SHELL

RAD IZ KOMANDNE LINIJE

Grafičko radno okruženje opterećuje procesor i povećava rizik u smislu sigurnosti sistema, tako da se, po pravilu, ne instalira na serverima. Tada sistem administratorima na raspolaganju ostaje **komandni interpreter (shell)** i prateći skup alata za rad sa datotekama.

KOMANDNI INTERPRETER (shell)

Shell je interfejs između korisnika i kernela, odnosno jezgra operativnog sistema. Shell prihvata komande koje korisnik zadaje, zatim ih interpretira i potom ih izvršava, pri čemu po potrebi pokreće odgovarajuće programe. Na UNIX sistemima postoji više različitih komandnih interpretera, a korisnici u toku rada po potrebi mogu preći iz jednog u drugi.

Komandni interpreter je proces koji obavlja sledeće funkcije:
- interpretaciju komandne linije,
- pokretanje programa,
- redirekciju ulaza i izlaza,
- povezivanje komandi u pipeline,
- rad sa datotekama iz komandne linije
- zamenu imena datoteka,
- rukovanje promenljivim i kontrolu okoline,
- shell programiranje.

INTERPRETACIJA KOMANDNE LINIJE

Kad se korisnik prijavi na sistem u kontekstu tekućeg login procesa izvršava se proces shell, odnosno komandni interpreter. Na ekranu se prikazuje komandni prompt (shell prompt), a to je najčešće znak **$, ukoliko se na sistem prijavi običan korisnik, odnosno #, ukoliko se na sistem prijavi root**. Kada korisnik zada neku komandu (odnosno otkuca neki tekst i pritisne Enter), shell to pokušava da interpretira. Tekst unet u shell prompt naziva se komandna linija (command line), čiji je opšti oblik:

```
$ command [opcije] [argumenti]
```

Znak $ je odzivni znak komandnog interpretera (shell prompt). Komanda može biti **interna (ugrađena u shell) ili eksterna (realizovana kao poseban program koji se nalazi u sistemskoj putanji)**. Opcije i argumenti su parametri koje shell prenosi komandi, pri čemu su argumenti najčešće obavezni i predstavljaju ime neke datoteke, direktorijuma, korisnika ili, na primer, identifikator procesa.
Ime komande, opcije i argumenti razdvajaju se razmakom. Shell interpretira razmak kao graničnik i na osnovu toga razdvaja argumente i opcije od imena komande. U jednu komandnu liniju može se uneti najviše **256 karaktera**. Imena većine UNIX komandi po pravilu se formiraju od malih slova (izuzeci su razni shell programi, poput /dev/MAKEDEV). Više UNIX komandi mogu se navesti u istoj komandnoj liniji ukoliko su razdvojene znakom tačka-zarez.

```
$   cal                 #  samo komanda
$   df /dev/sda         #  komanda (fd) i argument (/dev/sda)
$   cp 1.txt 2.txt      #  komanda (cp) i dva argumenta (1.txt i 2.txt)
$   date –u             #  komanda (date) i opcija (-u)
$   ls –l /etc          #  komanda (ls), opcija (-l) i argument (/etc)
$   clear ; date       #  dve komande koje se izvršavaju jedna za drugom
```

Opcije su osetljive na velika i mala slova (case-sensitive) i mogu se navesti na dva načina:
-x znak minus (-) praćen jednim slovom,
--option dva znaka minus (--) praćena punim imenom opcije.

INICIJALIZACIJA PROGRAMA

Nakon interpretacije komandne linije shell inicira izvršenje zadate komande. Ukoliko komanda nije interna (ugrađena u shell, poput komande cd) shell traži izvršnu datoteku koja odgovara imenu komande u direktorijumima navedenim u sistemskoj putanji. Nakon toga shell pokreće program i prosleđuje mu argumente i opcije navedene u komandnoj liniji.
Napomena: Ukoliko se izvršna datoteka nalazi u tekućem direktorijumu ili u nekom direktorijumu koji nije u sistemskoj putanji, ime komande se mora zadati sa putanjom. Slede i primeri koji ilustruju pokretanje programa koji se nalaze u tekućem direktorijumu i direktorijumu /usr/sbin:

```
$ ./myscript
$ /usr/sbin/useradd
```

REDIREKCIJA ULAZA I IZLAZA

UNIX komande primaju podatke sa **standardnog ulaza (stdin)**, rezultate izvršenja šalju na **standardni izlaz (stdout)**, a poruke o greškama na **standardni uređaj za greške (stderr)**. Većina UNIX komandi koristi tastaturu kao standardni ulaz, a monitor kao standardni izlaz i uređaj za greške.

