**Команди во Linux OS**

**Tee –** овозможува во новата датотека да се запишува и на крај таа да се испечати

Mkdir folder – kreirame nov folder

Rmdir – brisenje na direktorium

Rm fajl – brisenje na fajl

Mv fajl1.txt fajl2.txt – preimenuvanje na fajl1.txt vo fajl2.txt

Cd folder/ - vleguvame vnatre vo folderot (pr.[175009@os folder]$)

Pwd – ja kazuva patekata

Touch skripta.sh – kreiranje na skripta(na ist nacin moze I .txt, .pdf, .html, .odt,…….)

Wc -c < “skripta.sh” => kolku karakteri (bajti) ima taaa skripta

Cd .. – se vrakjame na pocetokot (pr. [175009 ~]$)

Cp lista.txt folder/ - ovozmozuva kopiranje na site txt datoteki da se prenesat vo folderot

Cp \*.txt folder/ - kopiraj gi site txt datoteki I stavi gi vo folder

Grep -l – gi printa datotekite koi sodrzat odreden zbor

Grep -L – gi printa datotekite koi ne sodrzat odreden zbor

Grep -v – se koristi za negacija

**~ => oznacuva deka se machuva nekoj patern**

-o gi dava samo detalite pr. Grep -o ‘^18’ – vrakja samo 18

-n gi dava na koi linii se naogjaat oderedeni zborvi (koi linii ja ispolnuvaat odredena komanda) (rednite broevi na liniite) => zavisno kade se izvrsuva ova vo koja komanda

-e => se koristi koga sakame vo ramkite na eden red da izvrsime povekje promeni so edna komanda

& => oznacuva deka pred/posle redot moze da dodame nekoj znak (pr. \$&/ na pocetok na redot stavi $)

Tr ‘a-z’ ‘A-Z’ – site mali bukvi gi zamenvi so golemi

Tr -s ‘ ‘ – ako ima posledovatelni prazni mesta gi pravi vo edno

Tr -s ‘E’ – ako ima posledovtelni bukvi e gi spojuv vo edno(-s znaci da se spoj/napravi vo nesto drugo so e definirano)

Tr -d a-z – gi brise site mali bukvi (-d znaci brisi)

Tr -c ‘a-z’ ‘ ‘ – se sto ne e bukva zameni go so prazno mesto (-c znaci negacija)

**Sed komanda - > najcesto se koristi koga treba da izvrsime zamena na nesto (pr. ( so prazno mesto)**

sed 's/sistemi/sis5/' novadat.txt => zamenuvame sistemi vo sis5 vo datotekata novadat.txt, pri toa mora da stavime napred s so toa se vrsi zamenata

DOKOLKU VO LINIJATA SE SREKAVAAT POVEKJE ZBOROVI SISTEMI TOGAS SO OVAA KOMANDAA KE NI BIDE ZAMENET SAMO PRVIOT ZBOR

**RESHENIE: sed 's/sistemi/sis5/g' novadat.txt => stavame/g sto znaci global pa ke ni izvrsi zamena na site zborovi soodvetno**

**Ignore case -i**

**sed '/0\.[0-9][0-9]$/p' fruit\_prices.txt** => /vo ovoj del pisuvame komanda/, /0 – znaci ni zapocnuva na 0,

/0**\.** Bidejki imame \. Toa znaci deka ke go zema znakot kako tocka a ne kako bilo kolku karakteri, potoa imame cifra I uste edna cifra, $ oznacuva da zavrsuva na broj. So p se printaat site onie linii koi ja zadovoluvaat ovaa komanda /komanda/p od dadena datoteka. **Bidejki nemame s toa znaci deka imame printanje ili kopiranje na linii koi ja zadovoluvaat taa komanda (ZNACI S SE KORISTI SAMO ZA ZAMENA).**

