

# CVS-RCS- HOW-TO dokument za Linux (sistem za kontrolo izvirne kode)

Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com*; prevedel Jernej Kovačič *jkovacic@email.si*; v9.0, 4. junij 2000, prevod 11. julij 2000

Ta dokument je "praktični vodnik" do hitre postavitve sistema za nadzor izvirne kode CVS/RCS. Dokument vsebuje tudi nekaj lupinskih skriptov, ki poenostavijo nekaj najpogostejše uporabljenih funkcij CVS. Ti skripti omogočajo lahek vmesnik do CVS. Informacije v tem dokumentu se nanašajo tako na Linux kot tudi na ostale vrste Unixa, kot so Solaris, HPUX, AIX, SCO, Sinix, BSD, SCO itd.

## Kazalo

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Kateri je zame? CVS ali RCS?</b>	<b>2</b>
<b>3</b>	<b>Postavljanje CVS</b>	<b>3</b>
3.1	Spremenljivke okolja . . . . .	3
3.2	Selitev od RCS k CVS . . . . .	4
<b>4</b>	<b>Lupinski skripti</b>	<b>5</b>
4.1	sget . . . . .	6
4.2	sedit . . . . .	8
4.3	scommit . . . . .	12
4.4	supdate . . . . .	15
4.5	sunlock . . . . .	16
4.6	slist . . . . .	19
4.7	sinfo . . . . .	20
4.8	slog . . . . .	22
4.9	sdif . . . . .	22
4.10	sadd . . . . .	23
4.11	sdelete . . . . .	25
4.12	sfreeze . . . . .	26
4.13	saddtree . . . . .	27
<b>5</b>	<b>Dokumentacija CVS</b>	<b>28</b>
<b>6</b>	<b>Urejevalnik Emacs</b>	<b>29</b>
<b>7</b>	<b>Sistem za sporočanje problemov</b>	<b>29</b>

8	Ostali formati tega dokumenta	30
9	Avtorske pravice in licenca	31

## 1 Uvod

Sistem za nadzor izvirne 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 izvirna koda najvitalnejša komponenta kateregakoli programskega projekta in ker razvoj programske opreme porabi ogromno časa in denarja, je zelo pomembno porabiti nekaj časa za **varovanje** izvirne kode z uporabo sistemov za kontrolo izvirne 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 izvirno kodo). Namesto vsake različice vsake datoteke, ki ste jo kadarkoli naredili, hrani CVS samo razlike med različicami. Med drugim vodi CVS 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 pravzaprav pod seboj uporablja RCS. CVS je precej zmogljivejše orodje in lahko nadzoruje popolno drevo izvirne kode. Uporaba CVS je **zelo** priporočena, ker si ga lahko prilagodite s skriptnimi jeziki, kot so npr. Perl, Korn ali Bash. Oglejte si vzorčne skripte za lupino Korn v poglavju o 4 (lupinskih skriptih).

### Prednosti CVS

- CVS je decentraliziran, uporabnik prevzema datoteke oz. imenike iz skladišča in ima svoje ločeno stabilno drevo imenikov izvirne kode.
- CVS lahko "označuje" izdaje celotnega projekta drevesa izvirne 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.

### Slabosti CVS

- Potrebneja je nekoliko več administriranja kot pri RCS.
- Zelo izpopolnjen in kompleksen sistem. Je "zadnji krik" tehnologije.
- Ima veliko število ukazov in ukaznih možnosti, zato se ga začetniki težje naučijo. Lupinski skripti v poglavju o 4 (lupinskih skriptih) lahko olajšajo uporabo.

#### Prednosti RCS

- RCS je zelo preprost za nastaviti, 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 ukaza 'make'.
- Ne more označevati izdaj celotnega programskega projekta.

Ta dokument vsebuje tudi lupinske skripte, ki omogočajo enostavno vlaganje, prevzemanje in udejanjanje datotek. Oglejte si jih v poglavju o 4 (lupinskih skriptih).

