

Operacijski sustavi (OS)

Nositelj: doc. dr. sc. Ivan Lorencin

Asistent: Luka Blašković, mag. inf.

Ustanova: Sveučilište Jurja Dobrile u Puli, Fakultet informatike u Puli



Fakultet informatike u Puli

(2) Zastavice CLI naredbi

#2

OS

Zastavice (*eng. flags*) su posebni argumenti koji modificiraju ponašanje naredbi u CLI sučelju. Do sad smo koristili argumente za prosljeđivanje putanje datoteka i direktorija, a danas ćemo se upoznati s zastavicama koje dodaju funkcionalnosti naredbama. Zastavice se često navode jednom (-) te služe prvenstveno za prilagodbu ponašanja naredbi, a ne za prosljeđivanje podataka. U ovom poglavlju studenti će se upoznati s osnovnim zastavicama naredbi koje smo već obradili i na taj način naučiti fleksibilnije i učinkovitije koristiti CLI sučelje.

 Posljednje ažurirano: 19.3.2025.

- skripta nije dovršena!  SOON

Sadržaj

- [Operacijski sustavi \(OS\)](#)
- [\(2\) Zastavice CLI naredbi](#)
 - [Sadržaj](#)
- [1. Uvod](#)
- [2. Zastavice naredbe](#) 1s
 -  [Zastavica: -a](#)
 -  [Zastavica: -l](#)
 -  [Zastavica: -R](#)
 - [2.1 Ostale zastavice naredbe](#) 1s
 - [2.2 Tablica čestih zastavica naredbe](#) 1s
- [3. Zastavice naredbi](#) cd, pwd, mkdir i rmdir
- [Zadatak 1: Vježba osnovnih zastavica](#)

- [4. Zastavice naredbi](#) `cp`, `mv` i `rm`
 -  Zastavica: `-i`
 -  Zastavica: `-r`
 -  Zastavica: `-v`
- [Zadatak 2: Vježba složenijih zastavica](#)

1. Uvod

Na prošlim vježbama naučili smo osnovne bash naredbe za rad s datotekama i direktorijima unutar CLI sučelja. CLI omogućava korisnicima interakciju s operacijskim sustavom putem teksta, a interpretaciju samih naredbi obavlja ljuska (eng. *shell*).

Vidjeli smo što su **apsolutna** i **relativna** putanja te kako ih koristimo kao argumente za naredbe. Također, naučili smo kako navigirati kroz direktorije, stvarati nove datoteke i direktorije te ih premještati i brisati.

U nastavku su navedene osnovne naredbe koje smo obradili, zajedno s opisima:

- `pwd` - ispisuje apsolutnu putanju trenutnog (radnog) direktorija
- `ls` - ispisuje sadržaj direktorija
- `cd` - mijenja trenutni direktorij
- `mkdir` - stvara novi direktorij
- `touch` - stvara novu datoteku
- `rm` - briše datoteku
- `rmdir` - briše direktorij (ako je prazan)
- `cp` - kopira datoteku ili direktorij
- `mv` - premješta datoteku ili direktorij
- `echo` - ispisuje tekst u terminal sučelju

U ovoj skripti studenti će se upoznati s konceptom **opcija/zastavica** (eng. *flags*) 

U bash skriptiranju, **zastavice** ili **opcije** (eng. *flags*) su posebni argumenti koji modificiraju ponašanje naredbi.

Zastavice se obično koriste kako bi se naredbama dodale funkcionalnosti ili kako bi se **promijenili neki interni parametri naredbe**.

Do sad nismo koristili zastavice, već smo "nadopunjavali" naredbe koristeći argumente (npr. putanje do datoteka i direktorija).

Argumenti (parametri) naredbi se obično koriste za prosljeđivanje podataka naredbi i navodimo ih nakon inicijalne naredbe:

```
→ naredba argument1 argument2 argument3 ... argumentN
```

Primjeri:

```
→ ls /mnt/c/Users  
  
→ cp datoteka1.txt /mnt/c/Users  
  
→ rm /mnt/c/Users/datoteka1.txt
```

Zastavice (opcije) se koriste za izmjenu ponašanja naredbi i navodimo ih nakon inicijalne naredbe, a obično se označavaju s `-` ili `--` (crtica ili dvostruka crtica):

- `-z` (kratka zastavica/kratki format)
- `--zastavica` (duga zastavica/dugi format)

```
→ naredba -z --zastavica2 -z3 ... --zastavicaN # uočite da zastavice mogu biti kratke (-z)  
ili duge (--zastavica)
```

Osim toga, moguće je i **vrlo često ćemo kombinirati argumente i zastavice**:

```
→ naredba -z1 --zastavica2 argument1 argument2
```

U pravilu se zastavice pišu **prije argumenata**, ali to nije uvijek slučaj. Neki programi mogu imati različite konvencije i dozvoljavaju različite redoslijede zastavica i argumenata.

Dakle, moguće je i sljedeće:

```
→ naredba argument1 -z1 argument2 --zastavica2 argument3
```

Primjeri:

```
→ ls -l /mnt/c/Users  
  
→ cp -r vazni_dokumenti /mnt/c/Users/Desktop  
  
→ mv -i script.js /home/user/Desktop/zadaca
```

Sintaksu koju smo pokazali na prošlim vježbama, na ovim vježbama ćemo proširiti dodavanjem zastavica. Dakle, naglasak će biti na zastavice osnovnih naredbi za rad s datotekama i direktorijima.

