CVS-RCS-HOW-TO za Linux (sistem za kontrolo izvorne kode)

Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com*; prevedel Jernej Kovačič *jkovacic@email.si*; v21.4, 25. avgust 2001, prevod 27. september 2001

Ta dokument je "praktični vodnik" do hitre postavitve sistema za nadzor izvorne kode CVS/RCS. Dokument vsebuje tudi nekaj lupinskih skriptov, ki poenostavijo nekaj najpogosteje uporabljanih funkcij CVS. Ti skripti omogočajo lahek vmesnik do CVS. Na voljo je tudi nekaj skriptov, ki poenostavijo delo z RCS. Informacije v tem dokumentu se nanašajo tako na Linux kot tudi na ostale vrste Unixa, kot so Solaris, HPUX, AIX, SCO, Sinix, BSD, SCO, Apple Macintosh (ki je BSD unix)... in BeOS.

Kazalo

1 Uvod

Sistem za nadzor izvorne kode je NUJEN za nadzor nad spremembami, do katerih prihaja med razvojem programskih projektov. Razvijalci potrebujejo popolno zgodovino sprememb, da se lahko v primeru kakršnihkoli problemov vrnejo k prejšnjim različicam. Ker je izvorna koda najvitalnejša komponenta kateregakoli programskega projekta in ker razvoj programske opreme porabi ogromno časa in denarja, je zelo pomembno porabiti nekaj časa za ***varovanje*** izvorne kode z uporabo sistemov za kontrolo izvorne kode, kot sta npr. CVS in RCS.

CVS (angl. Concurrent Version Control System, sistem za kontrolo hkratnih različic) je zmogljivo orodje, ki omogoča večim uporabnikom hkraten razvoj programske opreme. Je nadgradnja RCS in je nekakšen vmesnik na aplikacijskem nivoju do RCS.

CVS lahko zapisuje zgodovino vaših datotek (ponavadi, vendar ne zmeraj, gre za izvorno kodo). Namesto vsake različice vsake datoteke, ki ste jo kadarkoli naredili, hrani samo razlike med različicami. Med drugim vodi tudi evidenco, kdo, kdaj in zakaj je povzročil kakšno spremembo.

CVS je v veliko pomoč pri upravljanju z izdajami in nadzoru hkratnih urejanj izvornih datotek med različnimi avtorji. Namesto omogočanja kontrole različic za zbirko datotek v posameznem imeniku nam CVS omogoča kontrolo različic za hierarhično zbirko imenikov, ki vsebujejo revizije nadzorovanih datotek.

Te imenike in datoteke lahko potem kombiniramo, da oblikujemo izdajo programske opreme.

S CVS lahko hranimo datoteke "C", "C++", Java, Perl, HTML in druge.

2 Kateri je zame? CVS ali RCS?

CVS je pravzaprav nadgrajen RCS. CVS je precej zmogljivejše orodje in lahko nadzoruje celotno drevo izvorne kode. Uporaba CVS je ***zelo*** priporočena, ker si delo z njim lahko poenostavite z uporabo skriptnih jezikov, kot so npr. Perl, korn ali bash. Oglejte si vzorčne skripte za lupino Korn v poglavju o 6 (lupinskih skriptih).

Prednosti CVS:

- CVS je decentraliziran, uporabnik prevzema datoteke oz. imenike iz skladišča in ima svoje ločeno stabilno drevo imenikov izvorne kode.
- CVS lahko "označuje"izdaje celotnega projekta drevesa izvorne kode.
- CVS omogoča hkratno urejanje datotek.
- CVS si lahko prilagodimo, da nam omogoča zmogljivo zaklepanje datotek ali hkratno urejanje datotek z uporabo lupinskih skriptov ali Perla. CVS podpira šibko zaklepanje (ukaz 'cvs watches'), mogoče pa je tudi hkratno urejanje datotek brez zaklepanja.

Slabosti CVS:

- Potrebnega je nekoliko več administriranja kot pri RCS.
- Zelo izpopolnjen in kompleksen sistem. Je žadnji krik"tehnologije. CVS je zelo napreden in zmogljiv sistem, ki so ga razvijali zelo dolgo (več let!!).
- Ima veliko število ukazov in ukaznih možnosti, zato se ga začetniki težje privadijo. Lupinski skripti v poglavju o 6 (lupinskih skriptih) lahko olajšajo uporabo.

Prednosti RCS:

- RCS je zelo preprost za nastavljanje, manj dela je z administriranjem.
- RCS se uporablja v centraliziranih področjih, kjer vsak dela.
- RCS je koristen za preproste sisteme.
- Zelo zmogljivo zaklepanje datotek - hkratno delo je onemogočeno.

Slabosti RCS:

- Hkraten razvoj večih razvijalcev ni mogoč zaradi zaklepanja datotek in omejenosti na posamezni delovni imenik. Zaradi omejitve na posamezni delovni imenik lahko spremembe datotek, povzročene od več razvijalcev, povzročijo napako ob izvedbi ukaza 'make'.
- Ne more označevati izdaj celotnega programskega projekta.

Ta dokument vsebuje tudi lupinske skripte, ki omogočajo enostavno vlaganje (angl. check in), prevzemanje (angl. check out) in udejanjanje (angl. commit) datotek. Oglejte si jih v poglavju o 6 (lupinskih skriptih).

Za navodila o uporabi RCS si oglejte RCS mini-howto na CDju z Linuxom:

```
cd /mnt/cdrom/Redhat/RPMS
ls -l howto-6.0-*.noarch.rpm
rpm -qpl howto-6* | grep -i rcs
```

ali obiščite <http://www.LinuxDoc.org/HOWTO/mini/RCS.html>

Oglejte si tudi poglavje o 12 (lupinskih skriptih za RCS).

3 Postavljanje CVS

Najprej morate namestiti paket CVS. Če imate Redhat Linux, izvedite

```
cd /mnt/cdrom/Redhat/RPMS
rpm -i rcs*.rpm
rpm -i cvs*.rpm
Spisek nameščenih datotek si lahko ogledate z -
rpm -qpl cvs*.rpm | less
```

in pobrskajte po izhodu z j,k, CTRL+f, CTRL+D, CTRL+B, CTRL+U, page up/down ali smerniškimi tipkami. Oglejte si 'man less'.

Pri drugih vrstah Unixa boste morda morali najprej dobiti datoteke v zapisu tar z RCS in CVS ter slediti datotekama README in INSTALL za postavitev CVS. Obiščite <http://www.cyclic.com> in <http://www.loria.fr/~molli/cvs-index.html>.

3.1 Spremenljivke okolja

Naslednje spremenljivke okolja morajo biti nastavljene v /etc/profile - to so privzete vrednosti zahtevane za vse uporabnike. Če niso nastavljene v /etc/profile, morate dodati tole v vašo lokalno datoteko za zagon ob prijavi: /.bash_profile.

```
export EDITOR=/bin/vi
export CVSROOT=/home/cvsroot
export CVSREAD=yes
```

Naredite imenik za skladiščenje izvorne kode ter dodelite dostop za branje in pisanje unixovi skupini/uporabniku. Prepričajte se tudi, da ime imenika pri spremenljivki CVSROOT ne vsebuje presledkov (vrednosti te spremenljivke ne smete nastaviti na npr. '/home/moj cvsroot'.

```
bash$ su - root
bash# export CVSROOT=/home/cvsroot
bash# groupadd --help
bash# groupadd cvs
bash# useradd --help
bash# useradd -g cvs -d /home/cvsroot cvs

bash# ls -ld $CVSROOT   ... (morali bi videti tudi seznam datotek)
bash# chmod o-rwx $CVSROOT
bash# chmod ug+rwx $CVSROOT

# Pripravite si skladišče CVS in vanj shranite datoteke:
bash# cvs init

# V skupino cvs dodajte uporabniška imena uporabnikov. Naredite dodatno skupino za uporabnike.
# Upoštevajte, da za vejicami, ki ločujejo imena skupin pri izbiri -G, NE SME
```

```
# biti presledkov.
# V spodnjem primeru uporabnik tom pripada skupinam cvs, users in staff,
# uporabnik johnson pa pripada samo skupini cvs.
bash# usermod --help
bash# usermod -G cvs unixova_uporabniška_imena
bash# usermod -G cvs,users,staff tom
bash# usermod -G cvs,users,staroffice billclinton
bash# usermod -G cvs johnson
bash# exit    .... (odjavite se z računa sistemskega uporabnika (root))

# Prijavite se kot uporabnik in uvozite datoteke v CVS....
bash$ su - billclinton
bash$ export EDITOR=/bin/vi
bash$ export CVSROOT=/home/cvsroot
bash$ export CVSREAD=yes

# Sprememba delovnega imenika je obvezna
bash$ cd $HOME/moj_imenik_z_izvorno_kodo

# Podati je treba oznako proizvajalca in revizije
cvs import moj_imenik_z_izvorno_kodo Vendor1_0 Rev1_0

# Npr. jaz sem naredil tole:
bash$ cd $HOME/foobar
bash$ cvs import foobar Vendor1_0 Rev1_0

# Primer testiranja in verifikacije:
bash$ cd $HOME/foobar
bash$ cvs checkout myfoo.java
```

POGOSTA NAPAKA: ko izvedete 'cvs checkout', lahko dobite opozorilo, da je modul neznan. Pogosta napaka je, da ob 'cvs import' ne zamenjate imenika. ***OBVEZNO morate*** iti v imenik z izvorno kodo in šele potem izvedite 'cvs import'. Na primer:

```
bash$ cd $HOME/foobardir
bash$ cvs import foobardir Vendor1_0 Rev1_0
```

3.2 Selitev od RCS k CVS

Ze preselitev obstoječih datotek iz RCS v CVS uporabite naslednji skript. Prepričajte se, da imate nameščen paket z lupino Korn (pdksh*.rpm s CDja Linux Contrib).

Opomba: ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux Contrib.***

```
#!/bin/ksh
```

```
#############################################################
# Program za selitev obstoječih datotek izvorne kode iz RCS v CVS
#
# Potreben je RPM paket pdksh*.rpm s Cdja Linux contrib
#############################################################

#
# rcs2cvs - pretvori drevo izvorne kode iz RCS v CVS
#

# projekt za pretvorbo
PROJECT='project'

# trenutni izhodiščni imenik RCS
RCSROOT="$HOME/rcs"

if cd "$RCSROOT/$PROJECT"
then
        cd "$RCSROOT"
else
        echo >&2 "`basename "$0"`: ne morem priti v RCS-jev imenik '$RCSROOT/$PROJECT'."
        exit 1
fi

# trenutni izhodiščni imenik CVS
CVSROOT="$HOME/cvs"

# naredi nov imenik v CVS za projekt 'project'
if mkdir "$CVSROOT/$PROJECT"
then
        :
else
        echo >&2 "`basename "$0"`: ne morem narediti CVS-jevega imenika '$CVSROOT/$PROJECT'."
        exit 2
fi

# naredi projektno drevo CVS iz drevesa RCS
find "$PROJECT" -type d -name RCS -print |
while read RCS
do
        CVS="`dirname "$RCS"`"
        (if cd "$RCS"
        then
#               if find . -type f -name '*,v' -print | cpio -pdmv "$CVSROOT/$CVS"
                if find . -type f -print | cpio -pdmv "$CVSROOT/$CVS"
                then
                        :
```

```
                else
                        echo >&2 "`basename "$0"`: ne morem pretvoriti RCS-jevega podimenika '$RCSROOT/$RCS' v CVS-jev podimenik '$CVSROOT/$CVS'."
                fi
        else
                echo >&2 "`basename "$0"`: ne morem priti v  RCS-jev podimenik '$RCSROOT/$RCS'."
        fi)
done
```

Sedaj smo preselili projekt iz RCS v CVS kot 'project'. Pričnete lahko uporabljati ukaze CVS na modulu 'project'.

4 Uvod v ukaze CVS

V CVS nam je na voljo precej ukazov (v pregledu cvs_ukaz), izmed katerih nam jih precej nudi še nemajhno število izbir, ki zadostijo precej potrebam upravljanja z izvorno kodo v porazdeljenih okoljih. Vendar nam za koristno delo s CVS ni potrebno obvladati prav vseh podrobnosti. Pravzaprav nam za uporabo in prispevanje v skladišče zadostuje že pet ukazov. Najpogosteje se uporabljajo tile ukazi: **checkout**, **update**, **add**, **remove**, **commit** in **diff**.

