**OS VAJE**

1. **TEDEN**

> ssh remote@212.235.182.231

**Geslo:** vaje123

relativna pot: se nahajš nekje… da greš v drugo mapo se premikaš s cd .. in to

absolutna pot: greš od korena ( / ) dalje…

man ls (odpre manual)

man - če ni built in uporabiš man

help - če je built in

**NALOGE 1**

**1. Preverite, ali se nahajate v vašem domačem imeniku (/home/student)! V nasprotnem primeru pojdite v vaš domači imenik!**

* pwd
* cd /home/student

**2. Izpišite vsebino imenika!**

* ls

**3. V imeniku so tu tudi skrite datoteke. Izpišite jih!**

* ls -a

**4. Pojdite v korenski imenik diska:**

a. Z zaporedjem ukazov

* cd ..
* cd ..

b. Z absolutnim naslavljanjem (1 ukaz)

* cd /

c. Z relativnim naslavljanjem (1 ukaz)

* cd ../../

**5. Iz korenskega imenika se vrnite v domači imenik**

a. Po korakih z zaporedjem ukazov

* cd home
* cd student

b. Z absolutnim naslavljanjem

* cd /home/student

c. Z relativnim naslavljanjem

* cd home/student

d. Na najkrajši možni način

* cd ~

**6. Iz domačega imenika pojdite v imenik /etc/init.d**

a. Z zaporedjem ukazov

* cd /
* cd etc
* cd init.d

b. Z absolutnim naslavljanjem

* cd /etc/init.d

c. Z relativnim naslavljanjem

* cd ../../etc/init.dmdm

**7. Z ukazom ls izpišite vsebino korenskega imenika in rekurzivno vsebino vseh podimenikov korenskega imenika.**

* ls / --recursive **ali** ls / -R

**8. V obliki drevesa izpišite vsebino korenskega imenika in rekurzivno vsebino vseh podimenikov korenskega imenika do globine 2. Enkrat izpišite vse zapise v imenikih, enkrat samo imenike.**

* sudo apt install tree
* tree -a -L 2 /

Vsi zapisi:

* tree -L 2 /

Samo imeniki:

* tree -L 2 -d /

**NALOGE 2**

**1. V vašem domačem imeniku ustvarite podimenik vaje!**

**Izhodišče za naslednje naloge podimenik vaje v vašem domačem imeniku.**

* mkdir vaje

**2. Ustvarite podimenike** OS**,** os **in** Os**!**

* cd vaje
* mkdir OS
* mkdir os
* mkdir Os

**3. Ustvarite podimenik** operacijski sistemi**!**

* mkdir operacijski\ sistemi **ali** mkdir “operacijski sistemi”

**4. V imeniku** os **ustvarite podimenik** test**, pri čemer ukaz izvedete iz imenika vaje.**

* mkdir os/test

**5. Z enim ukazom odstranite celotno pot** os/test**!**

* rm -r os/test
* rmdir -p os/test

**6. Ustvarite datoteko** vaje.txt**! Namig: uporabite ukaz** touch**.**

* touch vaje.txt

**7. Kaj je glavna naloga ukaza** touch**?**

* Kreiranje datotek
* čas dostopa spremeni //nevem ki je sploh s tem mislu

**8. Prekopirajte** vaje.txt **v podimenik** OS**!**

* cp vaje.txt OS

**9. Prekopirajte** vaje.txt **še enkrat v podimenik** OS**, vendar tokrat datoteko hkrati preimenujte v** vaje2.txt**!**

* cp vaje.txt OS/vaje2.txt

**10. Datoteko** vaje.txt **v imeniku vaje preimenujte v** vaje3**!**

* mv vaje.txt vaje3

**11. V imeniku** vaje **poskusite ustvariti podimenik** vaje3**.**

* mkdir vaje3 (error: file exists)

**12. Datoteko** vaje3 **v imeniku vaje skopirajte v datoteko** test3.txt **(prav tako v imeniku** vaje**)!**

* cp vaje3 test3.txt

**13. Iz imenika** OS **z enim ukazom prekopirajte vse datoteke v imenik** Os**!**

* cp OS/\* Os

**14. Imenik** Os **v celoti prestavite v imenik** OS **(**Os **bo sedaj podimenik** OS**)!**

* mv Os OS

**15. Vsebino celotnega imenika** OS **z vsemi podimeniki z enim ukazom prekopirajte v imenik** operacijski sistemi**!**

* cp -r OS operacijski\ sistemi

**16. Iz imenika** vaje **z enim ukazom izbrišite vse datoteke, ki vsebujejo številko 3! Brisanje izvedite tako, da je potrebno brisanje vsake datoteke potrditi.**

* rm -i \*3\*

**Naloge 3**

**1. Napišite skripto** ustvari.sh**, ki v podanem imeniku (imenik podamo kot prvi argument) ustvari datoteke** test1.txt**,** test2.txt **in** test3.txt**!**

**Skripto poženite z imenikom** vaje **kot argument. Znotraj skripte so argumenti, ki smo jih podali, shranjeni v spremenljivkah** $# **(kjer** # **pomeni zaporedno številko argumenta – prvi argument npr. dobimo v spremenljivki $**1**).**

**V ustvarjene datoteke napišite poljubno vsebino (gedit, vi).**

* touch ustvari.sh
* nano ustvari.sh

#!/bin/bash **# Bash interpreter**

touch "$1"/test1.txt "$1"/test2.txt "$1"/test3.txt

* file sharnimo z ctrl + O; enter; izhod: ctrl+x;
* chmod +x ustvari.sh **# Executable permissions**
* ./ustvari.sh **# Poženemo - relativna pot**