Krenimo sa zastavicama! ▶▶▶

2. Zastavice naredbe `ls`

Sintaksa:

```
→ ls [FLAGS] <putanja>
```

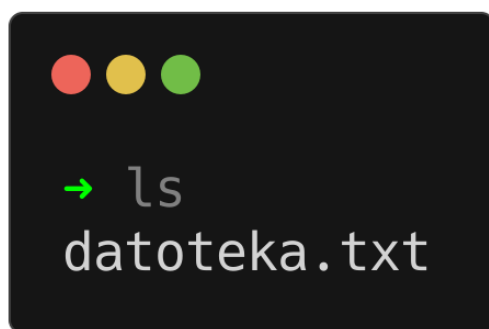
- zastavice [FLAGS] koristimo **prije putanje** kako bismo modificirali ispis
- <putanja> može biti apsolutna ili relativna putanja do direktorija

▶ Zastavica: -a

-a (zapamti kao "all") - lista sve datoteke, uključujući i one skrivene (koje počinju s .)

Primjer:

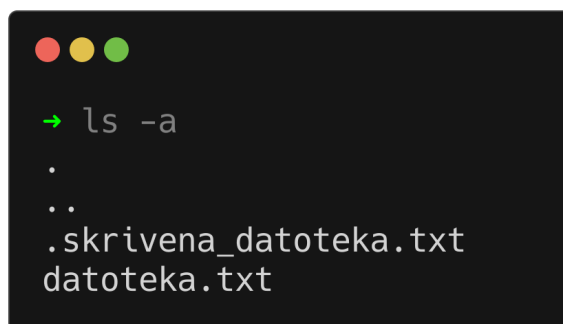
```
→ mkdir vjezba_2
→ touch vjezba_2/datoteka.txt
→ ls
```



🖼️ Naredba `ls` bez zastavice

Kako bismo dodali skrivenu datoteku, jednostavno dodajemo datoteku s točkom na početku:

```
→ touch .skrivena_datoteka.txt
→ ls # ne vidimo skrivenu datoteku
→ ls -a # vidimo skrivenu datoteku "skrivena_datoteka.txt"
```



🖼️ Naredba `ls` s zastavicom `-a` ispisuje skrivene datoteke

Ako bolje pogledate, osim skrivene datoteke `.skrivena_datoteka.txt`, vidimo i dva posebna direktorija: `.` i `..`.

Ovo su **specijalni direktoriji** koji predstavljaju:

- **trenutni** direktorij (`.`),
- **roditeljski** direktorij (`..`)

te se nalaze u svakom direktoriju datotečnog sustava. Mi smo ih do sada koristili u kontekstu naredbe `cd` kada smo se željeli vratiti u prethodni direktorij (`cd ..`).

Ove specijalne direktorije moguće je upotrebljavati i s drugim naredbama, kao što je `ls`.

Primjerice, ako se nalazimo unutar direktorija `vjezba_2` i želimo ispisati sadržaj roditeljskog direktorija, možemo koristiti:

```
→ ls ..
```

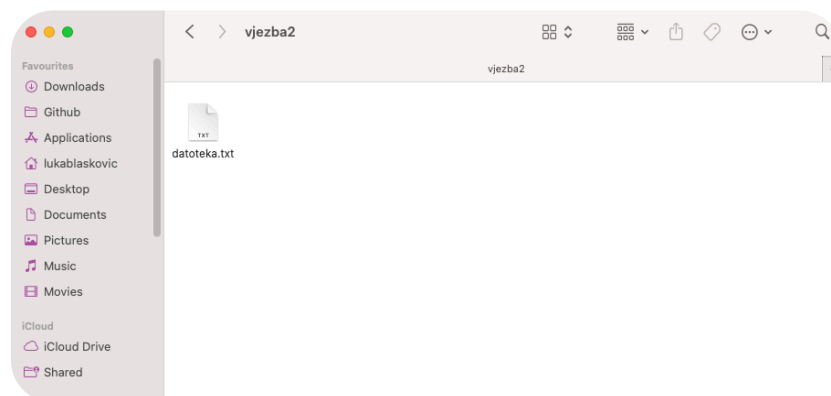
Ako bismo htjeli ispisati sadržaj roditeljskog direktorija roditeljskog direktorija (dakle, "grandparent" direktorij), koristili bismo:

```
→ ls ../..
```

Također, sljedeće naredbe su ekvivalentne:

```
→ ls
→ ls .
```

Ako koristite zadane postavke unutar GUI, skrivene datoteke i direktoriji **obično nisu vidljivi**.



Datoteka `".skrivena_datoteka.txt"` nije vidljiva unutar GUI-a, ako koristite zadane postavke

- [Kako prikazati skrivene datoteke na Windows OS GUI](#)
- [Kako prikazati skrivene datoteke/direktorije na macOS GUI](#)
- [Kako prikazati skrivene datoteke na Ubuntu/Linux OS GUI](#)

Postoji varijanta ove zastavice s velikim slovom `-A` koja također ispisuje sve datoteke, ali ne ispisuje `.` i `..` direktorije.

```
→ ls -A # ne ispisuje skrivene direktorije "." i ".."
```

🚩 Zastavica: -l

-l (zapamti kao "long") - ispisuje detaljan ispis datoteka i direktorija

```
→ cd vjezba_2  
→ ls -l
```



```
→ ls -l  
total 0  
-rw-r--r-- 1 lukablaskovic staff 0 Mar 18 21:52 datoteka.txt
```

