

Linux ფაილებთან მუშაობა - სრული გზამკვლევი სტუდენტებისთვის

სარჩევი

1. ფაილების და დირექტორიების შექმნა

- 1.1. ცარიელი ფაილის შექმნა - touch
- 1.2. დირექტორიის შექმნა - mkdir
- 1.3. ტექსტური ფაილის შექმნა შინაარსით

2. ფაილების კოპირება - cp

- 2.1. ფაილების კოპირება
- 2.2. დირექტორიების კოპირება
- 2.3. სასარგებლო ოპციები

3. ფაილების გადატანა/გადარქმევა - mv

- 3.1. ფაილის გადარქმევა
- 3.2. ფაილის გადატანა
- 3.3. სასარგებლო ოპციები

4. ფაილების წაშლა - rm

- 4.1. ფაილების წაშლა
- 4.2. დირექტორიების წაშლა
- 4.3. სასარგებლო ოპციები და უსაფრთხოება

5. ფაილების ნახვა

- 5.1. ფაილის შინაარსის ნახვა - cat
- 5.2. დიდი ფაილების ნახვა - less და more
- 5.3. ფაილის დასაწყისის/დასასრულის ნახვა

6. დირექტორიებში ნავიგაცია

- 6.1. მიმდინარე დირექტორიის ნახვა - pwd
- 6.2. დირექტორიის შიგთავსის ნახვა - ls

- 6.3. დირექტორიის შეცვლა - cd

7. ჯოკერები (Wildcards)

- 7.1. * - ნებისმიერი სიმბოლო(ები)
- 7.2. ? - ერთი ნებისმიერი სიმბოლო
- 7.3. [] - სიმბოლო დიაპაზონიდან

8. ფაილების უფლებები

- 8.1. უფლებების ნახვა
- 8.2. უფლებების შეცვლა - chmod

9. ხშირად დასმული კითხვები (FAQ)

10. სასარგებლო რჩევები

11. შეჯამება

შესავალი

ეს სახელმძღვანელო დაგეხმარებათ ისწავლოთ როგორ ვიმუშაოთ ფაილებთან Linux-ის ტერმინალში Ubuntu 18.04 ოპერაციულ სისტემაში. ყველაფერი ახსნილია მარტივად, ნაბიჯ-ნაბიჯ, მრავალი პრაქტიკული მაგალითით.

რას ისწავლით ამ გზამკვლევიდან:

- ფაილების და დირექტორიების შექმნა
- ფაილების კოპირება და გადატანა
- ფაილების წაშლა უსაფრთხოდ
- ფაილების ნახვა და რედაქტირება
- ფაილურ სისტემაში ნავიგაცია
- ჯოკერების (wildcards) გამოყენება
- ფაილების უფლებების მართვა

Linux-ის ტერმინალი ძლიერი ინსტრუმენტია და ამ უნარების დაუფლება დაგეხმარებათ ეფექტურად იმუშაოთ ფაილებთან.

1. ფაილების და დირექტორიების შექმნა

1.1. ცარიელი ფაილის შექმნა - touch

ბრძანება `touch` ქმნის ახალ ცარიელ ფაილს.

სინტაქსი:

```
touch ფაილის_სახელი
```

მაგალითები:

```
# ერთი ფაილის შექმნა
touch myfile.txt

# რამდენიმე ფაილის ერთდროულად შექმნა
touch file1.txt file2.txt file3.txt

# სხვადასხვა გაფართოებით ფაილების შექმნა
touch document.pdf script.sh photo.jpg
```

რა ხდება:

- თუ ფაილი არ არსებობს - იქმნება ახალი ცარიელი ფაილი
- თუ ფაილი უკვე არსებობს - განახლდება მისი ბოლო ცვლილების დრო

1.2. დირექტორიის შექმნა - mkdir

ბრძანება `mkdir` (make directory) ქმნის ახალ საქაღალდეს.