Za RCS si oglejte RCS mini-howto na Linux CDju

---

```
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>>

## 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
To see the list of files installed do -
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 tar datoteke za RCS in CVS ter slediti datotekama README in INSTALL za postavitve 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 izvirne kode ter dodelite dostop za branje in pisanje unixovi skupini/uporabniku.:

---

```
export CVSROOT=/home/cvsroot
mkdir $CVSROOT
chmod o-rwx $CVSROOT
chmod ug+rwx $CVSROOT
```

---

Za začetek dela s CVS in vlogo datotek izvirne kode storite naslednje:

---

```
cvs init

# Zamenjava imenika je OBVEZNA
cd $HOME/my_source_code_dir

# Potrebno je dati oznako izdelovalca in revizije
cvs import my_source_code_dir V1_0 R1_0
```

---

### 3.2 Selitev od RCS k CVS

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

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

---

```
#!/bin/ksh

#####
# Program za selitev datotek izvirne kode iz RCS v CVS
#
# Potreben je RPM paket pdcksh*.rpm z Linux contrib CDroma
#####

#
# rcs2cvs - pretvori drevo izvirne 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 "$0"`: ne morem priti v RCS imenik '$RCSROOT/$PROJECT'."
        exit 1
    fi

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

    # naredi nov imeniuk CVS za projekt 'projekt'
    if mkdir "$CVSROOT/$PROJECT"
    then
        :
    else
        echo >&2 "`basename "$0"`: ne morem priti v CVS imenik '$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 "$0"`: ne morem pretvoriti RCS podimenika '$RCSROOT/$RCS' v
            fi
        else
            echo >&2 "`basename "$0"`: ne morem priti v RCS podimenik '$RCSROOT/$RCS'."
        fi)
    done
done

```

---

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

## 4 Lupinski skripti

Sledijo skripti okoli osnovnih ukazov 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.

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 4.1 ()

2. **sedit** [-r številka\_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 4.2 ()
3. **scommit** [-r številka\_revizije] <ime\_datoteke> Udejanite spremembe v datoteko ali ves imenik. Odložite spremembe v CVS. Kliknite na 4.3 ()
4. **supdate** <ime\_datoteke/imenika> Posodobite datoteko ali ves imenik tako, da dobite najnoveše datoteke iz CVS. Kliknite na 4.4 ()
5. **sunlock** [-r številka\_revizije] <ime\_datoteke> Odklepanje datoteke, dobljene s sedit. Kliknite na 4.5 ()
6. **slist** Ogled datotek, ki jih trenutno urejate. Izvrši ukaz 'ls -l | grep | ...'. Kliknite na 4.6 ()
7. **sinfo** <ime\_datoteke/imenika> Informacije o spremembah/revizijah datoteke. Kliknite na 4.7 ()
8. **slog** <ime\_datoteke> Zgodovina sprememb/revizij datoteke iz CVS. Kliknite na 4.8 ()
9. **sdif** <ime\_datoteke> **sdif** -r revizija1 -r revizija2 <ime datoteke> Razlike med dvema revizijama datoteke v CVS. Kliknite na 4.9 ()  
 OPOMBA: sdif ima samo en 'f', ker že obstaja drug Unixov ukaz 'sdiff'
10. **sadd** <ime\_datoteke> Doda novo datoteko v skladišče CVS. Kliknite na 4.10 ()
11. **sdelete** <ime\_datoteke> Brisanje datoteke iz skladišča CVS. Kliknite na 4.11 ()
12. **sfreeze** <ime\_revizije> <ime\_imenika> Zamrznitev kode, izdaja celotnega izvornega drevesa. Kliknite na 4.12 ()
13. **saddtree** <ime\_revizije> <ime\_imenika> Dodajanje drevesa imenikov v CVS. Kliknite na 4.13 () 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

```
*****
```

## 4.1 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
```

```
cmdname=`basename $0`
```

```
Usage()
```

```
{
```

```
    print "\nUporaba: $cmdname [-r številka_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 "Izvelece po simbolici oznaki revizije kot - "
    print " $cmdname -r REVISION_1 some_directory "
    print " "
    exit
}

# Ukaz getopt ne bo vec podprt v naslednji izdaji.
# Namesto njega uporabljamo getopt.
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

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

# Preveri, ali datoteka ze obstaja ....
if [ -f $1 ]; then
    user_perms=" "
    group_perms=" "
    other_perms=" "
    user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
    group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
    other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
    if [ "$user_perms" = "w" -o "$group_perms" = "w" \
        -o "$other_perms" = "w" ]; then
        print "\nNapaka: Datoteko se sme spreminjati. Opuscam $cmdname ....."
        print "      Naredite rezervno kopijo, izvedite scommit ali izbrisite datoteko in"
        print "      poskusite s $cmdname znova\n"
        exit
    fi
fi