**2. Ustvarite mehko povezavo konfiguracije na imenik /etc! Preverite, če povezava deluje! Kako to preverite?**

* ln -s /etc
* ls -l

**3. Ustvarite mehko povezavo z imenom simbolp na test1.txt! Spreminjajte vsebino datoteke test1.txt in simbolp!**

* ln -s test1.txt simbolp

Obe datoteki se spremenita zaradi mehke povezave

**4. Zbrišite datoteko test1.txt! Preverite, kako deluje povezava.**

* rm test1.txt

Povezava ne deluje več

**5. Ustvarite trdo povezavo z imenom trdap na test2.txt! Spreminjajte vsebino datoteke test2.txt in trdap!**

* ln test2.txt trdap

Obe datoteki se spremenita zaradi trde povezave

**6. Zbrišite datoteko test2.txt! Preverite, kako deluje povezava.**

* rm test2.txt

Povezava še zmeraj deluje

**7. Ustvarite trdo povezavo test2.txt na datoteko trdap!**

* ln trdap test2.txt

**8. S pomočjo ukaza stat poglejte, na katere inode kažejo datoteke test2.txt, test3.txt in trdap!**

stat <datoteka>

test2.txt: **670274**

test3.txt: 666742

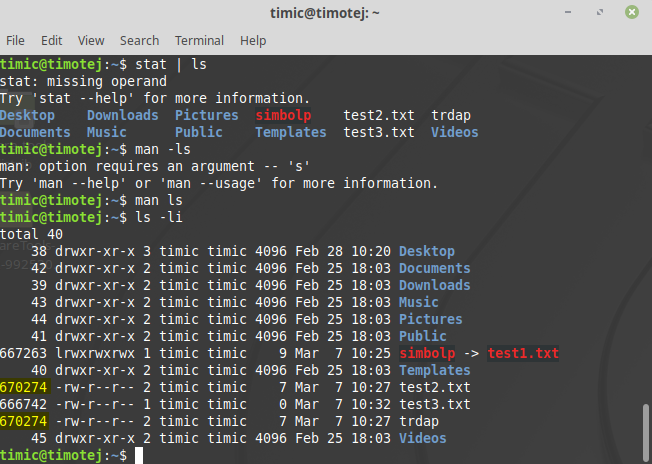
trdap: **670274**

stat test2.txt test3.txt trdap

**9. S pomočjo ukaza ls za vsak zapis (tudi za skrite) poiščite, koliko zapisov na datotečnem sistemu kaže na isti inode.**

ls -li

Dve kažeta na isti inode (označena z rumeno)



**10. Koliko prostora zavzame jedro linuxa? Kako lahko to pogledamo?**

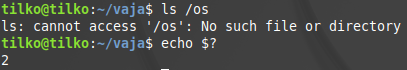
ls -lh /boot/vmlinuz\*

1. **TEDEN**

**1. Kakšen izhodni status vrne ukaz ls ~**

* izvedemo ukaz **ls ~** in nato še **echo $?** (nam pove status zadnjega ukaza ki se je izvršil)
* Izhodni status je **0** (OK)

**2. Kakšen izhodni status vrne ukaz ls /os**

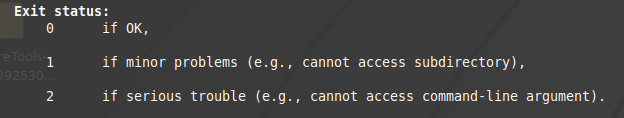
****

* Izhodni status je **2** (Misuse of shell builtins)

**3. Kakšen izhodni status vrne ukaz ls –R /, če ga prekinemo s ctrl+c**

* Izhodni status je **130** (Script terminated by Control-C)

**4. V man ls poglejte, kaj pomeni posamezni izhodni status.**

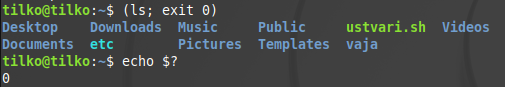


**5. Kakšen izhodni status vrne echo "OS"**

* Izhodni status je **0**

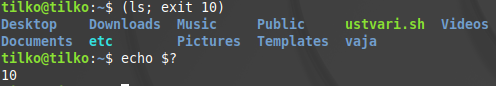
**6. Kakšen izhodni status vrne ( ls; exit 0 )**

* Izhodni status je **0**

****

**7. Kakšen izhodni status vrne ( ls; exit 10 )**

* Izhodni status je **10**

****

**8. Kaj se zgodi, če poženemo { ls; exit 0; }**

* Izhodni status je **0**

**9. Napišite skripto vsebina.sh, ki sprejme kot prvi argument imenik. Skripta naj izpiše kanonično pot do imenika (s pomočjo ukazov readlink ali realpath) in podrobnosti njegove vsebine (s pomočjo ukaza ls). Če se skripta zaključi uspešno, naj vrne status 0, če se zaključi neuspešno, naj vrne status 10.**

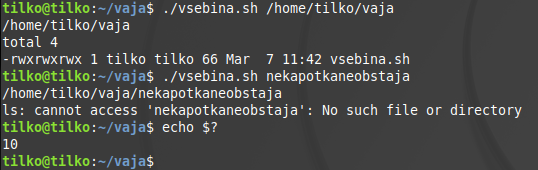
touch vsebina.sh

nano vsebina.sh

#!/bin/bash

readlink -f "$1" && ls -l "$1" && exit 0 || exit 10;

ec



**10. V imeniku ~/vaje ustvarite datoteko vaje3.txt in jo premaknite v imenik ~, če bo premik uspešen izpišite niz "datoteka uspešno premaknjena"**

mv vaje3.txt ~ && echo "datoteka uspešno premaknjena" || echo "datoteka ni uspešno premaknjena"