Ulaz komande preusmerava se pomoću znaka < (manje od) na sledeći način:

```
$ command < inputdevice
```

Primer:

```
$ wc -l < /tmp/jsnmith.dat
```

Za redirekciju izlaza se koristi **znak >**. Ukoliko se redirekcija vrši u postojeću datoteku, datoteka se briše, a zatim se kreira nova u koju se smešta rezultat izvršenja komande. Za dodavanje izlaza na postojeću datoteku koristi se **znak >>**.

```
$  sort kyuss.txt > /dev/lp0
$  ls -l /home/jsmith > myfile
$  ls -l /tmp/jsmith >> myfile
$  >emptyfile
```

POVEZIVANJE KOMANDI U PIPELINE

Pipeline funkcioniše na sledeći način: standardni izlaz komande sa leve strane znaka **pipe (|)** postaje standardni ulaz komande sa desne strane znaka. Znak pipe zahteva komande i sa leve i sa desne strane, a razmaci između znaka i komande su proizvoljni.
Primer.

```
$ ls -l /etc > /tmp/files_in_etc
$ wc -l < /tmp/files_in_etc
145
```



```
$ ls -l /etc | wc -l
145
```

ZAMENA IMENA DATOTEKA – JOKER znaci

Džoker karakteri: ***, ? i []**. Argument komande koji sadrži džoker karakter zamenjuje se odgovarajućom listom datoteka shodno pravilima zamene. Komandni interpreter izvršava ovu zamenu pre izvršavanja same komande, odnosno pre pokretanja programa.

```
$ echo *
myfile1 kyuss.txt file3 anotherfile3 file4
```

- **karakter *** menja bilo koji niz znakova proizvoljne dužine
- **karakter ?** menja bilo koji znak (tačno jedan znak)
- **opseg [poc-kraj] menja** tačno jedan znak koji pripada datom opsegu.

Opseg se ne sme zadati u opadajućem redu.

```
# ls -d /etc/[a-d][a-d]*
/etc/adduser.conf       /etc/bash.bashrc
/etc/bash_completion.d
/etc/adjtime            /etc/bash_completion          /etc/calendar
```

RUKOVANJE PROMENLJIVAMA I KONTROLA OKRUZENJA

Da bi komandni interpreter bio fleksibilniji i lakši za korišćenje, u shell je uveden koncept okruženja. Okruženje je skup promenljivih (kao što je, na primer, sistemska putanja) čije vrednosti korisnici mogu menjati i na taj način prilagoditi radno okruženje svojim potrebama. Dodatno, korisnici mogu definisati nove promenljive i brisati postojeće.

SHELL PROGRAMIRANJE

Komandni interpreteri nude specifičan jezik za pisanje shell programa (**shell script**), koji se mogu koristit za automatizovanje raznih administrativnih zadataka.

DODATNE MOGUCNOSTI BOURNE-AGAIN (BASH) SHELLA

Korišćenje kontrolinih karaktera
Kontrolni karakteri se zadaju: <Ctrl> + karakter (<Ctrl> se na ekranu prikazuje kao simbol ^ (carret)).
Kontrolni karakteri Bourne-again shella koji se najčešće koriste su:

<Ctrl-c> prekida izvršenje procesa koji radi u prvom planu;
<Ctrl-d> označava kraj datoteke; napuštanje programa koji podatke čitaju sa standardnog ulaza
<Ctrl-u> briše celu komandnu liniju;
<Ctrl-w> briše zadnju reč u komandnoj liniji;
<Ctrl-s> privremeno zaustavlja izvršenje procesa u prvom planu. Može se koristiti prilikom pregledanja sadržine nekog velikog direktorijuma komandom ls ili ukoliko se neka datoteka prikazuje na ekranu programom cat;
<Ctrl-g> nastavlja se izvršenje procesa u prvom planu.

