

Linux-ის ls ბრძანება

სარჩევი

1. ls ბრძანების ძირითადი გამოყენება
2. ოპცია -l (დეტალური სია)
 - o 2.1 შედეგის ინტერპრეტაცია
 - o 2.2 უფლებების გაშიფრვა
3. ოპცია -a (ყველა ფაილი)
4. ოპცია -h (ადამიანისთვის გასაგები ფორმატი)
5. ოპცია -R (რეკურსიული ჩვენება)
6. ოპცია -t (დროის მიხედვით დალაგება)
7. ოპცია -S (ზომის მიხედვით დალაგება)
8. ოპცია -r (რევერსული დალაგება)
9. ოპცია -i (inode-ების ჩვენება)
 - o 9.1 რა არის inode?
 - o 9.2 inode-ების პრაქტიკული მაგალითები
10. ოპცია -d (საქაღალდის ინფორმაცია)
11. Wildcards (ნაცვალ სიმბოლოები)
12. ოპციების კომბინირება
13. ფერადი გამოსახულება
14. დამატებითი სასარგებლო ინტერაქციები

შესავალი

ls ბრძანება ერთ-ერთი ყველაზე ხშირად გამოყენებული ბრძანებაა Linux-ში. მისი სახელი მოდის ინგლისური სიტყვებიდან "list" (ჩამოთვლა). ეს ბრძანება გამოიყენება საქაღალდეებში არსებული ფაილებისა და ქვესაქაღალდეების სიის სანახავად.

ძირითადი სინტაქსი:

```
ls [ოპციები] [საქაღალდე]
```

თუ საქალალდეს არ მიუთითებთ, `ls` გამოაჩენს იმ საქალალდის შიგთავსს, სადაც ამჟამად იმყოფებით.

1. `ls` ბრძანების ძირითადი გამოყენება

უმარტივესი ფორმით, `ls` ბრძანება აჩვენებს მიმდინარე საქალალდის შიგთავს.

მაგალითი:

```
ls
```

შედეგი:

```
Desktop Documents Downloads Music Pictures Videos
```

ეს ბრძანება აჩვენებს ყველა ფაილს და საქალალდეს თქვენს მიმდინარე დირექტორიაში.

2. ოპცია -l (დეტალური სია)

ყველაზე მნიშვნელოვანი და ინფორმაციული ოპციაა `-l` (long format).

მაგალითი:

```
ls -l
```

შედეგი:

```
total 28
drwxr-xr-x 2 student student 4096 Jan 15 10:30 Desktop
drwxr-xr-x 3 student student 4096 Jan 14 14:22 Documents
drwxr-xr-x 2 student student 4096 Jan 16 09:15 Downloads
-rw-r--r-- 1 student student 245 Jan 10 12:05 notes.txt
-rwxr-xr-x 1 student student 8192 Jan 12 16:40 program.sh
```

2.1 შედეგის ინტერპრეტაცია

თითოეული ხაზი შეიცავს 7 ძირითად ელემენტს:

```
-rw-r--r-- 1 student student 245 Jan 10 12:05 notes.txt
```

① ② ③ ④ ⑤ ⑥ ⑦

1. **უფლებები** (-rw-r--r--) - ვის შეუძლია ფაილის წაკითხვა/ჩაწერა/გაშვება
2. **ბმულების რაოდენობა** (1) - რამდენი hard link არსებობს
3. **მფლობელი** (student) - ფაილის მფლობელი მომხმარებელი
4. **ჯგუფი** (student) - ფაილის ჯგუფი
5. **ზომა ბაიტებში** (245) - ფაილის ზომა
6. **თარიღი და დრო** (Jan 10 12:05) - ბოლო ცვლილების დრო
7. **სახელი** (notes.txt) - ფაილის ან საქაღალდის სახელი

