1. **Termin**

**Veze (links)**

**Promenljive okruženja u shell-u**

Linus Torvalds vs. Chuck Norris

Linus Torvalds didn’t design Linux to run on the 386. Intel designed the 386 to run Linux.

1. **Linkovi – veze**
2. **Zadatak**

Za početak kreirati stablo kao na slici u početnom direktorijumu (voda je tekstualni fajl):

~

SRPSKI

MAGYAR

ENGLISH

voda

[student@drvo1 ~]$ **mkdir MAGYAR SRPSKI ENGLISH**

[student@drvo1 ~]$ **cat >SRPSKI/voda**

H2O

[student@drvo1 ~]$ **ls -lR**

.:

total 12

drwxr-xr-x 2 student student 4096 Apr 27 01:37 ENGLISH

drwxr-xr-x 2 student student 4096 Apr 27 01:37 MAGYAR

drwxr-xr-x 2 student student 4096 Apr 27 01:37 SRPSKI

./ENGLISH:

total 0

./MAGYAR:

total 0

./SRPSKI:

total 4

-rw-r--r-- 1 student student 4 Apr 27 01:37 voda

***Tvrde veze (hard links)***

ln <postojece ime> <novo ime>

1. **Zadatak**

Napraviti dva tvrda linka pod imenom *MAGYAR/viz* i *ENGLISH/water*! Oba linka treba da pokazuju na fajl *voda*. Šta se promenilo kod prvobitnog fajla?

[student@drvo1 ~]$ **ln SRPSKI/voda MAGYAR/viz**

[student@drvo1 ~]$ **ln SRPSKI/voda ENGLISH/water**

[student@drvo1 ~]$ **ls -lR**

.:

total 12

drwxr-xr-x 2 student student 4096 Apr 27 01:45 ENGLISH

drwxr-xr-x 2 student student 4096 Apr 27 01:44 MAGYAR

drwxr-xr-x 2 student student 4096 Apr 27 01:44 SRPSKI

./ENGLISH:

total 4

-rw-r--r-- **3** student student 4 Apr 27 01:44 water

./MAGYAR:

total 4

-rw-r--r-- **3** student student 4 Apr 27 01:44 viz

./SRPSKI:

total 4

-rw-r--r-- **3** student student 4 Apr 27 01:44 voda

Obratiti pažnju na brojeve tvrdih veza i vreme poslednje promene fajlova! Broj tvrdih linkova se povećao na tri jer sada tri a ne samo jedno ime pokazuje na isti fajl.

1. **Zadatak**

Promeniti sadržaj fajla *viz*: dopisati jedan red teksta! Proveriti sadržaj fajla *voda*!

[student@drvo1 ~]$ **cat >> MAGYAR/viz**

nije H3O

[student@drvo1 ~]$ **cat MAGYAR/viz**

H2O

nije H3O

[student@drvo1 ~]$ **cat SRPSKI/voda**

H2O

nije H3O

Pod svakim imenom se vidi isti sadržaj fajla: sadržaj se može promeniti pod bilo kojim imenom, isto će se naći pod drugim imenima.

originalno ime fajla

i-node

sadržaj fajla

tvrdi link

Kako se briše tvrda veza? Naravno tako kao do sada: pomoću naredbe *rm* briše se tvrdi link i ako više nijedno ime ne pokazuje na fajl, tada se ustvari briše fajl.

***Simbolički linkovi (symbolic links ili soft links)***

originalno ime fajla

i-node

sadržaj fajla

tvrdi link

simbolički link

1. **Zadatak**

Napraviti simbolički link pod imenom *SRPSKI/more* tako da pokazuje na fajl *voda*. Šta se promenio kod originalnog fajla?

[student@drvo1 ~]$ **ln -s SRPSKI/voda SRPSKI/more**

[student@drvo1 ~]$ **cat SRPSKI/more**

cat: SRPSKI/more: No such file or directory

Naredba *cat* nije pronašao fajl *SRPSKI/more*. Parametri pogrešno kreiranog linka *more*:

[student@drvo1 ~]$ **ls –l SRPSKI**

total 4

lrwxrwxrwx 1 student student 10 Apr 27 02:06 **more-> SRPSKI/voda**

-rw-r--r-- 3 student student 12 Apr 27 01:50 voda

Listanje direktorijuma ispisuje sledeće: link *more* pokazuje na fajl *voda* koji se nalazi u direktorijumu *SRPSKI* u aktivnom direktorijumu a aktivni direktorijum ne sadrži folder *SRPSKI*. Neće doći do greške ako se koristi apsolutna umesto relativne putanje kod fajla na koji pokazuje simbolički link. Briše se loš link i kreira se ispravan:

