

ლინუქსის ფაილების უფლებები

სარჩევი

1. ფაილების უფლებების სამი ძირითადი ტიპი

- 1.1. წაკითხვა (Read - r)
- 1.2. ჩაწერა (Write - w)
- 1.3. გაშვება (Execute - x)

2. მომხმარებელთა სამი კატეგორია

3. უფლებების ნაწვა

- 3.1. ბრძანება `ls -l`
- 3.2. როგორ ვიკითხოთ ეს ინფორმაცია

4. უფლებების შეცვლა - `chmod` ბრძანება

- 4.1. სიმბოლური მეთოდი
- 4.2. რიცხვითი მეთოდი
- 4.3. რეკურსიული შეცვლა

5. მფლობელის შეცვლა - `chown` ბრძანება

6. ჯგუფის შეცვლა - `chgrp` ბრძანება

7. ტიპიური შეცდომები და მათი გამოსწორება

- 7.1. შეცდომა 1: "Permission denied"
- 7.2. შეცდომა 2: ფაილის წაშლა შეუძლებელია
- 7.3. შეცდომა 3: 777 უფლებების გამოყენება

8. სასარგებლო რჩევები

9. შეჯამება - სწრაფი ცნობარი

შესავალი

Linux-ში ყველა ფაილს და დირექტორიას აქვს უფლებები, რომლებიც განსაზღვრავენ თუ ვინ შეუძლია მათი წაკითხვა, ჩაწერა ან გაშვება. ეს არის უსაფრთხოების მთავარი მექანიზმი.

1. ფაილების უფლებების სამი ძირითადი ტიპი

1.1. წაკითხვა (Read - r)

- ფაილისთვის: შინაარსის ნახვის უფლება
- დირექტორიისთვის: შიგთავსის სიის ნახვის უფლება

1.2. ჩაწერა (Write - w)

- ფაილისთვის: შინაარსის შეცვლის/წაშლის უფლება
- დირექტორიისთვის: ფაილების შექმნის/წაშლის უფლება

1.3. გაშვება (Execute - x)

- ფაილისთვის: პროგრამის გაშვების უფლება
- დირექტორიისთვის: დირექტორიაში შესვლის უფლება

2. მომხმარებელთა სამი კატეგორია

ყველა ფაილისთვის უფლებები განსაზღვრულია სამი ჯგუფისთვის:

- მფლობელი (Owner/User - u) - ფაილის შემქმნელი
- ჯგუფი (Group - g) - მომხმარებელთა ჯგუფი
- სხვები (Others - o) - დანარჩენი ყველა

3. უფლებების ნახვა

3.1. ბრძანება `ls -l`

```
ls -l
```

შედეგის მაგალითი:

```
-rw-r--r-- 1 user group 1234 08:59 19 10:30 file.txt  
drwxr-xr-x 2 user group 4096 08:59 19 10:30 myfolder
```

3.2. როგორ ვიკითხოთ ეს ინფორმაცია:

პრაქტიკული მაგალითი:

- `rwx` = წაკითხვა + ჩაწერა + გაშვება (ყველაფერი შეუძლია)
 - `r--` = მხოლოდ წაკითხვა
 - `---` = არანაირი უფლება

4. უფლებების შეცვლა - chmod ბრძანება

4.1. სიმბოლური მეთოდი (მარტივი და გასაგები)

სინტაქსი:

chmod [ସବୁ] + [ରା ଲ୍ୟାପ୍ଟୋପ୍ଟା] ଫାଇଲ୍ୟ

მაგალითები:

```

# მფლობელს დავუმატოთ გაშვების უფლება
chmod u+x script.sh

# ჯგუფს წავართვათ ჩაწერის უფლება
chmod g-w file.txt

# სხვებს დავუმატოთ წაკითხვის უფლება
chmod o+r document.txt

# ყველას დავუმატოთ გაშვების უფლება
chmod a+x program

# ჯგუფს და სხვებს წავართვათ ყველა უფლება
chmod go-rwx private.txt

# მფლობელს დავუმატოთ წაკითხვა და ჩაწერა ერთად
chmod u+rwx file.txt

```

სიმბოლოების განმარტება:

- `u` = user (მფლობელი)
- `g` = group (ჯგუფი)
- `o` = others (სხვები)
- `a` = all (ყველა)
- `+` = უფლების დამატება
- `-` = უფლების წაშლა
- `=` = ზუსტად ეს უფლებების დაყენება

4.2. რიცხვითი მეთოდი (სწრაფი, მაგრამ მოითხოვს დამახსოვრებას)

