Linux Ubuntu 18.10, terminal komande (4)

Komande za rad sa dozvolama i vlasništvima nad fajlovima

- **Sudo** komanda i tipovi naloga
- Komanda **id** (identitet korisnika)
- Komanda **chmod** (promena moda fajla)
- Komanda **umask** (podešavanje default dozvola)
- Komanda **chown** (promena vlasništva fajla)
- Komanda **chgrp** (promena grupe)
- Komanda **passwd** (izmena lozinke)

Objašnjenje **sudo** mehanizma

- Na Linux (kao i na Windows) OS-u, postoje dve osnovne vrste korisničkih naloga: **administratorski** i **obični** (korisnički) Obični korisnici imaju mnoge restrikcije (ograničen pristup podešavanjima, aplikacijama i objetkima fajl sistema), dok je administratoru dozvoljeno sve (npr. instalacija aplikacija, pravljenje i brisanje naloga, pristup svim datotekama fajl sistema i slično)
- Na Linuxu se podrazumevani administratorski nalog zove **root**
- Na Ubuntu distribuciji se pri instalaciji kreira **root nalog** (na Windows-u je to nalog Administrator); ovaj nalog postoji sa svim svojim pravima, ali je onemogućen (nije mu dodeljena lozinka) pa samim tim Linux ne dopušta da se pristupi sa ovim imenom

Objašnjenje **sudo** mehanizma

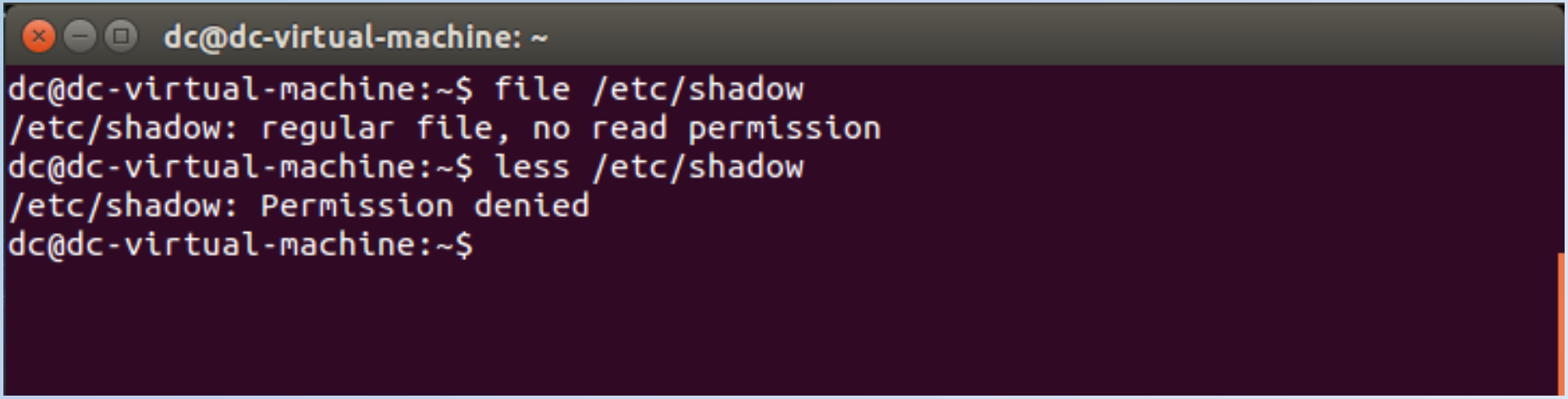
- Za administraciju sistema ne koristi se neposredno root nalog, već običan korisnik kome su dodeljena prava za administrativne akcije, pomoću komande **sudo** (Super User DO) preuzima funkciju administrator
- Običan korisnik pokreće program koji zahteva root privilegije tako što zadaje komandu **sudo** i ime traženog programa; zatim se od njega očekuje da unese lozinku, ali korisnik ne unosi root lozinku (ona i ne postoji), već svoju šifru za pristup sistemu
- Smatra se da nije mudro biti ulogovan duže vreme kao super korisnik; root korisnik ima potpunu i apsolutnu kontrolu nad celim sistemom, što sa sobom nosi i veliki rizik, a **sudo** komanda daje privremene privilegije super korisnika dok se izvršava jedna komanda, nakon čega vraća korisnika u neprivilegovan status.

Linux korisnici i nalozi

- Linux je nasledio od Unix-a karakteristiku da korisnici mogu imati vlasništvo (*owner*) nad fajlovima ili folderima
- Vlasništvo nad fajlovima znači da korisnik može kontrolisati pristup istom
- Korisnik može biti član grupe koju čini više korisnika i koji mogu određivati prava pristupa fajlovima ili folderima

Linux zaštićeni model pristupa

- U Linuxu se po default podešavanjima ne može svakom fajlu ili folderu pristupiti jer korisnik nema sva prava pristupa
- Primer: kao regularni korisnici, nemamo prava pristupa fajlu **/etc/shadow** (gde se čuvaju sve kriptografski zaštićene lozinke)



Linux korisnici i grupe

- U Linuxu datoteka **/etc/passwd** (ili *password file*) je baza podataka koja opisuje sve korisnike sistema
- Svaki korisnik Linux sistema mora pripadati najmanje jednoj grupi i GID ove grupe naveden je u datoteci **/etc/passwd**. Ova grupa se zove primarna grupa korisnika. Primarnoj grupi korisnika se dodeljuju objekti fajl-sistema koje korisnik kreira
- Osim primarnoj, korisnik može pripadati većem broju grupa i u svima je ravnopravan član
- Grupe mogu biti:

a) sistemske (nastaju prilikom instalacije Linux-a i služe za specijalne namene (npr. grupa **root**)

b) regularne (kreira ih root a koriste se u svrhe lakše administracije)

Linux korisnički nalozi

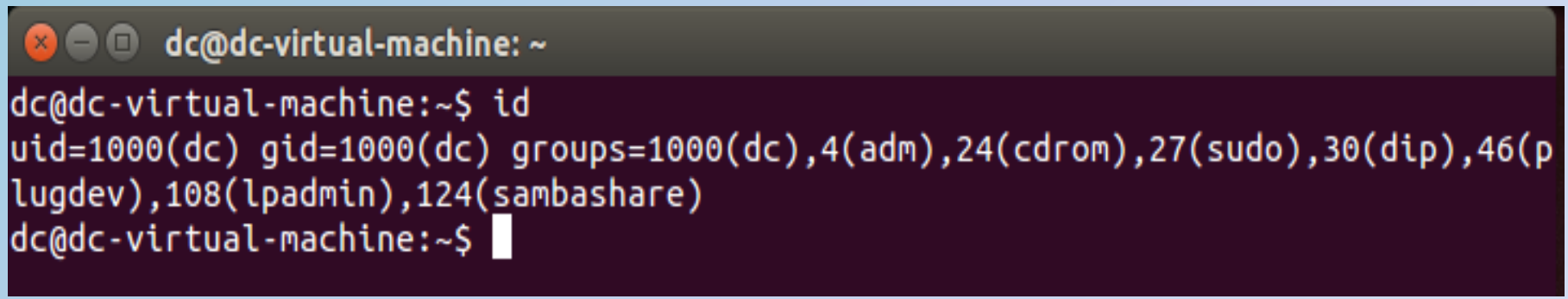
- Linux korisnički nalozi mogu biti:

a) Sistemski (nastaju prilikom instalacije OS; služe za specijalne namene, a ne za login; jedini sistemski korisnik koji se može prijaviti na sistem je **root** koji ima sve privilegije i služi za administraciju sistema; ovaj korisnik može drugima dati privilegije da obavljaju neke administrativne akcije pomoću **sudo** mehanizma (datoteka /etc/sudoers)

b) Regularni (kreira ih root ili neki drugi ovlašćeni korisnik; služe za prijavljivanje korisnika na sistem)

id komanda

- Komanda za informacije o identitetu korisnika; navodi se unosom komande **id**



- Korisnik dobija jedinstven UID (*uid*) koji se mapira u korisničko ime; korisnik se dodeljuje primarnoj grupi i dobija ID grupe (*gid*) a najčešće je i član drugih grupa koje se identifikuju svojim ID brojevima (Ubuntu ID brojeve dodeljuje od broja 1000)
- Probati i komandu **whoami**

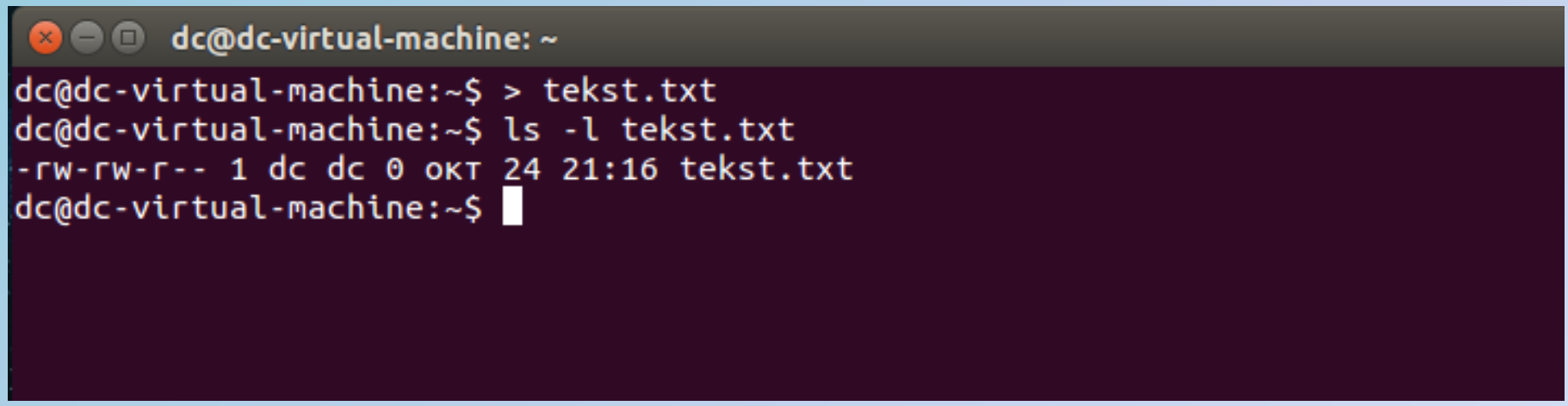
Ovlašćenja i prava pristupa

- **Vlasnik** (*owner*) je korisnik koji je kreirao objekat, odnosno korisnik kome je **root** dodelio vlasništvo
- **Grupa** (*group*) je primarna grupa korisnika koji je objekat kreirao ili grupa kojoj je root dodelio vlasništvo; primarna grupa korisnika je grupa koja mora da se navede kada se kreira korisnik
- **Svi ostali** korisnici koji nisu ni vlasnik objekta, niti pripadaju grupi kojoj objekat pripada spadaju u kategoriju ostali (*world* ili *others*)

Ovlašćenja i prava pristupa

- Ovlašćenja možemo postaviti za:
  1. **vlasnika fajla – u**
  2. **grupu – g**
  3. **za ostale – o**
- Prava pristupa koja možemo podesiti su:
  1. **pravo čitanja (*read*) - r**
  2. **pravo pisanja (*write*) - w**
  3. **pravo izvršavanja (*execute*) - x**