 Naredba `ls -l` ispisuje detaljan ispis datoteka i direktorija

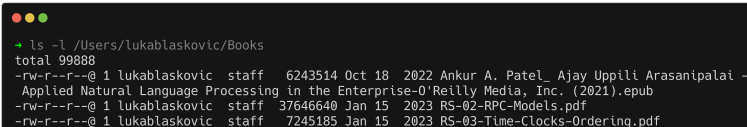
U detaljnom ispisu, svaki redak predstavlja jednu datoteku ili direktorij. Svaki redak sadrži sljedeće informacije:

1. **Tip datoteke i dozvole** - prva znamenka označava tip datoteke (`d` → direktorij, `l` → simbolička veza, `-` → obična datoteka, ...), a sljedeće tri znamenke predstavljaju dozvole za vlasnika, grupu i ostale korisnike (npr. `rw-r--r--`)
2. **Broj čvrstih veza** (eng. *Hard links*) - broj datoteka koje pokazuju na isti čvor (čvor predstavlja jedinstveni identifikator datoteke/direktorija, odnosno datoteku ili direktorij unutar datotečnog sustava, *googlati*: "unix inode")
3. **Vlasnik** - računalni korisnik koji je vlasnik datoteke/direktorija
4. **Grupa** - grupa kojoj datoteka/direktorij pripada
5. **Veličina** - veličina datoteke u bajtovima (`B`)
6. **Datum i vrijeme posljednje izmjene** - datum i vrijeme kada je datoteka/direktorij zadnji put izmijenjen
7. **Naziv direktorija/datoteke** - naziv direktorija ili datoteke

Primjer:

Želimo pogledati detaljan ispis direktorija `Books` koji se nalazi u home direktoriju korisnika `username`:

```
→ ls -l /mnt/c/Users/username/Books
```



```
→ ls -l /Users/lukablaskovic/Books  
total 99888  
-rw-r--r--@ 1 lukablaskovic staff 6243514 Oct 18 2022 Ankur A. Patel_ Ajay Uppili Arasanipalai -  
Applied Natural Language Processing in the Enterprise-0'Reilly Media, Inc. (2021).epub  
-rw-r--r--@ 1 lukablaskovic staff 37646640 Jan 15 2023 RS-02-RPC-Models.pdf  
-rw-r--r--@ 1 lukablaskovic staff 7245185 Jan 15 2023 RS-03-Time-Clocks-Ordering.pdf
```

 Naredba `ls -l` za detaljni ispis sadržaja direktorija `Books`

Vrijednost `total` koja se ispisuje na početku detaljnog ispisa ne predstavlja broj datoteka/direktorija unutar radnog direktorija. Ova vrijednost predstavlja **ukupnu veličinu svih datoteka unutar direktorija u blokovima** (eng. *disc blocks*), a 1 blok obično ima veličinu od 1024 ili 512 bajtova, ovisno o datotečnom sustavu.

Ako bismo željeli uključiti i skrivene datoteke u naš detaljni ispis, jednostavno dodajemo zastavicu `-a`:

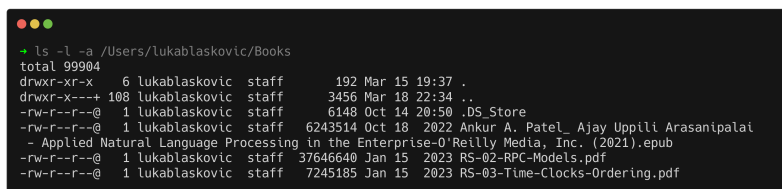
```
→ ls -l -a
```

Redoslijed zastavica vrlo često nije bitan, pa možemo koristiti i sljedeće:

```
→ ls -a -l
# ekvivalentno
→ ls -l -a
```

Ili specijalnu zastavicu koja uključuje oba ova ponašanja (`-la`):

```
→ ls -la
```



```
→ ls -l -a /Users/lukablaskovic/Books
total 99904
drwxr-xr-x  6 lukablaskovic  staff   192 Mar 15 19:37 .
drwxr-xr-x 108 lukablaskovic  staff  3456 Mar 18 22:34 ..
-rw-r--r--@ 1 lukablaskovic  staff   6148 Oct 14 20:50 .DS_Store
-rw-r--r--@ 1 lukablaskovic  staff 6243514 Oct 18 2022 Ankur A. Patel_Ajay Upplli Arasanipalal
- Applied Natural Language Processing in the Enterprise-0'Reilly Media, Inc. (2021).epub
-rw-r--r--@ 1 lukablaskovic  staff 37646640 Jan 15 2023 RS-02-RPC-Models.pdf
-rw-r--r--@ 1 lukablaskovic  staff 7245185 Jan 15 2023 RS-03-Time-Clocks-Ordering.pdf
```



Naredba `ls -l -a` za detaljni ispis sadržaja direktorija `Books` uključujući i skrivene datoteke

U ovoj skripti nećemo se detaljno baviti dozvolama, međutim za sad možete uočiti neke osnovne informacije iz ovog ispisa:

- `-` (crtica) - obična datoteka
- `d` (direktorij) - direktorij
- `r` (read) - dozvola za čitanje
- `w` (write) - dozvola za pisanje

Na primjeru od ranije:

- `drwxr-xr-x` - uočite da je prvi znak `d`, što znači da se radi o **direktoriju**, uočite i oznake `rw` što znači da vlasnik ima dozvolu za **čitanje i pisanje**
- `-rw-r--r--@` - uočite da je prvi znak `-`, što znači da se radi o **datoteci**, uočite i oznake `rw` što znači da vlasnik ima dozvolu za **čitanje i pisanje**

Za sada toliko! 😎

▶ Zastavica: `-R`

`-R` (zapamti kao "recursive") - rekurzivno ispisuje sadržaj ugniježđenih direktorija (eng. *subdirectories*)