**Sed -n '/0\.[0-9][0-9]$/p' fruit\_prices.txt => ni gid ava samo onie koi go ispolnuvaat uslovot**

**Awk komanda**

**\***

**cat fruit.txt | awk '{if($2 >=1) {print $0 "\*"} else print $0}'**

**REORDER**

**cat fruit.txt | awk '{if($3<=75) {print $0 "REORDER"} else print $0}'**

**awk '{ if($0 ~ /[1-9][0-9]\*\.[0-9][0-9]/) print($0, "\*"); else print($0); }' fruit\_prices.txt**

* So if proveruvame dali nekoja linija pocnuva so 1 pa ima opcionalno uste edna cifra posle toa imame slash(\.) sto ni oznacuva deka ima tocka pa posle nea sleduvaat 2 cifri od 0-9. Ako ova e zadovoleno togas na tie linii pokraj celata linija ($0) dodadi \* inaku ispecati ja samo linijata.

**awk '{ if ($3 < 75) printf("%s\t%s\n", $0, "REORDER"); else print($0);}' fruit\_prices.txt**

**awk '{ if (length($0) > 19) printf("%s\t%s\n", $0, "REORDER"); else print($0);}' fruit\_prices.txt**

**Команди**

команда за листање на сите датотеки чии имиња завршуваат на .txt (пр. zad1.txt, datoteka.txt)

**ls | grep .txt**

команда за листање на сите именици чии имиња се составени само од букви (големи или мали) и цифри (пр. OSgr1, zad1t1)

**ls -l | grep ^d | awk '{ print $9 }' | grep [A-Z]\*[a-z]\*[0-9]\***

команда за листање на сите датотеки чие име е со должина 10 (пр. Operativni, zad1t1a.sh)

**ls | awk 'length($0)==10 {print $0}' => printanje na site datoteki so dolzina 10**

команда за листање на сите именици чие име е од следниов облик: почнува на f, следуваат произволен број цифри, потоа gr и на крај големи букви (пр. f123456grKNI, f11223grKE)

**ls -l | grep 'f[0-9]\*gr[A-Z]\*' | awk '{print $10;}'**

команда за листање на сите датотеки чие име има повеќе од 5 карактери, каде крајниот е c, а содржат и точка (.) (пр. zad1.c, .javac)

**ls | awk 'length($0)>=5 { if ($0 ~ ".") print $0}' | grep c$**

**Sort komanda**

Primer, sortiranje na informaciite spored golemina gledame koja kolona e goleminata pa zatoa pisuvame -k I reden broj na kolonata koja ja dava goleminata (dokolku stavime sort -nk 6 ni gi podreduva od najmalata do najgolemata golemina). No ako vaka go ostam ovaj sort go zema vo predvid od 6 pa do kraj celiot red, ako sakame da ja zema vo predvid samo 6 kolona stavam zapirka I povtorno se pisuva redniot broj na kolonata 6, t.e sort -nk 6, 6

**Uniq komanda**

**Otstranuva duplikati I moze da se koristi vo kombinacija so sort primer,**

**1, 2, 3, 8, 3, 4, 11, 4, 6, 7**

**Sort -n | uniq ili sort -un**

**1, 2, 3, 4, 6, 7, 8, 11**

**Zadaca ispitna reshena na auditoriski**

**Da gi isfiltrirame site najaveni student cij indeks zapocnuva so 18, ip adresata e 77. I da gi sortira spored vreme**

1. **Who | grep ‘^18’**
2. **Who | grep ‘^18’ | awk ‘{if ($NF ~ /(77.\*)/) print ; }‘**
3. **Who | grep ‘^18’ | awk ‘{if ($NF ~ /(77.\*)/) print ; }‘ | sort -k4,4**

**So $NF pristapuvame do posledniot element (vo zadacava namesto $NF moze da se pristapi I direktno spored redniot broj na kolonata odnosno $5)**