სინტაქსი:

```
mkdir დირექტორიის_სახელი
```

მაგალითები:

```
# ერთი დირექტორიის შექმნა
mkdir documents

# რამდენიმე დირექტორიის ერთდროულად შექმნა
mkdir photos videos music

# დირექტორია სივრცეებით სახელში (გამოიყენეთ ბრჭყალები)
mkdir "My Documents"
mkdir 'My Photos'

# ქვედირექტორიების იერარქიის შექმნა -p ოპციით
mkdir -p projects/website/css
mkdir -p documents/2024/january

# ვერბოზური რეჟიმი (აჩვენებს რას აკეთებს)
mkdir -v my_folder
```

-p ოპციის მაგალითი:

```
# ეს შექმნის projects-ს, შიგ website-ს და შიგ css-ს
mkdir -p projects/website/css

# ეს შექმნის მთელ იერარქიას: home → student → documents → 2024 → reports
mkdir -p /home/student/documents/2024/reports
```

ხშირი შეცდომა:

```
# არასწორი - სივრცეები გარეშე ბრჭყალებისა
mkdir My Documents # შექმნის ორ საქალაქს: "My" და "Documents"

# სწორი
mkdir "My Documents" # შექმნის ერთ საქალაქს სახელით "My Documents"
```

1.3. ტექსტური ფაილის შექმნა შინაარსით

მეთოდი 1: echo ბრძანება (სწრაფი გზა)

```
# ერთი ხაზის ფაილი
echo "Hello World" > myfile.txt

# რამდენიმე სიტყვა
echo "ეს არის ჩემი პირველი ფაილი" > test.txt

# ფაილში ტექსტის დამატება (არ წაშლის არსებულს)
echo "ახალი ხაზი" >> myfile.txt
```

განსხვავება > და >> შორის:

- `>` - წაშლის ძველ შინაარსს და წერს ახალს
- `>>` - უმატებს ახალ ტექსტს ძველს

მეთოდი 2: cat ბრძანება (მრავალხაზიანი ტექსტი)

```
cat > myfile.txt
ეს არის პირველი ხაზი
ეს არის მეორე ხაზი
ეს არის მესამე ხაზი
[დააჭირეთ Ctrl+D როცა დაასრულებთ]
```

მეთოდი 3: ტექსტური რედაქტორები

```
# nano რედაქტორი (მარტივი და მოსახერხებელი დამწყებთათვის)
nano myfile.txt

# nano-ში მუშაობა:
# - დაწერეთ ტექსტი
# - Ctrl+O = შენახვა
# - Ctrl+X = გასვლა

# vim რედაქტორი (უფრო რთული, მაგრამ ძლიერი)
vim myfile.txt

# vim-ში მუშაობა:
# - დააჭირეთ 'i' რომ დაიწყოთ წერა
# - Esc = გასვლა წერის რეჟიმიდან
# - :wq = შენახვა და გასვლა
# - :q! = გასვლა შენახვის გარეშე
```

2. ფაილების კოპირება - cp

ბრძანება `cp` (copy) აკოპირებს ფაილებს ან დირექტორიებს.

სინტაქსი:

```
cp საიდან სად
```

2.1. ფაილების კოპირება

მაგალითები:

```
# ფაილის კოპირება იმავე დირექტორიაში ახალი სახელით
cp file.txt file_copy.txt

# ფაილის კოპირება სხვა დირექტორიაში
cp file.txt /home/student/documents/

# ფაილის კოპირება სხვა დირექტორიაში ახალი სახელით
cp file.txt /home/student/documents/new_name.txt

# რამდენიმე ფაილის კოპირება ერთ დირექტორიაში
cp file1.txt file2.txt file3.txt /home/student/backup/

# ვერბოზური რეჟიმი (აჩვენებს რას აკეთებს)
cp -v file.txt backup/

# ინტერაქტიული რეჟიმი (ეკითხება თუ ფაილი არსებობს)
cp -i file.txt existing_file.txt
```

2.2. დირექტორიების კოპირება

დირექტორიის კოპირებისთვის საჭიროა `-r` (recursive) ოპცია:

```
# დირექტორიის კოპირება ყველა შინაარსით
cp -r my_folder backup_folder

# დირექტორიის კოპირება სხვა ადგილას
cp -r documents /home/student/backup/

# ვერბოზური რეკურსიული კოპირება
cp -rv photos /media/usb/
```

