapisnik
- Ukoliko se poruka ne zada Subversion automatski pokreće podrazumijevani program za uređivanje teksta.
- Za ručno pokretanje uređivača teksta potrebno je ukucati:

```
$ svn commit --editor-cmd ARG
```

- ARG se zamijenjuje nazivom nekog vanjskog uređivača, npr. vi.

Ažuriranje radne kopije (1)

- Ako Marko objavi svoje promjene na datoteci hello.c u repozitoriju, Sanja mora ažurirati (*eng. update*) svoju radnu kopiju (npr. promijenila se datoteka hello.c)
- Da bi se Sanjina radna kopija ažurirala sa promjenama iz repozitorija, treba ukucati:

```
$ pwd
/home/Sanja/radna_kopija/trunk

$ ls -AC
hello.c  opcenito.c  .svn

$ svn update
U    hello.c
Updated to revision 2.
```

Ažuriranje radne kopije (2)

- U prošlom primjeru se ažurirala datoteka hello.c
- Slovo U sa lijeve strane svake datoteke koja se ažurirala nakon naredba svn update
- Potrebno je napomenuti da Sanja nije trebala specificirati koje datoteke želi ažurirati, već je to Subversion učinio umjesto nje koristeći podatke u .svn direktoriju.
- Zadatak: Promijeniti sadržaj datoteke hello.c, te potom objaviti promjene u repozitoriju

Revizije (1)

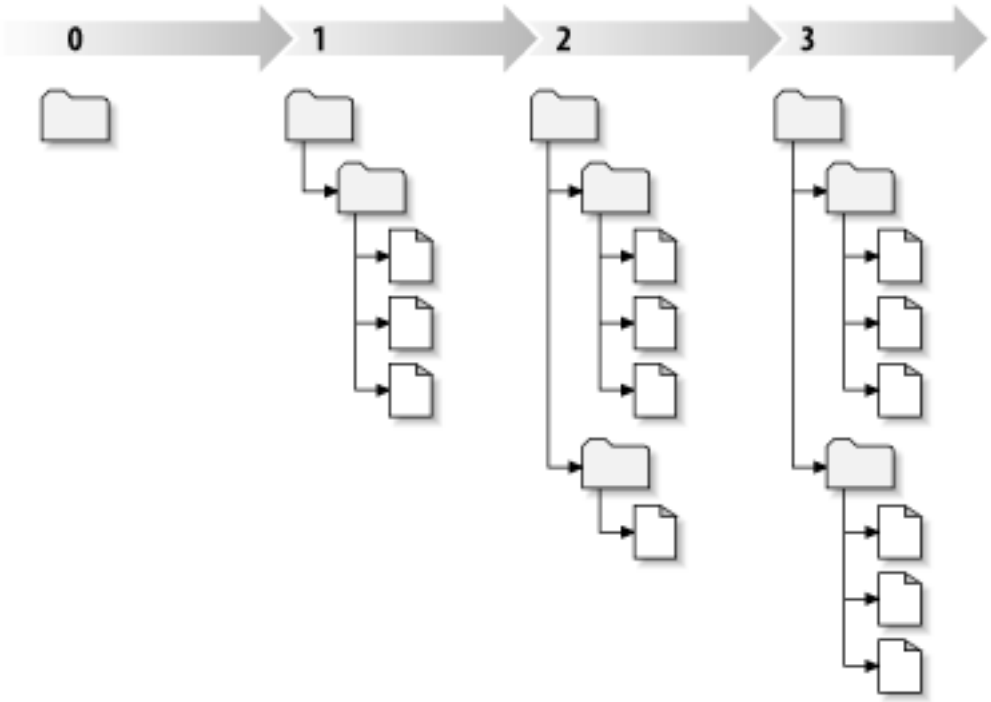
- Svaki puta kada repozitorij prihvati upis, stvara se novo stanje sistemskog datotečnog stabla (*eng. filesystem tree*) koje se naziva revizija (*eng. revision*).
- Svakoj reviziji je pridružen jedinstven prirodan broj, za jedan veći od prethodne revizije.
- Inicijalna revizija - broj 0
- Subversion primjenjuje (eng. apply) revizijske brojeve na cijelo stablo, a ne samo na pojedinačne datoteke

Revizije (2)

- N-ta revizija predstavlja stanje datotečnog sistema repozitorija nakon N-tog uspješnog pisanja u repozitorij
- Naredbe za objavljivanje u repozitoriju (svn commit) i ažuriranje radne kopije (svn update) ne povlače jedna drugu
- Radna kopija može sadržavati datoteke i direktorije koji imaju različite brojeve radnih revizija
- Moguće imati datoteku čija je radna revizija npr. N (N > M), dok je revizija radnog direktorija, u kojem se ta datoteka nalazi, M.