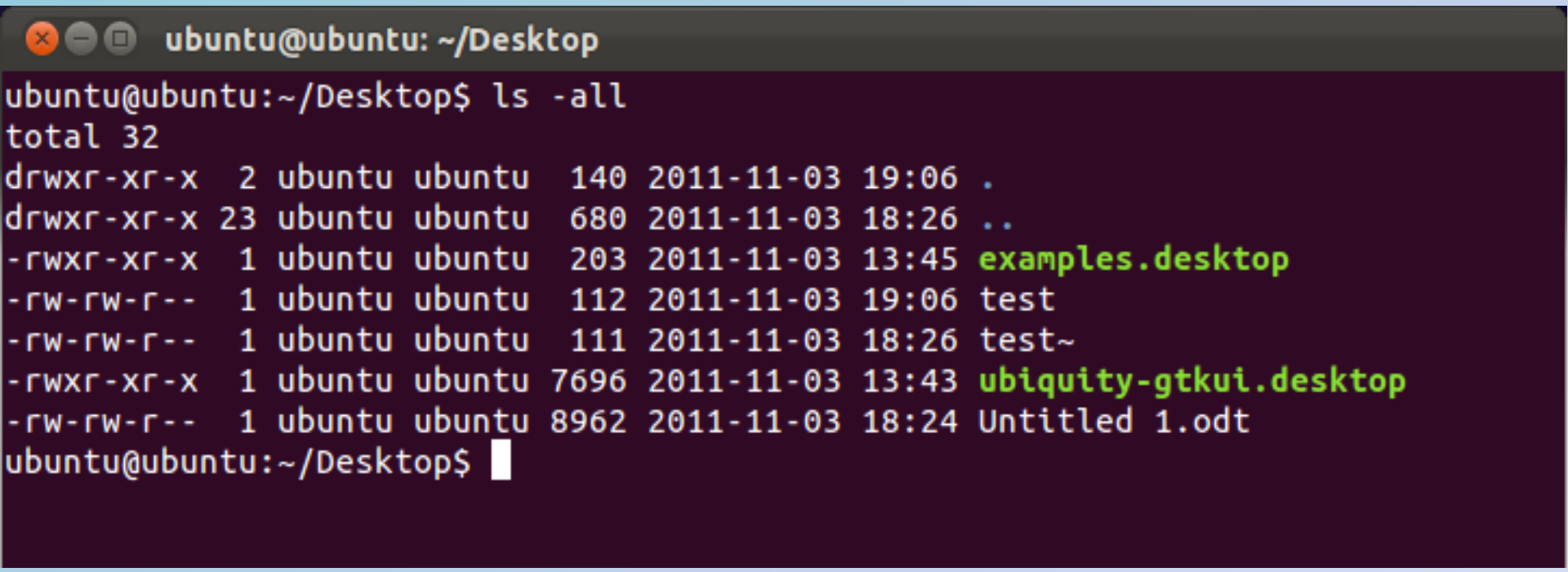
Atributi fajlova



- Prvih 10 karaktera određuju atribute fajla

-rw-rw-r--

Owner	Group	World
rwx	rwx	rwx



- Ovde je važno uočiti kolonu koja se nalazi skroz levo i koja ima 10 karaktera
-To su prava određenog fajla ili direktorijuma
-Svako slovo ima svoje tačno definisano značanje

Atributi fajlova

-rw-rw-r--

Prvi karakter određuje tip fajla koji može biti: regularni fajl (-), direktorijum (d), simbolički link (l), fajl specijalnih znakova (c) ili blok-uređaj (b)

Character	Type of the File
-	A regular file which may be empty, contain text or binary data.
d	A directory file which contains the names of other files and links to them.
l	A symbolic link is a file name that refers (points) to another file.
b	A block file is one that relates to a block hardware device where data is read in blocks of data.
c	A character file is one that relates to a character hardware device where data is read one byte at a time.
p	A pipe file works similar to the pipe symbol, allowing for the output of one process to communicate to another process through the pipe file, where the output of the one process is used as input for the other process.
s	A socket file allows two processes to communicate, where both processes are allowed to either send or receive data.

Prava pristupa fajlovima

-rw-rw-r--

Karakteri 2-10 određuju prava pristupa objektu za vlasnika, grupu i ostale

- Uvek je prva pozicija read, druga write i treća execute tj. "**rwx**".
- *Ukoliko se na nekoj poziciji umesto slova r, w ili x nalazi crtica ("-"), pravo pristupa je ukinuto*
- Da bi direktorijum na Linuxu bio upotrebljiv korisnicima treba dati prava "r" i "x"
- Pravo "w" dato nad datotekom ne znači da ona može da se obriše; da bi se datoteka obrisala, potrebno je pravo "w" nad direktorijumom u kome se ona nalazi

Prava pristupa fajlovima

- Pristupna prava fajla

r – čitanje sadržaja fajla (prikazivanje na ekranu, štampanje, kopiranje)

w – izmena sadržaja fajla (ne znači da korisnik može da obriše fajl!)

x – izvršavanje datoteke (ako je shell program ili binarna izvršna datoteka)

- Pristupna prava direktorijuma

r – čitanje sadržaja direktorijuma, tj. file-info struktura

w – izmena sadržaja direktorijuma (dodavanje novih i brisanje postojećih objekata u njemu)

x – pozicioniranje na dir. (cd), puni listing sadržaja (ls -l) i pretraživanje direktorijuma (find)

Prava pristupa fajlovima i folderima

Attribute	Files	Directories
r	Allows a file to be opened and read.	Allows a directory's contents to be listed if the execute attribute is also set.
w	Allows a file to be written to or truncated, however this attribute does not allow files to be renamed or deleted. The ability to delete or rename files is determined by directory attributes.	Allows files within a directory to be created, deleted, and renamed if the execute attribute is also set.
x	Allows a file to be treated as a program and executed. Program files written in scripting languages must also be set as readable to be executed.	Allows a directory to be entered, e.g., cd *directory*.

Prava pristupa - razumevanje

- Ako ste korisnik koji je kreirao fajl tj. vlasnik ste fajla, onda se prava pristupa primenjuju samo na vlasnika tj na prva 3 znaka **rwx**
- Ako niste vlasnik fajla ali pripadate grupi koja je njegov vlasnik, dozvole se primenjuju na druga 3 znaka **rwx**
- Ako niste ni vlasnik fajla ni član grupe koja je vlasnik istog, onda se dozvole primenjuju na poslednja tri znaka **rwx** (other)

Prava pristupa fajlovima-primeri

-rwx------ Regularni fajl koji vlasnik može čitati, menjati ili izvršavati; niko drugi nema pravo pristupa fajlu

-rw-r--r-- Regularni fajl koji vlasnik može čitati ili menjati; korisnici koji pripadaju grupi ili ostalima mogu samo čitati fajl

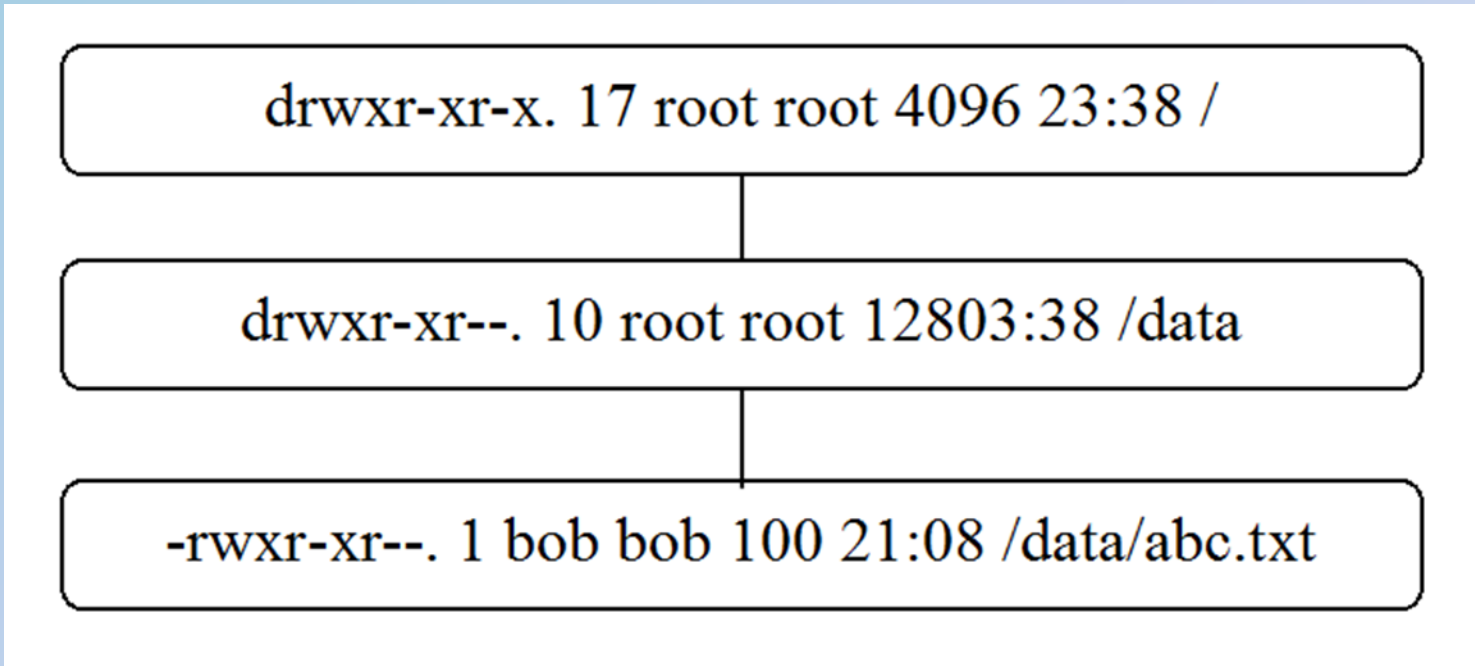
-rwxr-xr-x Regularni fajl koji vlasnik može čitati, menjati ili izvršavati; korisnici grupe i svi ostali mogu čitati ili izvršavati fajl

-rw-rw---- Regularni fajl koji samo vlasnik ili grupa mogu čitati i menjati

drwxr-x--- Označava direktorijum; vlasnik može ući u direktorijum i kreirati, preimenovati ili brisati fajlove u njemu; članovi grupe mogu ući u direktorijum ali ne mogu kreirati, brisati ili menjati imena fajlovima

Prava pristupa - razumevanje

Pitanje : Koji nivo pristupa ima korisnik bob za fajl /data/abc.txt



Nema prava pristupa jer bez dozvole execute za /data nema načina da bob pristupi fajlu `/data/abc.txt`

chmod komanda

- Sintaksa **chmod** (*change mode*) komande je:

chmod [opcije] mod file

- Pre nego što pristupimo promeni mod-a (dozvola, ovlašćenje, *permission*), najpre pogledamo kako izgleda neki fajl sa odredjenim dozvolama. Za to se najčešće koristi komanda **ls –all**
- Samo **vlasnik fajla** ili **root** može da promeni njegov mod

chmod komanda

- Komanda **chmod** podržava: oktalni (numerički) i simbolički način izmene ovlašćenja; sva prava koje može da sadrži jedan fajl/direktorijum možemo prikazati i u dekadnom i u binarnom obliku:

0	**---**	**000**
1	**--x**	**001**
2	**-w-**	**010**
3	**-wx**	**011**
4	**r--**	**100**
5	**r-x**	**101**
6	**rw-**	**110**
7	**rwx**	**111**

Numerički zapis ovlašćenja

Primeri:

7 (r = 4 w = 2 x=1)

4 (r = 4 w = 0 x=0)

onda je:

7 = 4 + 2 + 1 = **rwx**

6 = 4 + 2 + 0 = **rw-**

5 = 4 + 0 + 1 = **r-x**

4 = 4 + 0 + 0 = **r--**

3 = 0 + 2 + 1 = **-wx**

2 = 0 + 2 + 0 = **-w-**

1 = 0 + 0 + 1 = **--x**

0 = 0 + 0 + 0 = **---**

Primeri:

- 777 = rwxrwxrwx
- 775 = rwxrwxr-x
- 755 = rwxr-xr-x
- 700 = rwx------
- 664 = rw-rw-r--
- 640 = rw-r-----

Numerički zapis ovlašćenja

- `chmod 755 abc.sh` ili rwxr-xr-x
- `chmod 660 abc.txt` ili rw-rw----
- `chmod 771 somedir` ili rwxrwx--x
- `chmod 400 my.txt` ili r--------
- `chmod 700 userdir` ili rwx------