2.3. სასარგებლო ოპციები

```
# -r = რეკურსიული (დირექტორიებისთვის)
# -v = ვერბოზური (აჩვენებს პროცესს)
# -i = ინტერაქტიული (ეკითხება დადასტურებას)
# -u = განახლებს მხოლოდ ახალ ფაილებს
# -p = იცავს ატრიბუტებს (უფლებები, დრო)
```

```
# ოპციების კომბინაცია
```

```
cp -riv source_folder destination_folder
```

3. ფაილების გადატანა/გადარქმევა - mv

ბრძანება `mv` (move) გადააქვს ფაილს ან არქმევს მას.

სინტაქსი:

```
mv საიდან სად
```

3.1. ფაილის გადარქმევა

```
# ფაილის გადარქმევა
mv old_name.txt new_name.txt
```

```
# დირექტორიის გადარქმევა
mv old_folder new_folder
```

3.2. ფაილის გადატანა

```
# ფაილის გადატანა სხვა დირექტორიაში
mv file.txt /home/student/documents/
```

```
# ფაილის გადატანა და გადარქმევა ერთდროულად
mv file.txt /home/student/documents/renamed_file.txt
```

```
# რამდენიმე ფაილის გადატანა
mv file1.txt file2.txt file3.txt /home/student/backup/
```

```
# დირექტორიის გადატანა (რეკურსიული ოპცია არ სჭირდება)
mv my_folder /home/student/backup/
```

3.3. სასარგებლო ოპციები

```
# -v = ვერბოზური (აჩვენებს რას აკეთებს)
mv -v file.txt documents/

# -i = ინტერაქტიული (ეკითხება თუ ფაილი არსებობს)
mv -i file.txt existing_file.txt

# -u = გადააქვს მხოლოდ თუ ფაილი უფრო ახალია
mv -u file.txt backup/

# -n = არ გადააწერს არსებულ ფაილს
mv -n file.txt backup/
```

ხშირი შეცდომა:

```
# არასწორი - გადააწერს file2.txt-ს
mv file1.txt file2.txt

# თუ გინდათ ორივე ფაილი შევინახოთ:
mv file1.txt file2.txt backup/ # ორივე გადაინაცვლებს backup-ში
```

4. ფაილების წაშლა - rm

ბრძანება `rm` (remove) შლის ფაილებს და დირექტორიებს.

⚠ მნიშვნელოვანი გაფრთხილება: Linux-ში წაშლილი ფაილები არ გადადის "სანაგვე ყუთში" - ისინი სამუდამოდ იშლება!

სინტაქსი:

```
rm ფაილის_სახელი
```

4.1. ფაილების წაშლა

```
# ერთი ფაილის წაშლა
rm file.txt

# რამდენიმე ფაილის წაშლა
rm file1.txt file2.txt file3.txt

# ყველა .txt ფაილის წაშლა
rm *.txt

# ყველა .jpg და .png ფაილის წაშლა
rm *.jpg *.png

# ინტერაქტიული წაშლა (ეკითხება დადასტურებას)
rm -i file.txt

# ვერბოზური რეჟიმი
rm -v file.txt
```

4.2. დირექტორიების წაშლა

ცარიელი დირექტორიის წაშლა - rmdir:

```
# ცარიელი დირექტორიის წაშლა
rmdir empty_folder

# რამდენიმე ცარიელი დირექტორიის წაშლა
rmdir folder1 folder2 folder3
```

არაცარიელი დირექტორიის წაშლა - rm -r:

```
# დირექტორიის წაშლა ყველა შინაარსით
rm -r my_folder

# დირექტორიის წაშლა დადასტურების გარეშე
rm -rf my_folder

# ინტერაქტიული დირექტორიის წაშლა
rm -ri my_folder

# ვერბოზური რეკურსიული წაშლა
rm -rv my_folder
```