**So NR ni ja dava koj e redot od koj se printaat podatoci (vrakja cifri)**

**~ oznacuva patern odnosno koj e uslovot sto ke se napise**

**/(77.\*)/ oznacuva da se prikazat onie adresi koi zapocnuvaat na 77 pa imaat . a posle toa bilo sto**

**So sort -k4,4 gi sortira spored vremeto**

**Аудиториска вежба UNIX2**

**Задача 7**

#!/bin/bash

for var in A B C; do

echo $var

done

**Задача 8**

#!/bin/bash

fruits="Apple Pear Tomato Peach Grape"

for fruit in $fruits; do

if [ "$fruit" = "Tomato" ] || [ "$fruit" = "Peach" ]; then

echo ${fruit}es

else

echo ${fruit}s

fi

done

**Задача 9**

**#!/bin/bash**

**for file in $\*; do**

**if [ ! \_f $file ] || [ ! \_r $file ]; then**

**echo "Cannout view $file"**

**else**

**echo "====== $file ======="**

**cat $file**

**fi**

**done;**

**Задача 10**

#!/bin/bash

if [ $# -lt 3 ]; then

cat << HELP

./z3.sh -- preimenuva povekje fajlovi

USAGE: ./z3.sh 'regxp' 'replacement' files...

EXAMPE: x/z3.sh 'HTM$' 'html' \*.html

HELP

exit 1

fi

old="$1"

new="$2"

shift

shift

for file in $\*; do

if [ -f "$file" ]; then

newfile=`echo "file" | sed sed "/${old}/${new}/g"`;

if [ -f "$newfile" ]; then

echo "ERROR: File exists"

else

mv "$file" "$newFIle"

echo "Success!"

fi

fi

done

**Задача 11**

#!/bin/bash

price=$1

price=`echo "scale=2; 3.5 \* $price" | bc`

echo $price

**Задача 12**

#!/bin/bash

toLower() {

echo $@ | tr '[A-Z]' '[a-z]';

}

toLower $@

**Задача 13**

#!/bin/bash

all=`ls`

OIFS="$IFS"

IFS=$'\n'

select file in $all "exit"; do

if [ -z "$file" ]; then

continue;

fi

if [ "$file" = "exit" ]; then

break;

fi

if [ ! -f "$file" ]; then

echo "Not a file";

continue;

fi

echo "======= $file ======"

cat $file

done

IFS="$OIFS"

**Задача 14**

**ls -lh | sed -e '/^[^-]/d' -e 's/\s\+[0-9]\+.\*:[0-9]\{2\}\s/ /'**

Излистај ги сите фајлови, отстрани ги оние кои не почнуваат со – па додаваме d, т.е означува бришење на тие фајлови. Потоа соодветно ја правиме замената т.е, гледаме да ни почнува со празно место односно тоа го бележиме со /ѕ потоа бараме после празното место да имаме една цифра, со .\* кажуваме дека се што се наоѓа измеѓу без разлика што и да е да се скокне се додека не дојдеме до : па да има после тоа уште 2 цифри. \s/ / со ова кажуваме сето тоа што сакаме да се скипне да го замениме со едно празно место.

**ls -lh | sed -e '/^[^-]/d' -e 's/\s\+[0-9]\+.\*\([0-9]\{4\}\|:[0-9]\{2\}\)\s/ /'**

Истото објаснување како и претходно само што сега имаме или услов односно да зема во предвид или 4 цифри или : па 2 цифри. Овозможува мечување на фајлови кои имаат само година или точно време.

**Задача 15**

#!/bin/bash

if [ $# -lt 1 ]; then

echo "USAGE: $0 <filename>"

exit 1

fi

awk ' { for (i=NF; i>=1; i--) { printf("%s ", $i); } printf("\n"); } ' $1

NF – number fields (kolku argumenti ima na vlez)

**Задача 16**

#!/bin/bash

Прам директориуми посебни за парни и непарни папки и на почетокот ги означувам како hidden за да не дојде до грешки кога paren ќе треба да се премести во парен или непарен. Затоа после листањето на сите фолдери прам mv .paren paren I mv .neparen neparen.

mkdir .paren .neparen

for file in \*; do

case `wc-c < "$file"` in

\*[2468]) mv "$file" .paren ;;

\*[13579])mv "$file" .neparen ;;