Primjer:

Unutar direktorija `vjezba_2` definirat ćemo sljedeću strukturu direktorija i datoteka:

```
→ mkdir dir_1_razina_1
→ mkdir dir_2_razina_1
→ mkdir dir_3_razina_1

→ touch datoteka_1_razina_1.txt
→ touch datoteka_2_razina_1.txt

→ touch dir_1_razina_1/datoteka_1_razina_2.js
→ touch dir_1_razina_1/datoteka_2_razina_2.html
→ mkdir dir_1_razina_1/dir_1_razina_2

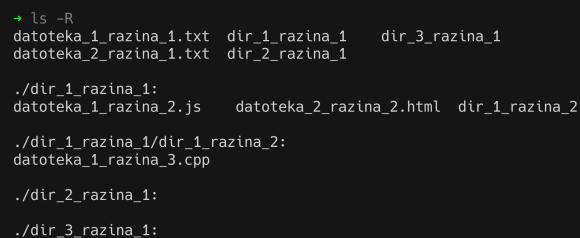
→ touch dir_2_razina_1/dir_1_razina_2/datoteka_1_razina_3.cpp
```

Očekujemo sljedeću strukturu:

```
[ 224] .
├─ [  0] datoteka_1_razina_1.txt      # 1. razina
├─ [  0] datoteka_2_razina_1.txt      # 1. razina
├─ [ 160] dir_1_razina_1              # 1. razina
│   ├─ [  0] datoteka_1_razina_2.js   # 2. razina
│   ├─ [  0] datoteka_2_razina_2.html # 2. razina
│   └─ [  96] dir_1_razina_2          # 2. razina
│       └─ [  0] datoteka_1_razina_3.cpp # 3. razina
├─ [  64] dir_2_razina_1              # 1. razina
└─ [  64] dir_3_razina_1              # 1. razina
```

Naredbom `ls -R` ispisujemo sadržaj direktorija `vjezba_2` **rekurzivno**:

```
→ ls -R
```



```
→ ls -R
datoteka_1_razina_1.txt  dir_1_razina_1  dir_3_razina_1
datoteka_2_razina_1.txt  dir_2_razina_1

./dir_1_razina_1:
datoteka_1_razina_2.js  datoteka_2_razina_2.html  dir_1_razina_2

./dir_1_razina_1/dir_1_razina_2:
datoteka_1_razina_3.cpp

./dir_2_razina_1:

./dir_3_razina_1:
```



Naredba `ls -R` će za radni direktorij ispisati sve ugniježdene direktorije i datoteke unutar njih, rekurzivno

2.1 Ostale zastavice naredbe `ls`

Zastavica ima jako puno, pokazat ćemo još nekoliko korisnih koje se često koriste.

- `-h` (zapamti kao "human"): ispisuje veličine datoteka u ljudski čitljivom formatu (npr. `1K`, `36M`, `3G`). Kombinira se s `-l` zastavicom.

```
→ ls -l -h /mnt/c/Users/username/Books
```


```
# ili
```

```
→ ls -lh /mnt/c/Users/username/Books # kombiniranje zastavica -l i -h u -lh
```

```
→ ls -l -h /Users/lukablaskovic/Books
total 99888
-rw-r--r--@ 1 lukablaskovic  staff   6.0M Oct 18  2022 Ankur A. Patel_ Ajay Uppili Arasanipalai - Applied Natural Language Processing in the Enterprise-O'Reilly Media, Inc. (2021).epub
-rw-r--r--@ 1 lukablaskovic  staff   36M Jan 15  2023 RS-02-RPC-Models.pdf
-rw-r--r--@ 1 lukablaskovic  staff   6.9M Jan 15  2023 RS-03-Time-Clocks-Ordering.pdf
```

 Naredba `ls -l -h` će ispisati veličine datoteka u ljudski čitljivom formatu

- `-t` (zapamti kao "time") sortira datoteke po vremenu zadnje izmjene (od **najnovijih** prema **najstarijima**).
- `-s` (zapamti kao "Size") sortira datoteke po veličini (od **najvećih** prema **najmanjima**).
- `-l` (zapamti kao "one") ispisuje svaku datoteku u zasebnom redu (korisno za ispis u skriptama).
- `--color` obojit će ispisane datoteke i direktorije (zadano: direktoriji su plave boje, datoteke crne, ...)

 Hint: Kod naredbe `ls`, gotovo sve zastavice je moguće međusobno kombinirati (npr. `-la`, `-lR`, `-laR`). Ovo je najbolje naučiti *trial-and-error* metodom.

Primjeri kombiniranja zastavica:

1. Detaljni ispis svih datoteka i direktorija u trenutnom direktoriju, sortiranih prema vremenu zadnje izmjene:

```
→ ls -lt # kombiniranjem: -l (detaljan ispis), -t (sortiranje prema vremenu)
```

2. Detaljni ispis svih datoteka i direktorija u trenutnom direktoriju, uključujući i skrivene datoteke, sortiranih po veličini:

```
→ ls -laS # kombiniranjem: -l (detaljan ispis), -a (sve datoteke), -S (sortiranje po veličini)
```

3. Rekurzivni detaljni ispis svih datoteka i direktorija u trenutnom direktoriju, sortiranih prema vremenu zadnje izmjene:

