**Linux**

გამოსადეგი შემოკლებები (გამოიყენება ბრძანებების არგუმენტებში):

**.**  მიმდინარე დირექტორია

**..** მიმდინარე დირექტორიის დედა-დირექტორია

~ მომხმარებლის საწყისი დირექტორია (როგორც წესი, მისი აბსოლუტური მისამართია /home/*username* (*username*-ს მაგივრად იწერება მომხმარებლის სახელი))

**/** "საძირკველი" დირექტორია (root directory)

**\*** ნებისმიერი რაოდენობის ასო-ციფრი-ნიშანი (გარდა ფრჩხილების, ბრჭყალების და ინტერვალისა),

**?** ერთი ასო-ციფრი-ნიშანი (გარდა ფრჩხილების, ბრჭყალების და ინტერვალისა)

ბრძანებები:

(***დახრილად***  დაწერილი ტექსტი ჩაანაცვლეთ კონკრეტული დასახელებით ან სრული მისამართით)

**cd** *dir* გადადი *dir* დირექტორიაში (თუ *dir* მითითებული არ არის, გადავა მომხმარებლის საწყის (~) დირექტორიაში);

**chmod a+x** *file* მიეცი ყველა მომხმარებელს *file* ფაილის (ან დირექტორიის) გაშვების უფლება;

**chmod a+w** *file* მიეცი ყველა მომხმარებელს *file* ფაილის (ან დირექტორიის) შეცვლის უფლება;

**chmod a+r** *file*  მიეცი ყველა მომხმარებელს *file* ფაილის (ან დირექტორიის) წაკითხვის უფლება;

თუ ფაილის (დირექტორიის) შეცვლის უფლება თქვენ მანამდე არ გქონდათ, chmod ბრძანება საჭიროებს ადმინისტრატორის (root) პრივილეგიებს;

**cp** *file dir* ჩაწერე *file*-ის ასლი *dir* დირექტორიაში. თუ ამ დირექტორიაში ამავე დასახელების (მაგრამ სხვა შინაარსის) ფაილი უკვე არსებობდა, ის გაქრება. თუ დირექტორია *dir* არ არსებობს, ბრძანება არ შესრულდება.

ამ ბრძანებაში შეიძლება ერთდროულად რამოდენიმე ფაილიც მიუთითოთ (დასახელებები უნდა დაშორდეს ინტერვალით). ყველა ჩაიწერება *dir* დირექტორიაში;

**cp** *file1 file2*  შექმენი *file1*-ს ასლი და დაარქვი მას *file2*. შედეგი იქნება ორი ერთნაირი შემცველობის ფაილი დასახელებებით *file1* და *file2*. თუ მანამდე სხვა შინაარსის *file2* ამ მისამართზე არსებობდა, ის გაქრება;

**cp** *dir1 dir2* იმის მიხედვით, არსებობს თუ არა *dir2* დირექტორია, ამ ბრძანებას განსხვავებული შინაარსი აქვს:

- თუ dir2 არსებობს, ბრძანება ნიშნავს: შექმენი *dir1*-ის ასლი *dir2*დირექტორიაში. ანუ შეიქმნება *dir2*-ის ქვედირექტორია *dir1*(მთელი თავისი შემცველობით). თუ ასეთი დასახელების (და განსხვავებული შინაარსის) ქვედირექტორია უკვე არსებობდა, ის გაქრება;

- თუ dir2 არ არსებობს, ბრძანება ნიშნავს: შექმენი *dir1*-ის ასლი და დაარქვი მას *dir2* ;

**exit** გამოდი ტერმინალიდან (და გადადი გრაფიკულ ინტერფეისში);

**ls** ჩამოწერე მიმდინარე დირექტორიის შემადგენლობა (არ აჩვენებს დამალულ ფაილებს);

**ls -a**  ჩამოწერე მიმდინარე დირექტორიის შემადგენლობა სრულად (აჩვენებს დამალულ ფაილებს);

**ls -l**  ჩამოწერე მიმდინარე დირექტორიის შემადგენლობა გრძელ ფორმატში;

**lshw** ჩამოწერე ამ კომპიუტერის შემადგენელი ნაწილების დეტალური აღწერა;

**man** *command* მაჩვენე ბრძანება *command*-ის აღწერა მანუალში;

**q** გამოდი მანუალიდან (და დაბრუნდი ტერმინალის ბრძანებების რეჟიმში);

**mkdir** *dir* შექმენი დირექტორია *dir* (თუ ახალი დირექტორიის მისამართი მითითებული არ არის, შეიქმნება მიმდინარე დირექტორიის ქვე-დირექტორია *dir*);