Numerički zapis ovlašćenja

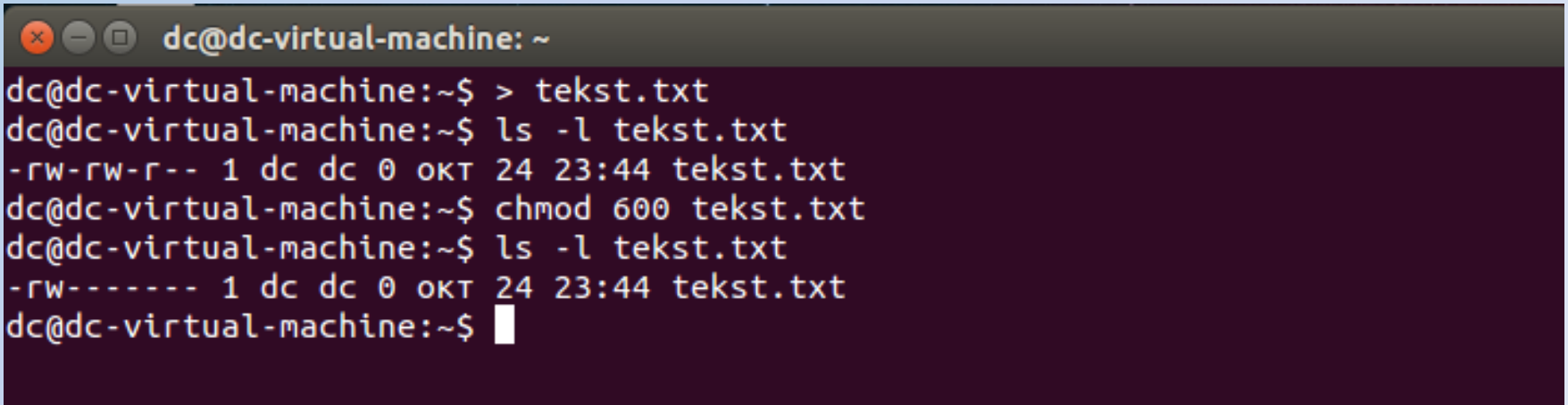
Primeri komandi:

chmod 644 (**rw-r--r--** vlasnik ima pravo upisa i čitanja, grupa i ostali pravo čitanja)

chmod 750 (**rwxr-x---** vlasnik ima sva prava, grupa prava čitanja i izvršavanja, ostali ništa)

chmod 754 (**rwxr-xr--** vlasnik ima sva prava, grupa prava čitanja i izvršavanja, ostali imaju samo pravo čitanja fajla)

chmod 600 (**rw-------** vlasnik ima pravo čitanja i upisa, ostali nemaju prava pristupa)



Simbolički zapis ovlašćenja

- Simboličko označavanje ovlašćenja mogu se postaviti za: vlasnika fajla/direktorijuma (**u**), vlasnika grupe (**g**), za ostale (**o**) i za sve (kombinacija **u+g+o**)
- Za dodeljivanje i oduzimanje ovlašćenja koriste se reč **chmod** uz operatore **+, -, =**
- Koristi se kombinovanje oznaka za vlasništvo, operatora i **rwx** oznaka za prava pristupa
- Ako želimo da omogućimo vlasniku pravo izvršavanja fajla **test** komanda bi bila

chmod u+x test

Simbolički zapis ovlašćenja

***who*: specificira čije dozvole smeju da se menjaju:**

u korisnik

g grupa

o ostali

a svi

operator: specificira da li dodati, izbrisati ili dodeliti dozvole:

+ dodavanje

- uklanjanje

= dodeljivanje

what: specificira tip dozvole na fajlu ili folderu: :

r čitanje

w upis

x izvršavanje

- ukinuto

Simbolički zapis ovlašćenja

u+x Omogućiti vlasniku pravo izvršavanja fajla

u-x Onemogućiti vlasniku pravo izvršavanja fajla

+x Omogućiti vlasniku, grupi i ostalima pravo izvršavanja fajla; ekvivalentno je sa **a+x**

o-rw Onemogućiti pravo čitanja i upisa za vlasnika i grupu

go=rw Omogućiti pravo čitanja i upisa za grupu i ostale; ako su prethodno imali dozvolu izvršavanja ona se uklanja

u+x, go=rx Omogućiti vlasniku pravo izvršavanja i omogućiti dozvolu čitanja i izvršavanja za grupu i ostale

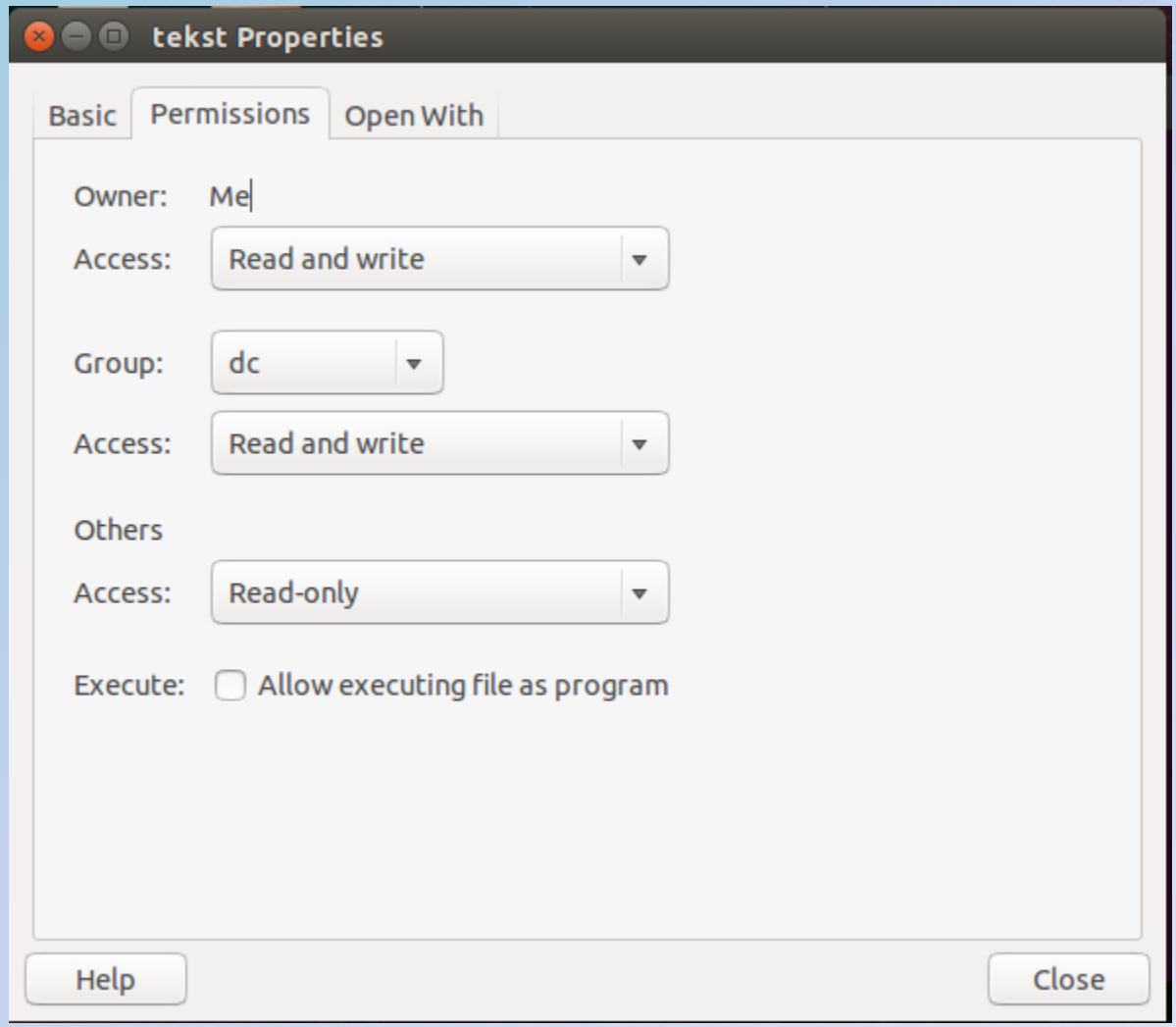
Simbolički zapis ovlašćenja

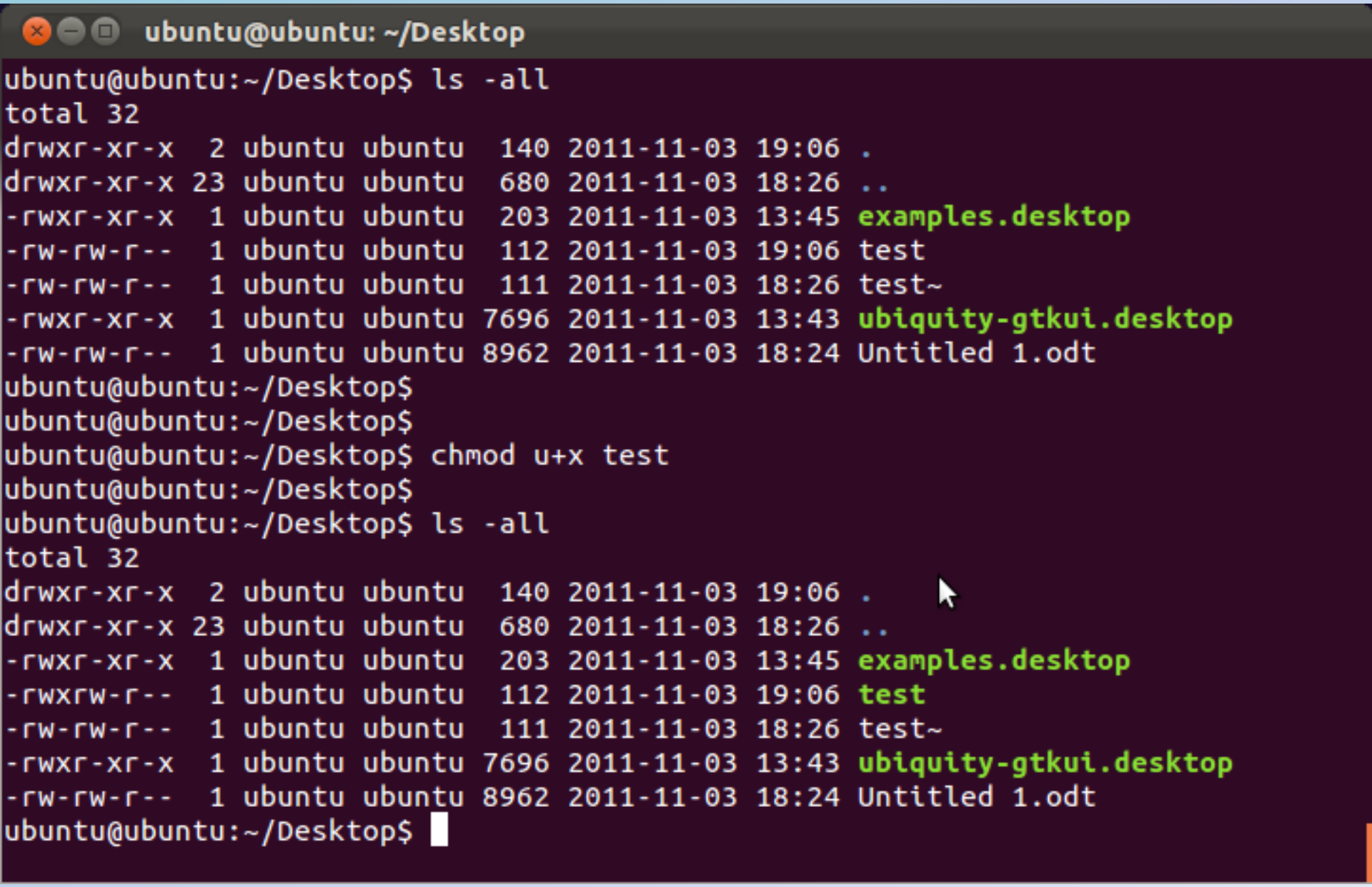
`chmod u+x abc.txt` označava da se vlasniku fajla **abc.txt** daje dozvola izvršavanja

`chmod go-rx abc.txt` će izbrisati read i execute za grupu i ostale

`chmod u+wx,g=rx,o-r abc.txt` će promeniti write i execute dozvole za vlasnika (read se ne menja), postaviće r-x za grupu i promeniti/izbrisati read dozvolu za ostale