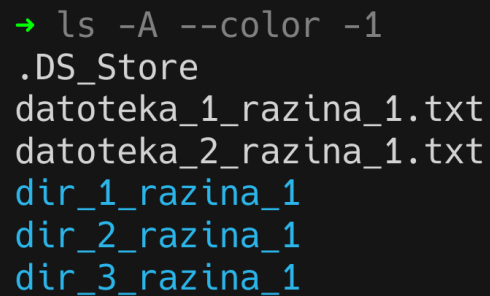
```
→ ls -lRt # kombiniranjem: -l (detaljan ispis), -R (rekurzivno), -t (sortiranje prema vremenu)
```

4. Običan ispis svih datoteka i direktorija u trenutnom direktoriju, svaki zapis u novom redu:


```
→ ls -l
```

5. Ispis svih datoteka i direktorija, uključujući skrivene ali bez `.` i `..` direktorija, obojene, svaki zapis u novom redu:

```
# nalazimo se unutar direktorija: "vjezba_2"  
→ ls -A --color -l
```



```
→ ls -A --color -l  
.DS_Store  
datoteka_1_razina_1.txt  
datoteka_2_razina_1.txt  
dir_1_razina_1  
dir_2_razina_1  
dir_3_razina_1
```

 Naredba `ls -A --color -l` će ispisati sve datoteke i direktorije, uključujući skrivene ali bez `.` i `..`, obojene i svaki podatak u zasebnom redu.

2.2 Tablica čestih zastavica naredbe `ls`

Zastavica	Sintaksa	Objašnjenje
<code>-l</code>	<code>ls -l / ls -l</code> <code>-h / ls -lh</code>	Detaljan popis datoteka/direktorija s dozvolama, vlasnikom, grupom, veličinom i datumom izmjene i drugim detaljima.
<code>-a</code>	<code>ls -a</code>	Prikazuje sve datoteke, uključujući skrivene (<code>.</code> i <code>..</code>).
<code>-A</code>	<code>ls -A</code>	Prikazuje skrivene datoteke, ali bez <code>.</code> i <code>..</code> .
<code>-h</code>	<code>ls -h / ls -l</code> <code>-h / ls -lh</code>	Prikazuje veličine u čitljivom formatu (<code>B</code> , <code>K</code> , <code>M</code> , <code>G</code>).
<code>-t</code>	<code>ls -t / ls -l</code> <code>-t / ls -lt</code>	Sortira datoteke prema vremenu izmjene (najnovije prve).
<code>-r</code>	<code>ls -r / ls -l</code> <code>-r / ls -lr</code>	Prikazuje popis datoteka/direktorija obrnutim redoslijedom.
<code>-R</code>	<code>ls -R / ls -l</code> <code>-R / ls -lR</code>	Rekurzivno ispisuje sadržaj svih poddirektorija.
<code>-1</code>	<code>ls -1</code>	Prikazuje popis datoteka/direktorija u jednom stupcu (svaku u posebni redak)
<code>-X</code>	<code>ls -X / ls -l</code> <code>-X / ls -lX</code>	Sortira datoteke prema ekstenziji.
<code>--color</code>	<code>ls --color</code>	Prikazuje datoteke u boji prema tipu datoteke.

Napomena: zastavice treba izvežbati i koristiti u praksi kako bi se bolje razumjele i zapamtile. Neke zastavice nije moguće kombinirati, primjerice zastavice `d` i `l` ne mogu se koristiti zajedno jer su međusobno kontradiktorne.

💡 Hint: Kod svih bash naredbi je moguće koristiti uputstva ili *manual* naredbom `man`, npr. `man ls` za detaljne upute kako koristiti naredbu `ls` i koje su sve zastavice dostupne uključujući njihova ograničenja. Iz manuala se izlazi pritiskom tipke `q`

3. Zastavice naredbi `cd`, `pwd`, `mkdir` i `rmdir`

Naredba `cd` nema zastavica i koristi se isključivo s argumentima (putanjama).

Naredba `pwd` ima dvije zastavice (`-L` i `-P`) međutim ne koriste se često pa ih nećemo niti spominjati.

Naredbe `mkdir` i `rmdir` imaju korisnu zastavicu `-p` koja omogućava **stvaranje/brisanje više ugniježđenih direktorija odjednom**.

U prošloj skripti, rekli smo da ne možemo napraviti direktorij unutar nepostojećeg direktorija, odnosno:

```
# ne možemo stvoriti direktorij "test" unutar nepostojećeg direktorija
"files_manipulation"
→ mkdir files_manipulation/test
```

- Ipak, zastavica `-p` će nam isto omogućiti 😊

```
→ mkdir -p files_manipulation/test

# pa i više ugniježđenih direktorija
# Primjer: stvaranje ugniježđenih direktorija "dir1/dir2/dir3"
→ mkdir -p dir1/dir2/dir3
```

Ekvivalentno možemo koristiti i naredbu `rmdir -p` za brisanje ugniježđenih direktorija, ako su prazni:

```
→ rmdir -p dir1/dir2/dir3 # briše sva 3 direktorija, ako su prazni
```

Zadatak 1: Vježba osnovnih zastavica

1. Stvorite direktorij `vjezba_1s` i unutar njega stvorite sljedeću strukturu direktorija i datoteka koristeći isključivo `mkdir` i `touch` naredbe:
 - o ugniježđene direktorije stvorite koristeći odgovarajuću zastavicu

```
[ 96] .
└─ [ 128] faks
    └─ [ 256] 1_semestar
        │   └─ [ 0] diferencijalni_i_integralni_racun.txt
        │   └─ [ 0] logika_i_diskretna_mat.txt
        │   └─ [ 0] multimedijalni_sustavi.txt
        │   └─ [ 0] osnove_ikt.txt
        │   └─ [ 0] osnove_podatkovne_znanosti.txt
        │   └─ [ 0] programiranje.txt
        └─ [ 256] 2_semestar
            │   └─ [ 0] baze_podataka_1.txt
            │   └─ [ 0] geometrija_i_linearna_algebra.txt
            │   └─ [ 0] informacijski_sustavi.txt
            │   └─ [ 0] matematicka_analiza.txt
            │   └─ [ 0] operacijski_sustavi.txt
            └─ [ 0] programiranje_u_skriptnim_jezicima.txt
```