Način vizualizacije repozitorija

- Samo pokretanjem naredbe za ažuriranje radne kopije će se cijela radna kopija dovesti u stanje M-te revizije
- Način vizualizacije repozitorija je prikazan na slici



Povezivanje repozitorija i radne kopije

- Dvije informacije u .svn direktoriju za svaku datoteku:
  - Revizija na kojoj se radna datoteka zasniva, tzv. radna revizija (*eng. working revision*)
  - Vrijeme zadnjeg ažuriranja lokalne kopije
- 4 stanja radne datoteke:
  - Nepromijenjena i tekuća
  - Lokalno promijenjena i tekuća
  - Nepromijenjena i zastarjela
  - Lokalno promijenjena i zastarjela

Ponovimo...

- Pristup repozitoriju - file:// i http://
- Pisanje u repozitorij
  - svn commit hello.c -m ”dodana linija printf”
- Ažuriranje radne kopije
  - svn update
- Revizija - broj koji predstavlja n-to stanje datotečnog sistema nakon n-tog uspješnog upisa u repozitorij

Sadržaj

- Uvod
- Osnovne naredbe
- Osnovni koncepti programa za kontrolu verzije
- **Naredbe**

Pomoć

- Popis svih naredbi Subversion-a

```
$ svn help
```

- Sintaksa pojedine naredbe Subversion-a:

```
$ svn help <pod_naredba>
```

- Ova naredba ispisuje sintaksu i ponašanje naredbe pod_naredba
- Zadaci:
  - Provjeriti ispis naredbe svn help
  - Ispisati sintaksu i ponašanje naredbe svn commit

Rad sa revizijama (1)

- Prekidač –-revision (kraće -r) – za odabir revizije
- Nakon prekidača -r slijedi cijeli broj ili neka od ugrađenih ključnih riječi:
  - HEAD - Najnovija revizija u repozitoriju
  - BASE - Broj revizije u radnoj kopiji bez lokalnih promjena
  - COMMITED - Revizija jednaka BASE, u kojoj postoje promjene
  - PREV - Broj revizije COMMITED -1
- Ove ključne riječi rade samo ako se repozitorij nalazi lokalno na računalu, ali ne i preko URL-a

Rad sa revizijama (2)

- Primjer:

```
$ svn checkout --revision 1 file:///put/do/repozitorija/projekt1 rev1
```

```
$ svn diff --revision PREV:COMMITTED proba.c
# prikazuje posljedne promjene objavljene u datoteci proba.c

$ svn log --revision HEAD
# prikazuje zapisnik o zadnjem objavljivanju u repozitorij
```

- Postoji i mogućnost specificiranja datuma:

```
$ svn checkout --revision {2002-02-17}
$ svn checkout --revision {"2002-02-17 15:30"}
```

- Datum zadan bez vremena – u biti (dan -1)

Rad sa revizijama (3)

Traženje u intervalu datuma:

```
$ svn log --revision {12-10-2006}:{15-10-2006}
```

- Ova naredba ispisuje zapisnik (eng. log) za sve revizije između 12.10 i 15.10 2006 godine
- Moguće miješati datume i brojeve revizija:

```
$ svn log --revision {12-10-2006}:13
```

- Zadaci:
  - Ispisati zapisnik o trećoj, četvrtoj i petoj reviziji direktorija trunk
  - Ispisati zapisnik o reviziji tekućeg datuma

Ažuriranje radne kopije projekta

- Za ažuriranje radne kopije sa repozitorijem:

```
$ svn update
U    proba.txt
Updated to revision 7.
```

- Popis i značenje slova u ispisu naredbe svn update:

Slovo	Značenje slova
U *a*	*a* uspješno ažuriran
A *a*	*a* je bio dodan radnoj kopiji
D *a*	*a* je bio izbrisan iz radne kopije
R *a*	*a* je bio zamijenjen
G *a*	*a* sadržavao promjene, ali su uspješno spojene
C *a*	konflikt u datoteci *a* kojeg korisnik treba riješiti