\*) echo "Error, NaN" ;;

esac

done

mv .paren paren

mv .neparen neparen

**Командни процедури UNIX3**

**Задача 1**

last | grep ^175009 | awk '{ print $10; }' | sed 's/(//' | sed 's/)//' | awk -F: 'BEGIN {total=0;} {total += $1\*60 + $2} END { print total; }'

1. Прв чекор треба да повикаме соодветна команда која дава кога корисникот бил последно најавен, тоа е командата **last**
2. После тоа треба да провериме за кој корисник се однесува тоа, т.е според кој индекс ќе се пребарува и тука ја користиме наредбата grep која врши филтрирање и плус ставаме и **^** кое означува да ги зема во предвид само оние редици кои почнуваат со тој број на индекс
3. Нареден чекор кој треба да го направиме е да ги земеме во предвид само времињата а тоа го правиме така што броиме кој е редниот број на колоната која го прикажуват времето, т.е тоа е колона бр.10, па со **awk** командата се принтаат само времињата
4. После тоа времињата бидејќи ни се дадени во формат (02:38) треба да се ослободиме од заградите па тоа го правиме така што користиме sed команда, односно **sed 's/(//**' што значи заградата ( замени ја со ништо (не смее да ставиме празно место меѓу //). На ист начин се прави истото и за отстранување на другата заграда
5. Бидејќи времињата се меѓусебно одвоени со : треба да воведиме друг сепаратор односно да ги одвоиме со празно место, тоа се прави со **-F:**
6. Наредно бидејќи се бара да се прикаже вкупното поминато време на корисникот во минути тоа може да се направи директно во awk командата на тој начин што првата колона т.е првите 2 цифри 02 се множат со 60 и ги собираме со останатите 2 цифри, **awk -F: '{ print $1\*60 + $2}'**. Каде што $1 и $2 соодветно ни ги даваат тие колони.
7. Бидејќи сите времиња сега треба да се соберат во една променлива тогаш тоа се прави со декларирање на варијабли. Варијабла се иницијализира на тој начин што користиме блок BEGIN кој се извршува пред да почне итерацијата низ секој ред и има блок END кој се евалуира откако ќе се изминат сите редови (при тоа овие блокови се пишуваат со големи букви). Кај блокот BEGIN дефинираме променлива која ќе биде сумата од сите времиња и ја иницијализираме на 0, т.е **BEGIN {total = 0;}**. Во END блокот само ја печатиме вредноста за total. **END { print total; }.** Но, проблем тука е тоа што мора на вредноста тотал да се додаваат претходните вредности. **{total += $1\*60 + $2}**

Ако истото се напише во shell скрипта тогаш кодот е следен:

#!/bin/bash

Со $# гледаме кои се аргументите кои ги препраќа корисникот, во нашиот случај ни треба само еден аргумет а тоа е корисичкото име(индексот). Па проверуваме дали не е = на 1 со -ne. Ако не е 1 тогаш се праќа информација како корисникот треба да ја користи скриптата (не е мн важно тоа како ќе биде формулирано доволно е само да има USAGEi I exit).

if [ $# -ne 1 ]

then

echo "USAGE: `basename $0` username"

exit 1

fi

Доколку излезната датотека постои, потребно е да се пребрише.

if [ -f out.txt ]

then

rm out.txt

fi

Ја повикуваме командата која претходно ја формулиравме

$1 е всушност индексот но го проследуваме како влезен аргумент

total=`last | grep ^$1 | awk '{ print $10; }' | sed 's/(//' | sed 's/)//' | awk -F: 'BEGIN {total=0;} {total += $1\*60 + $2} END { print total; }'`

во датотеката out.txt ќе го запише вкупното време

echo $total > out.txt

На крајот од скриптата да се прикаже содржината на датотеката out.txt.

cat out.txt

**Задача 2**

**Ps -ef = команда со која може да ги излистаме сите процеси кои што ги има еден корисник**

**#!/bin/bash**

**На почеток ни се праќа само еден аргумент на влез, а тоа е корисничкото име(индексот). Па затоа правиме проверка дали бројот на аргументи не е е 1 и ако не е соодветно се испишува порака за користење на скриптата**