```

```

fi

cur_dir=`pwd`
#echo $cur_dir

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

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

if [ "$subdir" = "" ]; then
    fdname=$1
else
    fdname=$subdir/"$1
fi

# Prestavi datoteko
touch $1 2>/dev/null
\mv -f $1 $1.$bkextn

# Naredi podlupino
(
cd $hme
#echo $fdname

# Uporabi možnost -A za izbris vseh zastavic
if [ "$FLAG1" = "" ]; then
    cvs -r checkout -A $fdname
else
    cvs -r checkout -A -$FLAG1 $OARG1 $fdname
fi
)
#pwd

if [ -f $1 ]; then
    print "\nPridobljena kopija datoteke $fdname samo za branje."
    print "Dokoncan $cmdname"
    #print "\nNamig (Uporaba): $cmdname <ime datoteke/imenika> \n"
fi

```

---

## 4.2 sedit

**Opomba :** *Lupino Korn /bin/ksh* dobite z namestitvijo paketa *pdcksh\*.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

```

```

cmdname='basename $0'

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

    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 " $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

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

bkextn=sedit_bak

cur_dir=`pwd`
#echo $cur_dir

```

```
len=${#hme}
len=$(( $len + 2 ))
#echo $len

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

if [ "$subdir" = "" ]; then
    fdname=$1
else
    fdname=$subdir/"$1
fi

# Ce je datoteko ze kdo prevzel....
cvs_root=` echo $CVSROOT | cut -f1 -d' ' `
if [ "$cvs_root" = "" ]; then
    print "\nNapaka: \"$CVSROOT\" ni nastavljen!\n"
    exit
fi
cldir=$CVSROOT/$subdir/Locks
mkdir $cldir 2>/dev/null
rcsfile=$CVSROOT/$subdir/$1,v
#echo $rcsfile

if [ ! -e $rcsfile ]; then
    print "\nNapaka: Datoteke $1 ni v skladiscu CVS!\n"
    exit
fi

# Dobi stevilko zadnje revizije datoteke....
# Uporabi tmpfile, ker arg ni mogoce nastavljati znotraj podlupine
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
    (
        cd $hme
        cvs log $fdname | head -6 | grep head: | awk '{print $2}' > $tmpfile
    )
    OARG1=`cat $tmpfile`
    \rm -f $tmpfile 2>/dev/null
fi

lockfile=$cldir/$1-$OARG1
#if [ -e $lockfile -a "$FLAG2" = "" ]; then
if [ -e $lockfile ]; then
    print "\nNapaka: Datoteka $1 revizije $OARG1 je ze zaklenjena!"
    aa=` ls -l $lockfile | awk '{print "Zaklenil jo je razvijalec z Unix uporabniskim imenom " $3}' `
    print $aa
    print "Ta razvijalec naj izvede scommit ALI sunlock, da jo bo odklenil"
```

```

        print " "
#       print "Uporabite lahko tudi možnost -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 možnost je koristna, ce delate na loceni"
#       print "C++ funkciji v datoteki, ki ne moti ostalih razvijalcev"
#       print " "
        exit
fi

# Sedaj dobi kopijo samo za branje....
if [ ! -e $1 ]; then
    (
        cd $hme
        cvs -r checkout $fdname 1>/dev/null
    )
fi

# Preveri, ali datoteka ze obstaja....
if [ -f $1 ]; then
    user_perms=" "
    group_perms=" "
    other_perms=" "
    user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
    group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
    other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
    if [ "$user_perms" = "w" -o "$group_perms" = "w" \
        -o "$other_perms" = "w" ]; then
        print "\nNapaka: Datoteko se sme spreminjati. Opuscam $cmdname ....."
        print "        Narediti morate rezervno kopijo, izvesti scommit ali zbrisati datoteko in"
        print "        poskusiti znova s $cmdname\n"
        exit
    fi
    #print "\nOpomba: Datoteka $1 je samo za branje."
    #print "Zato jo prestavljam v $1.$bkextn ....\n"
    \mv -f $1 $1.$bkextn
    chmod 444 $1.$bkextn
elif [ -d $1 ]; then
    print "\nNapaka: $1 je imenik in ne datoteka. Opuscam $cmdname ....\n"
    exit
fi

