GIT-DROPBOX

Dokumentacija

30.5.2011

Avtorja:
Rok Ritlop, Jaka Sivec

Kazalo

Uvod

Problem

Vsi smo se že znašli v situaciji, ko ustvarjamo kak dokument in ga moramo večkrat komu pokazati. Seveda dokument vsakič shranimo kot drugo verzijo in kaj kmalu imamo dokument, ki je končni končni verzija pet.

Problem je samo še hujši, če za delo uporabljamo več naprav, kar je s poplavo naprednih mobilnih naprav tudi pri ne-tehničnem prebivalstvu vse bolj pogosto.

Na srečo rešitev tega problema v tehničnih krogih obstaja že skoraj 30 let v obliki razno raznih t.i. "version control" sistemov, ki z različnimi prijemi omogočajo sočasno rabo različnih verzij istih datotek ter vračanje na starejša stanja, kadar se znajdemo v težavah.

Izkaže pa se, da ti sistemi izven programerskih krogov ne zaživijo predvsem, ker so za povprečnega uporabnika pretirano kompleksni, hkrati pa se večini ljudi uporaba takih sistemov preprosto ne zdi potrebna, ker le poredkoma sodelujejo v distribuiranih ekipah.

V zadnjem času se je pojavila okrnjena rešitev tega problema - Dropbox. Ta omogoča uporabo istih datotek na večih napravah, s čimer je zavedanje težave šele prišlo v zavest malce širše javnosti.

O programu

git-dropbox skuša svet sistemov za nadzor verzij približati slehernemu uporabniku, tako da učinkovit nadzor verzij naredi preprost, predvsem pa popolnoma avtomatičen.

Na kratko: združuje najboljše strani git-a in dropboxa. Po eni strani je vsestranski kot git, po drugi pa enostaven in lahek za uporabo kot dropbox.

Navodila za uporabo

Namestitev

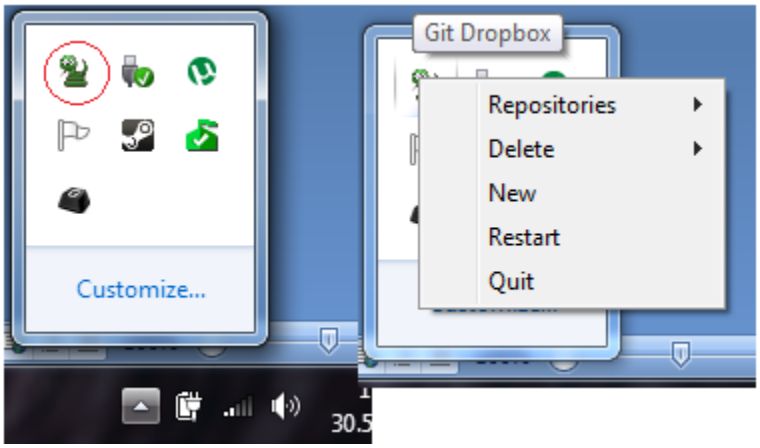
Za uporabo programa na sistemu potrebujemo python 2.7 ali več z nekaj knjižnicami:

- dulwich
- Pywin (windows gui)
- wxPython (windows gui)

Ko imamo vse potrebne knjižice, v direktoriju programa samo poženemo primerno python skripto za svoj sistem.

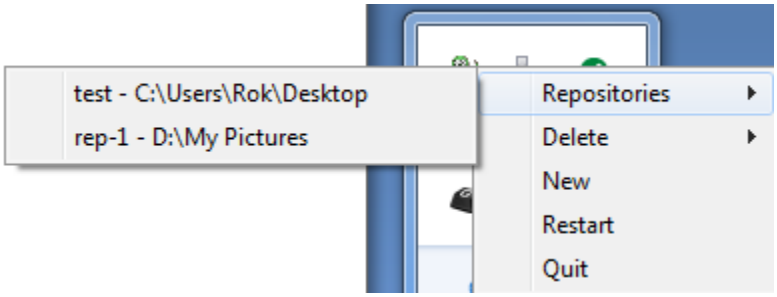
Uporaba

Opisana je uporaba grafičnega vmesnika na Windows okolju.