**if [ $# -ne 1 ]**

**then**

**echo "USAGE: `basename $0` username"**

**exit 1**

**fi**

**Доколку излезната датотека постои, потребно е да се пребрише**

**if [ -f out.txt ]**

**then**

**rm out.txt**

**fi**

**Во втората колона ни се наоѓаат процесите на корисникот затоа со awk ‘{print $2;}’ пристапуваме до таа колона**

**processes=`ps -ef | grep ^$1 | awk '{print $2;}'`**

**Со for циклус ги изминуваме сите процеси на даден процес и ги зачувуваме во променлива ps, додека пак во променливата count го зачувуваме бројот на појавувања на еден ист процес и со wc -l ни го дава соодветниот број на појавувања**

**$1 - корисникот**

**$3 – процесот**

**^${ps}$ - колку пати се содржи одреден процес во даден процес и при тоа да започнува и да завршува на даден број**

**^${ps}$ = ^${11353}$**

**for ps in $processes**

**do**

**count=`ps -ef | grep ^$1 | awk '{print $3}' | grep ^${ps}$ | wc -l`**

**Бидејќи имаме 2 вредности кои треба да се запишуваат во скрипта out.txt користиме >> за да не настане пребришување ( > користиме ако имаме само една вредност/променлива за запишување)**

**echo $ps $count >> out.txt**

**done**

**На крајот од скриптата да се прикаже содржината на датотеката out.txt**

**cat out.txt**

**Задача 3**

**#!/bin/bash**

**На почеток треба да бидат проследени 2 аргументи на влез, па правиме проверка дали бројот на внесени аргументи е помал од 2 ако е се испишува соодветна порака**

**if [ $# -lt 2 ]**

**then**

**echo "USAGE: `basename $0` ./src\_folder/ ./dest\_folder'"**

**exit 1**

**fi**

**Двата директориуми ги зачувуваме во соодветни променливи**

**from=$1**

**to=$2**

**Доколку директориоумот каде треба да се зачуваат датотеките не постои треба да се креира**

**if [ ! -d $to ]**

**then**

**mkdir -p $to**

**fi**

**Скрипта која ќе ги копира сите датотеки од именик зададен како прв аргумент кои започнуваат на број, по што следуваат само мали букви и имаат екстензија .out, во именик зададен како втор аргумент од командна линија.**

**files=`ls -l | grep ^- ${from} | awk '{print $10}' | grep '[0-9][a-z][a-z]\*\.out`**

**for file in files**

**do**

**Се копираат еден по еден фајл од даден директориум во некој краен**

**cp ${from}$file ${to}${file}**

**done**

**Потоа да се пресмета и да се испечати вкупната големина на ископираните датотеки за кои корисникот има привилегии за извршување.**

**total=`ls -l ${to} | grep ^-..x | awk '{print $6}' | awk 'BEGIN {total=0} {total+=$1} END { print total}'`**

**Се печати total**

**echo $total**

**ВТОР КОЛОКВИУМ ЗАДАЧИ**

**(25 poeni)**

1. **Даден е следниот код:**

**ls -l**

**total 12**

**drwxr-xr-x 2 175009 domain users 4096 Jul 5 00:44 asistent**

**drwxr-xr-x 2 175009 domain users 4096 Jul 5 00:45 asistent10**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent10.sh**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent1111.txt**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent.txt**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:46 Asistent.txt**

**drwxr-xr-x 2 175009 domain users 4096 Jul 5 00:46 nov\_asistent**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:46 permissions.java**

**Со извршување на наредбата ls -l | grep "\.[a-z]\{2,3\}$" што ќе се добие на излез?**

**Објаснување на наредбата: после . треба да има некоја буква од [a-z] и при тоа да е запазено дека после . максималната големина на датотеките да биде 2 или 3. Според тоа,**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent10.sh**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent1111.txt**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:45 asistent.txt**