# Naredi podlupino
print "\nSedaj dobivam datoteko $1 iz skladišca CVS ...\n"
(
    cd $hme
    #echo $fdname
    # Uporabljam možnost -A za brisanje oznake in da dobim
    # razlčico glavne revizije
    if [ "$FLAG1" = "" ]; then

```

```

        cvs -w checkout -A $fdname
else
        cvs -w checkout -A -$FLAG1 $OARG1 $fdname
fi
)

if [ -e $1 ]; then
    touch $lockfile
fi

#pwd

print "\nKoncano s $cmdname"
#print "\nNamig (Uporaba): $cmdname <filename> \n"

```

---

### 4.3 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

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 getopt.
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

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

# Poisci podimenik
cur_dir=`pwd`
#echo $cur_dir
len=${#hme}
len=$((len + 2))
#echo $len
subdir=` echo $cur_dir | cut -b $len-2000 `
#echo $subdir
if [ "$subdir" = "" ]; then
    fdname=$1
else
    fdname=$subdir/"$1
fi

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

# Ugotovi delovno stevilko revizije datoteke....
# Uporabi tmpfile, ker arg ne moremo nastaviti v podlupini
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
    (
        cd $hme
        cvs status $fdname 2>/dev/null | grep "Working revision:" | awk '{print $3}' >$tmpfile
    )
    OARG1=`cat $tmpfile`
    \rm -f $tmpfile 2>/dev/null
```

```
fi

if [ "$OARG1" = "" ]; then
    print "Datoteka $fdname je NOVA in je ni v skladiscu CVS"
else
    lockfile=$cldir/$1-$OARG1
    if [ -e $lockfile ]; then
        # Preveri, ali si lastnik revizije...
        aa=`ls -l $lockfile | awk '{print $3}' `
        userid=`id | cut -d'(' -f2 | cut -d')' -f1 `
        if [ "$aa" != "$userid" ]; then
            print " "
            print "Datoteke $fdname niste zaklenili vi!!"
            print "Zaklenil jo je uporabnik z up. imenom $aa, vi pa ste $userid"
#            print "Ce delate hkrati z drugim razvijalcem"
#            print "in ste pri sedit uporabili možnost -F,"
            print "morate pocakati, dokler drug razvijalec ne"
            print "izvede scommit ali sunlock"
            print "Opuscam $cmdname ...."
            print " "
            exit 2
        fi
    else
        if [ -f $CVSROOT/$subdir/$1,v ]; then
            print "Niste zaklenili datoteke $fdname s sedit!!"
            print "Opuscam $cmdname ...."
            exit 2
        else
            print "\nDatoteke $fdname se ni v skladiscu CVS!!"
            print "Morali bi izvesti sadd na $fdname ...."
        fi
    fi
fi

if [ -d $1 ]; then
    Usage
    exit 2
    # Za zdaj ne dovoli udejanjanja imenikov ...
    #cvs commit
else
    cvs commit $1
    exit_status=$?
fi

if [ $exit_status -eq 0 ]; then
    print "\nDokoncal $cmdname. $cmdname uspesen"
    #print "\nNamig (Uporaba): $cmdname <ime_datoteke/imenika>\n"
fi
```

---

## 4.4 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

cmdname=`basename $0`

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

# Preveri, ali datoteka ze obstaja....
if [ $# -gt 0 -a -f $1 ]; then
    user_perms=" "
    group_perms=" "
    other_perms=" "
    user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
    group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
    other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
    if [ "$user_perms" = "w" -o "$group_perms" = "w" \
        -o "$other_perms" = "w" ]; then
        while :
        do
            print "\n$cmdname bo naredil rezervno kopijo delovne datoteke"
            print "$1 v $1.supdate_bak pred kakrsnokoli spojitevijo"
            print "Ali ste prepričani, da zelite pripojiti spremembe iz"
            print -n "skladisca CVS v vaso delovno datoteko ? <d/n> [n]: "
            read ans
            if [ "$ans" = "d" -o "$ans" = "D" ]; then
                if [ -f $1.supdate_bak ]; then
                    print "\nOpozorilo: Datoteka $1.supdate_bak ze obstaja!!"
                    print "Prosim preglejte datoteko $1.supdate_bak in jo izbrisite"
                    print "in potem poskusite znova s $cmdname "
                    print "Opuscam $cmdname ...."
                    exit
                else
                    cp $1 $1.supdate_bak
                    break
                fi
            elif [ "$ans" = "n" -o "$ans" = "N" -o "$ans" = "" -o "$ans" = " " ]; then
                exit
            fi
        done
    fi
fi
```

