

Operacijski sistemi

vaje 4

pravice

1. Kot uporabnik `miha` ustvarite datoteko `~/vaje/test.txt`. Kakšne pravice nad to datoteko imajo posamezni uporabniki? Preizkusite spreminjati vsebino datoteke kot drug uporabnik.
2. Vsem uporabnikom omogočite pisanje v datoteko `test.txt`.
3. Ostalim uporabnikom (razen sebi in skupini) vzemite bralne in pisalne pravice nad datoteko
4. Omogočite, da bo uporabnik `student` (brez administratorskih pravic) poleg vas (uporabnik `miha`) lahko pisal in bral vašo datoteko, ostali uporabniki v splošnem pa ne.
5. Vsem uporabnikom odvzemite vse pravice nad datoteko ter prenesite lastništvo in skupino datoteke na uporabnika `administrator`. Ali lahko kot uporabnik `miha` pogledate in spreminjate vsebino datoteke? Ali jo lahko zbrisete? Zakaj?
6. Kako bi poskrbeli, da lahko v imenik `vaje` vsi uporabniki shranjujejo in brišejo svoje datoteke, ne morejo pa brisati datotek ostalih uporabnikov?

pravice

Pretvorite iz osmiškega zapisa v simbolični in obratno:

a) 660

b) 720

c) 100

č) 1777

d) `rwxr-x--x`

e) `r--r--r--`

f) `r-xrw--wx`

g) `rw-rw-rwt`

spremenljivke

- deklaracija:

```
ime_spremenljivke=vrednost
```

- uporaba:

```
${ime_spremenljivke}  
$ime_spremenljivke
```

primeri:

```
a = os
```

```
a=os
```

```
a=operacijski sistemi
```

```
a="operacijski sistemi"
```

```
a='operacijski sistemi'
```

```
echo a
```

```
echo $a
```

```
echo "$a"
```

```
echo '$a'
```

vgrajene spremenljivke

<code>\$_</code>	... zadnji argument predhodno izvedenega ukaza
<code>\$#</code>	... število podanih argumentov
<code>\$0</code>	... ime skripte
<code>\$1 \$2 ... \$ {n}</code>	... zaporedni argumenti skripte
<code>\$?</code>	... izhodni status zadnjega (v ospredju) izvedenega ukaza
<code>\$\$</code>	... PID lupine
<code>!</code>	... PID procesa, ki je bil zadnji zagnan v ozadju
<code>* , @</code>	... vsi argumenti skripte skupaj
<code>-</code>	... opcije podane lupini, ki poganja skripto

test, [], [[]]

- stikala za preverjanje datotek:

-e datoteka	... Ali datoteka obstaja?
-d datoteka	... Ali je datoteka imenik?
-f datoteka	... Ali je datoteka navadna datoteka?
-s datoteka	... Ali velikost datoteke ni enaka 0?
-b datoteka	... Ali je datoteka bločno-orientirana?
-c datoteka	... Ali je datoteka znakovno-orientirana?
-h datoteka	... Ali je datoteka simbolična povezava?
-L datoteka	... Ali je datoteka simbolična povezava?
-r datoteka	... Ali lahko datoteko (kot trenutni uporabnik) beremo?
-w datoteka	... Ali lahko v datoteko (kot trenutni uporabnik) zapisujemo?
-x datoteka	... Ali lahko datoteko (kot trenutni uporabnik) poganjamo?

test, [], [[]]

- preverjanje enakosti med števili

`x -eq y` ... preverimo, ali je `x` enak `y`

`x -ne y` ... preverimo, ali je `x` različen od `y`

`x -gt y` ... preverimo, ali je `x` večji kot `y`

`x -ge y` ... preverimo, ali je `x` večji ali enak kot `y`

`x -lt y` ... preverimo, ali je `x` manjši od `y`

`x -le y` ... preverimo, ali je `x` manjši ali enak od `y`

test, [], [[]]

- preverjanje enakosti med nizi

$x = y$... preverimo, če je niz x enak nizu y

$x \neq y$... preverimo, če x ni enak y

$\neg x$... preverimo, če x ni ničeln (null)

$\neg z\ x$... preverimo, če je x ničeln (null)

naloge

7. Napišite skripto, ki preveri, ali smo skripto pognali kot skrbnik sistema "super uporabnik" (angl. superuser)! Če nismo, izpiše opozorilo in se zaključi. Če smo, požene ukaz, ki izpiše zadnjih 5 uspešnih in 5 neuspešnih prijav v sistem.
8. V spremenljivko okolijsko spremenljivko `PATH` dodajte imenik, kjer se trenutno nahajate (ne trenutni imenik v splošnem) in podimenik `va` je v vašem domačem imeniku!
9. Napišite skripto, ki preveri, ali uporabnik, ki ga podamo kot argument skripti, obstaja (na našem sistemu)? Če uporabnik obstaja, preverite, ali obstaja njegov domači imenik! Če domači imenik obstaja, izpišite, če imate pravico za dostop do posameznih datotek v tem imeniku! Če uporabnik ne poda argumenta, ga opozorimo, da mora kot prvi argument podati uporabniško ime.