**-rw-r--r-- 1 175009 domain users 0 Jul 5 00:46 Asistent.txt**

**Подпрашање:**

**Сменете ги привилегиите така што корисникот ќе може да ја чита и извршува датотеката и при тоа да се користи октален запис.**

**chmod 500 permissions.java (Појаснување: 5=101=>r-x, ако ни бараше обичен запис тогаш ќе имавме chmod u=rx permissions.java)**

1. **Која команда ќе ги подреди само фајловите во еден директориум според времето на промена и ќе ги испечати само првите 2 записи (оние кои биле последно креирани).**

**ls -l | grep '^-' | sort -n -k 9 | head -2**

**-n означува нумеричко подредување, т.е од најмал кој најголем**

**-k означува која е колоната која ја проверуваме/сортираме**

**Испит Јули 2020**

**Задача4:**

**Да се напише едноставна shell скрипта која ќе има функција на калкулатор, така што како оператори се земаат (+,-,\*,/).**

**#!/bin/bash**

**if [ $# -ne 2 ]**

**then**

**"USAGE: `basename $0` <num1> <num2>"**

**exit 1;**

**fi**

**operand1=$1**

**operand2=$2**

**result=0**

**select operator in Addition Substraction Multiplication Division exit**

**do**

**case $operator in**

**Addition) result=$(($1 + $2)) ;;**

**Substraction) result=$(($1 - $2)) ;;**

**Multiplication) result=$(($1 \* $2)) ;;**

**Division) result=$(($1 / $2)) ;;**

**exit) echo "Ne moze da se presmeta" && break ;;**

**esac**

**echo "Result: $result"**

**done**

**Испит септември прв рок 2020**

**(35 поени) Задача4:**

**Да се напише shell скрипта passed\_students.sh која прима само датотеки со екстензија .csv. Доколку бројот на аргументи е 0 или поголем од 1 или докумнтот не е со екстензија .csv да се испише соодветна порака. Дадени се во скрипта име, презиме, индекс и број на поени од тест за студенти. Треба да се преименува датотеката од passed\_csv\_file.csv -> grades\_csv\_file каде во ноата датотека ќе ги има само оние студенти кои имаат барем 5 поени.**

**passed\_csv\_file.csv => grades\_csv\_file**

**Ime Prezime Indeks Poeni Ime Prezime Indeks Poeni**

**Petre Petrovski 123456 5 Petre Petrovski 123456 5**

**Ivan Ivanoski 123455 2 Simona Simonoska 123453 8**

**Trajce Trajceski 123454 3 Martin Marrtinoski 123452 9**

**Simona Simonoska 123453 8 Krste Krstevski 123451 5**

**Martin Marrtinoski 123452 9**

**Krste Krstevski 123451 5**

**#!/bin/bash**

**if [ $# -ne 1 ]**

**then**

**"USAGE: `basename $0` .csv file"**

**exit 1;**

**fi**

**# gi pecatime prvo so cat site studenti koi se naogjaat vo datotekata passed\_csv\_fle.csv**

**# potoa so awk komandata proveruvame koi studenti imaa povekje ili ednakvo na 5 poeni i ni$**

**students=`cat passed\_csv\_file.csv | awk '{ if($4 > 4) printf("%s %s %d %d\n", $1, $2, $3, $4);}'`**

**# studentit koi imaat povekje od 5 poeni gi stavame vo druga csv papka (ne mi gi printa ed$**

**echo $students > grades\_csv\_file**

**# so cat prikazuvame na ekran koj e izlezot**

**cat grades\_csv\_file**

**Испитни задачи 2018**

**Задача1:**

(15 поени) Да се напише една команда со која ќе се излистаат корисничките имиња на сите корисници кои во моментов се најавени на системот и припаѓаат на генерацијата студенти запишани во 2015 година. Студиската година на запишување на ФИНКИ се чита од индексот на студентот, каде првите две цифри ја означуваат годината: 151111, 152222, итн.