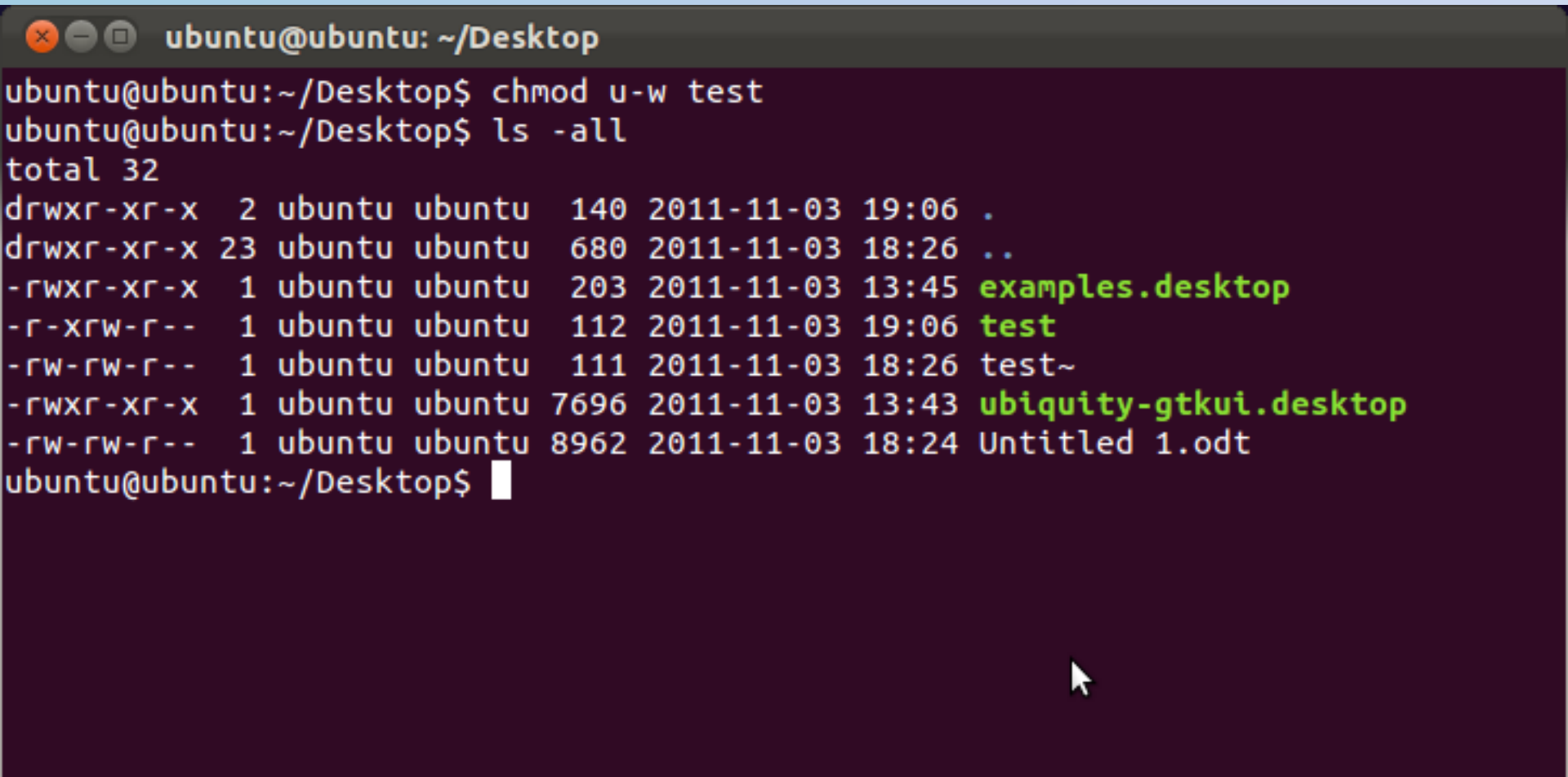
GUI podešavanje ovlašćenja




ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ ls -all
total 32
drwxr-xr-x 2 ubuntu ubuntu 140 2011-11-03 19:06 .
drwxr-xr-x 23 ubuntu ubuntu 680 2011-11-03 18:26 ..
-rwxr-xr-x 1 ubuntu ubuntu 203 2011-11-03 13:45 examples.desktop
-rw-rw-r-- 1 ubuntu ubuntu 112 2011-11-03 19:06 test
-rw-rw-r-- 1 ubuntu ubuntu 111 2011-11-03 18:26 test~
-rwxr-xr-x 1 ubuntu ubuntu 7696 2011-11-03 13:43 ubiquity-gtkui.desktop
-rw-rw-r-- 1 ubuntu ubuntu 8962 2011-11-03 18:24 Untitled 1.odt
ubuntu@ubuntu:~/Desktop$
ubuntu@ubuntu:~/Desktop$
ubuntu@ubuntu:~/Desktop$ chmod u+x test
ubuntu@ubuntu:~/Desktop$
ubuntu@ubuntu:~/Desktop$ ls -all
total 32
drwxr-xr-x 2 ubuntu ubuntu 140 2011-11-03 19:06 .
drwxr-xr-x 23 ubuntu ubuntu 680 2011-11-03 18:26 ..
-rwxr-xr-x 1 ubuntu ubuntu 203 2011-11-03 13:45 examples.desktop
-rwxrw-r-- 1 ubuntu ubuntu 112 2011-11-03 19:06 test
-rw-rw-r-- 1 ubuntu ubuntu 111 2011-11-03 18:26 test~
-rwxr-xr-x 1 ubuntu ubuntu 7696 2011-11-03 13:43 ubiquity-gtkui.desktop
-rw-rw-r-- 1 ubuntu ubuntu 8962 2011-11-03 18:24 Untitled 1.odt
ubuntu@ubuntu:~/Desktop$

- I suprotno: ako želimo da vlasniku oduzmemo pravo pisanja komanda bi glasila

chmod u-w test



umask komanda

- Linux pri kreiranju dodeljuje objektu vlasnika i grupu i postavlja podrazumevana prava pristupa:
  - Korisnik koji kreira objekat postaje njegov vlasnik
  - Objekat se dodeljuje primarnoj grupi tog korisnika
  - Podrazumevana prava pristupa zavise od promenljive **umask**
- Default prava pristupa dodeljuju se na osnovu vrednosti promenljive **umask**
- Promenljiva **umask** je specifična za svakog korisnika i postavlja se prilikom prijavljivanja na sistem (uobičajena vrednost je **022**)

umask komanda

- Namena umask promenljive je da **ukida prava pristupa** novokreiranim direktorijumima i datotekama

Primer:

umask 022 (ukida se pravo upisa za grupu i ostatak sveta)

umask 027 (ukida se pravo upisa za grupu i sva prava za ostatak sveta)

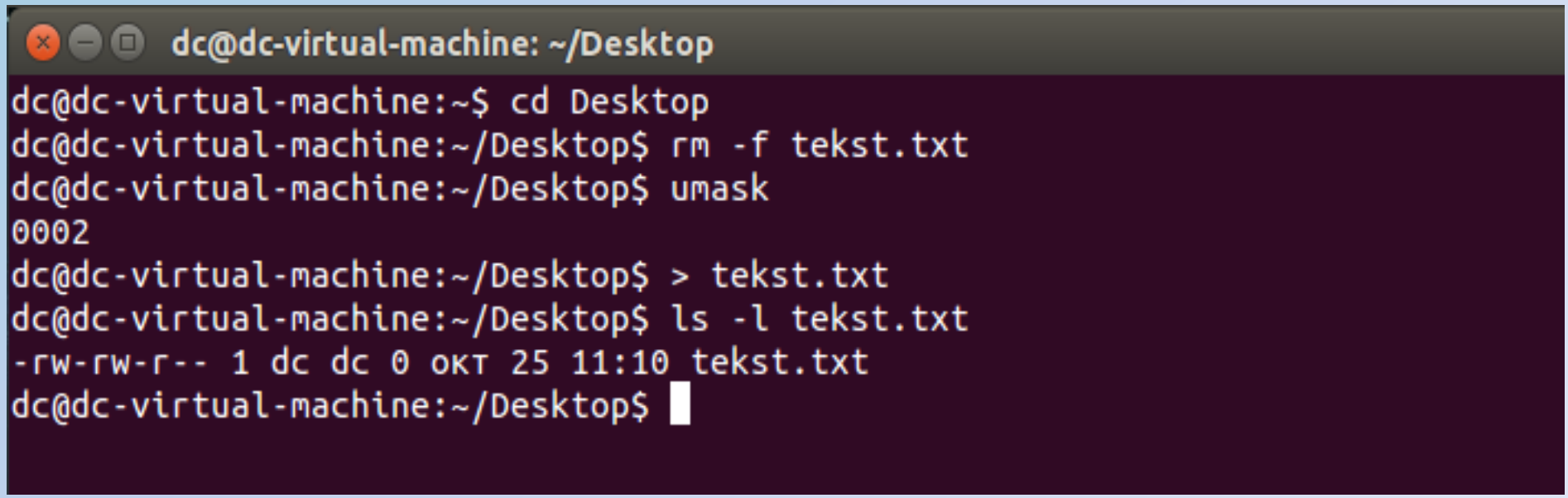
- Default prava pristupa pri kreiranju se razlikuju za direktorijume i datoteke:

a) za direktorijum se vrednost **umask** oduzima od 777 (**rwxrwxrwx**) (**777** je maksimalno dozvoljeno ovlašćenje)

b) za datoteku se vrednost umask oduzima od 666 (**rw-rw-rw-**), zato što se podrazumevano ne dodeljuje pravo izvršavanja, (**666** je maksimalno dozvoljeno ovlašćenje)

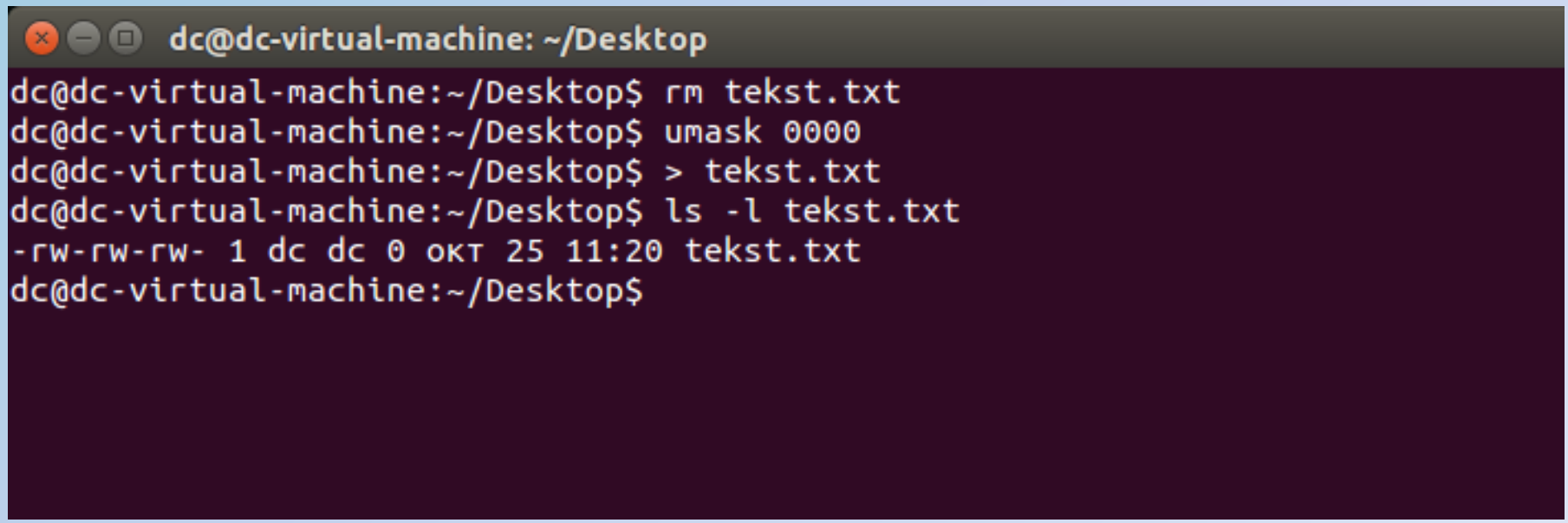
umask komanda

- Primer: obrisati sva ovlašćenja dokumenta tekst.txt koji je na desktopu (**rm –f tekst.txt**), komandom **umask** prikazati trenutni status (0002 je oktalni prikaz) i komandom **ls –l tekst.txt** prikazati ovlašćenja (**rw-rw-r--**)



umask komanda

- Primer: sada je potrebno postaviti novo ovlašćenje tako da svi korisnici imaju pravo upisa i čitanja tj. da bude (**rw-rw-rw-**); ovlašćenje ćemo izmeniti unosom vrednosti **0000**



umask komanda

- Primeri umask kmande za fajlove i direktorijume

Typical user umask				
	Directory		File	
Maxium Allowable Permission	rwxrwxrwx	777	rw-rw-rw-	666
umask value	-------w-	002	-------w-	002
Default permission	rwxrwxr-x	775	rw-rw-r--	664

umask komanda

- Za prikaz umask vrednosti potrebno je samo uneti reč **umask** bez argumenata ili **$ umask**
- Postavljanje umask vrednosti na 027 vrši se komandom $ **`umask 027`**
- Nova vrednost za **umask** će se primeniti samo tokom procesa logovanja
- Za postavljanje nove default vrednosti za **umask** treba izmeniti **`~/.bashrc`**
- Umask vrednost nema efekat na postojeće fajlove ili direktorijume već samo na novoformirane

umask komanda

Kako se dobija rezultujuća vrednost za umask?