2.2 უფლებების გაშიფრვა

```
-rw-r--r--
```

სხვების უფლებები (read)
ჯგუფის უფლებები (read)
მფლობელის უფლებები (read, write)
სპეციალური უფლებები
ფაილის ტიპი
(-) რეგულარული ფაილი, (d) საქაღალდე, (l) ბმული

აღნიშვნები:

- **r** = read (წაკითხვა)
- **w** = write (ჩაწერა)
- **x** = execute (გაშვება)
- **-** = უფლება არ არის მინიჭებული

3. ოპცია -a (ყველა ფაილი)

Linux-ში ფაილები, რომლებიც იწყება წერტილით (.), დამაღულია. ოპცია **-a** აჩვენებს ყველა ფაილს, დამაღულების ჩათვლით.

მაგალითი:

```
ls -a
```

შედეგი:

```
. .. .bash_history .bashrc Desktop Documents .hidden_folder notes.txt
```

განმარტება:

- `.` - მიმდინარე საქალალდე
- `..` - მშობელი საქალალდე
- `.bash_history`, `.bashrc` - დამალული ფაილები
- `.hidden_folder` - დამალული საქალალდე

4. ოპცია `-h` (ადამიანისთვის გასაგები ფორმატი)

როდესაც `-l` -ს ვიყენებთ, ფაილის ზომა ბაიტებშია. `-h` ოპცია გადააქცევს მას უფრო გასაგებ ფორმატში.

მაგალითი:

```
ls -lh
```

შედეგი:

```
total 28K
drwxr-xr-x 2 student student 4.0K Jan 15 10:30 Desktop
-rw-r--r-- 1 student student 245 Jan 10 12:05 notes.txt
-rw-r--r-- 1 student student 1.5M Jan 12 14:20 photo.jpg
-rwxr-xr-x 1 student student 8.0K Jan 12 16:40 program.sh
```

ახლა ზომები ნაჩვენებია როგორც: **K** (კილობაიტი), **M** (მეგაბაიტი), **G** (გიგაბაიტი).

5. ოპცია `-R` (რეკურსიული ჩვენება)

ეს ოპცია აჩვენებს არა მხოლოდ მიმდინარე საქალალდის, არამედ ყველა ქვესაქალალდის შიგთავსსაც.

მაგალითი:

```
ls -R Documents
```

შედეგი:

Documents:

file1.txt Projects

Documents/Projects:

project1 project2

Documents/Projects/project1:

main.py README.md

6. ოპცია -t (დროის მიხედვით დალაგება)

ფაილები დალაგდება ბოლო ცვლილების თარიღის მიხედვით, სადაც ყველაზე ახალი ფაილი პირველი იქნება.

მაგალითი:

```
ls -lt
```

შედეგი:

```
-rw-r--r-- 1 student student 180 Jan 16 09:15 new_file.txt
-rw-r--r-- 1 student student 245 Jan 15 10:30 notes.txt
-rw-r--r-- 1 student student 1024 Jan 14 14:22 old_file.txt
```

7. ოპცია -S (ზომის მიხედვით დალაგება)

ფაილები დალაგდება ზომის მიხედვით, სადაც ყველაზე დიდი ფაილი პირველი იქნება.

მაგალითი:

```
ls -lhS
```

შედეგი:

```
-rw-r--r-- 1 student student 5.2M Jan 15 10:30 video.mp4
-rw-r--r-- 1 student student 1.5M Jan 12 14:20 photo.jpg
-rw-r--r-- 1 student student 8.0K Jan 12 16:40 program.sh
-rw-r--r-- 1 student student 245 Jan 10 12:05 notes.txt
```

8. ოპცია -r (რევერსული დალაგება)

ნებისმიერი დალაგება შეიძლება შევატრიალოთ `-r` ოპციით.

მაგალითი:

```
ls -ltr
```

ეს აჩვენებს ფაილებს თარიღის მიხედვით, მაგრამ უძველესი ფაილი პირველი იქნება.