**last | grep '^15'**

б) (20 поени) Да се напише една команда која ќе го отпечати бројот на уникатни IP адреси од кои во изминатите денови се најавиле корисници на системот. Притоа, да се избројат само оние IP адреси кои не припаѓаат на лабораториите на ФИНКИ. Aдресите од лабораториите на ФИНКИ се во формат 10.10.X.Y. (**се падна и на испитот во септември прв рок беше дадено на заокружување)**

**last | awk '{print $3}' | grep -v '10.10'**

**Задача2:**

Да се испишат индексите на моментално најавените корисници кои имаат IP адреса која започнува на 10, вториот октет од адресата да содржи 2 или 3 бројки, а третиот октет да содржи точно 3 бројки од кои последната бројка е 0 и потоа индексите да се сортираат во растечки редослед, така што нема да има повторување на индексите.

**Пр. Доколку листата од моментално најавени корисници изгледа вака:**

222222 pts/3 2018-04-16 22:17 (10.205.120.182)

222222 pts/4 2018-04-16 21:31 (10.205.120.182)

333333 pts/5 2018-04-16 21:32 (45.205.122.40)

444444 pts/7 2018-04-16 23:29 (45.205.120.131)

888888 pts/8 2018-04-16 22:38 (55.205.120.145)

111111 pts/9 2018-04-16 23:01 (10.205.121.131)

999999 pts/15 2018-04-16 21:59 (10.205.120.162)

555555 pts/30 2018-04-16 23:05 (10.205.44.221)

333333 pts/34 2018-04-16 23:27 (10.205.124.45)

**Ќе се исфилтрираат само IP адресите:**

222222 pts/3 2018-04-16 22:17 (10.205.120.182)

222222 pts/4 2018-04-16 21:31 (10.205.120.182)

999999 pts/15 2018-04-16 21:59 (10.205.120.162)

**Излезот треба да изгледа вака:**

222222

999999

**Решение:**

**#!/bin/bash**

**if [ $# -ne 1 ]**

**then**

**"USAGE: `basename $0` .txt file"**

**exit 1**

**fi**

**korisnici=` cat lista.txt | awk '{ print $1, $5}' | grep '10.[0-9][0-9][0-9].[0-9][0-9]0' | sort | uniq`**

**echo $korisnici > out.txt**

**cat out.txt**

**Задача3:**

Да се прикажат во растечки редослед индексите на корисниците кои биле најавени во сабота, 14 април, така што индексите кои ќе се прикажат треба да завршуваат на парен број.

**Пр. Доколку листата од претходно најавени корисници изгледа вака:**

111111 pts/3 89.205.122.40 Sun Apr 15 22:43 - 02:05 (03:22)

222222 pts/3 79.126.216.207 Sun Apr 15 22:41 - 22:42 (00:01)

333333 pts/2 185.86.236.95 Sat Apr 14 22:35 - 22:45 (00:09)

444444 pts/2 79.126.216.207 Sat Apr 14 22:31 - 22:33 (00:02)

666666 pts/1 89.205.125.145 Sat Apr 14 22:26 - 00:37 (02:11)

**Излезот треба да изгледа вака:**

444444

666666

**Решение:**

**#!/bin/bash**

**if [ $# -ne 1 ]**

**then**

**"USAGE: `basename $0` <file.txt>"**

**exit 1;**

**fi**

**users=`cat korisnici.txt | awk '{ print $1, $5, $6 }' | grep 'Apr 14' | sort | uniq | awk '{ print $1;}`**

**cat lista1.txt | grep 'Apr 14' | awk '{print $1}' | grep '[02468]$'**

**cat lista1.txt | grep 'Apr 14' | awk '{if($1 % 2 == 0) print $1}' | sort**

**echo $users > out.txt**

**cat out.txt**

**Задача4:**

Со користење на филтри и регуларни изрази да се реши следната задача:

Да се излистаат сите процеси кои се извршуваат на системот (Командата е ps -ef). Од добиената листа на процеси мора да се избришат сите процеси на bash(доколку ги има), и да се најдат сите кои се започнати во 20та минута. За крај мора да се испечати најголемиот PID од добиената листа на процеси.