4.1 checkout

cvs checkout moduli... Potreben začetni ukaz, da deluje večina CVS: naredi vašo zasebno kopijo izvorne kode za module (ki se imenujejo zbirka izvorne kode; tu lahko tudi uporabite pot, ki je relativna na izvorno skladišče). S to kopijo lahko delate brez vmešavanje v delo ostalih. Vedno se naredi vsaj en podimenik.

```
bash$ cvs --help checkout
Usage:
  cvs checkout [-ANPRcflnps] [-r rev | -D date] [-d dir]
    [-j rev1] [-j rev2] [-k kopt] moduli...
        -A      Resetiraj vse lepljive oznake/datum/kopt
        -N      Če je dolčena izbira -d, ne skrajšuj poti v modulu.
        -P      Oklesti prazne imenike.
        -R      Imenike obdelaj rekurzivno.
        -c      Izpiši (z ukazom cat) podatkovno bazo modula.
        -f      Če ne najdeš oznake/datuma, se naj ujemata z glavno revizijo.
        -l      Samo lokalni imenik, brez rekurzije.
        -n      Ne poganjaj programov v modulu (če obstajajo).
        -p      Prevzemi datoteko na standardni izhod (v izogib lepljivosti).
        -s      Podobno kot -c, vendar vključi še modulov status.
        -r rev  Prevzemi revizijo ali oznako (izvede tudi -P) (je lepljiv).
        -D date Prevzemi revizije kot od datuma (izvede tudi -P) (je lepljiv)
        -d dir  Prevzemi v imenik dir namesto v ime modula.
        -k kopt Pri prevzemu uporabi RCS-jevo izbiro kopt -k.
        -j rev  Spoji spremembe, ki so nastale med trenutno revizijo in rev.
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.2 update

cvs update Ta ukaz izvedite znotraj vašega zasebnega imenika z izvorno kodo, kadar želite posodobiti vaše kopije izvornih datotek glede na spremembe, ki so jih drugi razvijalci naredili v izvorno kodo v skladišču.

```
bash$ cvs --help update
Usage: cvs update [-APdflRp] [-k kopt] [-r rev|-D date] [-j rev]
    [-I ign] [-W spec] [datoteke...]
        -A      Resetiraj vse lepljive oznake/datume/kopt.
        -P      Oklesti prazne imenike.
        -d      Zgradi imenike, podobo kot jih checkout.
        -f      Če ne najdeš oznake/datuma, se naj ujemata z glavno revizijo.
        -l      Samo lokalni imenik, brez rekurzije.
        -R      Obdelaj imenike rekurzivno.
        -p      Pošlji posodobitve na standardni izhod (v izogib lepljivosti).
        -k kopt Pri prevzemu uporabi RCSjevo izbiro kopt -k.
        -r rev  Posodobi in uporabi podano revizijo/oznako (je lepljiv).
        -D date Nastavi datum, glede na katerega naj se posodobi (je lepljiv).
        -j rev  Spoji spremembe med trenutno revizijo in rev.
        -I ign  Več datotek, ki naj se spregledajo (! za resetiranje).
        -W spec Vrstica za določitev ovitka.
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.3 add

cvs add datoteka... Ta ukaz uporabite za vpis novih datotek v CVS zapis vašega delovnega imenika. Datoteke se bodo dodale v skladišče ob naslednji izvedbi ukaza 'cvs commit'. Opomba: za prenos novih datotek v vaše zasebno skladišče uporabite 'cvs import'. 'cvs add' se uporablja samo za nove datoteke v že prevzetem imeniku.

```
bash$ cvs --help add
Usage: cvs add [-k rcs-kflag] [-m sporočilo] datoteke...
        -k      Pri dodajanju datoteke z določeno zastavico kflag uporabi "rcs-kflag".
        -m      V dnevnik zapiši sporočilo.
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.4 remove

cvs remove datoteka... Ta ukaz uporabite potem, ko ste zbrisali vse prej navedene datoteke. Z njim prijavite CVSju, da želite označene datoteke odstraniti iz skladišča. Odstranitev se ne pozna pri ostalih, vse dokler ne izvedete 'cvs commit'.

```
bash$ cvs --help remove
Usage: cvs remove [-flR] [datoteke...]
        -f      Briši datoteko preden jo odstraniš iz skladišča.
        -l      Obdelaj samo ta imenik (brez rekurzije).
        -R      Imenike obdelaj rekurzivno.
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.5 commit

cvs commit datoteka... Ta ukaz uporabite, kadar želite vnesti spremembe v skladišče in jih "objaviti" tudi za ostale razvijalce.

```
bash$ cvs --help commit
Usage: cvs commit [-nRlf] [-m spr. | -F dnevnik] [-r rev] datoteke...
        -n      Ne poganjaj programov v modulih (če obstajajo).
        -R      Imenike obdelaj rekurzivno.
        -l      Samo lokalni imenik (brez rekurzije).
        -f      Prisili datoteko, da bo udejanjena; onemogoči rekurzijo.
        -F dat. Preberi dnevniško sporočilo iz datoteke dat.
        -m spr. Log message.
        -r rev  Udejani v to vejo ali deblo revizije.
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.6 diff

cvs diff datoteka... Prikaže razlike med datotekami v delovnem imeniku in skladišču ali med dvema revizijama v skladišču. Ne spreminja datotek niti v skladišču niti v delovnem imeniku.

```
bash$ cvs --help diff
Usage: cvs diff [-lNR] [rcsdiff-options]
    [[-r rev1 | -D date1] [-r rev2 | -D date2]] [datoteke...]
        -l      Samo lokalni imenik, brez rekurzije.
        -R      Imenike obdelaj rekurzivno.
        -D d1   Izvedi diff med datoteko s podanim datumom glede na delovno datoteko.
        -D d2   Izvedi diff revizije rev1/date1 glede na date2.
        -N      vključi razlike za dodane in odstranjene datoteke.
        -r rev1 Izvedi diff med revizijo rev1 glede na delovno datoteko.
        -r rev2 Izvedi diff revizije rev1/date1 glede na revizijo rev2.
        --ifdef=arg     Izpisi razlike v zapisu ifdef.
(Glede izbir rcsdiff-options za si oglejte dokumentacijo za vaš program diff.
Najbolj priljubljena izbira je -c, je pa na voljo še veliko drugih).
(Vnesi izbiro --help global za seznam ostalih izbir)
```

4.7 Urejevalnik Emacs

Emacs je zmogljiv urejevalnik in podpira CVS/RCS - še posebej za spajanje in primerjavo revizij. Emacsova glavna domača stran je `<http://www.emacs.org>`.

5 Močno in šibko zaklepanje ali način brez zaklepanja

CVS je zmogljiv in zelo nastavljiv sistem. CVS podpira:

- Močno zaklepanje z "rezerviranimi prevzemiš pomočjo ukaza **cvs admin -l** ali 6 (lupinskega skripta). Preberite tudi spis *Reserved checkouts* `<http://www.cvshome.org/docs/inforeserve.html>`. Popravek Erica Griswolda o rezerviranih prevzemih najdete na naslovu `<http://www.cvshome.org/dev/patches/editf>`.
- Šibko zaklepanje s pomočjo značilnosti ukaza 'cvs watch'. Oglejte si tudi "cvs edit"glede opozorila (`<http://www.cvshome.org/dev/text2/res2>`), če kdo že ureja datoteko.
- Brez zaklepanja - privzeto je dovoljeno, da več razvijalcev hkrati ureja datoteke.

6 Lupinski skripti

Sledijo skripti za poenostavitev uporabe osnovnih ukazov CVS. Ti skripti vam dajo **prvi vpogled** v sistem CVS in so koristni, dokler se ne privadite na ukaze CVS. Skripti so napisani za lupino Korn, ker je ta lupina vedno na voljo za vse vrste Unixa, vedno pa jih lahko prevedete v Bash ali Perl, če je to potrebno. Prilagodite si jih lahko po svojem okusu. V osnovi so to ukazi CVS, dodane pa so še nekatere značilnosti, ki jih naredijo specifične glede na lokacijo. Na primer, skript sedit omogoča zaklepanje, tako uporabniki vedo, da nekdo že ureja to datoteko. Seveda lahko uporabniki neposredno uporabljajo ukaze CVS in tako obidejo te skripte. Ti skripti prikazujejo, kako si lahko prilagodite CVS.

OPOMBA: *Lupinski skripti privzamejo uporabnikov domači imenik za korenskega in prevzamejo izvorno drevo iz CVS tako, da ga naredijo neposredno pod uporabnikovim domačim imenikom.*

NASVET: *V teh lupinskih skriptih se vsako ime datoteke na vašem disku sestoji iz treh delov: domačega imenika, podimenika in imena datoteke. Polna pot do datoteke je sestavljena iz spremenljivk: $HOME/$subdir/$fname. V CVS je z uporabo spremenljivke $subdir ohranjena enaka struktura imenikov, torej bo tam izgledala nekako takole $CVSROOT/$subdir/$fname. V vseh skriptih igrajo spremenljivke $HOME, $CVSROOT, $subdir in $fname pomembno vlogo. Njihove vrednosti so npr. lahko HOME=/home/aldev, subdir=myproject/src, CVSROOT=/home/cvsroot, in fname=foo.cpp*

Skopirajte te skripte v imenik /usr/local/bin, ki naj bo v uporabnikovi spremenljivki okolja PATH.

1. **sget** [-r stevilka_revizije] <datoteka/imenik ime> Iz CVS dobite datoteko ali celoten imenik v načinu "samo-za-branje"(READ ONLY). Kliknite na 20 ().
2. **sedit** [-r stevilka_revizije] <ime_datoteke> Urejanje datoteke z namenom spremeniti vsebino. Datoteka se zaklene, tako, da je nihče ne more prevzeti. Seveda si skript lahko prilagodite svojim potrebam - brez zaklepanja, opozorilna sporočila ali zelo zmogljivo zaklepanje. Kliknite na 21 ().
3. **scommit** [-r stevilka_revizije] <ime_datoteke> Udejanite spremembe v datoteko ali ves imenik. Odložite spremembe v CVS. Kliknite na 22 ().
4. **supdate** <ime_datoteke/imenika> Posodobite datoteko ali ves imenik tako, da dobite najnoveše datoteke iz CVS. Kliknite na 23 ().
5. **sunlock** [-r stevilka_revizije] <ime_datoteke> Odklepanje datoteke, dobljene s sedit. Kliknite na 24 ().
6. **slist** Ogled datotek, ki jih trenutno urejate. Izvrši ukaz 'ls -l | grep | ...'. Kliknite na 25 (). Upoštevajte, da obstaja še en Unixov ukaz slist (ki izpiše seznam razpoložljivih strežnikov Netware). Prepričajte se, da je imenik s skriptom slist v spremenljivki PATH naveden pred drugimi imeniki s tem ukazom.
7. **sinfo** <ime_datoteke/imenika> Informacije o spremembah/revizijah datoteke. Kliknite na 26 ().
8. **slog** <ime_datoteke> Zgodovina sprememb/revizij datoteke iz CVS. Kliknite na 27 ().

9. **sdif** <ime_datoteke> **sdif** -r revizija1 -r revizija2 <ime datoteke> Razlike med dvema revizijama datoteke v CVS. Kliknite na 28 ().

   OPOMBA: ukaz sdif ima samo en 'f', ker že obstaja drug Unixov ukaz 'sdiff'

10. **sadd** <ime_datoteke> Doda novo datoteko v skladišče CVS. Kliknite na 29 ().

11. **sdelete** <ime_datoteke> Brisanje datoteke iz skladišča CVS. Kliknite na 30 ().

12. **sfreeze** <ime_revizije> <ime_imenika> Zamrznitev kode, izdaja celotnega izvornega drevesa. Kliknite na 31 (). Na primer:

