

# მომხმარებლის შექმნა და უფლებების მინიჭება Linux-ში

## სარჩევი

- შესავალი
- მომხმარებლის ცნება Linux-ში
- მომხმარებლის შექმნა
  - 3.1. useradd ბრძანება
  - 3.2. adduser ბრძანება
  - 3.3. განსხვავება useradd და adduser-ს შორის
- მომხმარებლის ინფორმაციის ნახვა
- მომხმარებლის პაროლის დაყენება
- მომხმარებლის წაშლა
- ჯგუფები Linux-ში
  - 7.1. ჯგუფის შექმნა
  - 7.2. მომხმარებლის ჯგუფში დამატება
  - 7.3. ჯგუფიდან მომხმარებლის წაშლა
- sudo უფლებების მინიჭება
  - 8.1. sudo ჯგუფში დამატება
  - 8.2. sudoers ფაილის რედაქტირება
- ფაილების და დირექტორიების უფლებები
  - 9.1. უფლებების ნახვა
  - 9.2. უფლებების შეცვლა chmod-ით
  - 9.3. მფლობელის შეცვლა chown-ით
- სასარგებლო რჩევები
- შეჯამება

## 1. შესავალი

Linux ოპერაციული სისტემა არის მრავალმომხმარებლიანი სისტემა, რაც ნიშნავს რომ ერთ კომპიუტერზე შეუძლია მუშაობდეს რამდენიმე მომხმარებელი. თითოეულ მომხმარებელს აქვს საკუთარი სახელი, პაროლი და უფლებები. ამ

თემაში ჩვენ ვისწავლით როგორ შევქმნათ მომხმარებლები, როგორ მივანიჭოთ მათ უფლებები და როგორ მოვახდინოთ სისტემის უსაფრთხო მართვა.

## 2. მომხმარებლის ცნება Linux-ში

Linux-ში არსებობს სამი ძირითადი ტიპის მომხმარებელი:

- **root** - სუპერმომხმარებელი, აქვს სრული უფლებები სისტემაზე
- **ჩვეულებრივი მომხმარებლები** - რეგულარული მომხმარებლები შეზღუდული უფლებებით
- **სისტემური მომხმარებლები** - სერვისებისა და პროცესებისთვის შექმნილი

თითოეულ მომხმარებელს აქვს უნიკალური ID (UID). root-ის UID არის 0.

## 3. მომხმარებლის შექმნა

### 3.1. useradd ბრძანება

`useradd` არის დაბალი დონის ბრძანება მომხმარებლის შესაქმნელად.

სინტაქსი:

```
sudo useradd [ოპციები] მომხმარებლის_სახელი
```

მაგალითი:

```
sudo useradd giorgi
```

ეს ბრძანება ქმნის მომხმარებელს სახელად "giorgi", მაგრამ:

- არ ქმნის home დირექტორიას
- არ აყენებს პაროლს
- არ აყენებს default shell-ს

სრული ვარიანტი ოპციებით:

```
sudo useradd -m -s /bin/bash -c "Giorgi Beridze" giorgi
```

ოპციების განმარტება:

- `-m` - ქმნის home დირექტორიას (/home/giorgi)
- `-s /bin/bash` - აყენებს bash-ს როგორც default shell-ს
- `-c "Giorgi Beridze"` - დააკომენტარების ველი (სრული სახელი)

ივარჯიშეთ:

```
sudo useradd -m -s /bin/bash student1
```

## 3.2. adduser ბრძანება

`adduser` არის მაღალი დონის, ინტერაქტიული ბრძანება Ubuntu-ზე.

მაგალითი:

```
sudo adduser student2
```

ეს ბრძანება:

- ავტომატურად ქმნის home დირექტორიას
- ითხოვს პაროლის შეყვანას
- ითხოვს დამატებით ინფორმაციას (სახელი, ტელეფონი და ა.შ.)
- აყენებს ყველა საჭირო პარამეტრს

ივარჯიშეთ:

```
sudo adduser student2
```

შეიყვანეთ პაროლი და შეავსეთ ინფორმაცია (შეგიძლიათ Enter-ით გამოტოვოთ).

## 3.3. განსხვავება useradd და adduser-ს შორის

useradd	adduser
დაბალი დონის ბრძანება	მაღალი დონის ბრძანება
მოითხოვს ოპციებს	ავტომატური კონფიგურაცია
არ არის ინტერაქტიული	ინტერაქტიულია
სწრაფი	მოსახერხებელი

**რჩევა:** დამწყებთათვის უმჯობესია `adduser` -ის გამოყენება.

## 4. მომხმარებლის ინფორმაციის ნახვა

მიმდინარე მომხმარებლის ნახვა:

```
whoami
```

ყველა მომხმარებლის სია:

```
cat /etc/passwd
```

კონკრეტული მომხმარებლის ინფორმაცია:

```
id student1
```

გამოსავალი მაგალითი:

```
uid=1001(student1) gid=1001(student1) groups=1001(student1)
```

ივარჯიშეთ:

```
whoami  
id student1  
id student2
```

## 5. მომხმარებლის პაროლის დაყენება

---

თუ `useradd` -ით შექმენით მომხმარებელს, საჭიროა პაროლის დაყენება:

```
sudo passwd student1
```

სისტემა ორჯერ მოგთხოვთ პაროლის შეყვანას.