**Пример Листа на тековни процеси**

UID PID PPID C STIME TTY TIME CMD

root 25433 2 0 11:19 ? 00:00:20 [kworker/3:2]

root 25525 2 0 11:24 ? 00:00:20 [kworker/0:1]

root 25533 2 0 11:24 ? 00:00:00 [kworker/1:0]

root 26103 2 0 11:31 ? 00:00:00 [kworker/2:0]

root 26426 2 0 11:36 ? 00:00:20 [kworker/3:0]

root 26522 2 0 11:44 ? 00:00:00 [kworker/u256:1]

root 26737 2 0 11:46 ? 00:00:00 [kworker/1:1]

root 26795 2 0 11:48 ? 00:00:00 [kworker/u256:2]

root 26803 2 0 11:48 ? 00:00:25 [kworker/0:2]

root 26811 1 0 11:49 ? 00:00:00 /lib/systemd/systemd-hostnamed

nasi 26826 1848 1 11:49 ? 00:00:15 gedit /home/nasi/Work/Zadaca

nasi 26843 1848 2 11:49 ? 00:00:00 /usr/lib/gnome-terminal/gnome-

nasi 26849 26843 0 11:49 pts/2 00:00:00 bash

nasi 26872 26843 0 11:49 pts/18 00:00:20 bash

nasi 26895 26843 0 11:49 pts/19 00:00:10 bash

nasi 26910 26849 0 11:49 pts/2 00:00:00 ps -ef

**Очекуван излез**

25533

**Решение:**

**#!/bin/bash**

**if [ $# -ne 1 ]**

**then**

**"USAGE: `basename $0` <file.txt>"**

**exit 1;**

**fi**

**processes=`cat procesi.txt | grep -v 'bash' | awk '{print($2, $5);}' | grep ':2[0-9]' | sort | tail -1 $**

**echo $processes > out.txt**

**cat out.txt**

**Задача5:**

(15 поени) Со користење на концептите на цевки, редирекција и / или командите за филтрирање на текст, напишете кратка командна процедура во UNIX која ќе направи преименување на сите датотеки од тековниот именик кои се зачувани со различни варијации на екстензијата .txt во правилни датотеки чија екстензија ќе биде точно .txt. (Пример: zad1.TxT -> zad1.txt, zad2.TXT -> zad2.txt, итн.)

Кратката командна процедура да се зачува под името zad3-t1.sh.

**Решение:**

**#!/bin/bash**

**lista=`ls -l | grep -i '.txt' | grep -v .txt`**

**for file in $lista**

**do**

**name=`echo $file | awk -F. '{ print $1; }'`**

**mv "$file" "$name.txt"**

**done**

**Задача6:**

Напишете скрипта dop.sh која ги печати имињата на сите извршни датотеки во тековниот директориум чие име е составено од точно 6 карактери, во датотека datoteka.txt (доколку датотеката не постои треба да се креира, а доколку постои истата мора да биде празна пред внесувањето на бараните податоци)

\*За да проверете дали една датотека е извршна, побарајте помош преку командата man test

**Решение:**

**Ova ne e moe reshenie ama raboti**

**lista=`ls -l | grep ^- | awk '{ print $1, $9 }'`**

**for f in $lista**

**do**

**if [ -x $f ]**

**then**

**x=`echo $f | awk -F"." '{ print $1 }' | wc -m`**

**target=$(($x - "1"))**

**if [ $target = "6" ]**

**then**

**echo $f > "datoteka.txt"**

**fi**

**fi**

**done**

**Мое решение:**

**#!/bin/bash**

**lista=`ls -l | grep '^-' | awk '{ print $10;}' | grep '[a-z]\{6\}'`**

**for file in $lista**

**do**

**if [ -x $file ]**

**then**

**echo $file > out.txt**

**fi**

**done**

**cat out.txt**