Primer. `bc` (basic calculator)

```
$ bc          # pokreće bc
100/5         # inicira operaciju deljenja
20            # program bc prikazuje rezultat prethodne operacije
<Ctrl-d>      # napuštanje programa i povratak u shell
$
```

ALTERNATIVNO IME KOMANDE (ALIAS)

Alias je način dodele kraćeg imena pomoću kog se određena komanda, ili niz komandi, može pozvati iz komandnog interpretera. Na primer, može se dodeliti alias ll (long listing) koji izvršava komandu ls –l. Alias je **aktivan samo u komandnom interpreteru za koji je napravljen**. Za korn i bash alias se dodeljuje na sledeći način:

```
$ alias aliasname=value
```

Primeri.

- komandi se može dodeliti kraće alternativno ime

```
$ alias h=history
```

```
$ alias c=clear
```

- jednom komandom se može zameniti sekvenca komandi

```
$ alias home="cd;ls"
```

- može se kreirati jednostavno ime za izvršavanje komandi sa određenim parametrima

```
$ alias ls="ls -l"
$ alias copy="cp -i"
```

PONAVLJANJE KOMANDNE LINIJE (HISTORY)

Komandni interpreter bash upisuje svaku komandnu liniju u history datoteku. Ovo omogućava da se prethodne komande ponove, pri čemu se pre ponovnog izvršavanja mogu i izmeniti. Komande se takođe mogu ponavljati na osnovu rednog broja koji im je pridružen u history datoteci. Bash shell history datoteku smešta u home direktorijum korisnika (**~/.bash_history**), i u njoj podrazumevano čuva 1000 prethodno izvršenih komandi.

Broj komandi koje se mogu smestiti u ovu datoteku može se promeniti pomoću promenljive HISTSIZE - na primer, ako je HISTSIZE=500, to zna i da se u datoteku ~/.bash_history mogu smestiti 500 prethodno izvršenih komandi.

Komanda **history** u bash shellu prikazuje prethodno izvršene komande:

```
$  history 3
   331 finger
   332 mail
   333 history 5
```

KOMPLETIRANJE IMENA DATOTEKA

```
$ ls -l /etc/pas<Tab>
$ ls -l /etc/passwd
```

Ukoliko shell u tekućem direktorijumu pronađe više od jedne datoteke čije ime počinje tim karakterima, korisnik će morati da unese još nekoliko karaktera u imenu datoteke, a zatim da ponovo pritisne taster <Tab>. Dodatno, ako korisnik dva puta pritisne <Tab>, shell će prikazati listu datoteka čije ime odgovara početku imena koje je korisnik uneo.

POREDENJE POZNATIH KOMANDNIH INTERPRETERA

Sedamdesetih godina pojavili su se Bourne Shell, Korn Shell i C Shell, na osnovu kojih su kasnije formirane dve klase komandnih interpretera: klasa bazirana na **Bourne shell**-u i klasa bazirana na C shell-u.

Bourne shell (sh)

Stephen Bourne je razvio Bourne shell za AT&T UNIX okruženje. Bourne shell (sh) se smatra za originalni UNIX komandni interpreter. Postoji na svim UNIX/Linux sistemima, ali se svi noviji komandni interpreteri koriste kao podrazumevani, jer su osetno bolji. Bourne shell je poznat po tome što je uveo mnogo suštinskih ideja, kao što je, na primer, izlazni **status izvršenih komandi**, koji je praktično omogućio pisanje shell script programa.

C shell (csh)

C shell (csh) je razvijen s ciljem da pruži okruženje za pisanje skriptova i izvršavanje naredbi izvedenih iz sintakse popularnog jezika C. Kod osnovnog C shella ne postoji mogućnost modifikacije komandne linije, ali postoji mogućnost ponavljanja komandi, kao i mogućnost kreiranja aliasa. Većina Linux sistema, nudi poboljšanu varijantu C shella, koja se naziva Enhanced C shell (**tcsh**) koji omogućava modifikaciju komandi. Pored sličnosti sa C sintaksom, C shell ima ugrađenu aritmetiku i funkciju poređenja, dok interpreteri bazirani na Bourne shell-u u te svrhe moraju pozivati eksterne komande (expr, bc).

Bourne-again shell (bash)

Bourne-again shell (bash) je najčešće korišćeni komandni interpreter pod Linux sistemima i prestavlja poboljšanu verziju Bourne shella, koja pruža mnoge dodatne mogućnosti kao što je ponavljanje i modifikovanje komandi i kompletiranje imena datoteka.