2. Unutar direktorija `vjezba_1s` ispišite rekurzivno sadržaj direktorija `faks`, u boji, svaku datoteku u zasebnom redu bez skrivenih datoteka.

```
faks

./faks:
1_semestar
2_semestar

./faks/1_semestar:
diferencijalni_i_integralni_racun.txt
logika_i_diskretna_mat.txt
multimedijalni_sustavi.txt
osnove_ikt.txt
osnove_podatkovne_znanosti.txt
programiranje.txt

./faks/2_semestar:
baze_podataka_1.txt
geometrija_i_linearna_algebra.txt
informacijski_sustavi.txt
matematicka_analiza.txt
operacijski_sustavi.txt
programiranje_u_skriptnim_jezicima.txt
```

3. Unutar direktorija `vjeba_1s` ispišite **detaljne** informacije datoteka iz 1. semestra, sortiranih po datumu zadnje izmjene - od najnovije prema najstarijoj.

```
total 0
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:07 osnove_ikt.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:07 multimedijalni_sustavi.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:07 programiranje.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:07 logika_i_diskretna_mat.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:07 diferencijalni_i_integralni_racun.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 osnove_podatkovne_znanosti.txt
```

4. U drugi semestar dodajte tajni kolegij koji je skrivena datoteka i **detaljno** ispišite sve datoteke iz 2. semestra uključujući i skrivene datoteke, bez `.` i `..` direktorija.

```
total 0
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:15 .tajni_kolegij.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 baze_podataka_1.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 geometrija_i_linearna_algebra.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 informacijski_sustavi.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 matematicka_analiza.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:08 operacijski_sustavi.txt
-rw-r--r-- 1 lukablaskovic staff 0 Mar 19 00:09 programiranje_u_skriptnim_jezicima.txt
```

5. Ako se nalazite u direktoriju `faks/2_semestar`, objasnite 2 načina kako biste dodali novu datoteku `.tajni_kolegij.txt` u direktorij `faks/1_semestar` bez da se prebacujete u taj direktorij, tj. bez korištenja naredbe `cd`.

4. Zastavice naredbi `cp`, `mv` i `rm`

U ovom dijelu pokrit ćemo zastavice naredbi `cp` (kopiranje), `mv` (premještanje) i `rm` (brisanje), koje omogućuju precizniju kontrolu nad njihovim ponašanjem.

Ove tri naredbe imaju dosta zastavica koje se ponavljaju i međusobno su slične, stoga ćemo ih obraditi zajedno.

Sintaksa:

```
→ cp [FLAGS] <izvorna_datoteka> <ciljna_datoteka>

→ mv [FLAGS] <izvorna_datoteka> <ciljna_datoteka>

→ rm [FLAGS] <izvorna_datoteka>
```

- zastavice `[FLAGS]` koristimo **prije** izvorne datoteke
- `<izvorna_datoteka>` i `<ciljna_datoteka>` mogu biti apsolutne ili relativne putanje do datoteka

▶ Zastavica: `-i`

`-i` (zapamti kao "interactive") - prije kopiranja/premještanja/brisanja datoteke, **zastavica će pitati korisnika za potvrdu**. Zastavica je korisna u slučajevima kada želimo izbjeći slučajnu izmjenu važnih datoteka, pogotovo kad radimo s više datoteka odjednom ili kad radimo s datotekama koje su važne za rad operacijskog sustava.

- zastavica `-i` unutar `cp` će pitati korisnika za potvrdu **samo ako datoteka već postoji na odredištu** (eng. *overwrite*)
- zastavica `-i` unutar `mv` će pitati korisnika za potvrdu **samo ako datoteka već postoji na odredištu** (eng. *overwrite*)
- zastavica `-i` unutar `rm` će pitati korisnika za potvrdu **uvijek prije brisanja datoteke** (pitat će za svaku datoteku)

Primjeri:

```
→ cp -i datoteka.txt /mnt/c/Users/username/Desktop # naredba će pitati korisnika prije
kopiranja datoteke samo ako "datoteka.txt" već postoji na odredištu

# ili

→ mv -i datoteka.txt /mnt/c/Users/username/Desktop # pitaj korisnika prije premještanja
datoteke na odredište ako "datoteka.txt" već postoji

# ili

→ rm -i datoteka.txt # pitaj korisnika prije brisanja datoteke svaki put
```

Korisnik odgovara na pitanje s `y` (yes) ili `n` (no), odnosno **unosom odgovarajućeg slova** i pritiskom tipke `Enter`.