```
cd $HOME;
sfreeze REVIZIJA_1_0  izvdrevo
```

zamrzne kodo z oznako REVIZIJA_1_0, da lahko pozneje prevzamete celotno drevo z uporabo imena revizije.

7 Dokumentacija CVS

V Unixovi ukazni vrstici odtipkajte:

1. cvs –help
2. cvs –help-options
3. cvs –help-commands
4. cvs -H checkout
5. cvs -H commit
6. man cvs
7. man tkcvs
8. Obiščite `<http://www.cyclic.com>`
9. Obiščite `<http://www.loria.fr/~molli/cvs-index.html>`

Tkcvs (`<http://www.tkcvs.org>`) je Tcl/Tk grafični vmesnik do CVS. Vsebuje tudi sprotno pomoč.

- cd $HOME/src/foo.cpp
- tkcvs
- Kliknite na foo.cpp
- Kliknite na ikono 'Revision Log' ki se nahaja takoj za ikono 'spectacle'
- To bo v oknu prikazalo vejo drevesa. Sedaj z DESNIM miškinim gumbom kliknite na besedilo '1.3' in z LEVIM gumbom kliknite na '1.1'. Potem kliknite na gumb"Diff". Prikaže se okno z dvema razdelkoma!!
- Kliknite na gumb "Nextža sprehod čez še več razlik. Kliknite na "Centerža centriranje besedila.

Obstaja tudi CVS odjemalec za Windows 95, ki se imenuje WinCVS (oglejte si <http://www.wincvs.org> in stran *Cyclicsite* <http://www.cvshome.org/cyclic/cvs/soft-maccvs.html>). WinCVS se lahko uporablja skupaj s Sambo - <http://www.samba.org>

Potrebni ukazi so:

- cvs checkout <ime_datoteke>
- cvs update <ime_datoteke>
- cvs add <datoteka, ..>
- cvs remove <datoteka, ..>
- cvs commit <datoteka>
- cvs status <ime_datoteke>
- cvs log <ime_datoteke>
- cvs diff -r1.4 -r1.5 <ime_datoteke> To poda razlike med različicama datoteke 1.4 and 1.5.

7.1 Priložena dokumentacija

Na sistemih z Linuxom lahko najdete dokumentacijo za CVS v zapisu postscript v imeniku **/usr/doc/cvs*/*.ps**. Na voljo je tudi seznam odgovorov na pogosto zastavljana vprašanja in ostale koristne informacije.

```
bash# cd /usr/doc/cvs*
bash# gv cvs.ps
```

7.2 Dokumentacija organizacije „CVS Organisation“

Dokumentacijo organizacije "CVS Organisation"boste našli na naslovu <http://www.cvshome.org/docs>.

Uradni Cederquistov priročnik za CVS se nahaja na naslovu <http://www.cvshome.org/docs/manual/cvs.html>.

Spisek pogosto zastavljanih vprašanj za CVS boste našli na naslovu <http://www.cs.utah.edu/dept/old/texinfo/cvs/FAQ.txt>.

7.3 Učenje CVS

- <http://rpmfind.net/tools/CVS/training/cvstrain.html>
- <http://www.loria.fr/~molli/cvs/cvs-tut/cvs_tutorial_toc.html>
- <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/tools/srt/>
- <http://durak.org/cvswebsites/>
- <http://www-users.informatik.rwth-aachen.de/~wge/tools/cvs/cvsclient/cvsclient_toc.html>

- <http://www-users.informatik.rwth-aachen.de/~wge/tools/cvs.html> Splošni pripomočki za CVS, ki so jih napisali ostali avtorji:
- Knjiga "Open Source Development with CVS"Karla Fogela na naslovu <http://cvsbook.red-bean.com> ima *Poglavje Third-party-tools v uradnem priročniku* <http://cvsbook.red-bean.com/cvsbook.html> in zrcalne strani: *Zevils* <http://www.zevils.com/doc/cvsbook/cvsbook_8.html>
- <http://rcs.ee.washington.edu/spp/Projects/Manastash/Links/cvsbook_toc.html>

8 Grafični uporabniški vmesniki

Na voljo so naslednji grafični uporabniški vmesniki za CVS:

- CVS home.org: <http://www.cvshome.org/dev/addons.html>
- CVS Web for windows: <http://www.devguy.com/fp/cfgmgmt/cvs/cvs_admin_nt.htm> in <http://stud.fh-heilbronn.de/~zeller/cgi/cvsweb.cgi>
- TkCVS (<http://www.tkcvs.org>) je grafični vmesnik za CVS napisan v jeziku Tcl/Tk. Druga lokacija: *cyclicsite* <http://www.cvshome.org/cyclic/tkcvs/index.html>
- gCVS: prenosljiv grafični vmesnik za netehnične uporabnike CVS: <http://www.arachne.org/software/gcvs>
- jCVS je odjemalec za CVS, v celoti napisan v javi: <http://www.jcvs.org> in *cyclicsite* <http://www.cvshome.org/cyclic/jcvs/index.html>
- WinCVS: <http://www.cvshome.org/cyclic/cvs/soft-maccvs.html> in *cyclicsite* <http://www.cvshome.org/cyclic/cvs/soft-maccvs.html>
- Component soft Win CVS: <http://www.componentsoftware.com/cvs>
- JA-SIG UPortal CVS <http://www.mis3.udel.edu/~jlaker/development>
- <http://ppprs1.phy.tu-dresden.de/~trogisch/lincvs/lincvsen.html>
- <http://www.loria.fr/~molli/cvs/doc/cvs_toc.html>

Zelo vam priporočamo, da uporabljate *Sambo (samba*.rpm na cdromu)* <http://www.samba.org> in 9 (strežnik X) na računalnikih z operacijskim sistemom MS Windows 95/NT. Z uporabo Sambe bo imenik na računalniku z Unixom v MS Windows videti kot lokalna mapa. Oglejte si naslednji razdelek o 9 (strežnikih X).

9 CVS za MS Windows 95/98/NT/2000

ZELO vam priporočamo uporabo *Sambe (samba*.rpm na cdromu)* <http://www.samba.org> in VNC viewerja (ali strežnika X) na sistemih z MS Windows 95/NT. S Sambo bo CVS strežnik na Unixu/Linuxu deloval kot **datotečni strežnik**. Z uporabo Sambe bo imenik na oddaljenem računalniku z Unixom v MS Windows videti kot lokalna mapa na lokalnem disku. Namestite paket samba*.rpm na strežniku Unix/Linux (s skladiščem CVS) in namestite VNC viewer (ali strežnik X) na namizje MS Windows 95/NT/2000. Z uporabo VNC (ali strežnika X) se z lahkoto prijavite v sistem Unix in vlagate/prevzemate datoteke. Na sistemih MS Windows lahko tudi uporabljate orodja kot Java Visual Cafe ali Java JBuilder za urejanje datotek iz imenikov s sistema Unix/Linux (preko Sambe). Po urejanju lahko vložite datoteke v sistem Unix s pomočjo VNC ali strežnika X.

Prednosti uporabe CVS na Linuxu/Unixu preko MS Windows so:

- En sam na Linuxu postavljen datotečni strežnik (CVS strežnik) lahko streže veliko odjemalcem na sistemih MS Windows.
- Datotečni (CVS) strežnik pod Linuxom je zelo robusten, varen in zanesljiv.
- Potrebna je samo ena baterija UPS (za neprekinjeno napajanje) za strežnik z Linuxom.
- S pomočjo Sambe bodo Linuxovi imeniki videti kot imeniki na sistemu MS Windows.
- Linuxov datotečni (CVS) strežnik podpira centralizirano izdelavo varnostnih kopij s pomočjo orodij kot *Arkeia, Bru* `<http://www.aldev.8m.com>`, zrcalni strežniki tudi na `<http://aldev0.webjump.com>`, *angelfire* `<http://www.angelfire.com/country/aldev0>`, *geocities* `<http://www.geocities.com/alavoor/index.html>`, *virtualave* `<http://aldev0.virtualave.net>`, *50megs* `<http://aldev0.50megs.com>`, *theglobe* `<http://members.theglobe.com/aldev1/index.html>`, *NBCi* `<http://members.nbci.com/alavoor>`, *Terrashare* `<http://aldev.terrashare.com>`, *Fortunecity* `<http://members.fortunecity.com/aldev>`, *Freewebsites* `<http://aldev.freewebsites.com>`, *Tripod* `<http://members.tripod.lycos.com/aldev>`, *Spree* `<http://members.spree.com/technology/aldev>`, *Escalix* `<http://www.escalix.com/freepage/aldev>`, *Httpcity* `<http://www.httpcity.com/aldev/index.html>`, *Freeservers* `<http://aldev.freeservers.com>`.
- Linuxov datotečni (CVS) strežnik zahteva samo en majhen klimatiziran in brezprašen prostor. Majhen prostor zniža stroške hlajenja/ogrevanja.
- Linux datotečni (CVS) strežnik omogoča varnost preko Unixovih skupin in overovljanj uporabnikove identitete.

Najboljše orodje za dostopanje do oddaljenih računalnikov je VNC. VNC ne porabi veliko procesorske moči ter pomnilnika in je precej boljši od strežnikov X. ***Uporabo VNC priporočamo bolj kot strežnike X***. Oddaljene računalnike lahko dosežete s pomočjo:

- VNC (Virtual Network Computing) na `<http://www.uk.research.att.com/vnc>`. VNC ni strežnik X, vendar lahko prikaže namizje z oddaljenega sistema Unix na namizju Windows. VNC je najboljše orodje na tržišču za dostop do oddaljenih računalnikov, je zelo skromen s porabo procesorske moči in zelo zmogljiv program.

  Sledi seznam razpoložljivih strežnikov X:
- Najboljši, poceni in majhen (3 MB): pojdite na `<http://www.microimages.com>` in kliknite na "X-Server (MI/X) for Windows".
- Humming bird eXceed (14 MB): `<http://www.hummingbird.com>`
- Starnet (5.2 MB): `<http://www.starnet.com>`

  Na voljo je več kot 2 ducata dobaviteljev strežnikov X za Windows:
- X-win pro (6.34 MB): `<http://www.labf.com>`
- X-WinPro: `<http://lab-pro.com>`

- X-Link: <http://www.xlink.com/x.htm>
- Xoftware: <http://www.age.com> Strežniki X z univerz:
- Seznami univerz: <http://www.et.byu.edu/support/pc/xterm.html>
- Na disketi temelječ strežnik X: <http://mirriwinni.cse.rmit.edu.au/~brad/co338/sem1/floppy.html>

9.1 Odjemalci FTP za Windows 95/NT/2000

S skladišča CVS na Unixu/Linuxu lahko datoteke prenašate na MS Windows tudi preko protokola FTP:

- Pojdite na stran Tucows in iščite po ključni besedi "ftp toolsža MS Windows: <http://www.tucows.com>

9.2 Datoteke Visual Cafe(Java), JBuilder, MS Visual C++, HTML

Z uporabo Sambe in strežnika X je mogoče uprabljati CVS na sistemih Windows. CVS podpira tudi orodja, kot so Symantec Visual Cafe (Java), Inprise JBuilder, MS Visual C++ in druga.

Datoteke HTML lahko s pomočjo Sambe shranjujete v skladišče CVS in jih z lahkoto dosežete iz MS Windows.

9.3 Orodja za administriranje Sambe

Sambo administrirajte z orodji na naslovu <http://www.samba.org>. Pojdite tja in kliknite na "GUI Interfaces Tools".

10 Varnost skladišč CVS

Strežnik in skladišče CVS lahko zavarujete tako:

- Poženite CVS na samostojnem sistemu Linux/Unix, oglejte si še razdelek o 13 (izboljšanju učinkovitosti).
- Odstranite nepotrebne programske pakete z računalnika s CVS, da zunanjim vandalom preprečite, da bi jih pognali. Za vsak primer, če se jim že posreči vdreti v sistem, jim na ta način vsaj ne ponudite priložnosti, da bi pognali nevarne programe.
- Razmislite o uporabi SSH, kot je opisano v poglavju 11 (Skladišče za več uporabnikov).
- Razmislite o uporabi Kerberos, namestite paket cvs-*-kerberos*.rpm: <http://cvshome.org/dev/codelinux.html>.
- Obiščite <http://www.cvshome.org> in na naslovu *mailing list* <http://cvshome.org/communication.html> zastavite vprašanja o varnosti.

11 Večuporabniško skladišče CVS z dostopom z drugih računalnikov

Cederqvistov priročnik na naslovu `<http://cvshome.org/docs/manual/cvs_2.html>` opisuje, kako nastaviti CVS, da bo mogoče dostopati nanj z drugih računalnikov.

Da bo lahko CVS uporabljala skupina, je potrebno nastaviti sistem dovoljenj, ki bo dovoljeval ljudem dostop do sistema z drugih računalnikov. To lahko omogočite na tri načine in sicer z :server:, :pserver: ali :ext:. Mehanizem pserver in uporaba rsh nista varna. Samo :ext (z uporabo ssh) nam nudi dovolj varnosti.

Če spremenljivko CVS_RSH nastavite na SSH ali kakšen drug nadomestek rsh, **so lahko** navodila podobna tistim za '.rhosts', vendar prej preverite v dokumentaciji za vaš nadomestek za rsh.

Ssh dobite, če obiščete `<http://rpmfind.net>` in kot iskalni pojem vnesete "ssh". Lahko tudi obiščete `<http://www.redhat.com/apps/download>` in spet kot iskalni pojem vnesite "ssh". Prenesite in namestite ssh RPM in zatem nastavite CVS za uporabo z njim. Oglejte si tudi `<http://www.ssh.org>`.

Opomba: če nameravate nastaviti CVS za uporabo z rsh, potem MORATE narediti ta kritičen korak:

```
bash# chmod 6000 .rhosts
```

Oglejte si še skladišče CVS z JA-SIG UPortal: `<http://www.mis3.udel.edu/~jlaker/development>`.

12 Lupinski skripti za RCS

Če želite namesto CVS uporabljati RCS, si lahko pomagate z naslednjimi lupinskimi skripti.

12.1 cotree.sh

```
#!/bin/ksh

# cotree.sh (lupinski skript za prevzemanje drevesa)
# cotree.sh - Prevzame celotni imenik iz RCS

# Uporaba :
# Tako boste dobili vse imenike
#       unix> cotree.sh
#
# Tako pa boste dobili samo drevo iz enega imenika
#       unix> cotree.sh <ime imenika>

# Oglejte si tudi cofiles.sh

###############################################################
# Nastavljanje RCS (Revision Control System)
# Namestite si programe RCS - dobili boste ukaze co, ci, rcslog
# Naredite domači imenik za RCS, kamor boste odlagali datoteke
# iz skladišča izvorne kode. Recimo, da se imenuje
# $RCSDIR=/home/rcs_version_control
```

```
# Nastavite vrednost spremenljivki okolja RCSDIR=/home/rcs_version_control
# v datoteki $HOME/.profile. Npr.
#               export RCSDIR=/home/rcs_version_control
# Pod $RCSDIR naredite imeniško strukturo in vanjo vločite vse vaše
# datoteke z uporabo ci . Oglejte si ’man ci’
# Sedaj naredite povezavo iz vašega domačega imenika do vašega projekta
# pod $RCSDIR
#               cd $HOME
#               mkdir $HOME/myproject
#               cd $HOME/myproject
# in poženite ta skript, da boste dobili vse datoteke in drevo imenikov
#               cotree.sh
# Ta skript bo naredil celotno drevo izvorne kode v uporabnikov
# domači imenik in bo imel tudi mehko povezavo do imenikov RCS. Vsak
# uporabnik bo pognal ta skript v svojem domačem imeniku.
###############################################################

check_out_directory()
{
        # Korenski imenik RCS (revision control system)
        # npr.  RCSDIR=/home/rcs_version_control
        RCSDIR=$1
        DIRNAME=$2

        # Ime podanega imenika mora obstajati v korenskem imeniku RCS
        if [ "$DIRNAME" = "" -o  ! -d $RCSDIR/$DIRNAME ]; then
                print "\nImenik DIRNAME=$DIRNAME ne obstaja!!"
                print "\nPrekinjam program ... in končujem ...\n"
                exit
        fi

        mkdir -p $DIRNAME
        ln -s $RCSDIR/$DIRNAME/RCS $DIRNAME
        (
                cd $DIRNAME

                # To ne bo uspelo v primeru ime datoteke=vzorec,vv
                # kar bo znotraj RCS enako RCS/vzorec,vv,v
                # ls RCS | cut -d’,’ -f1 | xargs co
                # Uporabite ujemanje za konec imena $, kot je spodaj -
                # Uporabite ls RCS/*, da se znebite imen ./ in ../
                #ls RCS/* | cut -d’/’ -f2 | sed -e’s/,v$//g’ | xargs co
                if [ -d RCS ]; then
                        ls RCS/* | cut -d’/’ -f2 | sed -e’s/,v$//g’ | \
                        while read ii
                        do
                                #echo "ii je : $ii"
```

```
                                if [ -f "RCS/$ii,v" ]; then
                                        co $ii
                                fi
                        done
                fi
        )
}

# Korenski imenik (revision control system)
# npr. RCSDIR=/home/rcs_version_control
if [ "$RCSDIR" = "" -o ! -d $RCSDIR ]; then
        print "\nImenik RCSDIR=$RCSDIR ne obstaja!!"
        print "\nPrekinjam program ... in ga končujem...\n"
        exit
fi
#echo "RCSDIR je : $RCSDIR"

# Če je kot argument podano ime imenika, potem prevzemi vse
# datoteke samo za ta imenik in končaj.
if [ "$1" != "" ]; then
        (cd $RCSDIR; find $1 -type d -print ) |
        while read DIRNAME
        do
                #echo DIRNAME=$DIRNAME
                #DIRNAME=c_src
                #Korenski imenik RCS in ime imenika naj bosta relativna na rcs root dir
                tmpaa=` basename $DIRNAME `
                if [ "$tmpaa" != "RCS" ]; then
                        check_out_directory $RCSDIR $DIRNAME
                fi
        done
else
        (cd $RCSDIR; find * -type d -print ) |
        while read DIRNAME
        do
                echo DIRNAME=$DIRNAME
                #DIRNAME=c_src
                # Korenski imenik RCS in ime imenika naj bosta relativna na rcs root dir
                tmpaa=` basename $DIRNAME `
                if [ "$tmpaa" != "RCS" ]; then
                        check_out_directory $RCSDIR $DIRNAME
                fi
        done
fi
```

12.2 cofiles.sh

```
#!/bin/ksh

# cofiles.sh (Lupinski skript za prevzemanje datotek)
# cofiles.sh - Prevzemi vse datoteke v trenutnem imeniku iz RCS
# Oglejte si tudi cotree.sh in 'man rcsclean'

if [ ! -d RCS ]; then
        print "\nImenik RCS ne obstaja!!"
        print "\nPrekinjam program ... in ga končujem...\n"
        exit
fi

#echo "No. of args = " $# " and all args " $@

while true
do
        print -n "\n\nPrevzamem vse datoteke s pravicami branja in pisanja (read-write)? <d/n> [n]: "
        read ans
        if [ "$ans" = "" -o "$ans" = "n" -o "$ans" = "N" ]; then
                ans="N"
                break
        elif [ "$ans" = "d" -o "$ans" = "D" ]; then
                ans="D"
                break
        else
                print "\nNapačen vnos! Poskusite znova!!"
        fi
done
#echo "The ans is : " $ans

if [ $# -eq 0 ]; then
        # 'ls RCS' ne uspe v primeru ime_datoteke=vzorec,vv v RCS/vzorec,vv,v
        # ls RCS | cut -d',' -f1 | xargs co
        # Uporabite ujemanje, da se ime konča z $, kot je razvidno spodaj -
        if [ "$ans" = "D" ]; then
                ls RCS | sed -e's/,v$//g' | xargs co -l
        else
                ls RCS | sed -e's/,v$//g' | xargs co
        fi
elif [ $# -eq 1 ]; then
        if [ -f "RCS/$1,v" ]; then
                # V tem primeru bo $1 kot dbalter.sql
                # in ne kot db*.sql....
                #echo "One arg, no. of args = " $# " and all args " $@
                if [ "$ans" = "D" ]; then
                        co -l "$1"
                else
```

```
                        co "$1"
                fi
        else
                # V primeru, ko je $1=db*.sql in v trenutnem imeniku
                # ni nobenega db*.sql
                #echo "No files... no. of args = " $# " and all args " $@
                tmpaa="RCS/$1,v"  # bo kot RCS/db*.sql,v
                ls $tmpaa | \
                while read ii
                do
                        #echo "ii is : $ii"
                        if [ "$ans" = "D" ]; then
                                co -l "$ii"
                        else
                                co "$ii"
                        fi
                done
        fi
else
        for ii in $@
        do
                #echo "ii is : $ii,v"
                if [ "$ans" = "D" ]; then
                        co -l "$ii"
                else
                        co "$ii"
                fi
        done
fi
```

12.3 ciall.sh

```
#!/bin/ksh

# ciall.sh (Lupinski skript za vlaganje datotek)
# ciall.sh - Vloži vse datoteke v trenutnem imeniku v RCS
# Ta skript je zelo koristen za vlaganje ogromnega števila novih datotek
# v RCS. Prihrani čas, ker ni potrebno tipkati "opisa" za vsako datoteko.
# Za datoteke, ki so že v RCS, izvede običajen ukaz check-in.


# Pretvorba imen datotek v male črke:
# 'tr', see 'man tr'
#ls * | \
#while read ii
#do
#       jj=`echo $ii | tr [A-Z] [a-z] `
```

```
#       echo "ii is : $ii"
#       echo "jj is : $jj"
#       mv $ii $jj
#done

if [ ! -d RCS ]; then
        print "\nImenik RCS ne obstaja!!"
        print "\nImenik RCS bo narejen sedaj ...\n"
        mkdir RCS
fi

print "\n\nOpomba: To ni dnevniško sporočilo!"
print "Prosim, vnesite opis (uporabil se bo za"
print -n "vse vložene datoteke) : "
read description

# Izbira prune ne deluje, uporabi -maxdepth 0
#find * -prune -type f |

# Število podanih izbir je nič ali več....
if  [ $# -eq 0 ]; then
        listoffiles="*"
else
        listoffiles="$@"
fi

# Izbira prune ne deluje, uporabi -maxdepth 0
#find $listoffiles -prune -type f |
find $listoffiles -maxdepth 0 -type f |
while read ii
do
        #echo $ii
        if [ -f "RCS/$ii,v" ]; then
                #print "The file $ii already in RCS"
                ci -m"$description" $ii
        else
                #print "The file $ii is new file"
                ci $ii << EOF
$description
EOF
        fi
done
```

13 Izboljšanje učinkovitosti strežnika CVS

Za optimalno učinkovitost mora strežnik CVS teči na samostojnem računalniku z Linuxom/Unixom.

Da boste iztisnili čimveč procesorske moči, naredite naslednje:

- Ponovno prevedite Linuxovo jedro, da bo manjše in zanesljivejše. Odstranite dele, ki se ne uporabljajo. Oglejte si spis Kernel-HOWTO na naslovu `<http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html>`
- Izključite nepotrebne Unixove procese. Na sistemih z Linuxom/Unixom poženite chkconfig.