OSNOVNE KOMANDE ZA RAD SA DATOTEKAMA

Dobijanje pomoći

- navođenje opcije --help samoj komandi. Na primer:

```
$ mkdir --help
Usage: mkdir [OPTION] DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.
   -m, --mode=MODE        set permission mode (as in chmod), not rwxrwxrwx - umask
   -p, --parents           no error if existing, make parent directories
as needed
   -v, --verbose           print a message for each created directory
         --help            display this help and exit
         --version         output version information and exit
Report bugs to <bug-fileutils@gnu.org>.
```

  Ispisuje na ekranu sintaksu i objašnjenja za odgovarajuće argumente i opcije, bez detaljnijeg opisa same komande. Ukoliko objašnjenje ne može stati na jedan ekran - pipeline sa komandom less (command **`--help |less`**).

- Jedan od najkompletnijih izvora pomoći (ponekad i jako komplikovan i nejasan) su stranice uputstva za korišćenje komande (manual page, odnosno man page).

```
$ man command
```

Lokatori komandi

- **whereis**
  prikazuje lokaciju izvršnih datoteka, izvornog koda i prateće dokumentacije programa

```
$ whereis [-bms] command
```

  bez parametara prikazuje lokacije svih elemenata programa
  -b izvršne datoteke
  -m uputstva
  -s izvorni kôd

  Primer.

```
$ whereis insmod
insmod: /sbin/insmod /usr/share/man/man8/insmod.8.gz
```

  upotreba komande whereis za pronalaženje lokacije programa insmod (koji se koristi za dodavanje modula u aktivno Linux jezgro)

- **which**
  prikazuje samo lokaciju izvršnih datoteka; traži izvršnu datoteku u direktorijumima navedenim u sistemskoj putanji i ukoliko je nađe, prikazuje putanju i ime prve pronađene komande

```
$ which [-a] command
```

  a primeri komande kada je poyiva root korisnik (jer navedene komande nisu dostupne običnom korisniku) su:

```
# which insmod
/sbin/insmod
# which fdisk
/sbin/fdisk
```

  `which insmod` može pozvati samo root korisnik, jer samo on i ima pravo da izvši `insmod`

- **apropos**

na ekranu prikazuje ime i opis svih komandi koje u opisu imaju zadati string.

```
$ apropos whoami
ldapwhoami           (1)  - LDAP who am i? tool
whoami               (1)  - print effective userid
```

ODREĐIVANJE TIPA DATOTEKE

Programi koji rade u UNIX komandnoj liniji ne prepoznaju datoteke na osnovu ekstenzija.

```
$ file kk.c
kk.c: ASCII C program text
```

Na UNIX sistemima postoji nekoliko osnovnih tipova datoteka:
- tekstualne datoteke - ASCII (neformatiran tekst), English text (tekst sa interpunkcijskim karakterima) i izvršni shell programi.
- izvršne (binarne) datoteke;
- datoteke u koje su smešteni podaci (na primer, Open Office Writer dokument).

KOPIRANJE, POMERANJE I BRISANJE DATOTEKA

Command	Description
pwd	Print working directory: shows the current directory the user is in.
ls	List: shows the contents of the directory specified.
cd	Change directory: moves you to the directory identified.
mkdir	Make directory: creates the specified directory.
rmdir	Remove directory: removes a directory.
cp	Copy: copies one file/directory to specified location.

```
$ cp /home/a.a /tmp/b.b
$ cp a* /tmp
$ cp /etc/[a-d][1-5]* .
$ cp -r /etc /tmp/oldconfig
```

kopiranje direktorijuma /etc sa svim poddirektorijumima i datotekama u direktorijum /tmp/oldconfig/etc (datoteka /etc/passwd kopira se u /tmp/oldconfig/etc/passwd),

```
$ cp -r /etc/* /tmp/oldconfig
```

kopiranje kompletnog sadržaja direktorijuma /etc u direktorijum /tmp/oldconfig (datoteka /etc/passwd kopira se u /tmp/oldconfig/passwd),

```
$ cp -r a* /tmp/mybackup
```

kopiranje datoteka čije ime počinje sa a iz tekućeg direktorijuma i svih poddirektorijuma u direktorijum /tmp/mybackup.