Primjer:

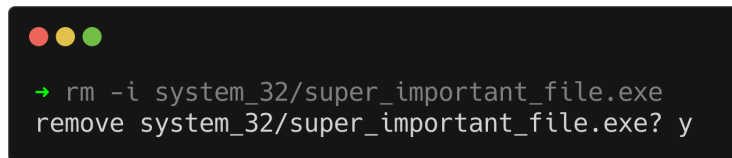
```
→ mkdir system_32
→ touch system_32/super_important_file.exe

→ rm -i system_32/super_important_file.exe
```

Nakon izvršavanja naredbe, korisnik će dobiti sljedeći ispis:

```
→ rm: remove system_32/super_important_file.exe?
```

- Ako korisnik odgovori s `y` i pritisne `Enter`, datoteka će biti obrisana
- Ako korisnik odgovori s `n` ili bilo kojim drugim znakom i pritisne `Enter`, datoteka neće biti obrisana



```
→ rm -i system_32/super_important_file.exe
remove system_32/super_important_file.exe? y
```



Naredba `rm -i` će pitati korisnika je li siguran u brisanje datoteke, svaki put

🚩 Zastavica: `-r`

`-r` (zapamti kao "recursive") - kopira/briše direktorij i **sav njegov sadržaj rekurzivno**. Pomoću ove zastavice moguće je kopirati/brisati direktorije i sve datoteke/poddirektorije unutar njih u jednom koraku.

- ovo ponašanje je zadano kod naredbe `mv` (premještanje) pa iz tog razloga nema zastavice `-r` kod naredbe `mv`
- zastavicu `-r` je moguće pisati i velikim slovom: `-R` (kod naredbe `ls` ove zastavice nisu ekvivalentne)

Kod **kopiranja** (`cp`) smo rekli da možemo kopirati određenu datoteku ili direktorij iz mjesta `<izvor>` u mjesto `<odredište>`:

```
→ cp <izvor> <odredište>
```

- gdje izvor i odredište mogu biti relativne ili apsolutne putanje

Do sad smo vidjeli primjere gdje kopiramo:

- jedan direktorij u drugi direktorij → (`cp dir dir2`)
- jednu datoteku u drugi direktorij → (`cp datoteka.txt dir`)
- jednu datoteku u drugu datoteku (s istim ili različitim nazivom) → (`cp datoteka.txt datoteka2.txt`)

Međutim, **što ako želimo kopirati cijeli direktorij, uključujući sav njegov sadržaj**, u drugi direktorij? Tada je potrebno koristiti rekurzivnu zastavicu `-r`:

```
→ cp -r <izvor> <odredište>
```

Primjer s rekurzivnim kopiranjem:

```
→ mkdir vjezba_cp_r

→ mkdir vjezba_cp_r/dir1

→ cd vjezba_cp_r/dir1
→ touch datoteka1.txt datoteka2.txt datoteka3.txt

→ ls -l

# Primjer: kopirat ćemo sav sadržaj direktorija "dir1" u direktorij "vjezba_cp_r/dir2"

→ cp -r . ../dir2 # "dir2" će biti stvoren unutar "vjezba_cp_r", ako ne postoji
```

Kod **premještanja** (`mv`), rekurzivno ponašanje je zadano i ne navodi se eksplicitno ovom zastavicom:

```
mv <izvor> <odredište> # nema zastavice -r
```

Primjer:

```
→ mkdir vjezba_mv

→ mkdir vjezba_mv/dir1

→ cd vjezba_mv/dir1
→ touch script1.js script2.js script3.js

→ mkdir ../dir2

# nalazimo se unutar: /vjezba2/vjezba_mv
# Primjer: premještanje cijelog direktorija sa sadržajem u drugi direktorij (dir1 -> dir2)
→ mv dir1 dir2 # premješta "dir1" u "dir2"
```

Ipak, ako bi htjeli premjestiti samo sadržaj direktorija `dir1` u `dir2`, a ne cijeli direktorij, moramo koristiti tzv. "wildcard" - znak `*`.

Općenito, [Wildcard](#) se koristi kad želimo radnju izvršiti nad više datoteka odjednom (konkretno: wildcard `*` predstavlja **sve datoteke** unutar nekog direktorija).

Sintaksa:

```
naredba dir1/dir2/wildcard # wildcard nije zastavica, već poseban znak unutar putanje!  
→ mv dir1/* dir2 # premješta sve datoteke iz "dir1" u "dir2"  
  
# "dir1" je sad prazan  
# "dir2" sadrži sve datoteke iz "dir1"
```


- Ako postoji previše datoteka unutar direktorija, wildcard `*` može dati grešku: "Argument list too long".

Kod **brisanja** (`rm`) moramo biti posebno oprezni jer rekurzivno brisanje direktorija i njegovog sadržaja može biti **nepovratno**.

```
rm -r <direktorij>
```

Primjer rekurzivnog brisanja:

```
→ mkdir vjezba_rm_r  
  
→ mkdir vjezba_rm_r/dir1  
  
→ cd vjezba_rm_r/dir1  
  
→ touch touch cache1.txt cache2.txt cache3.txt cache4.txt  
  
→ ls -l  
  
# nalazimo se unutar: /vjezba2/vjezba_rm_r  
# Primjer: rekurzivno brisanje direktorija "dir1" i svih datoteka unutar njega  
→ rm -r dir1
```

 **Oprez: Rekurzivno brisanje direktorija i njegovog sadržaja može biti opasno**, stoga je potrebno biti oprezan i provjeriti je li odabrani direktorij ispravan prije brisanja. Ipak, dobra praksa je kombinirati zastavicu `r` zastavicom `i` koja će u tom slučaju pitati korisnika je li siguran u brisanje za svaku datoteku. Ipak, većina modernih Linux distribucija ima neki oblik zaštita (*safeguards*) koje preveniraju rekurzivno brisanje korijenskog direktorija.