umask 022

	rw-rw-rw-	666 (podrazumevano pravo)
oduzmi:	----w--w-	022 (umask)
dobija se:	rw-r-----	644 (rezultat)

umask 027

	rw-rw-rw-	666 (podrazumevano pravo)
oduzmi:	----w--w-	027 (umask)
dobija se:	rw-r-----	**639** (rezultat) !!!!
		(broj 9 ne postoji za dodelu prava)

Vrednost će biti rw-r----- 640

chown komanda

- Pri formiranju nove datoteke ili direktorijuma je podrazumevano da onaj koji kreira samu datoteku ili direktorijum postane i njen vlasnik. Sa **chown** komandom je omogućeno sistem administratoru da menja vlasništvo i grupnu pripadnost svake datoteke ili dir. na sistemu
- Sintaksa ove komande je:

chown [-R] user filename (promena vlasnika)

chgrp [-R] group filename (promena grupe)

user, group – novi vlasnik i nova grupa

filename – ime objekta kojem se menja vlasništvo

[-R] - inicira promenu vlasništva direktorijuma i svih njegovih objekata

chown komanda

- Pri formiranju nove datoteke ili direktorijuma je podrazumevano da onaj koji kreira samu datoteku ili

```
chown [-c|--changes] [-v|--verbose] [-f|--silent|--quiet] [--dereference]
      [-h|--no-dereference] [--preserve-root]
      [--from=currentowner:currentgroup] [--no-preserve-root]
      [-R|--recursive] [--preserve-root] [-H] [-L] [-P]
      {new-owner|--reference=ref-file} file ...
```

chown komanda

- U Ubuntu Linuxu, **chown** i **chgrp** mogu da koriste samo sudo i root korisnci (ukoliko im je dodeljena lozinka); unosi se prvo sudo pa ime komande.

sudo chown marko my_file

(dodeljuje se datoteka my_file korsniku marko)

sudo chgrp grupa1 my_file

(dodeljuje se datoteka my_file grupi korisnika pod imenom „grupa1“)

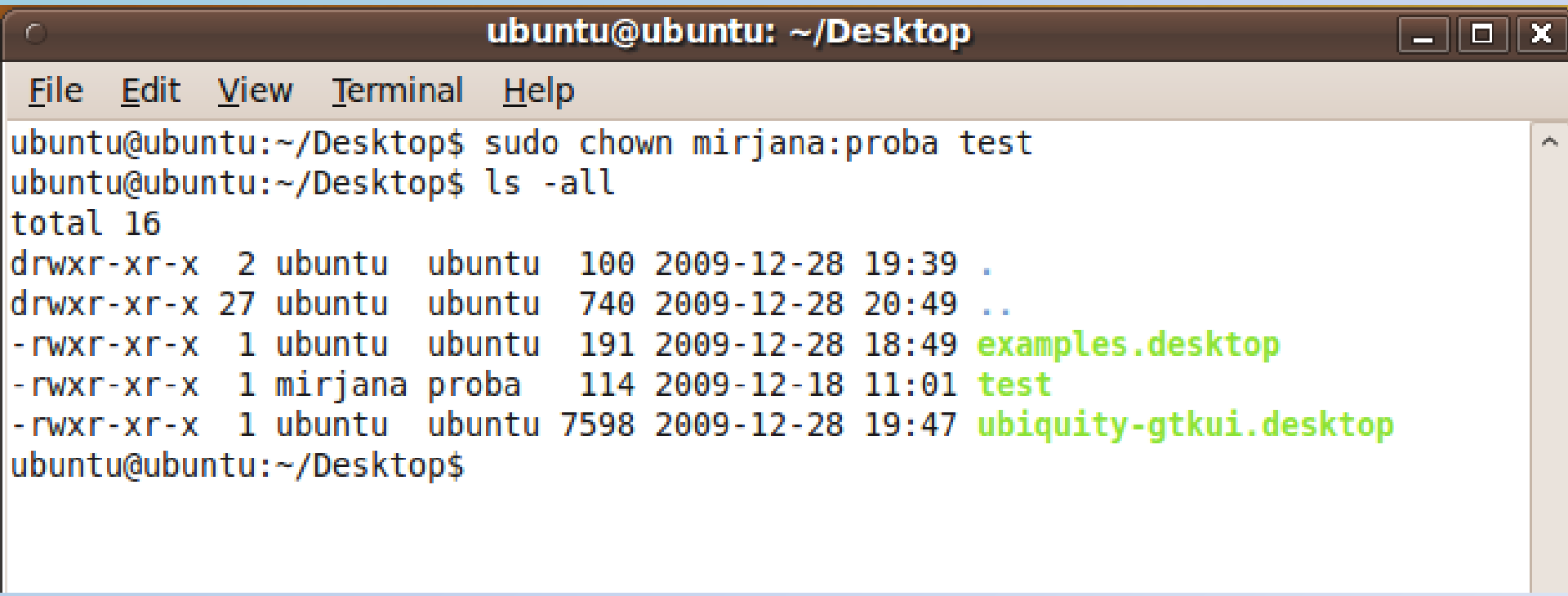
Parametre vlasništva proveriti komandom **ls –l my_file**

chown komanda opcije

chown [owner][:[group]] file...

Argument	Results
`bob`	Changes the ownership of the file from its current owner to user `bob`.
`bob:users`	Changes the ownership of the file from its current owner to user `bob` and changes the file group owner to group users.
`:admins`	Changes the group owner to the group `admins`. The file owner is unchanged.
`bob:`	Change the file owner from the current owner to user `bob` and changes the group owner to the login group of user `bob`.

Promena vlasnika fajla



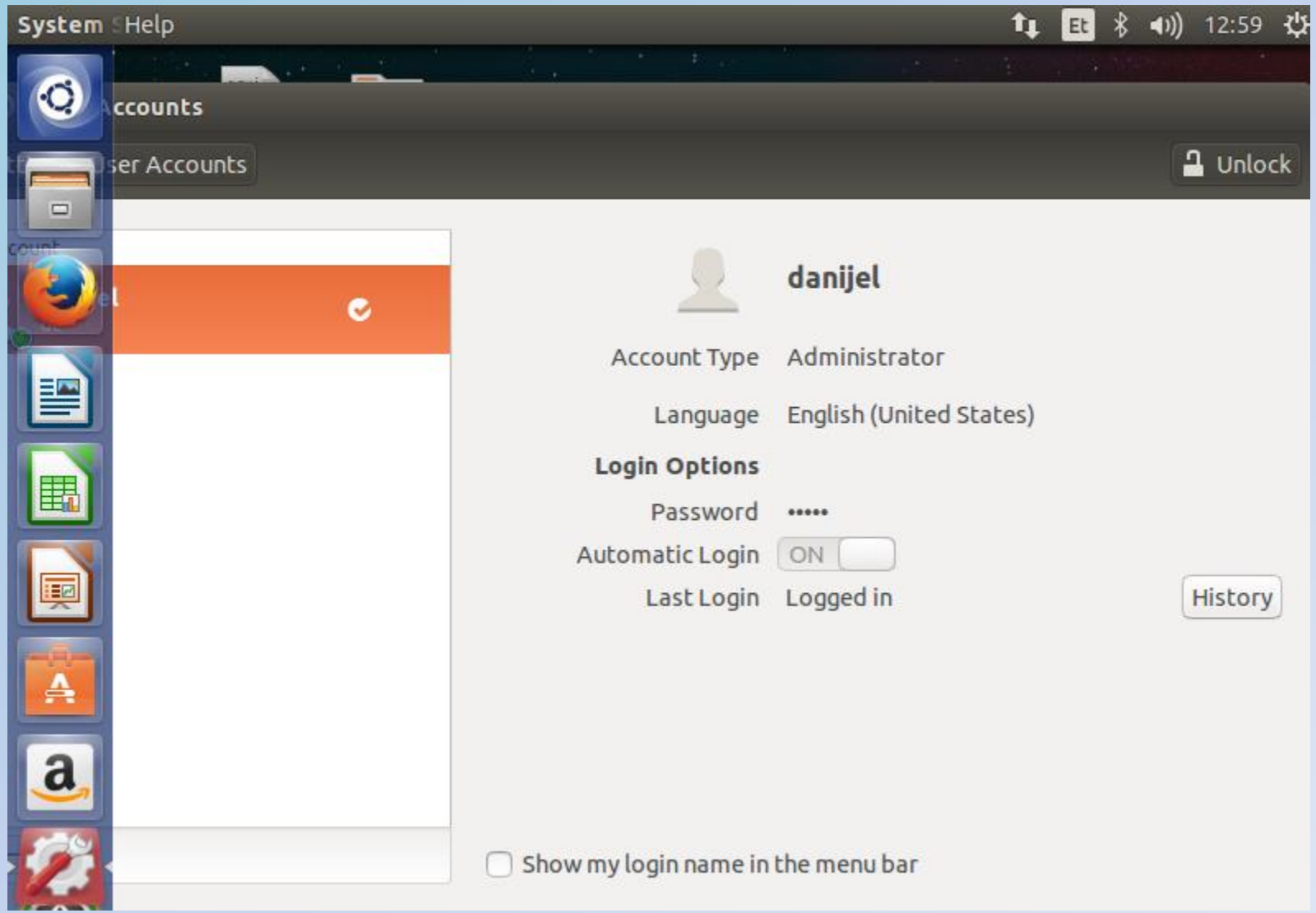
Kreiranje i brisanje naloga

- Kreiranje regularnih korisničkih naloga sadrži četiri osnovna koraka:

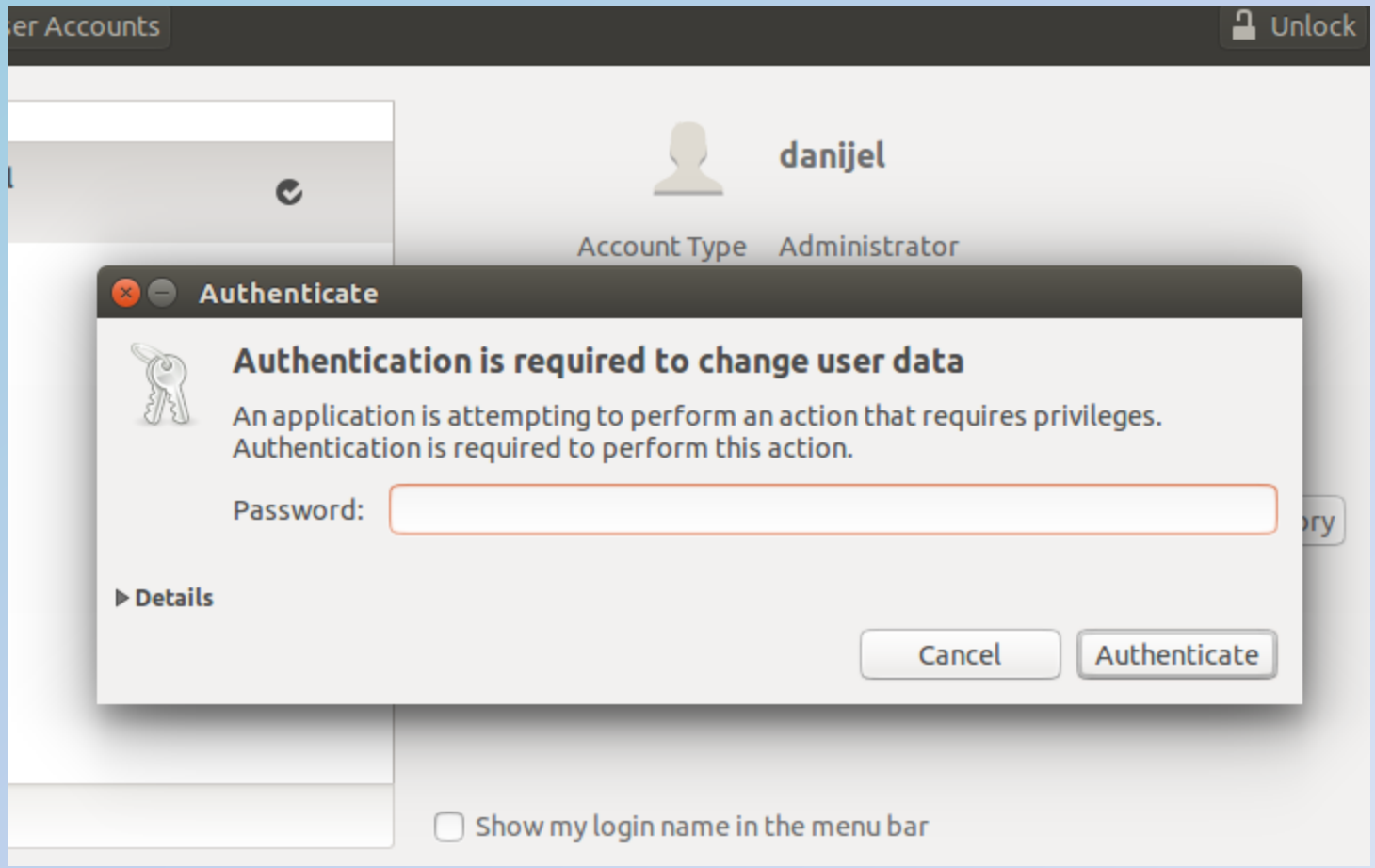
1) dodavanje informacija o korisniku u **/etc/passwd**
2) kreiranje ličnog direktorijuma i kopiranje inicijalnog profila
3) postavljanje ovlašćenja
4) dodela inicijalne lozinke, koju kasnije korisnik menja

- Samo administrator sistema može da kreira druge korisničke naloge (root ili sudoer)!