**mv** *file dir*  გადაიტანე *file* დასახელების ფაილი *dir* დირექტორიაში. თუ ასეთი დასახელების (მაგრამ სხვა შინაარსის) ფაილი უკვე არსებობდა, ის გაქრება. თუ დირექტორია dir არ არსებობს, ბრძანება არ შესრულდება;

**mv** *file1 file2* გადააწერე *file1* *file2*-ს. შედეგი იქნება ფაილი, რომლის დასახელება შეესაბამება *file2*-ს, შემადგენლობა კი - *file1*-ს. თუ *file2* დასახელების (მაგრამ განსხვავებული შინაარსის) ფაილი ამ მისამართზე უკვე არსებობდა, ის გაქრება;

**mv** *dir1 dir2* იმის მიხედვით, ბრძანების მომენტისთვის არსებობს თუ არა *dir2* დირექტორია, ამ ბრძანებას განსხვავებული შინაარსი აქვს:

- თუ *dir2* არსებობს, ბრძანება ნიშნავს: გადაიტანე *dir1*დირექტორია *dir2*-ში. ანუ *dir1* (მთელი თავისი შემადგენლობით) გახდება არსებული *dir2* დირექტორიის ქვედირექტორია;

- თუ *dir2* არ არსებობს, ბრძანება ნიშნავს: დაარქვი *dir1*-ს *dir2*.

**pwd** დაწერე მიმდინარე დირექტორიის მისამართი

**rm** *file* წაშალე *file* დასახელების ფაილი. თუ ფაილის მისამართი დაზუსტებული არ არის, წაიშლება ამ დასახელების ფაილი მიმდინარე დირექტორიაში;

**rm \*** წაშალე ყველა ფაილი მიმდინარე დირექტორიაში;

**rm -r \***  წაშალე მიმდინარე დირექტორიის ყველა ფაილი და ყველა ქვედირექტორია (მათი შემცველობითურთ);

**rmdir** *dir* წაშალე დირექტორია *dir*. წასაშლელი დირექტორია არ უნდა იყოს მიმდინარე დირექტორია. ეს ბრძანება შლის მხოლოდ ცარიელ დირექტორიებს.

**sudo** *command* დროებით მომეცი ადმინისტრატორის (root-ის) უფლებები *command* ბრძანების შესასრულებლად (*command* ბრძანება იწერება სრულად, არგუმენტებიანად). კომპიუტერი მოგთხოვთ თქვენს (და არა ადმინისტრატორის) პაროლს (შეყვანისას პაროლი არ გამოჩნდება). თუ პაროლი სწორია, command ბრძანება შესრულდება, თუ არასწორია - არ შესრულდება (შეტყობინება იმის შესახებ, რომ პაროლი არასწორია არ დაიწერება).

თუ პაროლი სწორად შეიყვანეთ, შემდეგი 15 წუთის განმავლობაში შეგიძლიათ sudo ბრძანება პაროლის გარეშე გამოიყენოთ. ამის შემდეგ კომპიუტერი კვლავ მოგთხოვთ თქვენს პაროლს. და ა.შ.

**touch** *file* შექმენი ფაილი დასახელებით *file*. შექმნის ცარიელ ფაილს

**GIT**

**git init** გითის რეპოზიტორიის ინიციალიზება

**git status** გვიჩვენებს რეპოზიტორიის მიმდინარე მდგომარეობას (ცვლილებებს/წაშლას)

**git add** *file* ფაილის დამატება გითის რეპოზოტორიაში რომ მოხდეს ამ ფაილის განახლების ნებისმიერი შესწორების დაფიქსრება. ცვლილებების ნახვა შეგვიძლია **git status** ბრძანებით.

**git commit -m** “*message*” ეს ბრძანება გვეხმარება იმაში რომ ჩვენი ცვლილებები შევინახოთ გითის ისტორიაში და შესაბამისის ცვლილელბის ამსახველი კომენტარიც მივუწეროთ რათა შემდგომში მარტივი იყოს მოძებნა.

**git push** ბრძანება გამოიყენება იმისთვის რომ მოხდეს საკუთარი კომიტების დამატება მთავარ სერვერზე

**git branch** ბრძანება გვეხმარება იმაში რომ ვნახოთ თუ პროექტის რომელ ვერსიაზე ვმუშაოთ მიმდინარე რეალურ თუ სატესტო და ა.შ. **git branch** *name* ამ ბრძანებით ხდება ახალი ბრენჩის შექმნა.

**git checkout** *branchname* ბრძანება გვაძლევს საშუალებს რომ გადავერთოთ ბრენჩებში. **პ.ს.** თითოეულ ბრენჩს შეიძლება გააჩნდეს ინდივიდუალური ფაილები ერთმანეთისგან განსხვავებულები.