Primjer rekurzivnog brisanja s potvrdom:

```
mkdir system_32  
  
touch system_32/super_important_file.exe  
touch system_32/another_important_file.exe  
touch system_32/settings.json  
  
# Primjer: rekurzivno brisanje direktorija "system_32" s potvrdom za svaku datoteku  
rm -ri system_32
```

Na ovaj način naredba `rm` će:

- pitati korisnika za pregled svakog direktorija (`examine`)

- pitati korisnika je li siguran u brisanje svake datoteke (`remove`)
- pitati korisnika je li siguran u brisanje ukupnog direktorija (`remove directory`)
- korisnik odgovara s `y` ili `n` i pritiskom tipke `Enter` na jednak način kao do sada

```
→ rm -ri system_32
examine files in directory system_32? y
remove system_32/settings.json? y
remove system_32/super_important_file.exe? y
remove system_32/another_important_file.exe? y
remove system_32? y
```



Naredba `rm -ri` će pitati korisnika je li siguran u brisanje svake datoteke i direktorija, svaki put.

🚩 Zastavica: `-v`

- `-v` (zapamti kao "verbose") - ispisuje detalje o radnji koja se izvršava (npr. ispisuje datoteke koje se kopiraju/premještaju/brišu)

Izraz *verbose* je općenit pojam koji označava "govorljivost" ili "detaljnost" ispisivanja informacija. U kontekstu naredbi `cp`, `mv` i `rm`, zastavica `-v` će ispisivati **detalje o radnji koja se izvršava**.

```
→ cp -v datoteka.txt /mnt/c/Users/username/Desktop # ispisuje detalje o kopiranju datoteke
```

```
→ mv -v datoteka.txt /mnt/c/Users/username/Desktop # ispisuje detalje o premještanju datoteke
```

```
→ rm -v datoteka.txt # ispisuje detalje o brisanju datoteke
```

Primjer s detaljima o kopiranju:

```
mkdir vjezba_v
```

```
touch vjezba_v/datoteka1.txt
```

```
cp -v vjezba_v/datoteka1.txt vjezba_v/datoteka2.txt # kopira datoteku i preimenuje je, ispisuje detalje o radnji
```

```
→ cp -v vjezba_v/datoteka1.txt vjezba_v/datoteka2.txt
vjezba_v/datoteka1.txt -> vjezba_v/datoteka2.txt
```



Zastavica `-v` će ispisati detaljne informacije o promjenama koje su se dogodile, konkretno:

```
vjezba_v/datoteka1.txt -> vjezba_v/datoteka2.txt
```

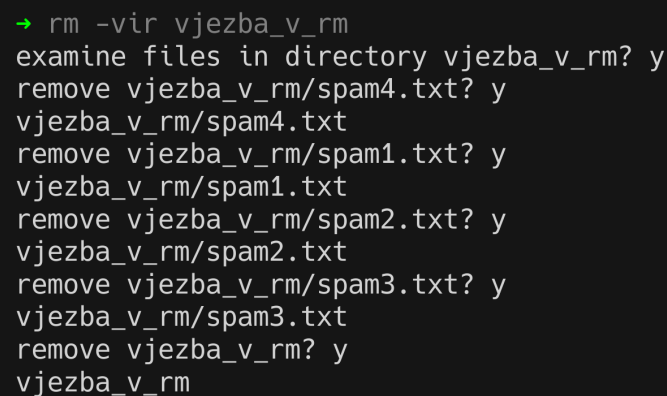
Primjer s detaljima o rekurzivnom brisanju i potvrdoma:

```
mkdir vjezba_v_rm

cd vjezba_v_rm

touch spam1.txt spam2.txt spam3.txt spam4.txt

# Primjer: kombinirat ćemo zastavice -v, -i i -r za upit prije brisanja svake datoteke i
# ispisati obrisanu datoteku/direktorij nakon svake operacije
rm -vir vjezba_v_rm
```



```
→ rm -vir vjezba_v_rm
examine files in directory vjezba_v_rm? y
remove vjezba_v_rm/spam4.txt? y
vjezba_v_rm/spam4.txt
remove vjezba_v_rm/spam1.txt? y
vjezba_v_rm/spam1.txt
remove vjezba_v_rm/spam2.txt? y
vjezba_v_rm/spam2.txt
remove vjezba_v_rm/spam3.txt? y
vjezba_v_rm/spam3.txt
remove vjezba_v_rm? y
vjezba_v_rm
```



Kombiniranje zastavica `-v`, `-i` i `-r` će ispisati detalje o brisanju svake datoteke i pitati korisnika za potvrdu svake radnje

Zadatak 2: Vježba složenijih zastavica

1. Stvorite direktorij `backup` unutar roditeljskog direktorija `zadatak2`. Unutar `backup` stvorite tri datoteke: `file1.txt`, `file2.txt`, `file3.txt`.
2. Jednom naredbom kopirajte sve datoteke iz `backup` u novi direktorij `backup/backup_copy`, koristeći odgovarajuću zastavicu da se prikaže svaka kopirana datoteka.
3. Premjestite `backup_copy` u direktorij `zadatak2` i preimenujte ga u `backup_moved`, koristeći odgovarajuću zastavicu da se prikaže svaka premještena datoteka.
4. Obrišite `backup_moved` direktorij i njegov sadržaj, koristeći zastavicu koja će tražiti potvrdu prije brisanja svake datoteke.
5. Izradite novi direktorij `data` unutar `zadatak2` i u njemu stvorite datoteku `.config.json` (skrivena datoteka). Zatim detaljno ispišite sve datoteke u `data`, uključujući i sve skrivene.
6. Ako se nalazite u `zadatak2/data`, kako biste kopirali `.config.json` u direktorij `backup` bez da mijenjate direktorij? Napišite dvije različite naredbe bez korištenja `cd`.