```

fi

if [ -d $1 ]; then
    print "\nPosodobitev imenika je onemogocena ker cvs update"
    print "spaja spremembe iz skladišca v vas delovni imenik."
    print "Zato podajte ime datoteke za posodobitev - kot je prikazano spodaj: "
    print " Uporaba: $cmdname <ime_datoteke>"
    exit
#     cvs update
else
    cvs update $1
fi

print "\nKoncan $cmdname. $cmdname uspesen"
#print "\nNamig (Uporaba): $cmdname <ime_datoteke/imenika>\n"

```

---

## 4.5 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 sprostitvev ključavnice, ki jo naredi sedit

cmdname=`basename $0`

Usage()
{
    print "\nUporaba: $cmdname [-r številka_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 getopt.
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

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

cur_dir=`pwd`
#echo $cur_dir

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

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

if [ "$subdir" = "" ]; then
    fdname=$1
else
    fdname=$subdir/"$1"
fi

# Ce je datoteko ze kdo prevzel....
cvs_root=` echo $CVSROOT | cut -f1 -d' ' `
if [ "$cvs_root" = "" ]; then
    print "\nNapaka: \$CVSROOT ni nastavljen!!\n"
    exit
fi
cldir=$CVSROOT/$subdir/Locks
rcsfile=$CVSROOT/$subdir/$1,v
#echo $rcsfile

if [ ! -e $rcsfile ]; then
    print "\nNapaka: datoteke $1 ni v CVS skladiscu!!\n"
    exit
fi

# Dobivam stevilko zadnje revizije datoteke....
# Uporabljam tmpfile, ker argumentov ni mogoce nastavljati v podlupini
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
    (
        cd $hme
        cvs log $fdname | head -6 | grep head: | awk '{print $2}' > $tmpfile
    )
fi
```

```

        OARG1='cat $tmpfile'
        \rm -f $tmpfile 2>/dev/null
fi

lockfile=$cldir/$1-$OARG1
#echo lockfile is : $lockfile
if [ ! -e $lockfile ]; then
    print "\nDatoteke $1 revizije $OARG1 ni nihce zaklenil"
    print " "
    exit
fi

ans=""
while :
do
    print "\n\n*****"
    print "OPOZORILO: $cmdname bo sprostil kljucavnico in omogocil ostalim"
    print "                razvijalcem, da urejajo to datoteko. Svetujem, da"
    print "                shranite spremembe z ukazom scommit"
    print "*****"
    print -n "\nAli ste prepričani, da zelite odkleniti datoteko <d/n>? [n]: "
    read ans
    if [ "$ans" = "" -o "$ans" = " " -o "$ans" = "n" -o "$ans" = "N" ]; then
        print "\nOpuscam $cmdname ...."
        exit
    fi
    if [ "$ans" = "d" -o "$ans" = "D" ]; then
        print "\n\n\n\n\n "
        print "POZOR: lahko izgubite vse spremembe v datoteki!!"
        print -n "Ali res zelite odkleniti datoteko <d/n>? [n]: "
        read ans
        if [ "$ans" = "d" -o "$ans" = "D" ]; then
            break
        else
            exit
        fi
    else
        print "\n\nNapacen vnos. Poskusite znova..."
        sleep 1
    fi
done

if [ -e $lockfile ]; then
    \rm -f $lockfile
    print "\nKoncan $cmdname"
else
    print "\nDatoteke $1 ni odklenil nihce"
    print " "
fi

```

---

## 4.6 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

#cmdname=`basename $0`

#echo "stevilo parametrov : " $#
#echo "vsi argumenti: " $@

recurse_flag=""

if [ "$1" = "" ]; then
    dir=.
    recurse_flag=""
else
    dir=$@
    recurse_flag=" -prune "
fi