[student@drvo1 ~]$ **rm SRPSKI/more**

[student@drvo1 ~]$ **ln** **-s /home/student/SRPSKI/voda SRPSKI/more**

[student@drvo1 ~]$ **cat SRPSKI/more**

H2O

nije H3O

[student@drvo1 ~]$ **ls -l SRPSKI**/

total 4

lrwxrwxrwx 1 student student 24 Apr 27 02:07 more -> /home/student/SRPSKI/voda

-rw-r--r-- 3 student student 12 Apr 27 01:50 voda

Obratiti pažnju na prvi karakter u dugačkoj listi u redu simboličkog linka: to je karakter *l* što znači da je reč o simboličkoj vezi. Veličina fajlova je različita, takođe je različito i vreme poslednje promene fajlova. Primećuje se kod broja tvrdih linkova da originalni fajl nije ni "primetio" da postoji neki simbolički link koji pokazuje na njega!

1. **Zadatak**

Obrisati originalni fajl na koji pokazuje simbolički link! Šta se dešava sa simboličkom vezom?

[student@drvo1 ~]$ **rm SRPSKI/voda**

[student@drvo1 ~]$ **cat SRPSKI/more**

cat: SRPSKI/more: No such file or directory

Mehanizam brisanja je takođe drugačiji od onog kod tvrdih veza: ako se briše simbolički link, originalni fajl ne primećuje kao što nije ni bio "svestan" toga da je simbolički link postojao.

Ako se briše originalni fajl simbolički link pokazuje na ništa. Ako se poziva na njega dobije se greška: No such file or directory.

Ustvari u ovom slučaju sadržaj originalnog fajla se nije obrisao jer još dva tvrda linka pokazuju na njega. Tvrdi linkovi su različita imena istog tog jednog fajla koji se može izbrisati samo u tom slučaju da se svako njegovo ime obriše.

1. **Zadatak**

Napraviti novi fajl *SRPSKI/voda*! Šta se dešava sa simboličkim linkom?

[student@drvo1 ~]$ **cat > SRPSKI/voda**

H2O

[student@drvo1 ~]$ **cat SRPSKI/more**

H2O

Ako se ponovo napravi fajl sa istim imenom simbolički link se "vaskrsava"!

1. **Zadatak**

Povezati sadržaj celog direktorijuma *SRPSKI* u aktivni folder (podsećanje: oznaka aktivnog direktorijuma je ".")!

[student@drvo1 ~]$ **ln -s SRPSKI/\* .**

[student@drvo1 ~]$ **ls -l**

total 12

drwxr-xr-x 2 student student 4096 Apr 27 01:45 ENGLISH

drwxr-xr-x 2 student student 4096 Apr 27 01:44 MAGYAR

drwxr-xr-x 2 student student 4096 Apr 27 02:26 SRPSKI

lrwxrwxrwx 1 student student 13 Apr 27 02:31 more -> SRPSKI/more

lrwxrwxrwx 1 student student 10 Apr 27 02:31 voda -> SRPSKI/voda

U ovom slučaju nije pogrešno kreiranje simboličkog linka pomoću relativne putanje jer su linkovi napravljeni u aktivnom folderu.

1. **Shell-ovi, shell-promenljive, promenljive korisnika, specijalne promenljive**

***Ljuske (shell-ovi)***

Važna stvar koja treba da se zapamti je da dosada nije neposredno korišćeno jezgro Unix/Linux-a, nisu pozivane neposredno Unix/Linux-ove sistemske komande. Dosad su korišćene samo ljuske. Jezgro Unix/Linux-a je duboko zakopano u utrobi računara (tačnije operativnog sistema) i ne udostojava se da se bavi tako beznačajnim stvarima kao što je utvrđivanje korisnikovih želja. Taj "prljavi" posao poveren je kategoriji programa poznatim kao ljuske. Shell je program koji čeka da se unese komanda koju zatim i izvršava tako da poziva sistemske komande. Sa Unix/Linux-ove tačke gledišta ljuska je samo prvi program koji Unix/Linux aktivira posle prijavljivanja korisnika. Naravno postoji i više ljuski, i kod najčešće korišćenih se upotrebljavaju uglavnom komande sa istim nazivima sa mogućnošću malih razlika. Najpoznatiji shell-ovi se mogu podeliti na dve grupe: ljuske tipa Bourne i tipa C. Lako se može utvrditi koji shell se koristi: ako se nakon prijavljivanja pojavi znak $ kao prompt, to je ljuska Bourne stila, a ako je znak % prompt, koristi se ljuska C.

Bourne je najkorišćeniji shell na sistemima Unix/Linux. Programsko ime ljuske Bourne je **sh**. Postoji i nekoliko alternativnih verzija ove ljuske, od kojih je najzapaženija Bourne Again (čije je programsko ime **bash**). Ljuska Korn (**ksh**) je izrasla isto iz shell-a Bourne tako da su dodate nove osobine. Programsko ime ljuske C je **csh**, i najuočljivija razlika između ovog shell-a i drugih poznatih vrsta bi mogla biti to da ljuska C koristi mnogo više specijalnih karaktera. Postoji više varijanta shell-a C: npr ljuska **tcsh** gde postoji mogućnost za izmenu unetih komandi pomoću tastera strelica.