- Vlasnik kopije je korisnik koji je pokrenuo komandu cp,
- Datoteka se dodeljuje primarnoj grupi korisnika koji je pokrenuo komandu cp,
- Pristupna prava kopije se dobijaju se logičkim množenjem bitova pristupnih prava originala i vrednosti promenljive umask. Na primer: ako su pristupna prava originalne datoteke 666, a vrednost promenljive umask 002, pristupna prava kopije biće 664.
- Sva tri vremena kopije (vreme kreiranja, poslednjeg pristupa i poslednje modifikacije) jednaka su vremenu pokretanja komande cp. Vreme poslednjeg pristupa originalne datoteke se takoće menja i jednako je vremenu pokretanja komande cp.

file	Identifies the file type (binary, text, etc).
cat	Concatenate: displays a file.
head	Shows the beginning of a file.
tail	Shows the end of a file.
less	Browses through a file from end or beginning.
more	Browses through a file from beginning to end.
touch	Creates a blank file or modifies an existing file's attributes.
mv	Move: moves the location of or renames a file/directory.
rm	Remove: removes a file.
wc	brojanje reči karaktera i linija **$ wc [-cwl] filename**

```
$ wc -l /etc/protocols
```

find Finds a file/directory.

traži datoteke čiji atributi zadovoljavaju kriterijume pretrage u direktorijumu koji je naveden kao početna tačka pretrage i svim poddirektorijumima, rekurzivno; ukoliko korisnik ne naznači komandi šta da uradi sa datotekama koje pronađe, komanda neće izvršiti nikakvu akciju.

```
$ find / -name urgent.txt –print
$ find /tmp -user jsmith -size +50 - print
$ find /home/jsmith -name "*.old" -print
```

Ostali kriterijumi pretrage su:

- **username uname**
- **groupname gname**
- **atime n** traže se datoteke kojima niko nije pristupio tačno n dana (n mora biti ceo broj, a dozvoljeni su i oblici -n i +n);
- **mtime n** traže se datoteke koje niko nije modifikovao -//-
- **perm mode** prava pristupa zadata u oktalnom obliku
- **links n** traže se sve datoteke sa n hard linkova (n mora biti ceo broj, a dozvoljeni su i -n i +n);
- **type x** traže se sve datoteke koje su tipa x, pri čemu x može biti b (blok uređaj), c (karakter uređaj), d (direktorijum), p (imenovani pipe);
- **inode n** traže se sve datoteke čiji je i-node n;
- **newer fname** traže se sve datoteke koje su modifikovane pre datoteke fname;
- **local** traže se sve datoteke koje se nalaze na lokalnim diskovima.

```
$ find /usr/home -name list.txt -exec rm {} \;
```

TRAŽENJE TEKSTA U DATOTECI

grep

```
$ grep only myfile
$ grep 'w.r' myfile
```

LINKOVI

- hard link,
- simbolički link (symbolic link).