9. ოპცია -i (inode-ების ჩვენება)

9.1 რა არის inode?

inode არის უნიკალური იდენტიფიკატორი, რომელიც Linux-ი ანიჭებს თითოეულ ფაილს ფაილურ სისტემაში. ეს არის თითქოს "პირადობის ნომერი" ფაილისთვის.

რისთვის გვჭირდება inode:

- ფაილის უნიკალურად იდენტიფიცირება
- Hard links-ის იდენტიფიცირება (ორი ფაილი ერთი inode-ით არის ერთი და იგივე ფაილი)
- სისტემური პრობლემების დიაგნოსტიკა

მაგალითი:

```
ls -li
```

შედეგი:

```
1048577 drwxr-xr-x 2 student student 4096 Jan 15 10:30 Desktop
1048578 drwxr-xr-x 3 student student 4096 Jan 14 14:22 Documents
1048580 -rw-r--r-- 1 student student 245 Jan 10 12:05 notes.txt
1048582 -rwxr-xr-x 1 student student 8192 Jan 12 16:40 program.sh
```

9.2 inode-ების პრაქტიკული მაგალითები

მაგალითი 1: ორიგინალური ფაილის inode

```
# შევქმნათ ფაილი
echo "Test content" > original.txt

# შევამოწმოთ inode
ls -li original.txt
```

შედეგი:

```
1048585 -rw-r--r-- 1 student student 13 Jan 16 10:00 original.txt
```

მაგალითი 2: Hard Link და inode

```
# შევქმნათ hard link
ln original.txt hardlink.txt

# შევამოწმოთ ორივე ფაილის inode
ls -li original.txt hardlink.txt
```

შედეგი:

```
1048585 -rw-r--r-- 2 student student 13 Jan 16 10:00 hardlink.txt
1048585 -rw-r--r-- 2 student student 13 Jan 16 10:00 original.txt
```

დაკვირდით: ორივე ფაილს აქვს ერთი და იგივე inode (1048585) - ეს ნიშნავს, რომ ისინი არიან ერთი და იგივე ფიზიკური ფაილი!

მაგალითი 3: Symbolic Link და inode

```
# შევქმნათ symbolic link  
ln -s original.txt symlink.txt  
  
# შევამოწმოთ ორივე ფაილის inode  
ls -li original.txt symlink.txt
```

შედეგი:

```
1048585 -rw-r--r-- 2 student student 13 Jan 16 10:00 original.txt  
1048590 lrwxrwxrwx 1 student student 12 Jan 16 10:01 symlink.txt -> original.txt
```

დაკვირდით: symlink-ს აქვს განსხვავებული inode (1048590) - ეს ახალი ფაილია, რომელიც მიუთითებს original.txt-ზე.

10. ოპცია -d (საქაღალდის ინფორმაცია)

ჩვეულებრივ `ls -l Documents` აჩვენებს რა არის Documents-ში. `-d` აჩვენებს თავად Documents-ის ინფორმაციას.

მაგალითი:

```
ls -ld Documents
```

შედეგი:

```
drwxr-xr-x 3 student student 4096 Jan 14 14:22 Documents
```

11. Wildcards (ნაცვალ სიმბოლოები)

შეგიძლიათ გამოიყენოთ სპეციალური სიმბოლოები ფაილების ფილტრაციისთვის.

მაგალითები:

```
# ყველა .txt ფაილი  
ls *.txt
```

შედეგი: notes.txt report.txt summary.txt

```
# ყველა ფაილი, რომელიც იწყება 'test'-ით  
ls test*
```

შედეგი: test1.py test2.py test_file.txt

```
# ყველა ფაილი, რომელიც სრულდება '.py'-ით  
ls *.py
```

შედეგი: main.py test1.py test2.py

```
# ყველა ფაილი, რომლის სახელში აქვს 'data'  
ls *data*
```

შედეგი: data.csv mydata.txt user_data.json

12. ოპტიკური კომპინირება

ოპტიკური შეიძლება გავაერთიანოთ ერთ ბრძანებაში.