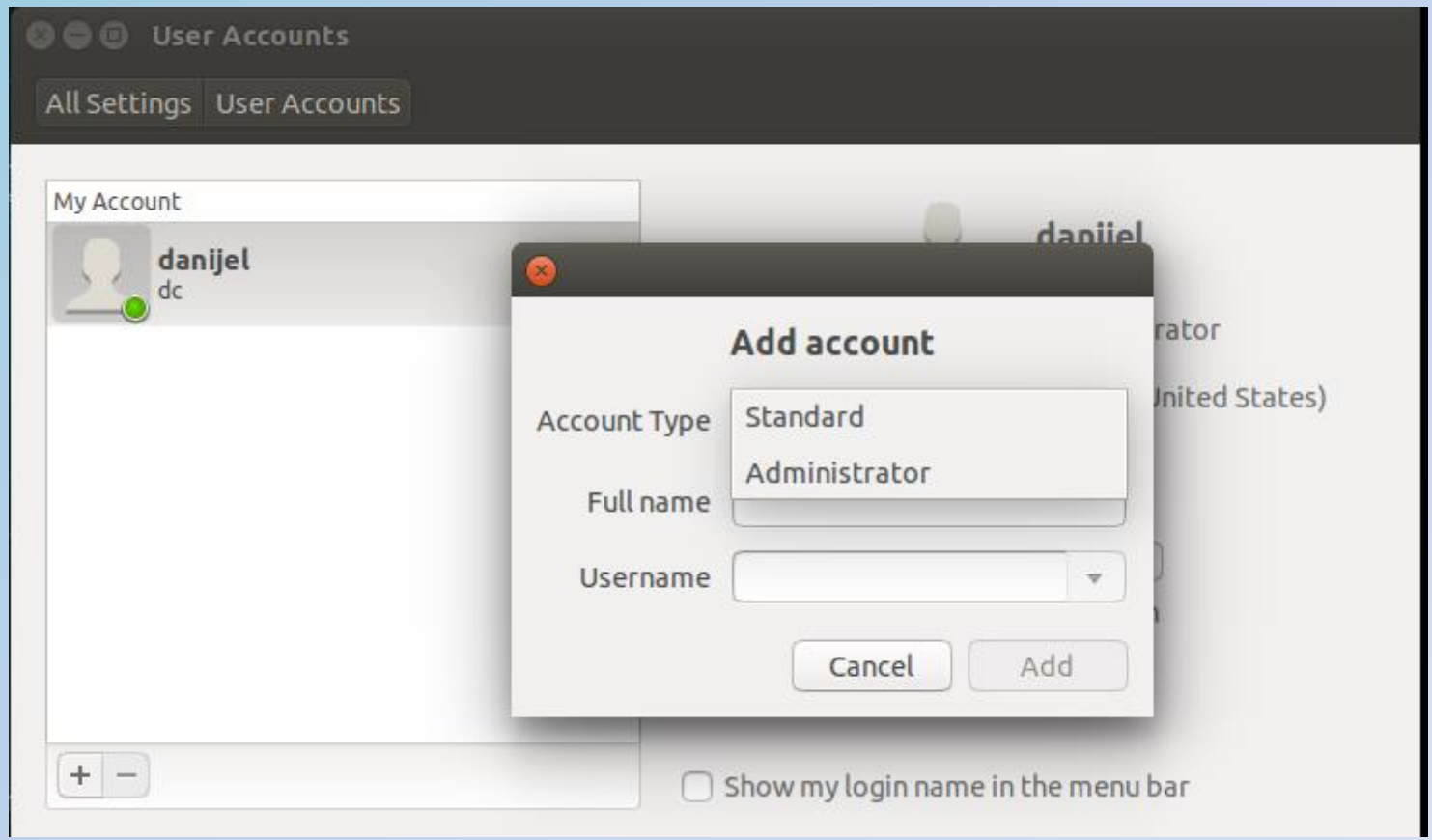
GUI dodavanje i brisanje naloga



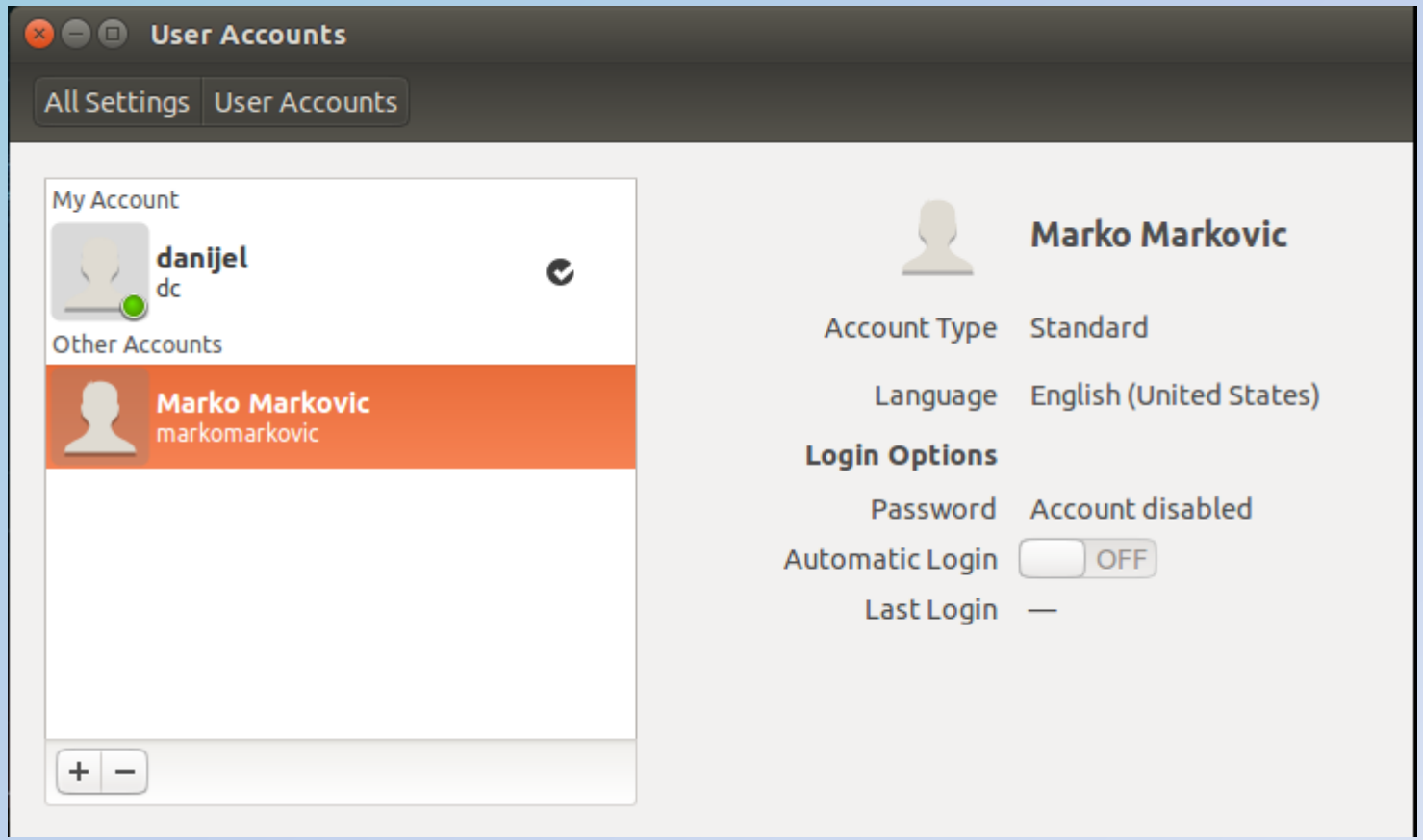
GUI dodavanje i brisanje naloga



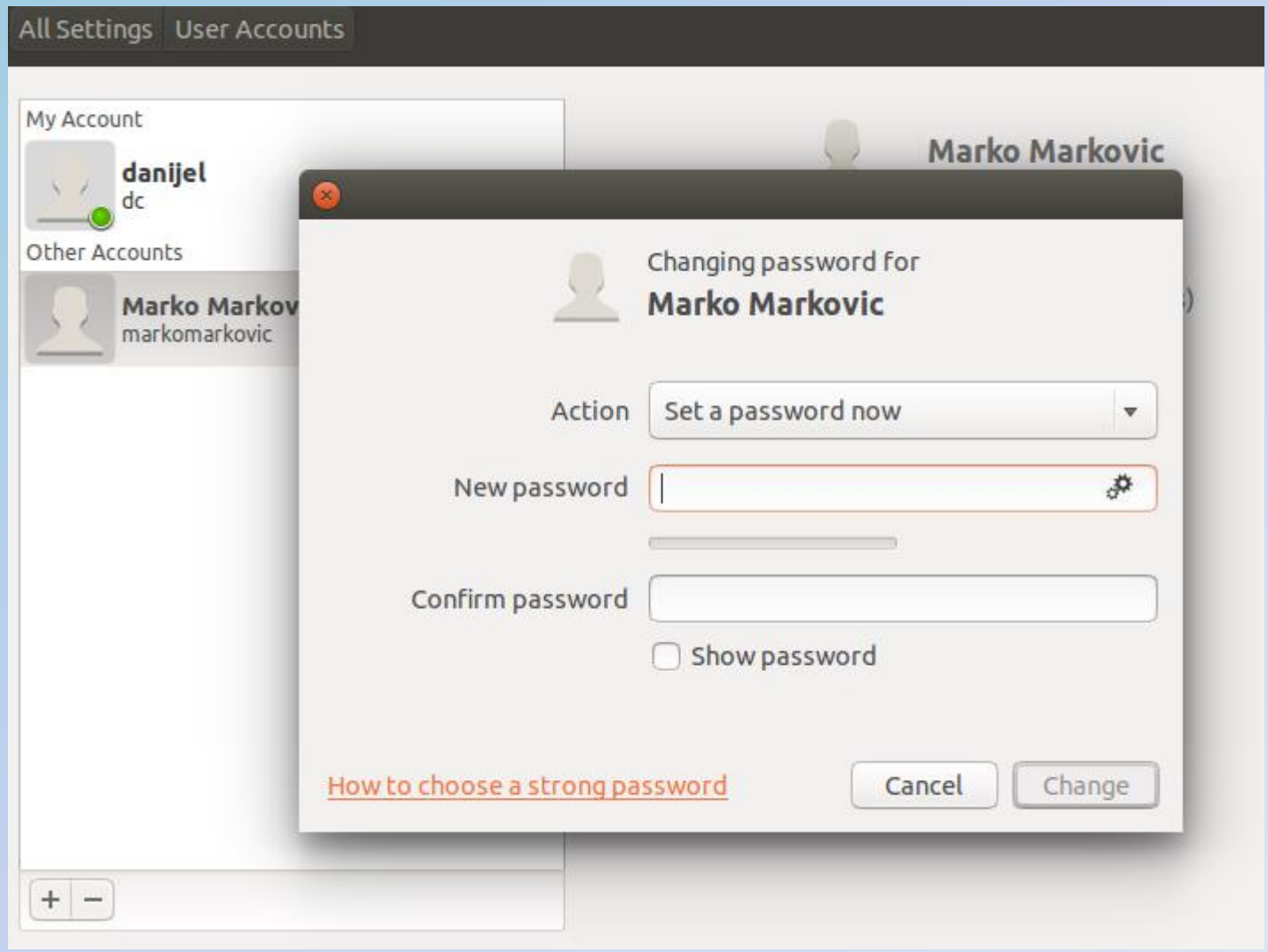
GUI dodavanje i brisanje naloga



GUI dodavanje i brisanje naloga



GUI dodavanje i brisanje naloga



Dodavanje naloga i grupe iz terminala

- Da bi se dodao novi nalog preko komandne linije, koristite komandu **adduser**
- Adduser je interaktivan program koji traži da sudo ili root korisnik navede korisničko ime, lozinku i opisne podatke
- Program adduser:

- upisuje jednu liniju u **/etc/passwd**
- opciono upisuje jednu liniju u **/etc/shadow**
- kreira lični direktorijum i podešava prava pristupa
- kopira inicijalni profil iz **/etc/skel**

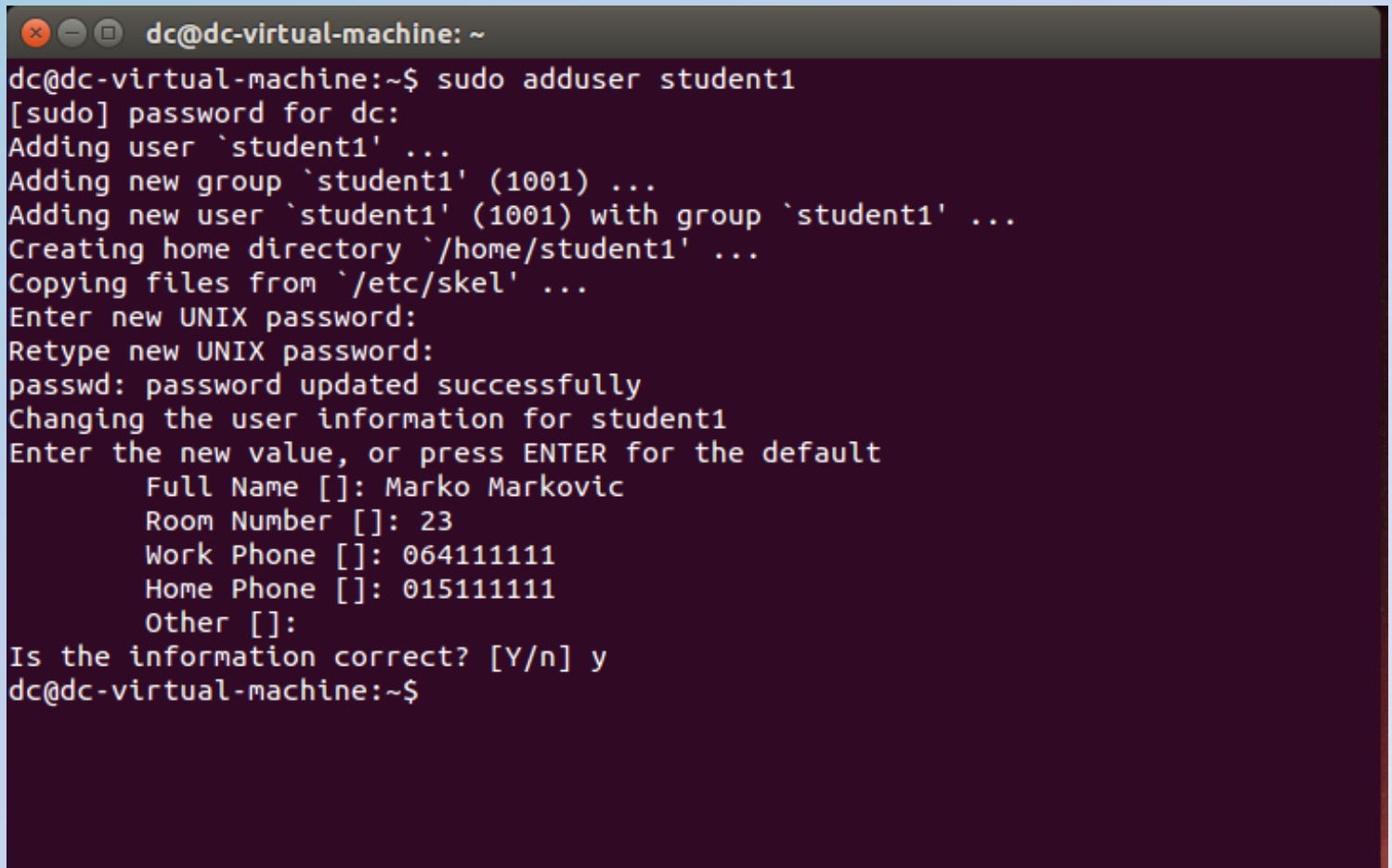
- Opšti oblik komande je:

sudo adduser korisnicko_ime

Dodavanje naloga i grupe iz terminala

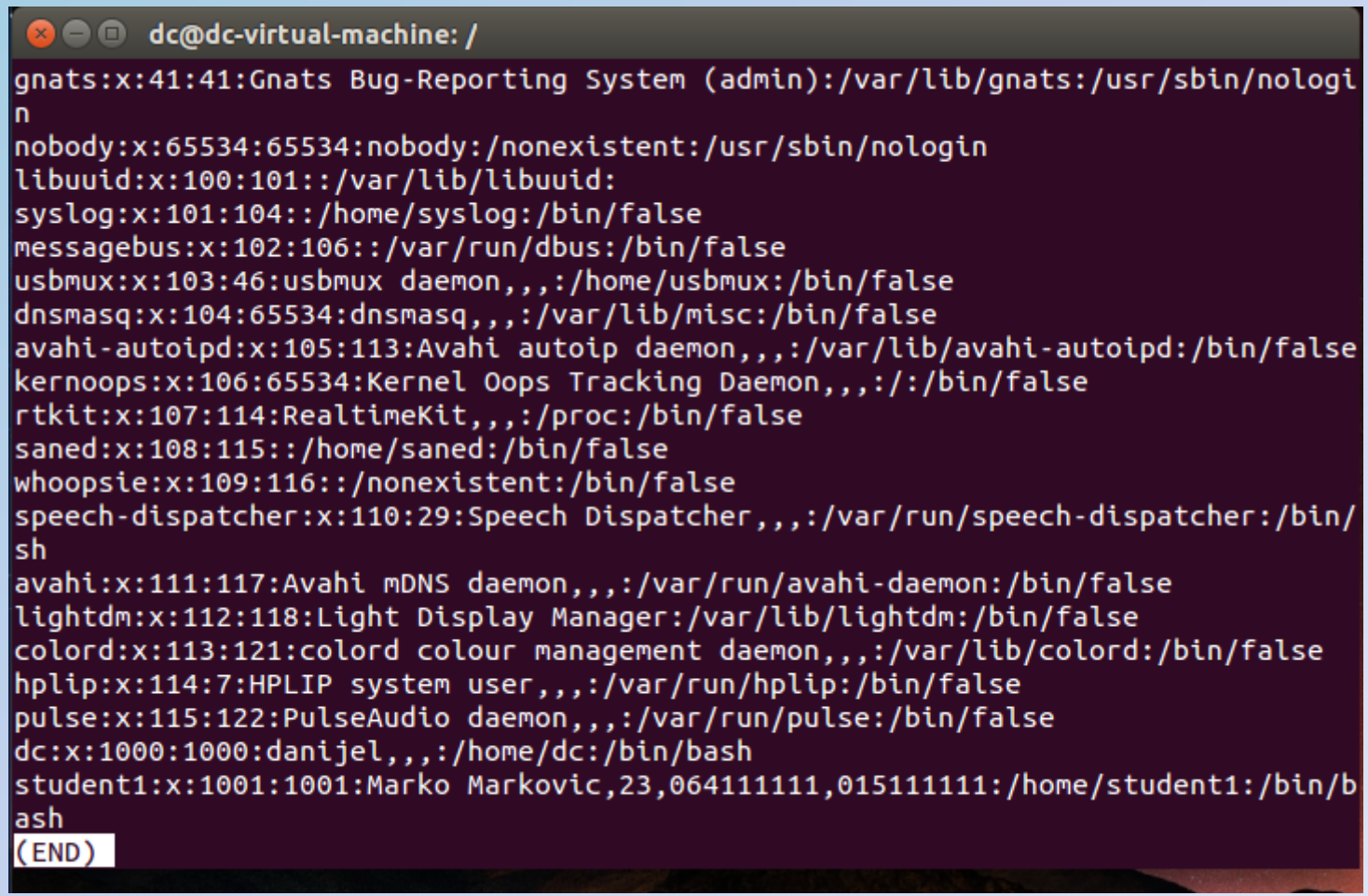
sudo adduser student1

(dodaje novog korisnika po imenu student1)