```
bash$ su - root
bash# man chkconfig
bash# chkconfig --help
bash# chkconfig --list | grep on | less
S seznama, ki ga dobite na zgornji način, odstranite procese, za katere ne
želite, da se zaženejo samodejno.
bash# chkconfig --level 0123456 <ime procesa> off
Ko naslednjič vključite računalnik, se ti procesi ne bodo več zagnali.
Sedaj vse procese, ki ste jih odstranile s seznama, odstranite še ročno.
bash# cd /etc/rc.d/init.d
bash# ./<ime procesa> stop
```

- Ne poganjajte ostalih nepotrebnih programov.
- Ne pustite sistema X Window, da teče brez potrebe, ker njegovi procesi zasedajo pomnilnik in prispevajo k obremenitvi procesorja. Lahko tudi predstavlja resno varnostno luknjo za zunanje napade. Ponavadi se uporabljajo okenski upravniki kot KDE, GNOME, CDE, XDM in drugi. Sistem X Window morate zapustiti takoj po uporabi in večino časa bi na računalniku s strežnikom CVS morali videti prijavni odzivnik v ukazni vrstici.

14 Sistem za sporočanje problemov

Skupaj s CVS boste morda želeli uporabljati sistem za sledenje projekta (angl. Project Tracking system) ali sistem za sporočanje problemov (angl. Problem Reporting system. Vsak programski projekt potebuje sistem za sporočanje problemov, kjer se sledijo hrošči in se jih posreduje različnim razvijalcem. Oglejte si projekt GNU gpl GNATS na naslovu `<http://www.gnu.org/software/gnats/gnats.html>` in `<http://dcl.sourceforge.net>`. Komercialni sistem za sporočanje problemov si lahko ogledate na `<http://www.stonekeep.com>`, poiščite še sistem za sledenje projekta.

15 Orodja za upravljanje s konfiguracijskami datotekami

Kaj je upravljanje konfiguracijskih datotek (angl. configuration management)?

Obstaja več različnih razlag. Nanaša se na sledenje in nadzor nad razvojem programske opreme ter s tem povezane aktivnosti. To pomeni upravljanje razvoja programskih projektov z vidiki kot so, da več razvijalcev istočasno dela z istimi deli kode, ki bo namenjena različnim končnim sistemom, podpora večim različicam in nadzor statusa kode (npr. preizkusne beta različice proti pravim različicam). Tudi v tem pogledu je več različnih razlag:

- Tradicionalen nadzor nad vlogami in prevzemi izvorne kode (in včasih tudi prevedenih programov) in možnost prevajanja entitet. Vključene so lahko tudi ostale funkcije.

- Upravljanje procesov - nadzor aktivnosti pri razvoju programske opreme. Na primer, možnost zagotoviti, da obstaja zahteva po spremembi, ki je bila odobrena in da so bile aktivnosti kot oblikovanje, dokumentacija in preizkus končane pred dovoljenjem za ponovno "vlaganje"kode.

Medtem ko sta upravljanje procesov in nadzor potrebna za ponavljajoč in optimiziran razvojni proces, so trdni temelji upravljanja s konfiguracijskimi datotekami nujni za ta proces.

Obiščite naslednje povezave:

- Odgovori na pogosto zastavljana vprašanja o orodjih za upravljanje s konfiguracijskimi datotekami: `<http://www.iac.honeywell.com/Pub/Tech/CM/CMFAQ.html>`
- Linux version control and configuration management tools `<http://linas.org/linux/cmvc.html>`
- Configuration Management systems `<http://www.cmtoday.com/yp/commercial.html>`
- Configuration Management Tools `<http://www.iac.honeywell.com/Pub/Tech/CM/CMTools.html>`
- DevGuy CVS config mgmt `<http://devguy.com/fp/cfgmgmt/cvs>`
- *Yahoo category site* `<http://dir.yahoo.com/Computers_and_Internet/software/programming_tools/software_engineering/configuration_management/cvs___concurrent_versions_system>`
- Free config mgmt tool `<http://www.canb.auug.org.au/~millerp/aegis/aegis.html>`
- Free CM tools `<http://www.loria.fr/cgi-bin/molli/cm/wilma/fcmt>`
- Rational ClearCase tool `<http://www.rational.com/products/clearcase/prodinfo.jsp>`

16 Podobne strani

Podobne strani lahko najdete na:

- Linux goodies `<http://www.aldev.8m.com>` in zrcalnih straneh: `<http://aldev0.webjump.com>`, *angelfire* `<http://www.angelfire.com/country/aldev0>`, *geocities* `<http://www.geocities.com/alavoor/index.html>`, *virtualave* `<http://aldev0.virtualave.net>`, *50megs* `<http://aldev0.50megs.com>`, *theglobe* `<http://members.theglobe.com/aldev1/index.html>`, *NBCi* `<http://members.nbci.com/alavoor>`, *Terrashare* `<http://aldev.terrashare.com>`, *Fortunecity* `<http://members.fortunecity.com/aldev>`, *Freewebsites* `<http://aldev.freewebsites.com>`, *Tripod* `<http://members.tripod.lycos.com/aldev>`, *Spree* `<http://members.spree.com/technology/aldev>`, *Escalix* `<http://www.escalix.com/freepage/aldev>`, *Httpcity* `<http://www.httpcity.com/aldev/index.html>`, *Freeservers* `<http://aldev.freeservers.com>`.
- CVS Bubbles `<http://www.loria.fr/~molli/cvs-index.html>`
- CSSC (SCCS like system) `<http://cssc.sourceforge.net>` in *zrcalni strani* `<ftp://alpha.gnu.org/pub/gnu/CSSC>`
- SCCS za Linux `<http://www.bitmover.com/bitkeeper>`

17 SCCS proti CVS-RCS

SCCS (angl. Source Code Control System, sistem za nadzor izvorne kode) se nič več ne razvija in izboljšuje. Splošno mnenje je, da je to orodje površno in ni primerno za veliko število uporabnikov, ki delajo na enem projektu. Pravzaprav SCCS prepleta vse različice, vendar lahko **občutno upočasni**. razvoj novih različic. Zatorej se sistem SCCS NE priporoča za začetek novih projektov. Kljub temu ga tu in tam še vedno srečamo zaradi podpore starejšim bazam izvorne kode v SCCS.

RCS (angl. Revision Control System, sistem za nadzor revizij) je po pogostnem mnenju boljši od SCCS. En vzrok tiči v dejstvu, da se hrani najnovejša različica in razlike za starejše različice glede na najnovejšo, zaradi česar je razvoj novih različic hitrejši. Dodatne razprave na temo SCCS proti RCS lahko najdete na naslovu `<http://www.faqs.org/faqs/unix-faq/faq/part7>`.

Upoštevajte, da se je RCS učil na napakah SCCS...

CVS, ki za delovanje zahteva RCS, razširi RCS tako, da nadzira hkratno urejanje izvorne kode večih uporabnikov, ki delajo na izdajah, zgrajenih iz hierarhičnih množic imenikov. "Če je RCS zbirni jezik, je CVS Pascal".

18 Ostali zapisi tega dokumenta

Ta dokument je izdan v 14 različnih zapisih in sicer: DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF(Rich Text Format), tekst, Unixove priročniške strani, datoteka HTML file, SGML (zapis Linuxdoc), SGML (zapis Docbook) in zapis MS WinHelp.

Ta dokument se nahaja na naslovu:

- `<http://www.linuxdoc.org>`, kliknite na HOWTOs in ga poiščite med dokumenti howto s pomocjo tipk CTRL+f ali ALT+f znotraj vašega brskalnika.

Ta dokument lahko najdete tudi na naslednjih naslovih:

- `<http://www.caldera.com/LDP/HOWTO>`
- `<http://www.linux.ucla.edu/LDP>`
- `<http://www.cc.gatech.edu/linux/LDP>`
- `<http://www.redhat.com/mirrors/LDP>`
- Ostale zrcalne strani v vaši bližini lahko najdete na `<http://www.linuxdoc.org/mirrors.html>`, kjer izberete najbližjo stran in greste v imenik /LDP/HOWTO/xxxxx-HOWTO.html.

- Ta dokument lahko dobite tudi v eni arhivski datoteki (tar) z zapisi HTML, DVI, Postscript ali SGML na naslovih: `<ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO/other-formats/>` in `<http://www.linuxdoc.org/docs.html>`.
- Dokument zapisan v golem tekstu najdete na: `<ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO>` in `<http://www.linuxdoc.org/docs.html>`.
- Celoten dokument v eni datoteki HTML se nahaja na: `<http://www.linuxdoc.org/docs.html>`. Enovito datoteko HTML dobite (glejte še man sgml2html) z ukazom: sgml2html -split 0 xxxxhowto.sgml

- Prevode dokumenta v druge jezike, npr. francoščino, nemščino, španščino, kitajščino in japonščino najdete na: <ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO> in <http://www.linuxdoc.org/docs.html> Vsakršna vaša pomoč pri prevajanju v druge jezike bo dobrodošla.

Dokument je napisan z uporabo orodij "SGML-Tools", ki jih lahko dobite na naslovu: <http://www.sgmltools.org>. Po prevajanju izvorne kode boste dobili množico ukazov, npr.:

- sgml2html xxxxhowto.sgml (za izdelavo datotke HTML)
- sgml2html -split 0 xxxxhowto.sgml (za izdelavo enovite datoteke HTML)
- sgml2rtf xxxxhowto.sgml (za izdelavo datoteke RTF)
- sgml2latex xxxxhowto.sgml (za izdelavo datoteke Latex)

18.1 Zapis Acrobat PDF

Datoteka v zapisu PDF se lahko izdela iz zapisa v Postscriptu z uporabo bodisi programa Acrobat **distill** bodisi z **Ghostscriptom**. Datoteko v zapisu Postscript izdelate iz zapisa DVI, le tega pa iz zapisa v Latexu. Programsko opremo Distill lahko poberete z naslova <http://www.adobe.com>. Spodaj je podan vzorčni postopek za izdelavo datoteke v zapisu PDF:

```
bash$ man sgml2latex
bash$ sgml2latex dokument.sgml
bash$ man dvips
bash$ dvips -o dokument.ps dokument.dvi
bash$ distill dokument.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf vhod.ps izhod.pdf
bash$ acroread izhod.pdf &
```

Lahko pa uporabite Ghostscriptov ukaz **Ps2pdf**. Ps2pdf je nekakšen nadomestek s skoraj vso funkcionalnostjo izdelka Adobe Acrobat Distiller: pretvarja datoteke v Postscriptu v zapis Portable Document Format (PDF). **Ps2pdf** je izveden kot zelo kratek ukazni skript (paketna datoteka), ki kliče Ghostscript in izbere posebno "izhodno napravož imenom **pdfwrite**. Da bo Ps2pdf deloval, mora biti "naprava"pdfwrite ob prevajanju Ghostscripta vključena v datoteko Makefile; glede podrobnosti si oglejte dokumentacijo o postavljanju Ghostscripta.

18.2 Pretvorba zapisa Linuxdoc v Docbook

Ta dokument je napisan v zapisu Linuxdoc SGML. Zapis Docbook SGML je njegov naslednik in ima precej več značilnosti kot Linuxdoc. Linuxdoc je zelo enostaven za uporabo. Če želite pretvoriti zapis Linuxdoc SGML v Docbook SGML, uporabite program **ld2db.sh** in nekaj skriptov v Perlu. Izpis programa Ld2db ni 100% čist, zato boste potrebovali skript **clean_ld2db.pl**, napisan v Perlu. Lahko se zgodi, da boste morali nekaj vrstic v dokumentu popraviti ročno.

- Poberite program Ld2db z naslova <http://www.dcs.gla.ac.uk/~rrt/docbook.html> ali s strani *Al Dev site* <http://www.aldev.8m.com/cppsrc.html>

- Poberite skript v Perlu cleanup_ld2db.pl s strani *Al Dev site* <http://www.aldev.8m.com/cppsrc.html>

Programcek ld2db.sh ni 100% čist, zato bo ob njegovem zagonu prišlo do nekaterih napak.