- *a* može biti direktorij, datoteka ili link

Mijenjanje radne kopije projekta

- Promjene radne kopije se dijele:
  - Datotečne promjene – Subversion ih automatski detektira
  - Promjene stabla – treba obavijestiti Subversion o promjeni
- Naredbe za mijenjanje radne kopije:
  - svn add
  - svn delete
  - svn copy
  - svn move

Dodavanje u repozitorij

```
$ svn add abc
```

- Dodaje datoteku direktorij ili simbolički link *abc* u repozitorij
- Napomena: abc mora postojati prije te naredbe, inače Subversion javlja grešku
- Ako je abc direktorij repozitoriju će se predati i svi njegovi poddirektoriji
- Za dodavanje direktorija bez njegovih poddirektorija potrebno je ukucati:

```
$ svn add abc --non-recursive
```


- Kraći zapis prekidača --non-recursive je -N

Brisanje iz repozitorija

```
$ svn delete abc
```

- Ovom se naredbom datoteka, direktorij ili simbolički link abc stavlja u listu čekanja za brisanje iz repozitorija
- Ukoliko je abc link onda je odmah obrisan iz radne kopije, a ako pak je direktorij, nije izbrisan već je stavljen u red čekanja za brisanje
- Nakon što korisnik objavi svoje promjene u repozitoriju, abc se briše iz repozitorija i radne kopije

Kopiranje u repozitoriju

```
$ svn copy abc def
```

- Nastaje kopija abc.
- Predmet (eng. item) def se automatski dodaje u red čekanja za dodavanje koji se nakon sljedećeg objavljivanja u repozitoriju (naredbom svn commit) dodaje u repozitorij iz liste čekanja

Premještanje u repozitoriju

```
$ svn move abc def
```

- Ova naredba ima isti učinak kao i naredbe svn copy abc def i svn delete abc pokrenute jedna za drugom, tj. def se stavlja u red čekanja za dodavanje, a abc u red čekanja za brisanje.
- Zadaci:
  - Dodati direktorij proba u repozitorij, i objaviti promjene
  - Iskopirati direktorij proba u direktorij proba2
  - Izbrisati direktorij proba iz repozitorija
  - Premjestiti direktorij proba2 u direktorij proba3

Ponovimo...

- Pomoć
  - svn help
- Revizije
  - prekidač --revision
- Ažuriranje radne kopije
  - svn update
- Mijenjanje radne kopije
  - svn add - dodavanje u repozitorij
  - svn delete - brisanje iz repozitorija
  - svn copy - kopiranje u repozitoriju
  - svn move - premještanje u repozitoriju

Ispitivanje promjena

- Naredbe za ispitivanje promjena su:
  - svn status
  - svn diff
  - svn revert
- Neovisnosne o sadržaju repozitorija, tj. naredbe ne moraju komunicirati sa repozitorijem da bi izvršile zadatak
- Brz i efikasan rad preko mreže, jer se sa poslužiteljem na kojem je repozitorij komunicira samo pri predaji i uzimanju podataka

Provjera promjena u radnoj kopiji

```
$ svn status
```

- Ispisuje promjene koje je korisnik napravio
- Popis i značenje znakova prve kolone ispisa

Slovo	Značenje slova
A dat	dat je stavljen u listu čekanja za dodavanje na repozitorij
R dat	dat je stavljen u listu čekanja za zamjenu
? dat	dat nije pod kontrolom Subversion-a
! dat	dat je pod kontrolom Subversiona, ali je nepotpun
I dat	dat nije pod kontrolom Subversion-a i on ga ignorira
D dat	dat je u stavljen u listu čekanja za brisanje
C dat	dat sadrži tekstualan konflikt

- Prihvača prekidač --verbose (kraće -v)

Ispis neažuriranih objekata

```
$ svn status --show-updates
```

- Ova naredba kontaktira repozitorij i ispisuje * pored objekata (*eng. item*) koji nisu ažurirani
- Kraći zapis prekidača --show-updates je -u
- Zadatak:
  - Ispisati promjene u radnoj kopiji
  - Ispisati promjene u radnoj kopiji kontaktirajući repozitorij