* v primeru napake izpiše datoteka ni uspešno premaknjena

**Naloge 2**

**1. Prijavite se v vsaj tri konzole (poskusite se prijaviti tudi z napačnim geslom ali neobstoječim uporabnikom)!**

ctrl+alt+F1

ctrl+alt+F2

ctrl+alt+F3

da prideš nazaj napišeš

ctrl+alt+F7

**2. Izpišite, kateri uporabniki so trenutno prijavljeni!**

* w

w - bolj podrobno

* users

**3. Kako izpišete samo podatke o vaši prijavi na konzoli, v kateri trenutno pišete ukaze?**

* whoami

**4. Izpišite, seznam zadnjih desetih prijav v sistem!**

* last -n 10
* sudo lastb - neuspešne prijave
* sudo lastb -n10

**5. Izpišite, seznam vseh uporabnikov, ki so registrirani v sistemu!**

* cat /etc/passwd

cat - izpiše vsebino tekstovne datoteke

**6. Kakšna imajo gesla?**

* sudo cat /etc/shadow

**7. Izpišite seznam vseh uporabnikov, vendar izpišite samo uporabnikovo uporabniško ime in domači imenik!**

* cat /etc/passwd | cut -d: -f1,6

d - kaj je ločnica (v našem primeru :)

f - izpiši prvi in šesti stolpec

**8. Poiščite, kateri uporabniki so včlanjeni v skupino sudo (kot sekundarno skupino)!**

* cat /etc/group | grep sudo
* grep "sudo" /etc/group (zgornja je lepša)

**9. V katere skupine je včlanjen uporabnik administrator?**

* groups administrator
* id administrator

id - pokaže tudi id grupe

**10. Katera je primarna skupina uporabnika administrator?**

* id administrator

**11. S pomočjo ukaza useradd ustvarite novega uporabnika miha (polno ime Miha Krajnc) skupaj z domačim imenikom /home/miha. V ukazu podajte tudi geslo.**

* sudo useradd -c "Miha Krajnc" -d /home/miha -m -p $(mkpasswd -m sha-512) miha

-m ustvari direktorij /home/miha (isto kot mkdir)

Geslo moramo kriptirati zato odpremo lupino in napišemo

* sudo apt install whois
* mkpasswd miha (ni preveč dobro zakriptirano)
* mkpasswd -m sha-256 miha

če damo v oklepaje izvedemo ukaz v podlupini

**12. Prijavite se kot uporabnik miha in preverite, če ste prijavljeni in pojdite v domači imenik uporabnika.**

* su miha
* whoami
* cd ~

**13. Ustvarite skupino uporabniki in vanjo včlanite uporabnika miha (uporabniki naj bo njegova sekundarna skupina). Prav tako uporabnika miha včlanite v skupino student.**

* sudo groupadd uporabniki
* sudo usermod -a -G uporabniki miha
* sudo usermod -a -G student miha

a (appenda k obstoječim skupinam, da pripada tudi teli skupini)

**14. Uporabniku miha dodajte pravice za izvajanje ukazov kot root (naj postane t.i. sudoer).**

* sudo usermod -a -G sudo miha

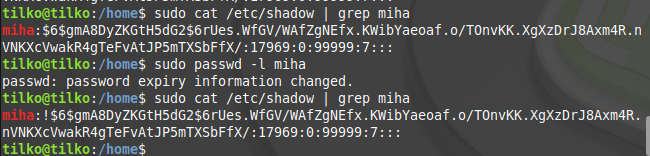


pripada skupini sudo

**15. Uporabniku miha onemogočite možnost prijave v sistem.**

tisti kriptirani password spremenimo

sudo passwd -l miha - doda **!** pred geslo, zaklene račun



sudo passwd -u miha (ga nazaj odklenemo)

**Naloge 4**



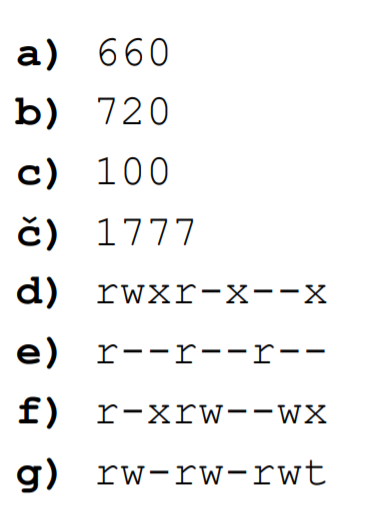
tip uporabnik skupina ostali

chmod 777 (r=4, w=2, x=1) - samo seštevamo (če dam 111 lahko samo izvršimo)

spreminjamo z ukazom chmod

* sudo usermod -s /bin/bash miha

zamenjamo interpreter/lupino v bash (da se lepše vidi)



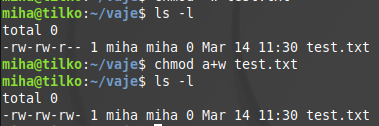
1. rw- rw- ---
2. rwxw-- ---
3. --x--- ---
4. 751
5. 444
6. 563
7. 1667 - 1 je spredi zaradi t-ja

**1. Kot uporabnik** miha **ustvarite datoteko** ~/vaje/test.txt**. Kakšne pravice nad to datoteko imajo posamezni uporabniki? Preizkusite spreminjati vsebino datoteke kot drug uporabnik.**

* su - miha
* mkdir vaje
* touch vaje/test.txt
* cd vaje
* -rw-rw-r--

**2. Vsem uporabnikom omogočite pisanje v datoteko** test.txt**.**

* chmod a+w test.txt (tudi +w dela)



**3. Ostalim uporabnikom (razen sebi in skupini) vzemite bralne in pisalne pravice nad datoteko**

* chmod o-rw test.txt

**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.**

* sudo usermod -aG miha student
* sudo chmod o-rw,u+rw,g+rw test.txt
* sudo chown miha:miha test.txt

ALI

* sudo chgrp student test.txt

**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 zbrišete? Zakaj?**

* me

**6. Kako bi poskrbeli, da lahko v imenik vaje vsi uporabniki shranjujejo in brišejo svoje datoteke, ne morejo pa brisati datotek ostalih uporabnikov?**

* chmod -r u+w,g+w,o-w vaje

**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.**

* touch preveri.sh
* nano preveri.sh

#!/bin/bash -x

test $(whoami) = root

test $? -eq 0 && last && lastb || echo nisi super user

-- damo -x če hočemo debuggat

#!/bin/bash

test $(whoami) = root && last && lastb || echo "Nisi sudo!"