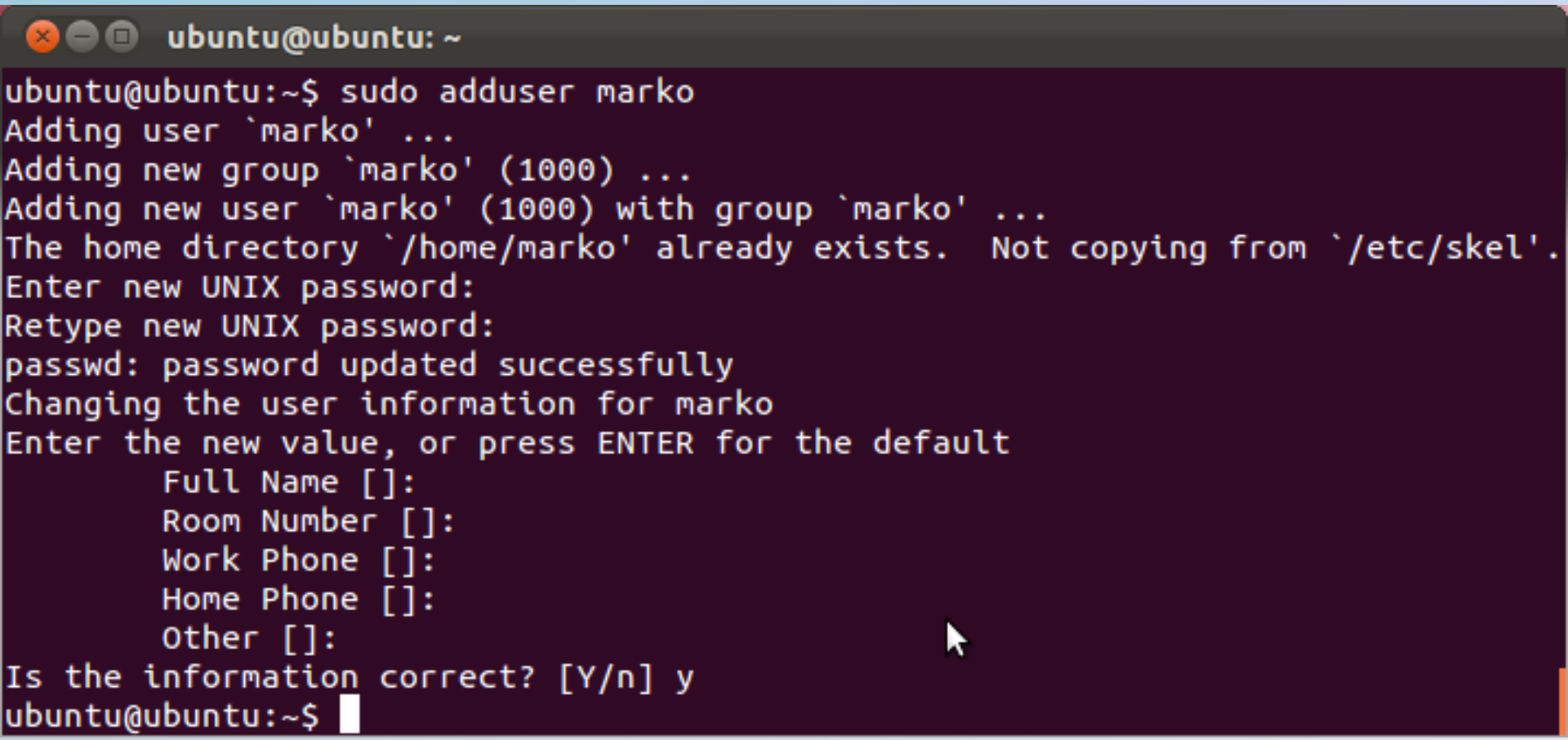


Dodavanje naloga i grupe iz terminala

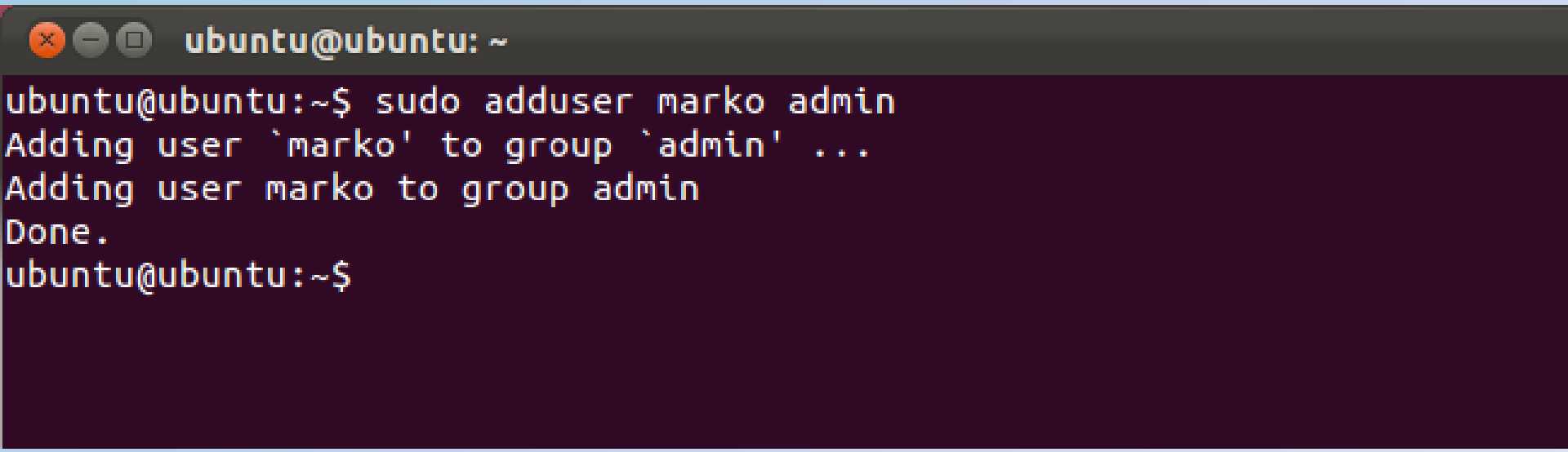
(preći u root i uneti komandu **less /etc/passwd**)



Dodavanje novih korisnika

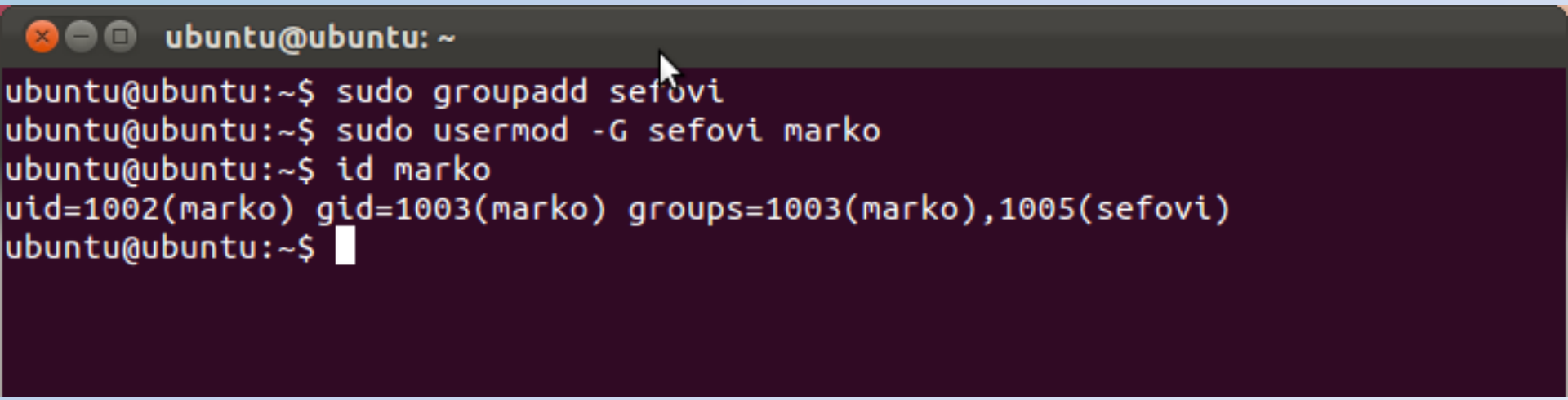


- U ovom primeru korisnika po imenu Marko prebacili smo u grupu administratora.



Dodavanje nove grupe i dodavanje korisnika u grupu

groupadd – kreiranje nove grupe korisnika
usermod – dodavanje korisnika u grupu



Promena parametara naloga

- Promena parametara korisničkih naloga svodi se na promenu polja u datotekama /etc/passwd i /etc/shadow
- Za promenu parametara koriste se sledeće komande:

chfn – promena punog imena korisnika (change full name)

passwd – promena lozinke korisnika (*password*)

chage – promena parametara lozinke (*change age*)

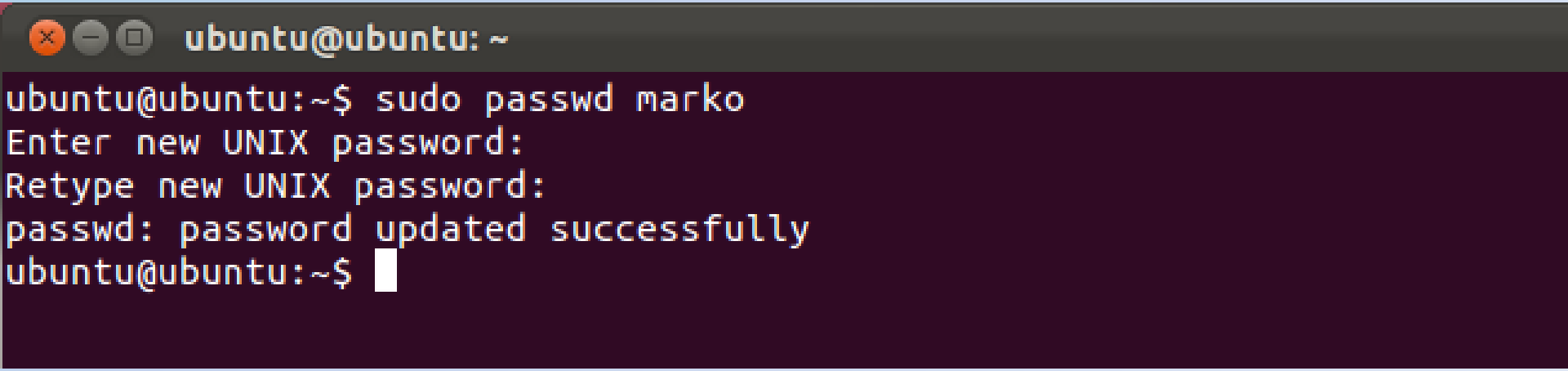
- regularni korisnici mogu promeniti svoju lozinku i svoje parametre, dok samo root ili sudo može da menja tuđe parametre i lozinke

Promena lozinke

- Sintaksa komande je:

  sudo passwd imekorisnika

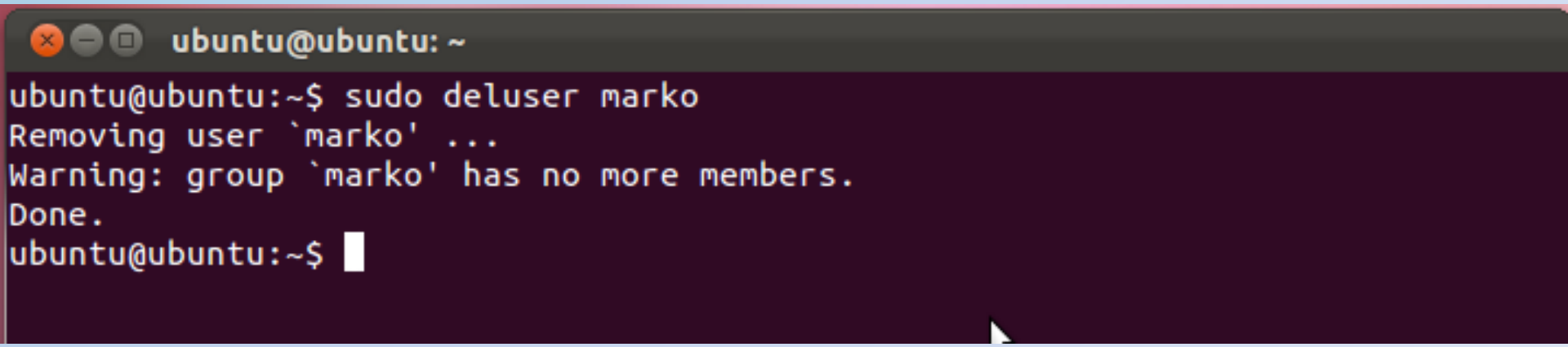
- Ovako napisana komanda omogućava promenu lozinke za navedenog korisnika, a potrebno je i restartovati računar



Brisanje korisnika

- Korisnike brišemo pomoću **deluser** komande
- Sintaksa komande je sledeća:

sudo deluser ime_korisnika



Brisanje grupe

- Grupe brišemo pomoću **delgroup** komande
- Sintaksa komande je sledeća:

sudo delgroup ime_grupe