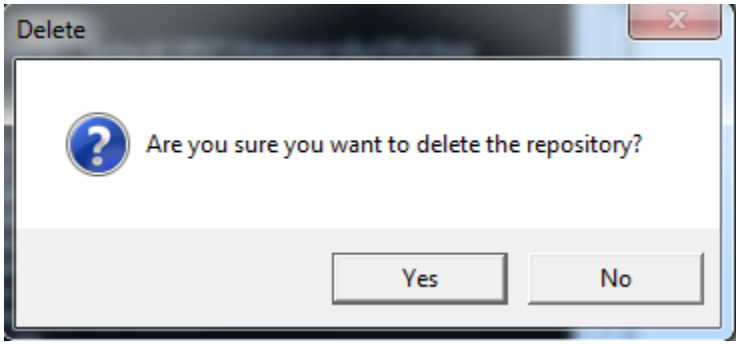


Vmesnik ima naslednje opcije

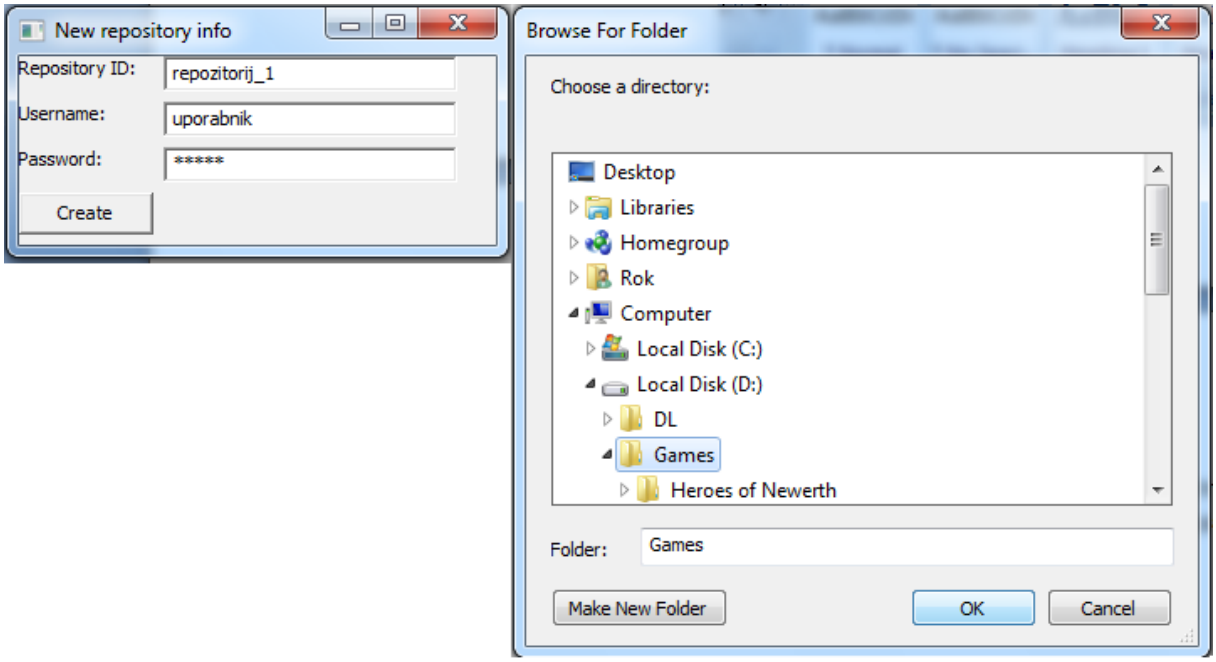
- Repositories: v podmeniju izpiše seznam repozitorijev, klik na enega od njih pa odpre novo okno windows explorerja na tisti lokaciji.



- Delete: ponovno se odpre enak podmenij kot pri prejšnji opciji, le da tokrat klik na željen repozitorij le tega izbriše. Pred izbrisom se pojavi tudi klasično YES/NO potrditveno okence.



- New: opcija za kreiranje novega repozitorija. Tu se pojavita dve okni; najprej okno s tremi polji za vnos, ki sprešuje po imenu repozitorija, uporabniškem imenu, in geslu, nato pa klasičen windows file browser. Po uspešnem vnosu podatkov se repozitorij kreira.



- Restart: resetira aplikacijo.
- Quit: zapre aplikacijo.

Tehnologija

Splošno

git-dropbox za svojo osnovo vzema sistem git, ki ga je Linus Torvalds v originalu razvil za potrebe razvoja Linux Kernela kot alternativo takrat popularnemu svn sistemu. V zadnjem času git hitro postaja najpopularnejši in splošno uporabljan sistem za nadzor verzij, sploh zaradi velikega uspeha github.com, ki je nekakšna moderna različica sourceforge-a.