მაგალითები:

```
# დეტალური, ყველა ფაილი, ადამიანისთვის გასაგები ზომა  
ls -lah

# დეტალური, დროის მიხედვით დალაგებული, რევერსული  
ls -ltr

# ყველა ფაილი, რეკურსიული  
ls -aR

# დეტალური, ყველა ფაილი, inode-ები, ადამიანისთვის გასაგები  
ls -lahi
```

უმსხვილესი ფაილების საპოვნელად:

```
# 10 ყველაზე დიდი ფაილი მიმდინარე საქაღალდეში  
ls -lhS | head -n 11
```

შედეგი:

```
total 25M
-rw-r--r-- 1 student student 10M Jan 15 10:30 bigvideo.mp4
-rw-r--r-- 1 student student 5.2M Jan 15 09:20 video.mp4
-rw-r--r-- 1 student student 3.8M Jan 14 14:22 archive.zip
...
```

13. ფურადი გამოსახულება

Ubuntu 18.04-ში, როდესაც იყენებთ `ls --color=auto`, ფაილები სხვადასხვა ფერითაა:

- **ლურჯი** = საქაღალდეები
- **შავი** = რეგულარული ფაილები
- **მწვანე** = გამშვები ფაილები
- **შითელი** = დაარქივებული ფაილები (.zip, .tar)
- **ლი ლურჯი** = symbolic links

Ubuntu 18.04-ში `ls` ბრძანება ავტომატურად იყენებს `--color=auto` ოფციას.

14. დამატებითი სასარგებლო ოპციები

ოპცია -1 (ერთი ფაილი თითო ხაზზე):

```
ls -1
```

შედეგი:

```
Desktop
Documents
Downloads
notes.txt
program.sh
```

ოპცია --help (დახმარება):

```
ls --help
```

აჩვენებს ყველა შესაძლო ოპციას დეტალური აღწერით.

ოპცია `--version` (ვერსია):

```
ls --version
```

შედეგი:

```
ls (GNU coreutils) 8.28
```

სასარგებლო რჩევები

1. უფლებების გაგება: მფლობელს შეუძლია წაკითხვა და ჩაწერა (`rw-`), ჯგუფსა და სხვებს - მხოლოდ წაკითხვა (`r--`).
2. დამალული ფაილების ძებნა: ყოველთვის გამოიყენეთ `-a` ოპცია, თუ დარწმუნებული არ ხართ არის თუ არა დამალული ფაილები.
3. საქაღალდის ინფორმაცია vs შიგთავსი:
 - o `ls -l Documents` - აჩვენებს რა არის Documents-ში
 - o `ls -ld Documents` - აჩვენებს თავად Documents-ის ინფორმაციას
4. ყველაზე ხშირად გამოყენებული კომბინაცია: `ls -lah` - აჩვენებს დეტალურ სის ყველა ფაილზე (დამალულების ჩათვლით) გასაგები ფორმატით.
5. inode-ების გამოყენება: გამოიყენეთ `ls -li` hard links-ის იდენტიფიცირებისთვის - ერთი inode ნიშნავს ერთი და იგივე ფიზიკურ ფაილს.

შეჯამება

`ls` ბრძანება არის თქვენი მთავარი ინსტრუმენტი ფაილური სისტემის ნავიგაციისთვის. აი ყველაზე მნიშვნელოვანი ოპციები:

ოპცია	დანიშნულება
-l	დეტალური ინფორმაცია
-a	ყველა ფაილი (დამალულების ჩათვლით)
-h	ადამიანისთვის გასაგები ზომები
-R	რეკურსიული ჩვენება
-t	დროის მიხედვით დალაგება
-s	ზომის მიხედვით დალაგება
-r	რევერსული დალაგება
-i	inode-ების ჩვენება
-d	საქაღალდის საკუთარი ინფორმაცია
-1	ერთი ფაილი თითო ხაზზე