# ali

test $(id -u) = 0 && last && lastb || echo "Nisi sudo!"

---

id -u je boljše za skripte, medtem ko id je uporabniku prijazno/berljivo

**8. V spremenljivko okolijsko spremenljivko PATH dodajte imenik, kjer se trenutno nahajate (ne trenutni imenik v splošnem) in podimenik vaje v vašem domačem imeniku!**

echo $PATH

-- na koncu bi radi dodali :in še pwd

PATH=echo ${PATH}:$(pwd) **# ko spremenimo PATH, lahko iz kjerkoli zaženemo skripte, ki so v našem imeniki (iskal ga bo po vseh imenikih v $PATH)**

****

**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.**

id uporabnik

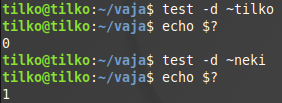
če obstaja bo 0, če ne bo 1 (ko napišemo echo $?)

cd ~uporabnik #nas poda v domači imenik

echo ~tilko

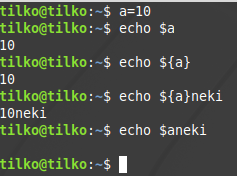
echo ~neki

-d preveri če imenik obstaja



---

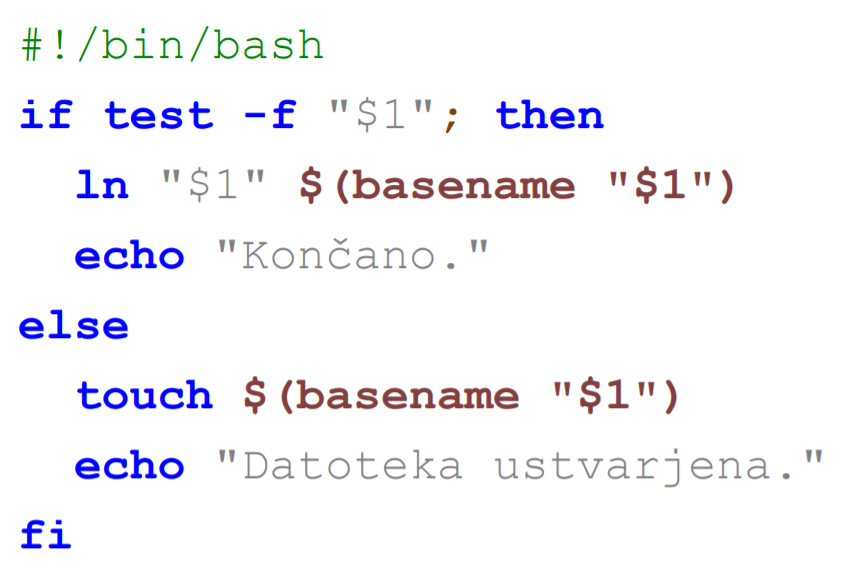
[**5. vaje**](https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=11189)



lahko poženemo z test ali pa z [ neki ] - uporabit morš [ in ]

test in [ sta vgrajena IN nevgrajena; vgrajena različica dela hitreje, ker je naložena v pomnilniku, nevgrajena pa se mora z diska pognat.

**1. Napišite skripto v BASH-u, ki naredi trdo povezavo na datoteko, ki jo podamo kot prvi argument ob klicu. Povezavo naredi v trenutnem imeniku in naj se imenuje enako kot izvorna datoteka, če pa izvorna datoteka ne obstaja, pa naj se ustvari prazna datoteka z istim imenom v trenutnem imeniku.**

****

**2. S pomočjo (a) zanke for in (b) zanke while napišite skriptni program, ki bo štel od 1 do 500 v enosekundnih intervalih.**

#!/bin/bash

for i in $(seq 1 500); do

echo -n "$i "

sleep 1

done

**3. Napišite skripto, ki bo združila vse navadne datoteke iz trenutnega imenika v pakete tar. V posameznem paketu naj bodo vse datoteke, ki imajo prvih n črk imena istih. Ime paketa naj bo sestavljeno iz prvih skupnih n črk in končnice tar. Število črk n podamo kot argument skripte.**

#!/bin/bash

for file in \*; do

if [ -f "$file" ]; then

start="${file:0:$1}"

if [ "$file" != "${start}.tar" ]; then

tar -uf "${start}.tar" "$file"

fi

fi

done

**4. Napišite skripto, ki bo združila vse navadne datoteke iz trenutnega imenika v pakete tar. V posameznem paketu naj bodo vse datoteke, ki imajo isto končnico (torej imajo isti niz za zadnjo piko). Ime paketa naj bo ime končnice. Če končnice ni, napišite null.**

**5. Napišite skripto, ki bo v podanem imeniku izpisala vse mehke povezave, ki ne kažejo na neko končno datoteko ali imenik (v verigi povezav manjka kakšen člen ali pa ciljna datoteka ne obstaja). Za vsako takšno povezavo naj skripta izpiše verigo povezav do člena, ki manjka. Imenik skripta prejme tako, da uporabniku izpiše poziv in čaka, da uporabnik vpiše imenik. Ko skripta obdela podani imenik in izpiše problematične mehke povezave, ponovno izpiše uporabniku poziv za vpis novega imenika itd. Če uporabnik ne vpiše imenika v petih sekundah, se skripta ustavi.**