***Promenljive shell-a i promenljive korisnika***

Cilj postojanja promenljivih u Unix/Linux-u (kao i svugde drugde) je skladištenje aktuelnih informacija. Nekoliko promenljivih sistem sam definiše i daje na raspolaganje a i korisnici mogu sami da prave promenljive za svoje potrebe. Po običaju promenljive sistema pišu se sa svim velikim slovima dok su promenljive korisnika sa svim malim slovima. Promenljive su tekstualnog tipa a program koji koristi datu promeljivu može i specijalno da je protumači (npr. kao broj). Promenljive sistema daju informacije komandama shell-a i samom shell-u.

Naredba *set* služi da se izlistaju sve promenljive ljuske. Promenljive u listi su poređane po abecedi po nazivima. Ako se želi promeniti vrednost neke promenljive i aktivni shell je Bourne, BASH ili Korn, komanda je: VARIABLENAME=nova\_vrednost, u shell-u C: set VARIABLENAME=nova\_vrednost.

1. **Zadatak**

Izlistati sve promenljive ljuske!

[student@drvo1 SRPSKI]$ **set**

BASH=/bin/bash

BASH\_ARGC=()

...

1. **Zadatak**

Analizirati vrednosti promenljivih *HOME*, *SHELL* i *PATH*!

[student@drvo1 SRPSKI]$ **echo $HOME**

/home/student

[student@drvo1 SRPSKI]$ **echo $SHELL**

/bin/bash

[student@drvo1 SRPSKI]$ **echo $PATH**

/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/student/bin

Korisniku nije potrebno da se brine o većini promenljivih, dok nekoliko promenljivih zahtevaje objašnjenje. Promenljiva **HOME** sadrži početni direktorijum korisnika u koji stiže nakon prijavljivanja na sistem. Ovo je sigurna tačka, pomoću koje se mogu zadati putanje fajlova koje važe i u slučaju da se fajlovi premeste na drugi sistem ili drugi računar. To ima poseban značaj kada se želi da programi rade isto tako dobro i u stranom okruženju.

Zadate komande ljuska traži na putanjama koje su nabrojane u promenljivoj **PATH**. Ako se npr. izdaje komanda *ls* za listanje direktorijuma, tada će shell na tim putanjama tražiti fajl pod imenom *ls*. Kada nađe prvi takav program shell "smatra" da u ovom fajlu se nalazi program koji treba pokrenuti. U aktuelnom direktorijumu će ljuska tražiti samo tada ako joj se to eksplicitno kaže u promenljivoj *PATH*. Aktuelni direktorijum treba staviti na kraj liste putanja, jer se može desiti da korisnik fajlu daje ime neke postojeće komande. Ako bi shell tražio prvo u aktuelnom direktorijumu, tada kada bi se ta komanda pozvala, ljuska bi pronašla ovaj korisnikov fajl i korisnik bi dobio grešku ili neočekivene rezultate.

Promenljiva **SHELL** sadrži putanju do komandnog interpretera koji se pokreće prilikom korisnikovog prijavljivanja na sistem.

Ako je potrebno za rad, korisnik sam može da definiše promenljive. Ako se zadaje vrednost promenljivoj koja do sada nije postojala, ta promenljiva istovremeno nastaje. Način difinisanja promenljive: <ime promenljive>=<vrednost>. Ako se poziva na promenljivu treba koristiti $<ime promenljive>.

1. **Zadatak**

Definisati promenljivu korisnika i ispisati njenu vrednost!

[student@drvo1 SRPSKI]$ **pisac=Ivo Andric**

-bash: Andric: command not found

Ako u vrednosti promenljive postoji razmak, vrednost mora da se stavi u navodnike, inače shell smatra da je druga reč posebna komanda i pokušava da je izvrši. Ispravno rešenje je:

[student@drvo1 SRPSKI]$ **pisac='Ivo Andric'**

[student@drvo1 SRPSKI]$ **echo $pisac**

1. **Zadatak**

Proveriti činjenicu da sadržaj promenljivih različite komande drugačije koriste: kreirati promenljivu *tacno* sa vrednošću *true*, napraviti fajl pod imenom *true* i posmatrati ponašanje promenljive *$tacno* u naredbama *cat* i *echo*!

[student@drvo1 ~]$ **cat >true**

Pozivamo se na promenljivu tacno.

^d

[student@drvo1 ~]$ **tacno=true**

[student@drvo1 ~]$ **cat $tacno**

Pozivamo se na promenljivu tacno.

[student@drvo1 ~]$ **echo $tacno**

true