```
bash$ ld2db.sh file-linuxdoc.sgml db.sgml
bash$ cleanup.pl db.sgml > db_clean.sgml
bash$ gvim db_clean.sgml
bash$ docbook2html db.sgml
```

Po zagonu skripta v Perlu boste morda morali ročno popraviti nekaj manjših napak. Na primer, za vsak < Listitem> boste morali vnesti zapiralmo oznako < /Para>.

18.3 Pretvorba v zapis MS WinHelp

Dokument v zapisu SGML lahko pretvorite v zapis Microsoft Windows Help. Najprej pretvorite SGML v HTML:

```
bash$ sgml2html xxxxhowto.sgml     (izdelava datotek HTML)
bash$ sgml2html -split 0   xxxxhowto.sgml (izdelava enovite datoteke HTML)
```

Zatem uporabite pripomoček *HtmlToHlp* <http://javadocs.planetmirror.com/htmltohlpe.html>. Lahko pa tudi izvedete Sgml2rtf in zatem zapis RTF pretvorite v winhelp.

18.4 Ogled različnih zapisov

Če si želite ogledati dokument v zapisu DVI, uporabite program Xdvi, ki ga dobite v paketu tetex-xdvi*.rpm (Redhat Linux) in najdete v meniju ControlPanel | Applications | Publishing | TeX. Dokument v zapisu DVI lahko berete z ukazom:

```
xdvi -geometry 80x90 howto.dvi
man xdvi
```

Velikost okna lahko spreminjate z miško. Po dokumentu se pomikate s smerniškimi tipkami, Page Up/Down, lahko pa tudi s tipkami 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' za pomik gor, navzdol, na sredino, naslednjo oz. prejšnjo stran ipd. S tipko 'x' izklopite meni.

Datoteko v zapisu Postscript si lahko ogledate s programoma 'Gv' (Ghostview) ali 'Ghostscript'. Program 'Ghostscript' se nahaja v paketu ghostscript*.rpm Gv pa v paketu gv*.rpm (Redhat Linux), do njiju pa pridete preko menijev ControlPanel | Applications | Graphics. Program Gv je uporabniku precej prijaznejši kot Ghostscript. Oba programa sta na voljo tudi za druge operacijske sisteme, npr. OS/2, Windows 95 in NT. Ta dokument si lahko ogledate tudi na omenjenih operacijskih sistemih.

- Ghostscript za Windows 95, OS/2, in vse druge operacijske sisteme dobite na naslovu <http://www.cs.wisc.edu/~ghost>.

Dokument v zapisu Postscript si lahko ogledate z ukazom:

```
gv howto.ps
ghostscript howto.ps
```

Dokument v zapisu HTML si lahko ogledate s pomočjo Netscape Navigatorja, Microsoft Internet Explorerja, Redhat Baron Web browserja ali kateregekoli izmed preostalih 10 spletnih brskalnikov.

Dokumente v zapisih Latex in LyX si lahko ogledate s programom LyX - vmesnikom za LaTeX v okolju X Window.

19 Avtorske pravice in licenca

20 sget

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program sget
# Program za prevzem datoteke v načinu "samo za branje"iz CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
        print "\nUporaba: $cmdname [-r stevilka_revizije/simbolicno_ime_oznake] <ime datoteke/imenika> "
        print "Moznost -r je neobvezna "
        print "Na primer - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
```

```
        print " $cmdname nek_imenik "
        print "Izvlece po simbolicni oznaki revizije kot - "
        print " $cmdname -r REVISION_1 some_directory "
        print " "
        exit
}

# Ukaz getopt ne bo vec podprt v naslednji izdaji.
# Namesto njega uporabljamo getopts.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
fi

bkextn=sget_bak

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
        fname=$1
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
```

```
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# Preveri, ali datoteka ze obstaja....
if [ -f "$HOME/$subdir/$fname" ]; then
        tmpaa="$HOME/$subdir/$fname"
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b3-3 `
        group_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b6-6 `
        other_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                        -o "$other_perms" = "w" ]; then
                print "\nNapaka: Datoteko je mogoce spreminjati. Opuscam $cmdname ......"
                print "         Naredite rezervno kopijo, izvedite scommit ali izbrisite datoteko in"
                print "         poskusite s $cmdname znova.\n"
                exit
        fi
fi

# Prestavi datoteko
mkdir -p "$HOME/$subdir"
touch "$HOME/$subdir/$fname" 2>/dev/null
\mv -f "$HOME/$subdir/$fname" "$HOME/$subdir/$fname.$bkextn"

# Naredi podlupino
(
        cd $homedir

        # Uporabi izbiro -A za izbris vseh lepljivih zastavic
        if [ "$FLAG1" = "" ]; then
                if [ "$subdir" = "." ]; then  # ne uporabi pike, ker bo zmedla CVS
                        cvs -r checkout -A $fname
                else
                        cvs -r checkout -A "$subdir/$fname"
                fi
        else
                if [ "$subdir" = "." ]; then  # ne uporabi pike, ker bo zmedla CVS
```

```
                        cvs -r checkout -A -$FLAG1 $OARG1 $fname
                else
                        cvs -r checkout -A -$FLAG1 $OARG1 "$subdir/$fname"
                fi
        fi
)
#pwd

if [ -f "$HOME/$subdir/$fname" ]; then
        print "\nPridobljena kopija datoteke $subdir/$fname samo za branje."
        print "Dokoncan $cmdname"
        #print "\nNasvet: $cmdname <ime datoteke/imenika> \n"
fi
```

21 sedit

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh
# CVS program sedit
# Program za prevzem datotek iz CVS v načinu branje in pisanje z zaklepanjem

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
#       print "\nUporaba: $cmdname [-r stevilka_revizije] [-F] <ime_datoteke>"
#       print "Izbiri -r, -F sta neobvezni"
#       print "Izbira -F prisili k urejanju tudi, ce je datoteko "
#       print "nekdo ze zaklenil"

        print "\nUporaba: $cmdname [-r stevilka_revizije] <ime_datoteke>"
```

```
        print "Izbira -r je neobvezna "

        print "Na primer - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
#       print " $cmdname -F foo.cpp "
        print " "
}

# Ukaz getopt ne bo vec podprt v naslednji izdaji.
# Namesto njega uporabimo getopts.
#while getopts r:F ii
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
#       F) FLAG2=$ii; OARG2="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
        exit
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

bkextn=sedit_bak

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
```

```
        fname=$1
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# Ce je datoteko ze kdo prevzel....
cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nNapaka: spremenljivka \$CVSROOT ni nastavljena!!\n"
        exit
fi
mkdir -p "$CVSROOT/$subdir/Locks" 2>/dev/null

if [ ! -e "$CVSROOT/$subdir/$fname,v" ]; then
        print "\nNapaka: datoteke $fname ni v skladiscu CVS!!\n"
        exit
fi

# Imenik CVS v vasem lokalnem imeniku je nujno potreben za delovanje vseh ukazov..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi izdelati CVS in v lokalnem imeniku s sget.
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
```

```
                        fi
                )
        fi
fi

# Dobi stevilko revizije datoteke....
# Uporabi tmpfile ker argumentov ni mogoce nastavljati znotraj podlupine.
tmpfile=$homedir/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
        (
        cd $homedir
                if [ "$subdir" = "." ]; then  # ne uporabi pike, ki lahko zmede CVS
                        cvs log $fname | head -6 | grep head: | awk '{print $2}' > $tmpfile
                else
                        cvs log "$subdir/$fname" | head -6 | grep head: | awk '{print $2}' > $tmpfile
                fi
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

lockfile="$CVSROOT/$subdir/Locks/$fname-$OARG1"
#echo "lockfile je : " $lockfile
#if [ -e $lockfile -a "$FLAG2" = "" ]; then
if [ -e $lockfile ]; then
        print "\nNapaka: Datoteka $fname revizije $OARG1 je ze zaklenjena!!"
        aa=` ls -l $lockfile | awk '{print "Zaklenil jo je " $3}' `
        print $aa
        print "Ta razvijalec naj izvede scommit ALI sunlock, da jo bo odklenil."
        print " "
#       print "Uporabite lahko tudi izbiro -F, da prisilite datoteko k urejanju,"
#       print "tudi ce jo je kdo drug zaklenil. Toda najprej se morate"
#       print "z njim pogovoriti, da lahko z datoteko delata hkrati."
#       print "Na primer - ta moznost je koristna, ce delate na loceni"
#       print "funkctiji v jeziku C++ v datoteki, ki ne moti ostalih"
#       print "razvijalcev."
#       print " "
        exit
fi

# Sedaj dobi kopijo samo za branje....
if [ ! -e "$HOME/$subdir/$fname" ]; then
        (
        cd $homedir
                if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                        cvs -r checkout $fname 1>/dev/null
```

```
                else
                        cvs -r checkout "$subdir/$fname" 1>/dev/null
                fi
        )
fi

# Preveri, ali datoteka ze obstaja....
tmpaa="$HOME/$subdir/$fname"
if [ -f $tmpaa ]; then
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b3-3 `
        group_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b6-6 `
        other_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                        -o "$other_perms" = "w" ]; then
                print "\nNapaka: datoteko je mogoce spreminjati. Opuscam $cmdname ......"
                print "         Narediti morate rezervno kopijo, izvesti scommit ali zbrisati datoteko in"
                print "         poskusiti s $cmdname znova.\n"
                exit
        fi
        #print "\nOpomba: datoteka $tmpaa je samo za branje."
        #print "Zato jo prestavljam v $tmpaa.$bkextn ....\n"
        \mv -f $tmpaa $tmpaa.$bkextn
        chmod 444 $tmpaa.$bkextn
elif [ -d $tmpaa ]; then
        print "\nNapaka: $tmpaa je imenik in NE datoteka. Opuscam $cmdname ....\n"
        exit
fi

# Naredi podlupino
print "\nSedaj dobivam datoteko $fname iz skladisca CVS ...\n"
(
        cd $homedir
        # Uporabljam izbiro -A, da zbrisem lepljivo oznako in da dobim
        # razlicico glavne revizije
        if [ "$FLAG1" = "" ]; then
                if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                        cvs -w checkout -A $fname
                else
                        cvs -w checkout -A "$subdir/$fname"
                fi
        else
                if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                        cvs -w checkout -A -$FLAG1 $OARG1 $fname
                else
```

```
                        cvs -w checkout -A -$FLAG1 $OARG1 "$subdir/$fname"
                fi
        fi
)

if [ -e "$HOME/$subdir/$fname" ]; then
        # Zaklenjena datoteka je $CVSROOT/$subdir/Locks/$fname-$OARG1
        touch $lockfile
        if [ -e $lockfile ]; then
                print "\nKoncan $cmdname."
        else
                print "\nUsodna napaka: datoteka $fname revizije $OARG1 ni zaklenjena !!"
                print "\nPreverite vzrok za to napako.. pred nadaljevanjem..."
        fi
fi

#pwd

#print "\nNasvet: $cmdname <ime_datoteke> \n"
```

22 scommit

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh
# CVS program scommit
# Program za udejanjenje sprememb in vnos datotek v CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
```

```
        print "\nUporaba: $cmdname [-r stevilka_revizije] <ime_datoteke>"
        print "Moznost -r je neobvezna"
        print "Na primer - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
        print " "
}

# Ukaz getopt v naslednji izdaji ne bo vec podprt.
# Namesto njega uporabljam getopts.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
        exit 2
fi

if [ -d $1 ]; then
        Usage
        exit 2
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

# Poisci podimenik
cur_dir=`pwd`
#echo $cur_dir
len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
```

```
        fname=$1
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
# echo "subdir je : " $subdir
# echo "fname je : " $fname

# Ce je datoteko ze kdo prevzel....
cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nNapaka: spremenljivka \$CVSROOT ni nastavljena!!\n"
        exit
fi
mkdir -p "$CVSROOT/$subdir/Locks" 2>/dev/null

# pri vseh ukazih je zahtevan imenik CVS v vasem lokalnem imeniku...
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Najprej morate izvesti sget na imeniku `basename $subdir`. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi izdelati CVS v lokalnem imeniku s sget
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi
```

```
# Ugotovi stevilko delovne revizije za datoteko....
# Uporaba tmpfile kot argumenta se ne more nastaviti znotraj podlupine
tmpfile=$homedir/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
        (
        cd $homedir
                if [ "$subdir" = "." ]; then  # ne uporabljaj pike, ki lahko zmede CVS
                        cvs status $fname 2>/dev/null | grep "Working revision:" | awk '{print $3}' >$tmpfile
                else
                        cvs status "$subdir/$fname" 2>/dev/null | grep "Working revision:" | awk '{print $3}' >$tmpfile
                fi
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

if [ "$OARG1" = "" ]; then
        print "Datoteka $subdir/$fname je NOVA, zato je ni v skladiscu CVS."
else
        lockfile="$CVSROOT/$subdir/Locks/$fname-$OARG1"

        if [ -e $lockfile ]; then
                # Preveri, ali revizija pripada vam...
                aa=` ls -l $lockfile | awk '{print $3}' `
                userid=`id | cut -d'(' -f2 | cut -d')' -f1 `
                if [ "$aa" != "$userid" ]; then
                        print " "
                        print "Datoteke $subdir/$fname NISTE zaklenili vi!!"
                        print "Zaklenil jo je uporabnik $aa , vase uporabnisko ime pa je $userid ."
#                       print "Ce delate socasno z drugim razvijalcem"
#                       print "in ste pri sedit uporabili izbiro -F,"
                        print "morate pocakati, da drug razvijalec izvede scommit"
                        print "ali sunlock."
                        print "Opuscam $cmdname ...."
                        print " "
                        exit 2
                fi
        else
                # Datoteka mora obstajati v CVS
                if [ -f "$CVSROOT/$subdir/$fname,v" ]; then
                        print "Datoteke $subdir/$fname niste zaklenili s sedit!!"
                        print "Opuscam $cmdname ...."
                        exit 2
                else
                        print "\nDatoteke $subdir/$fname se ni v skladiscu CVS!!"
                        print "Na datoteki $subdir/$fname bi morali izvesti sadd...."
```

```
                        exit 2
                fi
        fi
fi

# Deluj znotraj podlupine - in deluj iz korenskega imenika
(
        cd $homedir

        # Za zadaj ne dovoli udejanjanj imenikov ...
        #if [ -d "$subdir/$fname" ]; then
        #       cvs commit "$subdir/$fname"
        #fi

        if [ "$subdir" = "." ]; then  # ne uporabljaj oike, ki lahko zmede CVS
                cvs commit $fname
        else
                cvs commit "$subdir/$fname"
        fi
        exit_status=$?

        if [ $exit_status -eq 0 ]; then
                lockfile="$CVSROOT/$subdir/Locks/$fname-$OARG1"
                if [ -e $lockfile ]; then
                        \rm -f $lockfile
                fi

                # Potrebno je spremeniti dovoljenja datoteke za primer,
                # ko ni sprememb datoteke
                chmod a-w "$HOME/$subdir/$fname"
                print "\nKoncan $cmdname. $cmdname uspesen."
                #print "\nNasvet: $cmdname <ime datoteke/imenika>\n"
        fi
)
```