FOUT=slist_temporary_file.out

\rm -f $FOUT

find $dir $recurse_flag -type f -exec ls -ltr {} \; \
| grep -v "/CVS/" \
| grep ^\-rw \
| grep -v /\.o \
| grep -v /\.log \
| grep -v /\.out \
| grep -v /\.pid \
| awk '{ if ($NF != "tags") print $0 }' \
| awk '{ if ($NF != "a.out") print $0 }' \
| awk '{ if ($NF != "core") print $0 }' \
| awk '{ print $NF }' > $FOUT

aa=`cat $FOUT`
\rm -f $FOUT

for ii in $aa ; do
    ftype=" "
    ftype=`file $ii | awk '{print $2 }' `

    # find . -type f -exec file {} \;
    # 1)ELF 2)ukazi 3)[nt]roff, 4)c 5)anglescina 6)izvrsilna
```

```

# 7)ascii 8)tekoca 9)prazna
# ELF so prevedeni programi, lib.a so tekoce dat.
#
if [ "$ftype" = "ascii" -o "$ftype" = "commands" \
    -o "$ftype" = "[nt]roff," -o "$ftype" = "c" -o "$ftype" = "data" \
    -o "$ftype" = "English" -o "$ftype" = "executable" ]; then
    pcfile=` echo $ii | cut -d'.' -f1`
    pcfile=${pcfile}.pc"
    if [ ! -f $pcfile ]; then
        ls -l $ii
    else
        if [ "$ii" = "$pcfile" ]; then
            ls -l $ii
        fi
    fi
fi
done;

#| grep -v ^\-rwx \

#ls -l | grep ^\-rw | grep -v \\.o
#ls -l | grep ^\-rw | grep -v \\.o | awk '{ if ($NF != "tags") print $0 }'
#ls -l | grep ^\-rw | grep -v ^\-rwx | grep -v \\.o | awk '{ if ($NF != "tags") print $0 }' | awk '{ i

#print "\nKoncan $cmdname. $cmdname uspesen"
#print "\nNamig (Uporaba): $cmdname <ime_datoteke>\n"

```

---

## 4.7 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

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

hme=` echo $HOME | cut -f1 -d' ' `

```

```

if [ "$hme" = "" ]; then
    print "\nNapaka: \$HOME ni nastavljen!!\n"
    exit
fi

tmpfile=$hme/cvs_sinfo.tmp
rm -f $tmpfile

cur_dir=`pwd`
#echo $cur_dir

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

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

if [ "$subdir" = "" ]; then
    fdname=$1
else
    fdname=$subdir/"$1
fi

# Naredi podlupino
if [ -f $1 ]; then
    (
        cd $hme
        clear
        cvs status $fdname
    )
elif [ -d $1 ]; then
    (
        cd $hme
        clear
        echo " " >> $tmpfile
        echo " *****" >> $tmpfile
        echo "          Celoten status imenika >> $tmpfile
        echo " *****" >> $tmpfile
        cvs release $fdname 1>>$tmpfile 2>>$tmpfile << EOF
Y
EOF

echo "\n  -----" >> $tmpfile

aa=`cat $tmpfile | grep ^"M " | awk '{print $2}' `
for ii in $aa
do
    jj="(cd $hme; 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

```

---

## 4.8 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

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

cvs log $1 | /usr/local/bin/less

print "\nKoncan $cmdname. $cmdname uspesen"
#print "\nNamig (Uporaba): $cmdname <ime_datoteke>\n"

```

---

## 4.9 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

cmdname='basename $0'

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

if [ "$FLAG1" = "" ]; then
    cvs diff -r HEAD $1 | less
else
    cvs diff -$FLAG1 $OARG1 -$FLAG2 $OARG2 $1 | less
fi

```

---

#### 4.10 sadd

**Opomba :** *Lupino Korn /bin/ksh dobite z namestitvijo paketa `pdcksh*.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

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

# Preveri, ali datoteka obstaja ....
if [ -f $1 ]; then
    cvs add $1
    exit
fi

if [ ! -d $1 ]; then
    print "\nArgument $1 ni niti datoteka niti imenik!"
    print "Uporaba: $cmdname <ime datoteke/imenika> \n"
    exit
fi