4.3. სასარგებლო ოპციები და უსაფრთხოება

```
# -r = რეკურსიული (დირექტორიებისთვის)
# -f = ძალით (force), დადასტურების გარეშე
# -i = ინტერაქტიული (ეკითხება თითოეული ფაილისთვის)
# -v = ვერბოზური (აჩვენებს რას შლის)

# უსაფრთხო კომბინაცია (რეკომენდებული დამწყებთათვის)
rm -riv my_folder # რეკურსიული, ინტერაქტიული, ვერბოზური
```

⚠ საშიშარი ბრძანებები - არასოდეს გამოიყენოთ:

```
# ეს ბრძანებები შლის მთელ სისტემას!
rm -rf / # წაშლის მთელ სისტემას
rm -rf /* # წაშლის ყველაფერს root დირექტორიაში
rm -rf ~/* # წაშლის ყველა თქვენს ფაილს
```

5. ფაილების ნახვა

5.1. ფაილის შინაარსის ნახვა - cat

```
# ფაილის სრული შინაარსის ნახვა
cat file.txt

# რამდენიმე ფაილის შინაარსის ნახვა
cat file1.txt file2.txt

# ნომრებით ხაზები
cat -n file.txt
```

5.2. დიდი ფაილების ნახვა - less და more

```
# less (უფრო მოსახერხებელი)
less file.txt
# Space = შემდეგი გვერდი
# b = წინა გვერდი
# q = გასვლა

# more (ძველი ვერსია)
more file.txt
# Space = შემდეგი გვერდი
# q = გასვლა
```

5.3. ფაილის დასაწყისის/დასასრულის ნახვა

```
# პირველი 10 ხაზი
head file.txt

# პირველი 20 ხაზი
head -n 20 file.txt

# ბოლო 10 ხაზი
tail file.txt

# ბოლო 30 ხაზი
tail -n 30 file.txt

# ფაილის რეალურ დროში თვალყურის დევნება (ლოგებისთვის)
tail -f logfile.txt
```

6. დირექტორიებში ნავიგაცია

6.1. მიმდინარე დირექტორიის ნახვა - pwd

```
# აჩვენებს სად იმყოფებით
pwd
# მაგალითად: /home/student
```

6.2. დირექტორიის შიგთავსის ნახვა - ls

```
# მარტივი სია
ls

# დეტალური ინფორმაცია
ls -l

# ყველა ფაილი (დამალულების ჩათვლით)
ls -a

# დეტალური + ყველა ფაილი
ls -la

# ადამიანურად წაკითხვადი ზომები
ls -lh

# თარიღით დალაგება (ყველაზე ახალი ბოლოში)
ls -lt

# ზომის მიხედვით დალაგება
ls -ls

# კონკრეტული ფაილების ნახვა
ls *.txt
ls file*

# კონკრეტული დირექტორიის შიგთავსი
ls /home/student/documents
```

6.3. დირექტორიის შეცვლა - cd

```
# home დირექტორიაში გადასვლა
cd
cd ~

# კონკრეტულ დირექტორიაში გადასვლა
cd /home/student/documents

# ერთი დონით ზემოთ ასვლა
cd ..

# ორი დონით ზემოთ ასვლა
cd ../..

# წინა დირექტორიაში დაბრუნება
cd -

# შედარებითი გზა (მიმდინარე დირექტორიიდან)
cd documents/projects

# აბსოლუტური გზა (root-დან)
cd /home/student/documents/projects
```

7. ჯოკერები (Wildcards)

ჯოკერები გეხმარებათ იმუშაოთ რამდენიმე ფაილთან ერთდროულად.