საკუთარი პაროლის შეცვლა:

```
passwd
```

ივარჯიშეთ:

```
sudo passwd student1
```

შეიყვანეთ პაროლი: `Test1234`

---

## 6. მომხმარებლის წაშლა

---

მომხმარებლის წაშლა (home დირექტორიის შენარჩუნებით):

```
sudo userdel student1
```

მომხმარებლის წაშლა home დირექტორიით:

```
sudo userdel -r student1
```

ოპცია `-r` - შლის home დირექტორიას და mail spool-ს.

გაფრთხილება: წაშლილი მომხმარებლის აღდგენა შეუძლებელია!

---

## 7. ჯგუფები Linux-ში

---

ჯგუფები საშუალებას გაძლევთ რამდენიმე მომხმარებელს მიანიჭოთ ერთნაირი უფლებები.

## 7.1. ჯგუფის შექმნა

```
sudo groupadd developers
```

ჯგუფების სია:

```
cat /etc/group
```

ან:

```
getent group
```

ივარჯიშეთ:

```
sudo groupadd developers
sudo groupadd testers
```

## 7.2. მომხმარებლის ჯგუფში დამატება

არსებული მომხმარებლის ჯგუფში დამატება:

```
sudo usermod -aG developers student2
```

ოპციების განმარტება:

- `-a` - append (დაამატე, არ ჩაანაცვლო)
- `-G` - დამატებითი ჯგუფები

გაფრთხილება: თუ `-a` არ გამოიყენებთ, მომხმარებელი ყველა სხვა ჯგუფიდან წაიშლება!

მომხმარებლის ჯგუფების ნახვა:

```
groups student2
```

ივარჯიშეთ:

```
sudo usermod -aG developers student2  
groups student2
```

### 7.3. ჯგუფიდან მომხმარებლის წაშლა

```
sudo gpasswd -d student2 developers
```

ჯგუფის წაშლა:

```
sudo groupdel testers
```

---

## 8. sudo უფლებების მინიჭება

`sudo` (Super User DO) საშუალებას აძლევს ჩვეულებრივ მომხმარებელს შეასრულოს ბრძანებები root უფლებებით.

### 8.1. sudo ჯგუფში დამატება

Ubuntu-ზე `sudo` ჯგუფის წევრებს აქვთ sudo უფლებები:

```
sudo usermod -aG sudo student2
```

შემოწმება:

```
groups student2
```

უნდა დაინახოთ `sudo` ჯგუფში.

ივარჯიშეთ:

```
sudo usermod -aG sudo student2  
groups student2
```

### 8.2. sudoers ფაილის რედაქტირება

უფრო მოქნილი კონფიგურაციისთვის გამოიყენეთ:

```
sudo visudo
```

ეს ბრძანება უსაფრთხოდ ხსნის `/etc/sudoers` ფაილს რედაქტირებისთვის.

**მაგალითი - კონკრეტული ბრძანების ნებართვა:**

```
student2 ALL=(ALL) /usr/bin/apt-get
```

ეს ნიშნავს: student2-ს შეუძლია მხოლოდ apt-get ბრძანების გაშვება sudo-თი.

**მაგალითი - პაროლის გარეშე sudo:**

```
student2 ALL=(ALL) NOPASSWD: ALL
```

**გაფრთხილება:** ეს უსაფრთხოების რისკია! გამოიყენეთ მხოლოდ საჭიროების შემთხვევაში.

---

## 9. ფაილების და დირექტორიების უფლებები

---

### 9.1. უფლებების ნახვა

```
ls -l
```

გამოსავალი მაგალითი:

```
-rw-r--r-- 1 student2 student2 0 თებ  5 10:30 file.txt  
drwxr-xr-x 2 student2 student2 4096 თებ  5 10:31 folder
```

**განმარტება:**

- პირველი სიმბოლო: `-` (ფაილი) ან `d` (დირექტორია)
- შემდეგი 9 სიმბოლო: უფლებები (rwx rwx rwx)
  - `r` - read (წაკითხვა)
  - `w` - write (ჩაწერა)
  - `x` - execute (გაშვება)