HARD LINKOVI

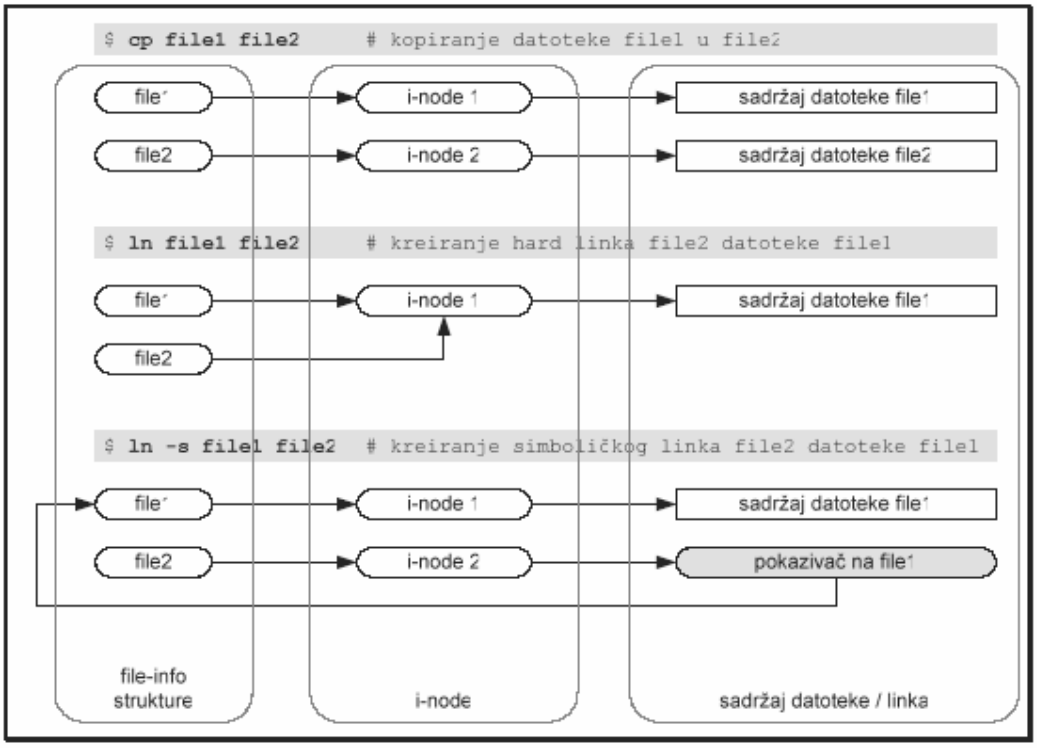
Kada korisnik pozove datoteku po imenu (na primer: cat tekst1), UNIX prevodi simboličko ime datoteke koje je naveo korisnik u interno ime, koje koristi operativni sistem. Zbog posebne interne reprezentacije, korisnici mogu datotekama dodeliti veći broj imena. Hard link je jedno od tih imena, odnosno alternativno ime datoteke.

```
$ ln file1 file2
$ ls file*
file1 file2
```

Ukoliko korisnik obriše datoteku file1, file2 se ne briše.
Osobine:

- link i original imaju isti i-node, tako da se moraju nalaziti na fizički istom sistemu datoteka (hard link se ne sme nalaziti na drugoj particiji ili na drugom disku). Ne mogu se linkovati datoteke sa mrežnog sistema datoteka (NFS);
- ne može se linkovati direktorijum niti nepostojeća datoteka;
- vlasnik, grupa i prava pristupa su isti za link i za original;
- slobodan prostor na disku neznatno se umanjuje (jedna dir-info struktura više za alternativno ime datoteke);

- broj linkova originalne datoteke uvećava se za jedan nakon linkovanja;
- datoteka sa hard linkovima se ne može obrisati sa diska sve dok se ne uklone svi hard linkovi koji upućuju na tu datoteku.



SIMBOLIČKI LINKOVI

Simbolički linkovi se mogu kreirati na dva načina:

```
$ ln -s original linkname
$ cp -s original linkname
```

osobine:

- svaki simbolički link koristi poseban i-node i jedan blok podataka u sistemu datoteka; mogu se kreirati nalaziti na fizički istom ili različitom sistemu datoteka, odnosno na istoj ili drugoj particiji (disku). Takođe, mogu se linkovati datoteke sa mrežnog sistema datoteka (NFS);
- može se linkovati direktorijum, kao i nepostojeća datoteka;
- u odnosu na original, link može imati različitog vlasnika, grupu i prava pristupa. Na korisnika koji datoteci ili direktorijumu pristupa putem simboličkog linka primenjuje se unija restrikcija (presek dozvola) linka i datoteke. Na primer, neka je korisnik user2 vlasnik linka link1 koji ukazuje na datoteku file1, i nek pripada grupi kojoj je ta datoteka formalno dodeljena. Ukoliko su pristupna prava za link i datoteku 777 i 640 respektivno, korisnik će imati samo pravo čitanja te datoteke;
- slobodan prostor na disku se umanjuje (za jedan blok podataka). Takođe, simbolički link troši jedan i-node iz i-node tabele;
- broj linkova originalne datoteke se ne uvećava za jedan nakon linkovanja, već ostaje isti kao pre linkovanja;
- s obzirom da simbolički link može ukazivati na nepostojeći objekat, originalna datoteka se može obrisati sa diska bez obzira na broj simboličkih linkova koji upućuju na nju.

```
$ ln -s /etc dir_etc
$ ls -l dir_etc
lrwxrwxrwx 1 root root 4 Sep 5 14:40 dir_etc -> /etc
$ ls -l unexist
ls: unexis: No such file or directory
$ ln -s unexist junk
$ ls -l junk
lrwxrwxrwx 1 root root 15 Sep 5 14:40 junk -> unexist
```