Ena glavnih prednosti git-a je delovanje na peer-to-peer sistemu brez centralne avtoritete, kar omogoča lažje sodelovanje med veliko razvijalci, saj omogoča direktno dostopanje do katerega koli repozitorija. Tako smo po starem vedno morali svojo kodo poslati na glavni repozitorij in so razvijalci čakali drug na drugega, da so lahko prišli do zadnjih sprememb. Hkrati so se ekipe, ki so razvijale različne aspekte istega projekta, lahko med sabo ovirale saj so vsi uporabljali isti centralni repozitorij. Zdaj pa lahko koherentna ekipa pošilja kodo samo med sabo na arbitrarno izbrani glavni repozitorij pa kodo pošljejo šele, ko je končana.

Druga velika prednost pa je odlična podpora za vejenje repozitorijev, kar v osnovi pomeni, da vedno ko začnemo nov set sprememb lahko odpremo novo vejo in, ako ugotovimo, da smo nekaj naredili hudo narobe, lahko celotno vejo preprosto ignoriramo.

git-dropbox vse to s pridom izkorišča predvsem za dodajanje arbitrarnega števila naprav, ki sodelujejo pri istih datotekah, hkrati pa bi potencialno lahko po nekih hevristikah računal kdaj ob novi spremembi začeti novo vejo v repozitoriju in jo potem primerno združil z glavnimi vejami, ko uporabnik zaključi z delom.

Zaradi kompatibilnosti s čim večjim razponom naprav je cel projekt narejen v pythonu, ki dokaj veselo deluje na prav vseh operacijskih sistemih in je dovolj razširjen, da ga ima večina računalnikov z unix-like okolji nameščenega že kar kot osnoven del sistema.

Za interakcijo z git-om uporabljamo dulwich[1], ki je čista python implementacija git protokola in nam tako nudi vso fleksibilnost in moč git sistema, brez da bi uporabnik sploh moral na sistemu imeti nameščen git. Seveda pa vsi repozitoriji, ki jih ustvari git-dropbox delujejo popolnoma enako kot vsak git repozitorij, kar pomeni, da lahko napredni uporabniki še vedno posegajo po svojih običajnih orodjih kadar si tega poželijo.

Osnovna ideja delovanja

Zasnova git-dropbox bazira na zaznavanju sprememb datotek in primernem komuniciranju z git repozitoriji kamor se datoteke shranjujejo. To dosežemo z uporabo dveh ločenih sistemov.

[1] http://samba.org/~jelmer/dulwich/

Zaznavanje sprememb v datotekah poteka preko python skripte, ki vedno teče v ozadju in spremlja kaj se dogaja v direktorijih, ki so označeni za uporabo z git-dropbox. Na unix okoljih to zaenkrat dosežemo s periodičnim preverjanjem vseh datotek, ki jih primerjamo s časom zadnje znane spremembe in izračunanim hashom. Na windows okolju pa znamo zaznavati obvestila operacijskega sistema o spremembah, kar zmanjša porabo sistemskih sredstev in je načeloma boljša rešitev.

Ko je zaznana sprememba, drugi podsistem naredi nov commit na repozitorij ter tako shrani novo verzijo datoteke v sistem za nadzor verzij. Da ne bi prišlo, do prevelikega števila commitov sistem uporablja hevristiko za grupiranje shranjenih verzij v commite - načeloma vsakih X sprememb v datoteki oz. po določenem pretečenem času.

S podobno hevristiko se sistem potem odloča kdaj nove commite poslati na sestrske repozitorije, s čimer poskrbi za backup podatkov, ker jih razprši na različne trdo ločene naprave.

Zaradi tega lahko v datotekah pride do konfliktov v primeru, da se iste datoteke spreminjajo na različnih napravah v istem časovnem oknu. Ta problem se zaenkrat rešuje, tako da se vzame verzija, ki je novejša, kar sicer ni optimalno vendar se večinoma izkaže za dovolj dobro rešitev, saj v primeru binarnih datotek smiselno merganje različnih verzij praktično ne obstaja. Razen seveda, če bi sistem šel globlje v poznavanje formata binarnih datotek, kar bi bilo praktično v primeru dokumentov, bi se pa lahko hitro zakompliciralo pri slikah in videih.