- პირველი ჯგუფი (rwx) - მფლობელის უფლებები
- მეორე ჯგუფი (r-x) - ჯგუფის უფლებები
- მესამე ჯგუფი (r-x) - სხვების უფლებები

ივარჯიშეთ:

```
touch testfile.txt
ls -l testfile.txt
```

## 9.2. უფლებების შეცვლა chmod-ით

სიმბოლური მეთოდი:

```
chmod u+x testfile.txt # მფლობელს ემატება execute უფლება
chmod g-w testfile.txt # ჯგუფს ეშლება write უფლება
chmod o+r testfile.txt # სხვებს ემატება read უფლება
chmod a+x testfile.txt # ყველას ემატება execute უფლება
```

სიმბოლოები:

- u - user (მფლობელი)
- g - group (ჯგუფი)
- o - others (სხვები)
- a - all (ყველა)

რიცხვითი მეთოდი:

თითოეული უფლება აქვს მნიშვნელობა:

- r = 4
- w = 2
- x = 1

```
chmod 755 testfile.txt
```

755-ის განმარტება:

- 7 (rwx) = 4+2+1 - მფლობელი
- 5 (r-x) = 4+0+1 - ჯგუფი
- 5 (r-x) = 4+0+1 - სხვები

ხშირად გამოყენებული კომბინაციები:

- 644 - rw-r--r-- (სტანდარტული ფაილისთვის)
- 755 - rwxr-xr-x (სკრიპტებისთვის)
- 777 - rwxrwxrwx (სრული უფლებები - არ არის რეკომენდებული!)

ივარჯიშეთ:

```
chmod 644 testfile.txt
ls -l testfile.txt
chmod 755 testfile.txt
ls -l testfile.txt
```

### 9.3. მფლობელის შეცვლა chown-ით

მფლობელის შეცვლა:

```
sudo chown student2 testfile.txt
```

მფლობელისა და ჯგუფის შეცვლა:

```
sudo chown student2:developers testfile.txt
```

მხოლოდ ჯგუფის შეცვლა:

```
sudo chown :developers testfile.txt
```

ან:

```
sudo chgrp developers testfile.txt
```

რეკურსიული შეცვლა (დირექტორია და ყველა შიგთავსი):

```
sudo chown -R student2:developers /home/student2/project
```

ივარჯიშეთ:

```
ls -l testfile.txt
sudo chown student2:developers testfile.txt
ls -l testfile.txt
```

## 10. სასარგებლო რჩევები

1. ყოველთვის იყენეთ `-a` ოპცია `usermod -aG` -სთან, თორემ მომხმარებელი სხვა ჯგუფებიდან წაიშლება.
2. `sudo` ჯგუფში დამატებისას მომხმარებელმა უნდა გამოვიდეს და ხელახლა შევიდეს სისტემაში, რომ ცვლილებები ძალაში შევიდეს.
3. არ გამოიყენოთ `chmod 777` - ეს უსაფრთხოების დიდი რისკია!
4. `visudo` გამოყენებისას თუ შეცდომა დაუშვით, რედაქტორი გაფრთხილებთ და არ შეინახავს არასწორ კონფიგურაციას.
5. `root`-ად პირდაპირ შესვლა არ არის რეკომენდებული. გამოიყენეთ `sudo`.
6. პაროლები უნდა იყოს ძლიერი: მინიმუმ 8 სიმბოლო, ასოები, რიცხვები და სპეციალური სიმბოლოები.
7. მომხმარებლის წაშლამდე დარწმუნდით, რომ მისი ფაილები საჭიროების შემთხვევაში დაარქივით.
8. უფლებების შემოწმება: რეგულარულად შეამოწმეთ მომხმარებლების უფლებები `id` და `groups` ბრძანებებით.

## 11. შეჯამება

ამ თემაში ვისწავლეთ:

- მომხმარებლის შექმნა `useradd` და `adduser` ბრძანებებით
- მომხმარებლის ინფორმაციის ნახვა და მართვა
- პაროლების დაყენება და შეცვლა
- მომხმარებლების წაშლა
- ჯგუფების შექმნა და მართვა
- მომხმარებლების ჯგუფებში დამატება და წაშლა

- sudo უფლებების მინიჭება
- ფაილებისა და დირექტორიების უფლებების ნახვა და შეცვლა `chmod` -ით
- მფლობელის შეცვლა `chown` -ით

Linux-ში მომხმარებლების და უფლებების სწორი მართვა არის სისტემის უსაფრთხოების საფუძველი. ივარჯიშეთ ამ ბრძანებებით Ubuntu 18.04 ტერმინალში და დაიმასსოვრეთ ძირითადი პრინციპები: მინიმალური საჭირო უფლებები, ძლიერი პაროლები და რეგულარული მონიტორინგი.