23 supdate

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program supdate
# Program za posodobitev datoteke iz CVS v nacinu beri/pisi
```

```
# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUporaba: $cmdname <ime_datoteke>"
        exit
fi

# Zaradi zascite presledkov v $1 uvedi dvojne narekovaje
tmpaa="$1"

# Preveri, ali datoteka ze obstaja....
if [ $# -gt 0 -a  -f $tmpaa ]; then
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b3-3 `
        group_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b6-6 `
        other_perms=`ls -l $tmpaa | awk '{print $tmpaa }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                        -o "$other_perms" = "w" ]; then
                while :
                do
                        print "\n$cmdname bo naredil rezervno kopijo delovne datoteke"
                        print "$tmpaa v $tmpaa.supdate_bak pred kakrsnimkoli spajanjem."
                        print "Ali ste prepricani, da zelite spojiti spremebe iz"
                        print -n "skladisca CVS v vaso delovno datoteko? <d/n> [n]: "
                        read ans
                        if [ "$ans" = "d" -o "$ans" = "D" ]; then
                                if [ -f $tmpaa.supdate_bak ]; then
                                        print "\nOpozorilo: datoteka $tmpaa.supdate_bak ze obstaja!!"
                                        print "Preglejte datoteko $tmpaa.supdate_bak in jo izbrisite."
                                        print "Zatem poskusite znova s $cmdname "
                                        print "Opuscam $cmdname ...."
                                        exit
                                else
```

```
                                        cp $tmpaa $tmpaa.supdate_bak
                                        break
                                fi
                        elif [ "$ans" = "n" -o "$ans" = "N" -o "$ans" = "" -o "$ans" = " " ]; then
                                exit
                        fi
                done
        fi
fi

if [ -d $tmpaa ]; then
        print "\nPosodobitev imenika je onemogocena, ker cvs update"
        print "spaja spremebe iz skladisca v delovni imenik."
        print "Zato podajte ime datoteke za posodobitev - kot je prikazano spodaj: "
        print " Uporaba: $cmdname <ime_datoteke>"
        exit
#       cvs update
else
        cvs update $tmpaa
fi

print "\nKoncan $cmdname. $cmdname uspesen."
print "\n\nRezervna kopija prvotne datoteke je v $tmpaa.supdate_bak ."
print "\nPrvotna datoteka je tako SHRANJENA v $tmpaa.supdate_bak ."
print "\n\n"
#print "\nNasvet: $cmdname <ime datoteke/imenika>\n"
```

---

24 sunlock

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh
# CVS program sunlock
# Program za odklepanje datotek, za sprostitev ključavnice, ki jo naredi sedit

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp
```

```
# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
        print "\nUporaba: $cmdname [-r stevilka_revizije] <ime_datoteke>"
        print " Izbira -r je neobvezna"
        print "Na primer - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
        print " "
}

# Ukaz getopt v naslednji izdaji ne bo vec podprt.
# Namesto njega uporabljam getopts.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

if [ $# -lt 1 ]; then
        Usage
        exit
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir
tmpaa=`dirname $1`
```

```
if [ "$tmpaa" = "." ]; then
        fname=$1
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nNapaka: spremenljivka \$CVSROOT ni nastavljena!!\n"
        exit
fi

if [ ! -e "$CVSROOT/$subdir/$fname,v" ]; then
        print "\nNapaka: datoteke $fname ni v skladiscu CVS!!\n"
        exit
fi

# imenik CVS v vasem lokalnem imeniku je potreben pri vseh ukazih..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja"
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj s pomocjo sget poskusi izdelati CVS
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ki lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
```

```
                )
        fi
fi

# Ugotovi stevilko zadnje revizije datoteke....
# Uporabi tmpfile ker argumentov ni mogoce nastavljati v podlupini
tmpfile=$homedir/sunlock-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
        # Deluj znotraj podlupine - iz korenskega imenika
        (
        cd $homedir
                if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ki lahko zmedejo CVS
                        cvs log $fname | head -6 | grep head: | awk '{print $2}' > $tmpfile
                else
                        cvs log "$subdir/$fname" | head -6 | grep head: | awk '{print $2}' > $tmpfile
                fi
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

lockfile="$CVSROOT/$subdir/Locks/$fname-$OARG1"
#echo lockfile is : $lockfile
if [ ! -e $lockfile ]; then
        print "\nDatoteke $fname revizije $OARG1 NI zaklenil nihce."
        print " "
        exit
fi

ans=""
while :
do
        print "\n\n********************************************************"
        print "POZOR: $cmdname bo sprostil kljucavnico in omogocil ostalim"
        print "       razvijalcem urejanje datoteke. Svetujemo, da"
        print "       shranite spremembe z ukazom scommit."
        print "*********************************************************"
        print -n "\nAli ste prepricani, da zelite odkleniti datoteko <d/n>? [n]: "
        read ans
        if [ "$ans" = "" -o "$ans" = " " -o "$ans" = "n" -o "$ans" = "N" ]; then
                print "\nOpuscam $cmdname ...."
                exit
        fi
        if [ "$ans" = "d" -o "$ans" = "D" ]; then
                print "\n\n\n\n\n "
                print "POZOR: izgubite lahko vse spremebe, ki ste jih naredili v datoteki!!"
```

```
                print -n "Ali ste prepricani? Ali res zelite odkleniti datoteko <d/n>? [n]: "
                read ans
                if [ "$ans" = "d" -o "$ans" = "D" ]; then
                        break
                elif [ "$ans" = "" -o "$ans" = " " -o "$ans" = "n" -o "$ans" = "N" ]; then
                        exit
                else
                        print "\n\nNapacen vnos. Poskusite znova..."
                        sleep 1
                fi
        else
                print "\n\nNapacen vnos. Poskusite znova..."
                sleep 1
        fi
done

if [ -e $lockfile ]; then
        \rm -f $lockfile
        print "\nKoncan $cmdname ."
else
        print "\nDatoteka $fname NI zaklenjena."
        print " "
fi
```

---

25 slist

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program slist
# Program za izpis vseh datotek iz CVS, ki se urejajo

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
```

```
#       ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

# Uporaba:
#               $ slist      (Vse datoteke in podimeniki)
#               $ slist *.*      (Vse datoteke)
#               $ slist *      (Vse datoteke in podimeniki)
#               $ slist ab*      (Vse datoteke, i se zacnejo z ab)

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nPozor: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir

# Ce je datoteko ze kdo prevzel....
cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nNapaka: spremenljivka \$CVSROOT ni nastavljena!!\n"
        exit
fi

# Ce trenutnega imeniskega drevesa ni v korenskem imeniku CVS, koncaj
if [ ! -d $CVSROOT/$subdir ]; then
        print "\nImenika $subdir ni bilo mogoce najti v $CVSROOT"
        exit
fi

#echo "st. parametrov : " $#
#echo "Argument $ 1 je : " $1
#echo "vsi argumenti : " $@

if [ $# -eq 0 ]; then
        #tmpbb=` find * -prune -type d `
        tmpbb=` find * -maxdepth 0 -type d `
elif [ $# -eq 1 ]; then
        if [ "$1" = "." ]; then
                #tmpbb=` find * -prune -type d `
```

```
                tmpbb=` find * -maxdepth 0 -type d `
        else
                if [ -d $1 -a ! -d $CVSROOT/$subdir/$1 ]; then
                        print "\nImenika $subdir/$1 ni v $CVSROOT"
                        exit
                fi
                tmpbb=$@
        fi
else
        tmpbb=$@
fi

#echo "tmpbb je : " $tmpbb

# Sedaj odstrani vsa imena imenikov, ki jih ni v korenskem imeniku CVS
dirnames=""
for ii in $tmpbb ; do
        if [ -d $CVSROOT/$subdir/$ii ]; then
                dirnames="$dirnames $ii "
        fi
done
#echo "dirnames je : " $dirnames

if [ "$dirnames" != "" ]; then
        find $dirnames  -type f |
        while read ii
        do
                # Prikazi samo tiste datoteke, ki so v sistemu CVS
                if [ -f "$CVSROOT/$subdir/$ii,v" ]; then
                        #echo "ii is : " $ii
                        ls -l $ii | grep ^\-rw
                fi
        done;
fi

# Ugotovi vse datoteke v trenutnem imeniku
listfiles=`ls $tmpbb `
# Izbira prune ne deluje, zato uporabi maxdepth
#find * -prune -type f |
find * -maxdepth 0 -type f |
while read ii
do
        for jj in $listfiles ; do
                if [ "$jj" = "$ii" ]; then
                        # Prikazi samo datoteke, ki so v sistemu CVS
                        if [ -f "$CVSROOT/$subdir/$ii,v" ]; then
                                #echo "ii je : " $ii
```

```
                                ls -l $ii | grep ^\-rw
                        fi
                fi
        done
done;
```

26 sinfo

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program sinfo
# Program za pridobitev statusa datotek v delovnem imeniku

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUporaba: $cmdname [ime datoteke/imenika] "
        print "Na primer - "
        print " $cmdname foo.cpp"
        print " $cmdname nek_imenik "
        print " "
        exit
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi
```

```
cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
        fname=$1
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# V lokalnem imeniku se od vseh ukazov zahteva imenik CVS..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja"
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj s pomocjo sget poskusi narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ki lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
```

```
                )
        fi
fi

# Naredi podlupino
if [ -f $1 ]; then
        (
        cd $homedir
                clear
                print "\nCVS status je : "
        cvs status "$subdir/$fname"
        )
elif [ -d $1 ]; then
        (
                cd $homedir
                clear
                print "\nCVS status je : "
                tmpfile="$homedir/cvs_sinfo.tmp"
                rm -f $tmpfile
        echo "  " >> $tmpfile
        echo "  ***************************************" >> $tmpfile
        echo "        Skupni status imenika" >> $tmpfile
        echo "  ***************************************" >> $tmpfile
        cvs release "$subdir/$fname" 1>>$tmpfile 2>>$tmpfile << EOF
N
EOF
        echo "\n   -------------------------------\n" >> $tmpfile

        aa=`cat $tmpfile | grep ^"M " | awk '{print $2}' `
        for ii in $aa
        do
                        jj="(cd $homedir; cvs status \"$subdir/$ii\" );"
                        echo $jj | /bin/sh  \
                                        | grep -v Sticky | awk '{if (NF != 0) print $0}' \
                                        1>>$tmpfile 2>>$tmpfile
        done

        cat $tmpfile | grep -v ^? | grep -v "Are you sure you want to release" \
        | less
        rm -f $tmpfile
        )
else
        print "\nArgument $1 ni datoteka ali imenik."
        exit
fi
```

27 slog

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program slog
# Program za izpis zgodovine datoteke v CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUporaba: $cmdname <ime_datoteke> \n"
        exit
fi