Ispis promjena unutar datoteke

```
$ svn diff
Index: hello.c
===================================================================
- - - hello.c     (revision 10)
+++ hello.c     (working copy)
@@ -2,6 +2,7 @@

 int main()
 {
-       printf("Hello!");
+       printf("Hello!\n");
+       printf("Dodao sam newline.\n");
        return 0;
 }
```

- Ispis promjena u unificiranom diff formatu
- Naredba svn diff kontaktira repozitorij

Vraćanje u prijašnje stanje

```
$ touch a.txt
$ svn status a.txt
?      a.txt

$ svn add a.txt
A         a.txt

$ svn revert a.txt
Reverted 'a.txt'

$ svn status a.txt
?      a.txt
```

- Vraćanje u stanje sačuvano u .svn direktoriju
- Zadaci:
  - Napraviti promjene na datoteci hello.c
  - Ispisati promjene nastale u datoteci hello.c
  - Vratiti datoteku hello.c u prijašnje stanje

Spajanje vlastitih promjena s promjenama drugih korisnika

- Poruka o staroj verziji datoteke

```
$ svn commit hello.c
svn: Commit failed (details follow):
svn: Out of date: '/svn/trunk/hello.c' in transaction '14-1'
```

- Ukoliko se Sanjine promjene ne preklapaju sa Markovim promjenama Subversion će naredbom svn update uspješno spojiti promjene (slovo G od *eng. merGe*)

Pojava konflikta

- Kod pojave konflikta Subversion pomaže na ovaj način:
  - Prilikom ažuriranja ispisuje slovo C, te pamti da je datoteka u stanju konflikta
  - Stavlja konfliktne markere (*eng. conflict markers*) u datoteku, koji pokazuju na mjesto konflikta u datoteci.
  - Konfliktni markeri se sastoje od znakova <, = i >

Privremene datoteke kod pojave konflikta

- Za svaku datoteku koja je u stanju konflikta, Subversion dodaje tri datoteke u radnu kopiju koje nisu pod kontrolom Subversion-a:
  - ime_datoteke.mine - datoteka identična onoj iz radne kopije prije nego se pokrenula naredba svn update
  - ime_datoteke.rStaraRev - datoteka koja je uzeta sa repozitorija prije ažuriranja radne kopije
  - ime_datoteke.rNovaRev - datoteka koju je Subversion skinuo sa repozitorija nakon ažuriranja radne kopije. To je u biti najnovija revizija repozitorija

```
$ ls
 hello.c  hello.c.mine  hello.c.r12  hello.c.r15
```

Rješavanje konflikta

- Konflikt se može riješiti na jedan od tri načina:
  - Ručno rješavanje konflikata, uz pomoć konfliktnih markera
  - Kopiranjem jedne od trenutnih datoteka na mjesto datoteke koja je u konfliktu
  - Korištenjem naredbe svn revert, tj. poništavanjem lokalnih promjena

Dojava o rješenju konflikta

```
$ svn resolved hello.c
Resolved conflicted state of 'hello.c'
$ ls
 hello.c
```

- Brišu se privremene datoteke
- Primjer datoteke u konfliktu:

```
Ovo je datoteka 1

<<<<<<< .mine
Ovdje će doći do konflikta1.
=======
Ovim retkom se stvara konflikt sa drugom verzijom ove datoteke!
>>>>>>> .r5

Ovo je kraj datoteke1
```

Ponovimo...

- Naredbe za ispitivanje promjena su:
  - svn status - provjera promjena u radnoj kopiji
  - svn status -u - ispis neažuriranih objekata
  - svn diff - ispis promjena unutar datoteke
- svn revert - vraćanje u prijašnje stanje
- svn resolved - dojava o rješenju konflikta

Ispis zapisnika

- Naredba svn log ispisuje zapisnik sa datutmom i autorom za pojedinu reviziju
- prihvača prekidač --revision

```
$ svn log --revision 5:3
```

- Općenito se može zadati --revision a:b, gdje se prvo ispisuje revizija a pa inkrementalno (ili dekrementalno ako je a>b) do uključno revizije b
- prekidač --verbose - ispisuje i datoteke kojima je izmjenjen put (*eng. path*)

Ostale naredbe

- svn cat - za dohvat i ispis bilo koje datoteke koja je postojala u nekoj reviziji
- svn blame - ispisuje reviziju, autora i promjene koje je on napravio
- svn list - ispisuje popis datoteka u direktoriju bilo koje revizije

Subversion

This is the end...