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

cur_dir='pwd'
len=${#hme}
len=$((len + 2))
subdir=' echo $cur_dir | cut -b $len-2000 '

if [ "$subdir" = "" ]; then
    if [ -d $CVSROOT/$1 ]; then
        print "\nImenik $1 ze obstaja v CVSROOT"
        exit
    else
        # Dodaj izvirni imenik $CVSROOT
        if [ "$2" = "" -o "$3" = "" ]; then
            print "\nUporaba: $cmdname <imenik> <oznaka izdelovalca> <oznaka izdaje>"
            print "Na primer - "
            print " $cmdname foo_imenik V_1_0 R_1_0"
            exit
        else
            (
                cd $1;
                cvs import $1 $2 $3
            )
        fi
    fi
fi
```

```

        )
    fi
fi
else
    # Ce trenutni imenik obstaja v CVS...
    if [ -d $CVSROOT/$subdir ]; then
        if [ -d $CVSROOT/$subdir/$1 ]; then
            print "\nImenik $1 je ze v CVS skladiscu!"
        else
            cvs add $1
        fi
    else
        print "\nPodimenik $subdir ne obstaja v CVS"
        print "Najprej morate dodati $subdir v CVS"
        exit
    fi
fi
fi

```

---

#### 4.11 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

cmdname=`basename $0`

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

# Preveri, ali datoteka ne obstaja....
if [ ! -f $1 ]; then
    # Poskusi jo dobiti iz CVS
    sget $1
    if [ ! -f $1 ]; then
        print "\nNapaka: $1 NE obstaja v skladiscu CVS. Opuscam $cmdname ....."
        exit
    fi
fi

bkextn=cvs_sdelete_safety_backup
\mv -f $1 $1.$bkextn

cvs remove $1

```

```

print "\nUkaz sdelete odstrani datoteko iz skladišca CVS"
print "in jo arhivira v imenik CVS Attic. V primeru, da to datoteko"
print "potrebujete v prihodnosti, se posvetujte s CVS administratorjem"
print " "

print "\nKoncan $cmdname. $cmdname uspesen"
#print "\nNamig (Uporaba): $cmdname <ime_datoteke>\n"
\mv -f $1.$bkextn $1

```

---

## 4.12 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 izvirnega drevesa iz CVS

cmdname=`basename $0`

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

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

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

# cvs rtag symbolic_tag <ime imenika>
cvs rtag $1 $2

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

---

### 4.13 saddtree

**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

#####
# Vzorčni program za vnasanje drevesa imenikov (recimo SAMP) v CVS
# Upostevajte, da ce imenika SAMP ni v CVS, potem morate uporabiti ukaz
# sadd in -
#     cd SAMP; cvs import SAMP V1_0 R1_0
# Ko pozenete ta program, izvedite -
#     cd $HOME/foo/SAMP
#     cvs import foo/SAMP V1_0 Rev1_0
#####

hme=`echo $HOME | cut -f1 -d' ' `
if [ "$hme" = "" ]; then
    print "\nNapaka: \"$HOME\" ni nastavljen!!\n"
    exit
fi
sampdir=$hme/foo/SAMP
```

```

check_out_files()
{
    # Sedaj si izposodi datoteke
    tmp2f=$hme/tmp2.baksamp.sh

    cd $hme
    \rm -rf foo/SAMP
    cvs -w checkout foo/SAMP
    cd $hme/foo
    find SAMP -type f -print > $tmp2f

    cd $hme
    for ii in `cat $tmp2f`
    do
        iidir=`dirname $ii`
        iifile=`basename $ii`
        if [ "$iifile" = "Root" -o "$iifile" = "Repository" -o "$iifile" = "Entries" ]; then
            continue
        fi
        jjdir=` echo $iidir | cut -d'/' -f2-1000 `

        cp $hme/foo/SAMP.tobe/$jjdir/$iifile $hme/foo/$iidir/$iifile
        echo "cp $hme/foo/SAMP.tobe/$jjdir/$iifile $hme/foo/$iidir/$iifile "

        cvs add foo/$iidir/$iifile
    done

    print
    print "======"
    print " Sedaj pozenite cvs commit foo/SAMP"
    print " Zatem izvedite - "
    print "          cd foo; rm -rf SAMP"
    print " in dobite svezo kopijo, sget SAMP"
    print " S slog filename.samp preverite novo revizijo"
    print "======"
    print
}