7.1. * - ნებისმიერი სიმბოლო(ები)

```
# ყველა .txt ფაილი
ls *.txt

# file-ით დაწყებული ყველა ფაილი
ls file*

# ყველა ფაილი რომელიც შეიცავს "test"-ს
ls *test*

# ყველა .jpg და .png ფაილის წაშლა
rm *.jpg *.png

# ყველა .txt ფაილის კოპირება
cp *.txt backup/
```

7.2. ? - ერთი ნებისმიერი სიმბოლო

```
# file1.txt, file2.txt, fileA.txt (მაგრამ არა file10.txt)
ls file?.txt

# test_1.txt, test_2.txt, test_a.txt
ls test_?.txt
```

7.3. [] - სიმბოლო დიაპაზონიდან

```
# file1.txt, file2.txt, file3.txt
ls file[123].txt

# file1.txt-დან file9.txt-მდე
ls file[1-9].txt

# fileA.txt-დან fileZ.txt-მდე
ls file[A-Z].txt

# file1.txt, file2.txt, filea.txt, fileb.txt
ls file[12ab].txt
```

პრაქტიკული მაგალითები:

```
# ყველა 2024 წლის ფაილი
ls report_2024*.txt

# test_1.txt-დან test_9.txt-მდე
ls test_[1-9].txt

# ყველა .txt ფაილის backup-ში კოპირება
cp *.txt backup/

# ყველა .tmp ფაილის წაშლა
rm *.tmp

# ყველა jpg, jpeg, png ფაილის გადატანა images ფოლდერში
mv *.jpg *.jpeg *.png images/
```

8. ფაილების უფლებები

Linux-ში თითოეულ ფაილს აქვს უფლებები: წაკითხვა (r), ჩაწერა (w), შესრულება (x).

8.1. უფლებების ნახვა

```
ls -l file.txt
# -rw-r--r-- 1 student student 1234 Jan 15 10:30 file.txt

# -rw-r--r-- ნიშნავს:
# - = რეგულარული ფაილი (d = დირექტორია)
# rw- = მფლობელს შეუძლია წაკითხვა და ჩაწერა
# r-- = ჯგუფს შეუძლია მხოლოდ წაკითხვა
# r-- = სხვებს შეუძლია მხოლოდ წაკითხვა
```

8.2. უფლებების შეცვლა - chmod

```
# ფაილის გახადეთ შესასრულებელი
chmod +x script.sh

# მხოლოდ მფლობელს მიეცით ჩაწერის უფლება
chmod 644 file.txt

# დირექტორიას და ყველა შინაარსს მიეცით უფლებები
chmod -R 755 my_folder
```

რიცხვითი უფლებები:

- 7 = rwx (წაკითხვა + ჩაწერა + შესრულება)
- 6 = rw- (წაკითხვა + ჩაწერა)
- 5 = r-x (წაკითხვა + შესრულება)
- 4 = r-- (მხოლოდ წაკითხვა)
- 0 = --- (არანაირი უფლება)

9. ხშირად დასმული კითხვები (FAQ)

1. როგორ დავაკოპირო ფაილი სხვა დირექტორიაში?

```
cp file.txt /path/to/destination/
```

2. როგორ წავშალო დირექტორია ყველა შინაარსით?

```
rm -r directory_name
```

3. როგორ გადავარქვა ფაილს?

```
mv old_name.txt new_name.txt
```

4. როგორ ვნახო დამალული ფაილები?

```
ls -a
```

5. როგორ შევქმნა რამდენიმე დირექტორია ერთდროულად?

```
mkdir folder1 folder2 folder3
```

6. რა განსხვავებაა cp და mv შორის?

- `cp` - აკოპირებს (ორიგინალი რჩება)
- `mv` - გადააქვს (ორიგინალი იშლება)

7. როგორ გავაუქმო ბრძანება? დააჭირეთ `Ctrl+C`

8. როგორ დავბრუნდე წინა დირექტორიაში?

```
cd -
```

9. როგორ შევქმნა ფაილი ტექსტით ერთი ბრძანებით?

```
echo "ტექსტი" > file.txt
```

10. რამდენად საშიშია `rm -rf`? ძალიან საშიშარია! იგი შლის ყველაფერს დადასტურების გარეშე. ყოველთვის გამოიყენეთ `-i` ოპცია: `rm -ri folder`

10. სასარგებლო რჩევები

Tab ავტოშევსება

დაწერეთ ფაილის/დირექტორიის სახელის დასაწყისი და დააჭირეთ Tab:

```
cd doc[Tab] # ავტომატურად შეავსებს "documents"
cat myfi[Tab] # ავტომატურად შეავსებს "myfile.txt"
```