#!/bin/bash

echo -n "Vpiši imenik: "

read -t 5 imenik || (exit && echo)

for datoteka in "$imenik"/\*; do

if [[ -L "$datoteka" ]] && [[ ! -e $(readlink -f "$datoteka") ]]; then

echo -n "$datoteka"

while [[ -h "$datoteka" ]]; do

link=$(readlink "$datoteka")

echo -n " -> $link"

datoteka="$imenik/$link"

done

echo

fi

done

**VAJE 6**

sudo find -name \*neki\* -type d -najde vse director

**1. Poiščite vse imenike, ki imajo lepljivi bit in pravico pisanja za ostale uporabnike.**

sudo find / -type d -perm -1002

sudo find / -type d -perm -1002 -exec stat -c "%A %n" {} \;-

**2. V domačem imeniku uporabnika student in vseh podimenikih (s potrjevanjem) pobrišite vse datoteke s končnico txt!**

sudo find ~/test -name \*.txt -exec rm -i {} \;

**3. Poiščite vse trde povezave na določeno datoteko. Namig pomagajte si z ukazom stat.**

sudo find -inum $(stat -c "%i" ab.tar)

std in - 0

std out - 1

std err - 2

ukaz >FILE -- Prenesi v FILE

ukaz <FILE -- Iz FILE v IN

ukaz 1>FILE -- Na std::out

ukaz 2>FILE -- Na std:err

za oboje

ukaz &>FILE

**Kako delujejo naslednje preusmeritve?**

touch vhod.txt

echo /etc/ > vhod.txt (Boljše, če daš “/etc/”)

**1. pwd <vhod.txt** (predhodno ustvarite datoteko vhod.txt, ki vsebuje niz /etc/)

Ignorira argument (ker pwd nima parametrov oz. ne prejema vhoda)

**2. pwd >izhod.txt** (standardni vhod je tipkovnica, preusmeri se v izhod.txt. Errorje pa šezmeri izpiše na zaslon)

Kar izpiše pwd (trenutni imenik), gre v izhod.txt

**3. ls -l izhod.txt**

Izpiše ls -l, kjer izpiše samo izhod.txt

**4. ls -l >izhod.txt**

Izpiše v datoteko izhod.txt

**5. ls -l 1>izhod.txt**

Standardni izhod izpiše v datoteko izhod.txt

**6. ls -l 2>izhod.txt**

Napake izpiše v izhod.txt

**7. ls -l vhod.txt >izhod.txt**

vhod.txt je podan kot argument ls-ja. Standardni izhod je izhod.txt standardni vhod je pa še vedno vhod.txt

**8. ls -l 2>>izhod.txt**

Napake vpiše (appenda!) na izhod.txt, drugo napiše na standardni izhod! (Terminal)

**9. ls -l 2>izhod.txt 1>&2**

1-ka kaže tam kjer je dvojka, posledično kaže na izhod. Sepravi vse izpiše na izhod.txt (vsebino in napake)

Standardni izhod kaže na REFERENCO standardne izhoda za napake

**10. ls -l 2>izhod.txt 1>2**

Isto kot prej, samo standardni izhod preusmerjamo v DATOTEKO “2”, ne pa na standardni izhod za napake!

**11. ls -l 1>&2 2>izhod.txt**

1ka kaže tam kamor 2ka (na zaslon) potem pa 2ka kaže na izhod. 1ka bo še vedno kazala na zaslon ker se ne spremeni.

**12. ls -l &>izhod.txt**

Vse preusmeri na izhod.txt

**13. cat <vhod.txt**

Izpiše vsebino datoteke vhod.txt

**14. cat <<konec**

Čaka na niz “konec” in potem konča

**15. cat >izhod.txt**

Kar vpišemo, se bo izpisalo v izhod.txt

**16. cat vhod.txt**

Izpiše vsebino datoteke vhod.txt

**17. cat vhod.txt vhod2.txt vhod3.txt >izhod.txt**

Izpiše vsebino vhod.txt, vhod2.txt, vhod3.txt v izhod.txt skupaj z novo vrstico vmes med vsebino teh

**18. cat <vhod.txt >izhod.txt**

Vsebino vhod.txt preusmeri v izhod.txt

**19. cat <izhod.txt >>izhod.txt**

To ne gre

**20. Ukazi pogosto izpisujejo v konzolo. Kako bi pognali ukaz, da bi preprečili te izpise?**

kar smo do zdej delali

**cevovod**

**1. V dveh konzolah poženite:**

cat >a.txt

cat <a.txt

**2. Ustvarite imenovan cevovod testpipe in poženite:**

cat >testpipe

cat <testpipe

**3. Kako delujejo naslednji ukazi in preusmeritve?**

• more /etc/passwd in cat /etc/passwd | more

• cat /etc/passwd | sort | more

• tree / | less

**4. Poiščite 5 največjih datotek v imeniku in njegovih vseh podimenikih.**

sudo find . -type f -exec stat -c "%s %n" {} r | head -n 5

sudo find . -type f -exec stat -c "%s %n" {} + | sort -n -r | head -n 5

**VAJE 7**

**1. Ali je uporabnik administrator prijavljen v sistem ali ne?**

who | grep -w administrator

echo $?