check_out_files

```

---

## 5 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 "Next" za sprehod čez še več razlik. Kliknite na "Center" za centriranje besedila.

Obstaja tudi CVS odjemalec za Windows 95, ki se imenuje WinCVS - <<http://www.wincvs.org>>. 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 <file, ...>
- 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.

## 6 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>>.

## 7 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 potrebuje sistem za sporočanje problemov, kjer se sledijo hrošči in se jih posreduje različnim razvijalcem. Obiščite stran <<http://www.stonekeep.com>> za sistem sledenja projekta.

## 8 Ostali formati tega dokumenta

Ta dokument je objavljen v 11 različnih formatih in sicer - DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF (Rich Text Format), ASCII tekst, Unix man strani in SGML.

- # spremenite naslednje formate v http iz LinuxDoc.org :) Ta HOWTO dokument lahko dobite v eni tar datoteki v formatih HTML, DVI, Postscript ali SGML s - `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/>` ali `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/>`
- Dokument v ASCII tekst formatu je na razpolago na: `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` ali `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>`
- Prevodi v druge jezike, kot so francoski, nemški, španski, kitajski, japonski idr., so na voljo na `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` ali `<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>` Kakršnakoli vaša pomoč pri prevajanju v preostale jezike je dobrodošla.

Ta dokument je napisan z uporabo orodja "SGML tool", ki je na razpolago s - `<http://www.xs4all.nl/~cg/sgmltools/>` Za prevajanje izvirnika dobite ukaze kot:

- `sgml2html cvs-rcs-howto.sgml` (izdelava html datoteke)
- `sgml2rtf cvs-rcs-howto.sgml` (izdelava RTF datoteke)
- `sgml2latex cvs-rcs-howto.sgml` (izdelava latex datoteke)

Ta dokument (angleški izvirnik) se nahaja na -

- `<http://sunsite.unc.edu/LDP/HOWTO/CVS-RCS-HOWTO.html>`

Najdete ga lahko tudi na naslednjih zrcalnih straneh -

- `<http://www.caldera.com/LDP/HOWTO/CVS-RCS-HOWTO.html>`
- `<http://www.WGS.com/LDP/HOWTO/CVS-RCS-HOWTO.html>`
- `<http://www.cc.gatech.edu/linux/LDP/HOWTO/CVS-RCS-HOWTO.html>`
- `<http://www.redhat.com/linux-info/ldp/HOWTO/CVS-RCS-HOWTO.html>`
- Ostale zrcalne strani bližje vam lahko najdete na `<http://sunsite.unc.edu/LDP/hmirrors.html>`. Izberite stran in pojdite v imenik `/LDP/HOWTO/CVS-RCS-HOWTO.html`

Če si želite ogledati ta dokument v formatu dvi, uporabite program xdvi. Program xdvi se nahaja v paketu tetex-xdvi\*.rpm Redhat Linuxa. Najdete ga lahko preko menijev ControlPanel | Applications | Publishing | TeX menu buttons.

```
Za branje dvi dokumentov izvedite ukaz -
      xdvi -geometry 80x90 howto.dvi
in spremenite velikost okna z miško. Oglejte si man strani za xdvi.
Za pomikanje uporabite smerniške tipke, Page Up, Page Down, lahko pa
uporabljate tudi tipke 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' za
premikanje gor, dol, na sredino, naslednjo in prejšnjo stran itd.
Meni izklopite s pritiskom na 'x'.
```

Postscript datoteke lahko berete s programom 'gv' (ghostview) ali 'ghostscript'. Program ghostscript se nahaja v paketu ghostscript\*.rpm, program gv se nahaja v paketu gv\*.rpm Redhat Linuxa. Najdete ga preko menijev Control-Panel | Applications | Graphics. Program gv je uporabniku precej prijaznejši kot ghostscript. Ghostscript in gv sta na voljo tudi za ostale operacijske sisteme, npr. OS/2, Windows 95 in NT.

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

Za branje postscript dokumenta izvršite ukaz -  
gv howto.ps

Za uporabo ghostscripta izvršite -  
ghostscript howto.ps

Dokumente v formatu HTML lahko berete s programi Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser ali katerikoli drugim internet brskalnikom.

Dokumente v formatu latex in lyx lahko berete z uporabo programa LyX - "X-Window" vmesnika za latex.

## 9 Avtorske pravice in licenca

(C) Al Dev (Alavoor Vasudevan) 1998-2000.

Licenca je GNU GPL, zahteva se še, da avtorjevo ime in elektronski naslov ostaneta na vseh izvodih.