ბრძანებების ისტორია

```
# ბოლო ბრძანებების ნახვა
history

# ბოლო 20 ბრძანება
history 20

# ისტორიიდან ძებნა
Ctrl+R # შემდეგ დაწერეთ ძებნის სიტყვა

# წინა ბრძანების გამეორება
!!

# კონკრეტული ბრძანების გამეორება
!номер # მაგ: !342
```

მრავალი ბრძანება ერთ ხაზზე

```
# ; - გაუშვით ერთის მეორის მიყოლებით
mkdir test; cd test; touch file.txt

# && - გაუშვით მხოლოდ თუ წინა ბრძანება წარმატებული იყო
mkdir test && cd test && touch file.txt

# || - გაუშვით თუ წინა ბრძანება წარუმატებელი იყო
mkdir test || echo "დირექტორია უკვე არსებობს"
```

გზების შემოკლებები

```
~ # home დირექტორია (/home/student)
. # მიმდინარე დირექტორია
.. # მშობელი დირექტორია
- # წინა დირექტორია
/ # root დირექტორია
```

ბრძანების გაუქმება

```
Ctrl+C # გაუშვებული ბრძანების გაუქმება
Ctrl+Z # პროცესის შეჩერება (არა გაუქმება)
Ctrl+D # ტერმინალიდან გასვლა
```

Linux-ში რეგისტრი მნიშვნელოვანია:

- `File.txt` ≠ `file.txt` ≠ `FILE.txt` - ეს სამი განსხვავებული ფაილია!

11. შეჯამება

ამ სახელმძღვანელოში შეისწავლეთ Linux-ის ფაილებთან მუშაობის ძირითადი ბრძანებები:

ფაილების შექმნა:

- `touch file.txt` - ცარიელი ფაილის შექმნა
- `mkdir folder` - დირექტორიის შექმნა
- `echo "text" > file.txt` - ფაილი ტექსტით

ფაილების კოპირება:

- `cp file.txt copy.txt` - ფაილის კოპირება
- `cp -r folder backup` - დირექტორიის კოპირება

ფაილების გადატანა/გადარქმევა:

- `mv old.txt new.txt` - გადარქმევა
- `mv file.txt /path/` - გადატანა

ფაილების წაშლა:

- `rm file.txt` - ფაილის წაშლა
- `rm -r folder` - დირექტორიის წაშლა
- `rmdir empty_folder` - ცარიელი დირექტორიის წაშლა

ფაილების ნახვა:

- `cat file.txt` - შინაარსის ნახვა
- `less file.txt` - დიდი ფაილის ნახვა
- `head file.txt` / `tail file.txt` - დასაწყისი/დასასრული

ნავიგაცია:

- `pwd` - მიმდინარე დირექტორია
- `ls` - დირექტორიის შიგთავსი
- `cd folder` - დირექტორიის შეცვლა

ჯოკერები:

- `*` - ნებისმიერი სიმბოლოები
- `?` - ერთი სიმბოლო
- `[]` - დიაპაზონი

რჩევები პრაქტიკისთვის:

1. ივარჯიშეთ ხშირად - შექმენით სატესტო დირექტორია და ექსპერიმენტები ჩაატარეთ
2. დაიწყეთ მარტივიდან - ჯერ ერთი ფაილი, შემდეგ რამდენიმე
3. გამოიყენეთ `-i` ოპცია - დამწყებთათვის ინტერაქტიული რეჟიმი უსაფრთხოა
4. გააკეთეთ backup - მნიშვნელოვანი ფაილების კოპირება ყოველთვის
5. წაკითხეთ შეცდომის შეტყობინებები - ისინი გეტყვიან რა არის პრობლემა

თუ რამე გაუგებარია - არ შეგეშინდეთ შეცდომების. სწორედ შეცდომებით ვსწავლობთ. შექმენით სატესტო დირექტორია და ივარჯიშეთ ყველა ბრძანებაზე!