who | cut -d" " -f1 | grep -w administrator | sort | uniq

**2. Izpišite število trenutnih prijav uporabnikov!**

who | wc -l

**3. Za vsakega uporabnika izpišite, kolikokrat je trenutno prijavljen (v kolikih konzolah).**

users | tr " " "\n" | uniq -c

**4. Preštejte število datotečnih zapisov v trenutnem imeniku.**

ls | wc -l

**5. Permutirajte zaporedje števil od 100-999 in izpišite prvih 5 števil permutiranega zaporedja.**

seq 100 999 | shuf | head -n 5

**6. Vse male črke niza "Operacijski sistemi" pretvorite v velike.**

echo "operacijski sistemi" | tr a-z A-Z

**7. Izpis ukaza tree, ki izpiše celotno datotečno strukturo sistema, preusmerite v datoteke po 1000 vrstic. Imena datotek naj bodo oblike drevo\_#, kjer je # tromestna številka.**

tree / | split -a 3 -l 1000 -d - drevo\_

**8. V pomoči ukaza find poiščite omembe besede inode. Izpišite dve vrstici pred in po vrstici, kjer se nahaja beseda. Vrstice naj bodo oštevilčene.**

man find | grep "inode" -C 4

**9. Za vsako datoteko v imeniku izpišite število vrstic, besed in znakov, ki jih vsebuje.**

wc \*

wc $(ls)

**10. Prepišite ukaz naslednji ukaz tako, da bo uporabljal xargs. find / –inum $(stat test.txt –c "%i") –exec rm -i {} \;**

sudo find . -samefile test | xargs -pI {} rm {}

**I** nam omogoča vrstico po vrstici

**11. Primerjajte datoteki a.txt in b.txt. Če sta njuni vsebini enaki, izbrišite eno od njiju. Če je brisanje uspešno, izpišite ustrezno sporočilo.**

cmp a.txt b.txt 1>/dev/null && rm a.txt && echo "Brisanje uspešno"

**12. Izpišite vsebino datoteke /etc/passwd tako, da bodo stolpci poravnani.**

cat /etc/passwd | column -t -s:

**13. Izpis prejšnjega ukaza izpišite na zaslon, shranite v datoteko uporabniki.txt, poleg tega pa tudi v datoteko uporabniki\_arhiv.txt.**

cat /etc/passwd | column -t -s: | tee uporabniki.txt uporabniki\_arhiv.txt

**14. Ustvarite naključno geslo s pomočjo generatorja psevdonaključnih števil. Geslo naj bo sestavljeno iz sedmih alfanumeričnih znakov.**

cat /dev/urandom | tr -c -d 0-9a-zA-Z | head -c 7 | xargs

**Napišite skripto, ki izračuna največji skupni delitelj treh naravnih števil, ki jih podamo kot argumente skripti.**

**Primer klica: najvecjiDelitelj.sh 27 15 45**

**• Največji skupni delitelj dveh naravnih števil lahko izračunamo rekurzivno:**

**int gcd(a, b) {**

**if (b == 0) {**

**return a;**

**} else {**

**return gcd(b, a%b);**

**}**

**}**

**• velja: gcd(a, b, c) = gcd(a, gcd(b, c)) = gcd(gcd(a, b), c) = gcd(gcd(a, c), b)**

#!/bin/bash

gcd() {

if [ $2 -eq 0 ]; then

echo $1

else

echo $(gcd $2 $(($1 % $2)))

fi

}

gcd $1 $(gcd $2 $3)

**VAJE 8**

**Prijavite se kot uporabnik student in kot uporabnik administrator v dve virtualni konzoli.**

**1. Izpišite PID-je terminalov bash.**

pidof bash

**2. Izpišite PID samo svojega terminala.**

echo $$

**3. Izpišite vse procese na sistemu. Koliko jih je trenutno?**

ps aux | wc -l

*Šteje se tudi glava, lahko odštejemo ali pa uporabimo stikalo za brez glave*

**4. Izpišite:**

**a) samo svoje procese;**

ps -u timic

**b) samo procese, ki so pognani iz trenutnega terminala.**

ps -q $$

**5. Izpišite vse procese uporabnika root in njihove podatke izpišite v daljši obliki.**

ps aux | grep root

*ali*

ps -u root all

*ali*

ps -u root -l

**6. Iz ukaza top ubijte enega od svojih procesov (prej seznam omejite samo na svoje procese).**

H za help

U za user

Press **K** to kill

**7. Izpišite drevo vseh podprocesov terminala, v katerem se nahajate.**

pstree $$

**8. Ustvarite cevovod testfifo. V eni konzoli poženite cat >testfifo, v drugi konzoli**

**poženite cat <testfifo | tee c.txt d.txt in poglejte datotečne deskriptorje**

**vseh procesov v imeniku /proc/.**

mkfifo testfifo

**T1** cat > testfifo

**T2** cat < testfifo | tee c.txt d.txt

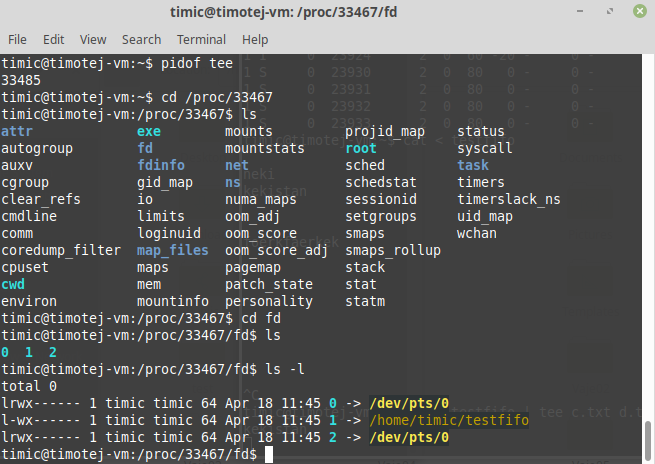
pidof cat

*cat 33484*

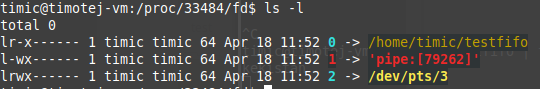
*cat 33467*

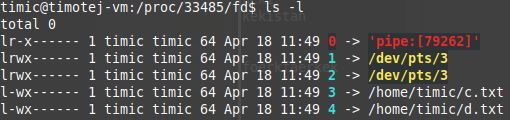
pidof tee

*tee 33485*



0 in 2 kažeta na naš terminal, 1 pa na cevovod *testfifo*. (Vhod)