# Preveri, ali datoteka ze obstaja....
if [ ! -f $1 ]; then
        print "\nNapaka: $1 NI datoteka. Opuscam $cmdname ......"
        exit
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
```

```
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir is : " $subdir
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
        fname="$1"
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# V lokalnem imeniku se zahteva imenik CVS, da bodo ukazi lahko delovali..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi s sget narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ker lahko zmedejo sistem
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi

# Deluj znotraj podlupine
(
```

```
        cd $homedir
        cvs log "$homedir/$subdir/$fname" | less
)

print "\nKoncan $cmdname. $cmdname uspesen"
#print "\nNasvet: $cmdname <ime_datoteke>\n"
```

28 sdif

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program sdif
# Program za prikaz razlik med delovno datoteko in kopijo v CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
        print "\nUporaba: $cmdname <ime_datoteke> "
        print "$cmdname -r<revizija1> -r<revizija2> <ime_datoteke> \n"
        exit
}

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

FLAG1=""
```

```
FLAG2=""
OARG1=""
OARG2=""
# Ukaz getopt v naslednji izdaji ne bo vec podprt.
# Namesto njega uporabljam getopts.
while getopts r:r: ii
do
        case $ii in
        r)
                if [ "$FLAG1" = "" ]; then
                        FLAG1=$ii;
                        OARG1="$OPTARG"
                else
                        FLAG2=$ii;
                        OARG2="$OPTARG"
                fi
                ;;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

if [ "$FLAG2" = "" ]; then
        FLAG2=r
        OARG2=HEAD
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#homedir}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir
tmpaa=`dirname $1`
if [ "$tmpaa" = "." ]; then
        fname="$1"
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename $1`
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
```

```
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# imenik CVS v lokalnem imeniku je nujno potreben, da bodo ukazi delovali..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi s sget narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabi pik, ker lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi

# Deluj znotraj podlupine
(
        cd $homedir
        if [ "$FLAG1" = "" ]; then
                cvs diff -r HEAD "$homedir/$subdir/$fname" | less
        else
                cvs diff -$FLAG1 $OARG1 -$FLAG2 $OARG2 "$homedir/$subdir/$fname" | less
        fi
)
```

29 sadd

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# test
# CVS program sadd
# Program za dodajanje datotek v CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`
if [ $# -lt 1 ]; then
        print "\nUporaba: $cmdname <ime datoteke/imenika> \n"
        exit
fi

onearg="$1"
if [ ! -f "$onearg" -a ! -d "$onearg" ]; then
        print "\nArgument $onearg ni ime datoteke oz. imenika!"
        print "Uporaba: $cmdname <ime datoteke/imenika> \n"
        exit
fi

# Argument je ime imenika .....
homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nNapaka: spremenljivka \$CVSROOT ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
len=${#homedir}
len=$(($len + 2))
```

```
subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir je : " $subdir
tmpaa=`dirname "$onearg" `
if [ "$tmpaa" = "." ]; then
        fname="$onearg"
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename "$onearg" `
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# imenik CVS v lokalnem imeniku je nujen za delovanje ukazov..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi s sget narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ker lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi

# Preveri, ali datoteka obstaja ....
if [ $# -eq 1 ]; then
        if [ -f "$onearg" ]; then
                cvs add "$onearg"
```

```
                exit
        fi
elif [ $# -gt 1 ]; then
        print "\n\n\nDodajam vse datoteke v trenutnem imeniku v CVS."
        print "Imeniki se ne bodo dodali."
        print -n "Pritisnite Enter za nadaljevanje ali CTRL+C za prekinitev..."
        read ans
        for ii in $@
        do
                if [ -f "$ii" ]; then
                        cvs add "$ii"
                fi
        done;
        exit
fi

# Ko je $subdir enak ".", ste v korenskem imeniku
if [ "$subdir" = "." ]; then
        # Na tej tocki je $onearg imenik in ne datoteka...
        if [ -d "$CVSROOT/$onearg" ]; then
                print "\nImenik $onearg ze obstaja v CVSROOT."
                exit
        else
                # Dodajate v korenski imenik $CVSROOT
                if [ "$2" = "" -o "$3" = "" ]; then
                        print "\nUporaba: $cmdname <imenik> <oznaka proizvajalca> <oznaka izdaje>"
                        print "Na primer - "
                        print " $cmdname foo_imenik V_1_0 R_1_0"
                        exit
                else
                        (
                                cd "$homedir/$subdir";
                                cvs import "$onearg" $2 $3
                        )
                fi
        fi
else
        # Ce trenutni imenik obstaja v CVS...
        if [ -d "$CVSROOT/$subdir/$onearg" ]; then
                print "\nImenik $onearg je ze v skladiscu CVS!"
                exit
        else
                (
                        if [ -d "$homedir/$subdir/$onearg/CVS" ]; then
                                print "\nNapaka: imenik $homedir/$subdir/$onearg/CVS ze obstaja!!"
                                print "\nKoncujem ...."
                                exit
```

```
                        fi

                        # za ukaz import MORATE spremeniti ciljni imenik
                        # in MORATE dolociti polno pot, ki se zacnr s $subdir
                        cd "$homedir/$subdir/$onearg";
                        cvs import "$subdir/$onearg" Ver_1 Rel_1
                )
        fi
fi
```

30 sdelete

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program sdelete
# Program za brisanje datotek iz CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUporaba: $cmdname <ime_datoteke> \n"
        exit
fi


homedir=` echo $HOME | cut -f1 -d' ' `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi
```

```
cur_dir=`pwd`
len=${#homedir}
len=$(($len + 2))
subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir is : " $subdir
tmpaa=`dirname "$onearg" `
if [ "$tmpaa" = "." ]; then
        fname="$onearg"
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        fi
else
        fname=`basename "$onearg" `
        if [ "$subdir" = "" ]; then
                subdir=$tmpaa
        else
                subdir="$subdir/$tmpaa"
        fi
fi
#echo "subdir je : " $subdir
#echo "fname je : " $fname

# imenik CVS v lokalnem imeniku je nujen za delovanje ukazov..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi s sget narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # nu uporabljaj pik, ker lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi
```

```
# Deluj znotraj podlupine ...
(
	cd $homedir

	# Check if file does not exist....
	if [ ! -f "$subdir/$fname" ]; then
		# Poskusi dobiti datoteko iz CVS
		sget "$subdir/$fname"
		if [ ! -f "$subdir/$fname" ]; then
			print "\nNapaka: $subdir/$fname NE obstaja v skladiscu CVS."
			print "\nKoncujem $cmdname ......"
			exit
		fi
	fi

	bkextn=cvs_sdelete_safety_backup
	\mv -f "$subdir/$fname" "$subdir/$fname.$bkextn"

	cvs remove "$subdir/$fname"

	print "\nUkaz sdelete odstrani datoteko iz skladisca CVS"
	print "in jo arhivira v imenik CVS Attic. V primeru,da v prihodnosti"
	print "potrebujete to datoteko, vzpostavite stik z administratorjem CVS."
	print " "

	print "\nKoncan $cmdname. $cmdname uspesen"
	print "Izvedite scommit na imeniku $homedir/$subdir/$fname , da"
	print "bo ta sprememba stalna."
	\mv -f "$subdir/$fname.$bkextn" "$subdir/$fname"
)
```

31 sfreeze

Opomba : ***Lupino Korn /bin/ksh dobite z namestitvijo paketa pdksh*.rpm s CDja Linux contrib***

Shranite to datoteko kot tekstovno datoteko in na njej izvedite chmod a+rx.

```
#!/bin/ksh

# CVS program sfreeze
# Program za zamrznitev in odrez izdaje izvornega drevesa iz CVS

# Vsako ime datoteke se sestoji iz treh delov: domačega imenika, podimenika
# in imena datoteke. Polna pot do datoteke je tako $HOME/$subdir/$fname.
# V CVS se ohrani enaka struktura imenikov (z vpeljavo spremenljivke
# $subdir), torej se bo pot v CVS glasila $CVSROOT/$subdir/$fname.
```

```
# V tem programu igrajo 4 spremenljivke ($HOME, $CVSROOT, $subdir in $fname)
# pomembno vlogo. Na primer, vzorčne vrednosti so lahko:
# HOME=/home/aldev, subdir=myproject/src CVSROOT=/home/cvsroot
# in fname=foo.cpp

# Pozor: Uporabljajte dvojne narekovaje, da se bodo v vrednostih spremenljivk
#        ohranili presledki, npr. "$HOME/$subdir", če je $subdir enak 'some foo.cpp'

cmdname=`basename $0`

Usage()
{
        print "\nUporaba: $cmdname simbolicna_oznaka <ime imenika> "

        print "\nNa primer: - "
        print "   cd \$HOME"
        print "   $cmdname  REVIZIJA_1   moj_imenik_z_izvorno_kodo_projekta"
        print "Za prikaz seznama revizij izvedite -"
        print "slog <ime_datoteke>, zapomnite si simbolicno ime in izvedite -"
        print "cvs history -T"

        print "\nDa naredite stranski poganjek iz glavnega debla, uporabite"
        print "izbiri -b in -r, ki naredita oznako kot oznako veje. To je"
        print "koristno za izdelavo popravkov prej izdane programske opreme"
        print "Na primer :- "
        print "   cd \$HOME"
        print "   cvs rtag -b -r REVIZIJA_1 REVIZIJA_1_1 moj_imenik_z_izvorno_kodo_projekta"
        print " "

#       print "\nInformacija o oznaki se nahaja v \$CVSROOT/CVSROOT/taginfo,v"
#       print "Lahko izvedete - cd $HOME; sget CVSROOT"
#       print "da vidite to datoteko"
        exit
}

# Ukaz getopt ne bo vec podprt v naslednji izdaji.
# Namesto njega uporabljam getopts.
#while getopts r: ii
#do
#       case $ii in
#       r) FLAG1=$ii; OARG1="$OPTARG";;
#       ?) Usage; exit 2;;
#       esac
#done
#shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1
```

```
if [ $# -lt 2 ]; then
        Usage
fi

if [ ! -d $2 ]; then
        print "\nNapaka: drugi argument $2 ni imenik!"
        print "          Opuscam $cmdname...."
        print " "
        exit
fi

homedir=` echo $HOME | cut -f1 -d' '  `
if [ "$homedir" = "" ]; then
        print "\nNapaka: spremenljivka \$HOME ni nastavljena!!\n"
        exit
fi

cur_dir=`pwd`
len=${#homedir}
len=$(($len + 2))
subdir=` echo $cur_dir | cut -b $len-2000 `
#echo "subdir is : " $subdir

# imenik CVS v lokalnem imeniku je nujen za delovanje ukazov..
if [ ! -d  "$homedir/$subdir/CVS" ]; then
        #tmpaa=` (cd "$CVSROOT/$subdir"; find * -prune -type f -print | head -1 ) `
        tmpaa=` (cd "$CVSROOT/$subdir"; find * -maxdepth 0 -type f -print | head -1 ) `
        tmpbb=`basename $tmpaa | cut -d',' -f1 `
        if [ "$tmpaa" = "" -o ! -f "$CVSROOT/$subdir/$tmpbb,v" ]; then
                print "\nImenik $homedir/$subdir/CVS ne obstaja."
                print "Na imeniku `basename $subdir` morate izvesti sget. Podajte -"
                print "       cd $homedir/`dirname $subdir` "
                print "       sget `basename $subdir` "
                exit
        else
                # Sedaj poskusi s sget narediti CVS v lokalnem imeniku
                (
                        cd "$homedir"
                        if [ "$subdir" = "." ]; then  # ne uporabljaj pik, ker lahko zmedejo CVS
                                cvs -r checkout -A $tmpbb
                        else
                                cvs -r checkout -A "$subdir/$tmpbb"
                        fi
                )
        fi
fi
```

```
if [ "$cur_dir" != "$homedir" ]; then
        print "\nNe nahajate se v domacem imeniku $homedir!!"
        print "Izvesti morate ukaz sfreeze"
        print "iz domacega imenika $homedir."
        exit
fi

# cvs rtag symbolic_tag <directory name>
cvs rtag $1 $2

print "\nKoncan $cmdname. $cmdname uspesen."
```