თითოეულ უფლებას აქვს რიცხვითი მნიშვნელობა:

- `r (read) = 4`
- `w (write) = 2`
- `x (execute) = 1`

როგორ ვთვლით:

- `rwx = 4+2+1 = 7`
- `rw- = 4+2+0 = 6`
- `r-x = 4+0+1 = 5`

- `r--` = $4+0+0 = 4$
- `---` = $0+0+0 = 0$

მაგალითები:

```
# 755 = rwxr-xr-x (ტიპიური პროგრამისთვის)
chmod 755 script.sh

# 644 = rw-r--r-- (ტიპიური ფაილისთვის)
chmod 644 document.txt

# 700 = rwx----- (მხოლოდ მფლობელისთვის)
chmod 700 private_script.sh

# 777 = rwxrwxrwx (ყველას ყველაფერი - არასაიმედო!)
chmod 777 file.txt
```

ნშირად გამოყენებული კომბინაციები:

- `755` - პროგრამები, სკრიპტები (მფლობელი: ყველაფერი, სხვები: წაკითხვა+გაშვება)
- `644` - ჩვეულებრივი ფაილები (მფლობელი: წაკითხვა+ჩაწერა, სხვები: წაკითხვა)
- `700` - პირადი ფაილები (მხოლოდ მფლობელი)
- `600` - პირადი დოკუმენტები (მფლობელი: წაკითხვა+ჩაწერა)

4.3. რეკურსიული შეცვლა (დირექტორიებისთვის)

```
# ყველა ფაილისთვის მთელ დირექტორიაში
chmod -R 755 myfolder/

# -R ნიშნავს recursive - ყველა ქვედირექტორიასა და ფაილზეც
```

5. მფლობელის შეცვლა - `chown` ბრძანება

```
# მფლობელის შეცვლა  
sudo chown newuser file.txt  
  
# მფლობელისა და ჯგუფის შეცვლა ერთად  
sudo chown newuser:newgroup file.txt  
  
# რეკურსიულად  
sudo chown -R user:group myfolder/
```

6. ჯგუფის შეცვლა - chgrp ბრძანება

```
# ფაილის ჯგუფის შეცვლა  
sudo chgrp developers file.txt  
  
# რეკურსიულად  
sudo chgrp -R developers myfolder/
```

7. ტიპიური შეცდომები და მათი გამოსწორება

7.1. შეცდომა 1: "Permission denied"

```
# თუ გამოჩნდა ეს შეცდომა:  
bash: ./script.sh: Permission denied  
  
# გამოსწორება:  
chmod +x script.sh
```

7.2. შეცდომა 2: ფაილის წაშლა შეუძლებელია

```
# თუ ფაილი არ იშლება:  
rm: cannot remove 'file.txt': Permission denied  
  
# შეამოწმეთ დირექტორიის უფლებები და სცადეთ:  
chmod u+w file.txt  
rm file.txt
```

7.3. შეცდომა 3: 777 უფლებების გამოყენება (სახითაო!)

```
# არასწორი (ყველას აძლევს ყველა უფლებას):
chmod 777 important_file.txt

# სწორი:
chmod 644 important_file.txt # ჩვეულებრივი ფაილისთვის
# ან
chmod 755 script.sh # სკრიპტისთვის
```

8. სასარგებლო რჩევები

1. ყოველთვის შეამოწმეთ უფლებები `ls -l` ბრძანებით
2. არ გამოიყენოთ 777 უფლებები (უსაფრთხოების რისკი)
3. სკრიპტებს დაუმატეთ +x უფლება გასაშვებად
4. პირად ფაილებს დაუყენეთ 600 ან 700
5. sudo გამოიყენეთ მხოლოდ საჭიროების შემთხვევაში

9. შეჯამება - სტრატი ცნობარი

უფლება	სიმბოლო	რიცხვი	აღწერა
წაკითხვა	r	4	ფაილის წაკითხვა/დირექტორიის სიის ნახვა
ჩაწერა	w	2	ფაილის შეცვლა/დირექტორიაში ფაილების შექმნა
გაშვება	x	1	პროგრამის გაშვება/დირექტორიაში შესვლა

ხშირი ბრძანებები:

- `ls -l` - უფლებების ნახვა
- `chmod u+x file` - მფლობელს გაშვების უფლება
- `chmod 755 file` - სტანდარტული პროგრამის უფლებები
- `chmod 644 file` - სტანდარტული ფაილის უფლებები
- `sudo chown user file` - მფლობელის შეცვლა