**9. V imeniku /proc/ poglejte tudi okoljske spremenljivke obeh ukazov, status (v statusu**

**poiščite PID očeta), lokacijo programa, ki je bil pognan, in argumente, ki so bili podani ob**

**zagonu programa.**

cat environ | tr "\0" "\n"

cat status | grep PPid *- za ID očeta*

ls -l exe *- lokacija programa, ki je bil pognan*

cat cmdline | tr "\0" " " *- argumenti ob zagonu*

**1. Iz konzole poženite program gedit. Kaj se zgodi s konzolo?**

Notr lahko pišemo.

**2. Začasno ustavite izvajanje procesa gedit in izpišite seznam trenutnih opravil.**

CTRL-Z za ustavitev, jobs za seznam opravil.

Ko je ustvaljen, ne moremo pisati notri.

**3. Proces gedit naj nadaljuje z izvajanjem, tokrat v ozadju.**

gedit &xd

**4. Kaj se zgodi s procesom gedit, če zapremo konzolo, iz katere smo ga pognali?**

se zapre

**5. Kako bi preprečili zapiranje programa, če zapremo njegovega očeta?**

nohup xed

**6. Poženite naslednje procese v ozadju: gedit, xeyes, xclock. Proces gedit, ki teče v ozadju,naj se ponovno izvaja v ospredju.**

xed &

xeyes &

xclock &

jobs

fg {job}

**7. Prekinite izvajanje procesa gedit iz druge konzole s pomočjo ukaza kill. Kakšen je izhodni status procesa gedit?**

pidof xed

kill 2022 - **$? = 0**

kill -2 2022 - **Interrupted $? = 130**

kill -19 2022 - **$? = 147**

echo $?

**8. Iz druge konzole začasno ustavite izvajanje procesa xeyes. Pomagajte si signalom STOP.**

kill -20 2046

**9. Iz druge konzole ponovno poženite ustavljen proces xeyes.**

kill -18 2046

**10. Poženite ukaz tree / >izpis.txt z najvišjo prioriteto. Koliko hitreje se bo ukaz izvedel, če je to edini proces, ki teče na procesorju?**

time sudo nice -20 tree / > izpis.txt - Največja prioriteta, čas se ne spremeni, ker pri prejšnem primeru procesor ni porabil zdelo

Procesor mora še zmeraj čakati na disk, ker je počasnejši

**11. Napišite program, ki izvaja neskončno zanko. Iz konzole ga dvakrat poženite v ozadju. Spreminjajte prioriteto procesov in opazujte delež procesorskega časa.**

Prioriteta gre od -20 (najvišja) do 19(najnižja)

sudo renice -20 neki - za to nastavimo najvišjo prioriteto

**VAJE 9**

**Naloge 2**

**1. V konzoli preverite datum in uro in ju nastavite na pravilne vrednosti, če trenutno niso pravilne.**

tzdata

**2. V datoteko ukaz.txt napišite ukaz, ki bo na konec datoteke status.txt dodal besedilo Trenutno je:**

echo 'echo Trenutno je $(date) >> /home/administrator/vaje/status.txt' >> ukaz.txt

**3. Napišite ukaz, ki bo dne 20.05.2017 izvedel ukaz iz datoteke ukaz.txt**

at -f ukaz.txt 21.12.2019

atq

**4. Ukaz iz datoteke ukaz.txt naj se izvede čez 1 minuto in ob 4h zjutraj čez 10 dni.**

at -f ukaz.txt now+1min

at -f ukaz.txt 4am+10days

**5. Kako bi pogledali, kdaj se bodo izvedli ukazi, katerih čas še ni prišel? Kateri ukazi se bodo izvedli?**

atq

1 Sat Dec 21 10:55:00 2019 a timic

-- tudi čez 1 minuto, samo je že potekel.

3 Sun May 26 04:00:00 2019 a timic

cat status.txt

**6. Iz vrste izbrišite ukaz, ki naj bi se izvedel ob 4h zjutraj čez 10 dni.**

atrm 3

atq

1 Sat Dec 21 10:55:00 2019 a timic

**7. Nek ukaz bi radi izvajali periodično. S katerim ukazom bomo to storili?**

crontab -e

**8. V crontab dodajte vrstico z ukazom, ki bo v datoteko status.txt vsako minuto dopisal trenuten datum in uro.**

\* \* \* \* \* echo Trenutno je $(date) >> /home/administrator/vaje/status.txt

**9. Kako bi napisali vrstico z ukazom v crontabu, ki bo pognala isti ukaz kot v prejšnji nalogi vsak ponedeljek, sredo, četrtek in petek, od februarja do aprila, 5 minut čez vsako sodo uro.**

5 \*/2 \* 2-4 1,3-5 echo Trenutno je $(date) >> /home/administrator/vaje/status.txt

**Naloge 3**

**1. Izpišite zasedenost priklopljenih datotečnih sistemov. Izpis naj vsebuje človeku lažje berljive enote (KB, MB itd.).**

df -h

**2. Koliko prostora zavzame na disku domači imenik uporabnika administrator (ali pa student) z vsemi podimeniki?**

du -h /home/administrator

**3. Koliko prostora na disku zavzame datoteka ~/status.txt? Kako velika je vsebina datoteke ~/status.txt?**

du -h status.txt # 4.0K

ls -lh status.txt # 546B

stat status.txt # 546B

Blok je velik 4.0K, zato zasede 4KB na disku, v resnici je pa velik 546B.

umount za varno izklopitev USB ključka (umount /media/administrator/data), še prej pa naredimo df -h, da vidimo kje je montiran (FS bi moral biti /dev/sdb1...)