Windows okolje

Sedaj ko smo opisali kako git-dropbox zgleda na zunaj, pa si lahko pogledamo kako deluje.

Ob zagonu programa se iz datoteke »repositories.dataz« preberejo podatki o vseh trenutno obstoječih repozitorijih. Če datoteke ni se kreira nova prazna datoteka.

Za vsak tako pridobljen repozitorij se potem zažene nov thread, ki na določeni lokaciji posluša za kakršnekoli spremembe datotek. Hkrati se iz podatkov o repozitorijih kreirajo potrebne podatkovne strukture za menije v system trayu. Nato se požene del kode ki naredi in operira s celotnim system trayem. Velika večina te kode je skopirana iz interneta in spremenjena za naše potrebe, zato je tudi hkrati edini večji kos kode v tem projektu ki ni dobro pokomentiran.

V tej točki program čaka na interakcijo z uporabnikom, threadi pa poslušajo svoje direktorije za spremembe. V primeru spremembe thread pokliče primerne funkcije za sinhronizacijo repozitorijev, ki so obravnavane v drugem delu te dokumentacije.

Ob izhodu iz aplikacije se najprej ugasne system tray ikona, nato pa se požene metoda ki konča vse threade.

Izzivi

V tem delu bom izpostavil nekaj tehničnih izzivov, s katerimi sem imel opravka.

Za še največji izziv se je izkazalo sprotno updatanje seznama repozitorijev v meniju po dodajanju oziroma brisanju repozitorijev. Edini način, ki sem ga našel, je da se ob vsaki spremembi seznama repozitorijev celoten system tray del programa ugasne in ponovno zažene.

Tak način »brute force updatanja« pa ni bil tako slab, saj tako lahko hkrati tudi ustavimo in ponovno zaženemo vse threade – z novimi podatki. Tako bo thread, ki nadzira repozitorij ki smo ga ravnokar odstranili, takoj ugasnjen, hkrati pa bo ravnokar kreiran repozitorij takoj dobil svoj thread.

Naslednji izziv se je pojavil zaradi načina implementacije threadov. Izkazalo se je, da so threadi z neskončno zanko zelo odporni na konvencionalne metode ugašanja. Tudi tu je bila na koncu implementirana bolj »brute force« metoda, ki povzroči izhod iz neskončne zanke, nato pa se thread izvede do konca in ustavi.

Na koncu pa je tu še edini trenutno znan bug tega dela seminarske: kakršnokoli resetiranje system tray ikone (torej kreiranje, brisanje ali pa reset opcija) povzroči crash pythona če poskušaš nato odpreti kakšnega od repozitorijev prek system tray menija. Ker sta ta dva dela kode (reset in odpiranje folderja) dve popolnoma različni in ločeni kodi resnično ne vem kaj bi ta bug lahko povzročalo, hkrati pa mi za nadaljnje testiranje zmanjkuje časa.

Nadaljnje delo

git-dropbox je trenutno še v zelo prototipni fazi in ni primeren za slehernega uporabnika, s čimer bi se lahko reklo, da je zgrešil enega od svojih osnovnih ciljev - biti popolnoma preprost za uporabo. Kljub temu pa ga je primeren za srednje-zahtevne uporabnike, ki so si pripravljeni vzeti malo časa za razumevanje delovanja in postavitev lokalnega sistema, potem pa se s tem več ne želijo ukvarjati in hočejo, da se vse dogaja samodejno.

Ako se odločiva z razvojem git-dropboxa nadaljevati bo potrebno predvsem delati na uporabniški izkušnji. Vse od [boljšega] avtomatskega odkrivanja oddaljenih repozitorijev s katerimi se syncati, do predstavitve različnih vej in verzij datotek uporabniku; predvsem znajo biti težave pri sočasnosti različnih verzij, kar se lahko kaj hitro pripeti.

Prav tako je v trenutnem stanju slabo poskrbljeno za sestavljanje datotek iz več sočasnih verzij, saj v primeru konfliktov preprosto vzamemo verzijo, ki je malce novejša, čeprav bi bilo verjetno primerneje za razrešitev konflikta poprositi uporabnika. Tu pa je težava kako to uporabniku predstaviti, ker se dodajanje v repozitorij in pošiljanje datotek dogaja avtomatsko brez sodelovanja uporabnika.