**VAJE 10 - Regularni izrazi**

**1. man grep | grep -C 100 "."**

Vsak (katerikoli) znak posebi - obarva vse

**2. man grep | grep -C 100 "\."**

Obarva vse pike, kjerkoli

**3. man grep | grep -C 100 "\.$"**

Vse kar se konča na piko

**4. man grep | grep -C 100 "a."**

Dolžine 2, prva črka je **a**, druga pa katerakoli

**5. man grep | grep -C 100 ".a."**

Dolžine 3, na sredini je **a**

**6. man grep | grep -C 100 "\Bce"**

**\B** pomeni znotraj besede (ne rob besede), dolžine 2, “ce” - na sredi besede (ne na začetku)

**7. man grep | grep -C 100 "ce\b"**

**\b** pomeni zaključek besede, najde samo “ce”, če je na koncu besede

**8. man grep | grep -C 100 "the"**

Označi vse “the”

**9. man grep | grep -C 100 "\"**

Error

**10. man grep | grep -C 100 "[yYnN]"**

Iščemo znake, katerekoli od naštetih v **[ ]**

**11. man grep | grep -C 100 "\b[yYnN]"**

Iščemo znake, ki se začnejo na črke v **[ ]**

**12. man grep | grep -C 100 "[a-z]"**

Iščemo črke od **a** do **z** (case-sensitive) na enem mestu

**13. man grep | grep -C 100 "[^a-z]"**

Vse kar ni mala črka (znak **^** negira)

**14. man grep | grep -C 100 "[a-z0-9]"**

Iščemo črke od **a** do **z** ter črke od **0** do **9** na enem mestu

**15. man grep | grep -C 100 "[-+0-9]"**

Najde **-** ali **+** ali **0** do **9**

**16. man grep | grep -C 100 "[^aeiou]"**

Vse razen samoglasnikov

**-- 2 --**

**1. man grep | grep -C 100 "[a-z][a-z]"**

Dolžine dva, od **a** do **z**

**2. man grep | grep -E -C 100 "[a-z]{2}"**

egrep, isto kot prej

**3. man grep | grep -E -C 100 "ti\*"**

Začne se s **ti**, ena ali več ponovitev (lahko je t, ti, tii, …)

**4. man grep | grep -E -C 100 "(ti)\*„**

Tako: ti, titi, tititi...

**5. man grep | grep -E -C 100 "([tc])([ho])"**

th

to

ch

co

**6. man grep | grep -E -C 100 "\b[A-Z][a-z]{2}"**

Začne na veliko začetnico, nato pa morata biti dve mali

**7. man grep | grep -E -C 100 "\b[A-Z]?[a-z]{2}"**

Lahko je velika začetnica, ali pa je ni

**8. man grep | grep -E -C 100 "th.\*s"**

Začne z th in konča z s-jem (poišče najdaljše nize)

**9. man grep | grep -E -C 100 "th[^s]\*s"**

Išči samo znake, ki nimajo s, ko pa najde s pa ustavi

**Naloge 1:**

**1. V pomoči ukaza grep poiščite vse nize, ki se začnejo s črko A ali a.**

man grep | grep -E -C 100 "\b[Aa]"

**2. V datoteki /etc/passwd poiščite vse vrstice, kjer se pojavi beseda admin.**

cat /etc/passwd | grep -E -C 100 "\badmin\b”

**3. V datoteki test.txt poiščite vse vrstice, kjer so za besedilom na koncu vrstice presledki.**

cat /etc/passwd | grep -E -C 100 " +$"

**Kaj predstavljajo naslednji regularni izrazi (s katerimi nizi se ujema)?**

**4. ".{6}"**

Katerikoli znak šestkrat

**5. ".{6,}"**

Šestkrat ali več

**6. "(19|20)[0-9]{2}"**

Letnice

2009, 1999…..

**7. "(+|-)?[0-9]+(\.[0-9]+)?"**

+, - ali nič

vsaj ena števka

lahko je cel blok ali pa ni (če je, mora biti na prvem mestu pika, nato številke do n)

**8. "(\w+)\1"**

Niz, ki se ponovi

**9. "\<([^ ]+)[ ]+\1"**

Najde dve ponovitvi iste besede

**<([^ ]+)** najde besedo

**[ ]** presledek

**10. Kaj predstavlja naslednji regularni izraz (s katerimi nizi se ujema)? Izraz poenostavite tako, da odstranite ponavljanja.**

**[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}**

0-9, od 1 do 3x

.

0-9, od 1 do 3x

.

0-9, od 1 do 3x

.

0-9, od 1 do 3x

Ujema IPje:

192.168.3.0

**Narobe:** Lahko tudi “999…”! Naredimo split

Dodamo **^***izraz***$**

**11. Kaj predstavlja naslednji regularni izraz (s katerimi nizi se ujema)?**

**^(0[1-9]|[12][0-9]|3[01])[-/.](0[1-9]|1[012])[-/.](19|20)[0-9][0-9]$**

Začne

0

1-9 ali [1 ali 2], 0-9 ali 3[0 ali 1]

[- ali / ali .]

0, 1-9 ali 1[0 ali 1 ali 2]

- ali / ali .

19 ali 20

0-9

0-9

Konča

To je kandidat za datum, lahko pride 99.99.2099…

**---**

Regularni izrazi nimajo sklada (ne znajo šteti), je en nivo višje

Če ne znamo rešit, gremo višje, ampak performance pade

HTML ne moremo parsat z regexom.