grep:

iš failo /data/ld/ld2/stud2001 išrinkit eilutes, kurių gale yra skaičius 7 (grep "7$" /data/ld/ld2/stud2001)

iš failo /data/ld/ld2/stud2001 išrinkit eilutes, kurių pradžioje yra skaičius 2 (grep "^2" /data/ld/ld2/stud2001)

iš failo stud2001 išrinkit eilutes, kurių gale yra raidė n arba d arba raidės nuo r iki v. (grep "d$\|n$\|[r-v]$" stud2001)

iš failo stud2001 išrinkit eilutes, kuriose yra įrašas apie studentus, kurių vardai sutampa su Jūsų vardu. (grep "Mildaras" stud2001)

iš failo stud2001 išrinkit eilutes, kuriose nėra IF 9/1 grupės studentų. (grep -v "IF 9/1" stud2001)

užrašykite filtrą, kuris ieškotų eilučių, kuriose nėra jokio skaičiaus. (grep -vE '[0-9]' stud2001)

užrašykite filtrą, kuris ieškotų eilutėse bet kokios raidės(tiek didžiosios, tiek mažosios). (grep -E '[A-Za-z]' stud2001)

užrašykite filtrą, kuris iš failo stud2001 išrinktų eilutes, kuriose po A raidės seka raidė n arba raidė u, o po jų seka raidės d ir r. (grep -E 'A[nu]dr' stud2001)

užrašykite filtrą, kuris iš failo stud2001 išrinktų eilutes, kuriose po A raidės seka raidė n arba raidė u, o po jų seka raidės d ir r, o po raidės r nėra raidės e. (grep -E 'A[nu]dr[^e]' stud2001)

užrašykite filtrą, kuris faile stud2001 ieškotų eilučių su įrašais apie 9/1 arba 9/2 grupių studentus. (grep -E 'IF 9/[12]' stud2001)

užrašykite filtrą, kuris ieškotų eilučių besibaigiančių seka andr arba mari. (grep -E 'andr$\|mari$' stud2001)

Ką reiškia išplėstos reguliarios išraiškos: a(i)+b, a(i)\*b, a(i)?b. Duomenų pavyzdys: source:shell|duom (Išplėstos reguliarios išraiškos yra specialios teksto paieškos sintaksės, kurios leidžia atlikti tikslesnes ir sudėtingesnes teksto paieškas.)

awk:

Sudarykite programą awk11, kuri iš failo /data/ld/ld2/stud2001, atrinktų studentus, kurių vardai sutampa su Jūsų vardu ir atspausdintų eilutės numerius kuriose jie rasti, jų pavardes bei vardus.

(awk -F : '{if ($4 == "Mildaras") {print NR, $4, $5}}' /data/ld/ld2/stud2001 | tail -1)

Sudarykite programą awk12, kuri iš failo stud2001 išrinktų tas eilutes, kuriose yra IF 9/2 grupės studentai ir gale atspausdintų, kiek eilučių rasta.

(awk '$2 == "IF 9/2" {print $0; count++} END {print "Number of lines found: " count}' stud2001)

Sudarykite programą awk13, kuri iš failo /data/ld/ld2/awk/preke išrinktų tas eilutes, kuriose yra informacija apie tas prekes, kurios visos yra parduotos.

(awk -F : '$1 ~ $2' /data/ld/ld2/awk/preke)

Sudarykite programą awk14, kuri iš failo /data/ld/ld2/awk/preke išrinktų maksimalias stulpelių reikšmes bei gale jas atspausdintų.

(

'BEGIN {max\_kaina = max\_gauta = max\_parduota = 0}

{if ($1 > max\_kaina) max\_kaina = $1;

if ($2 > max\_gauta) max\_gauta = $2;

if ($3 > max\_parduota) max\_parduota = $3}

END {print "Max kaina: " max\_kaina;

print "Max gauta: " max\_gauta;

print "Max parduota: " max\_parduota}' /data/ld/ld2/awk/preke

)

Sudarykite programą awk15, kuri iš šio failo suskaičiuotų, kiek paketų ir kiek baitų yra iškeliavę į Internetą.

(

#!/bin/sh

BEGIN {ip =" 62. 40.103.217"; Baitu\_suma = Paketu\_suma = 0;FS=":";}

{

if($3 == ip)

{

for(i=1;i<=NF;i++) Baitu\_suma += $6; Paketu\_suma += $5;

}

}

END { print "\nProgramos darbo pabaiga", "Baitu skaicius: ", Baitu\_suma, "Paketu skaiciu: ", Paketu\_suma;}

)

Sudarykite awk filtrą, kuris kaip duomenis imtų ls -l rezultatus ir suskaičiuotų/atspausdintų bendrą katalogo failų dydį (užtenka vertint ls išvedamą dydį, t.y. kietų nuorodų, failų tipų ir kt. vertint nereikia)

(

ls -l | awk '{sum += $5} END {print "Total size: " sum}'

)

Bendros užduotys:

awk '{count[$4]++} END{for (i in count) {print i, count[i]}}' /data/ld/ld2/stud2001 | sort -k2,2nr

awk '{print $1}' /data/ld/ld1/Solaris\_access\_log | sort | uniq -c | sort -nr | head -10

awk '{arr[$1]+=$10} END {for (ip in arr) print arr[ip], ip}' /data/ld/ld1/Solaris\_access\_log | sort -nr | head -10

awk -F\" '($2 ~ /^GET/) {print $2}' /data/ld/ld1/Solaris\_access\_log | cut -d' ' -f2 | sort | uniq -c | sort -nr | head -10

awk '($9 == 404) {print $1}' /data/ld/ld1/Solaris\_access\_log | sort | uniq -c | sort -nr

awk -F\" '($9 == 404) {print $1, $2}' /data/ld/ld1/Solaris\_access\_log | sort | uniq -c | sort -nr | head -10

...

...

1.

#!/bin/sh

echo "Ivesk savo varda: "

read v

echo "Grupe: "

read g

echo "Vartotojo numeri: "

read n

echo $v, $g, $n

2.

#!/bin/sh

echo "Elementu kiekis: " $#

echo "Visi elementai: " $\*

3.

1.

#!/bin/sh

# Check if the number of arguments is not equal to 2

if [ $# -ne 2 ]; then

echo "Error: Please provide exactly two arguments."

exit 2

fi

file1="$1"

file2="$2"

# Check if both files exist

if [ ! -e "$file1" ] || [ ! -e "$file2" ]; then

echo "Error: One or both files do not exist."

exit 1

fi

# Get the file owners

owner1=$(stat -c '%U' "$file1")

owner2=$(stat -c '%U' "$file2")

# Check if the file owners are the same

if [ "$owner1" == "$owner2" ]; then

echo "The file owners are the same: $owner1"

exit 0

else

echo "The file owners are different: $owner1 and $owner2"

exit 1

fi

2.

#!/bin/bash

if [ $# -eq 0 ]; then

echo "Klaida: nėra argumentų."

exit 1

else

echo "Argumentų skaičius: $#"

echo "Argumentų sąrašas: $@"

fi

#!/bin/bash

if [ $# -eq 0 ]; then

echo "Klaida: nėra argumentų."

exit 1

elif [ $# -eq 1 ] && [ -d "$1" ]; then

echo "Katalogo \"$1\" turinys:"

ls -1 "$1"

else

echo "Argumentų skaičius: $#"

echo "Argumentų sąrašas: $@"

fi

#!/bin/bash

if [ $# -eq 0 ]; then

echo "Klaida: nėra argumentų."

exit 1

elif [ $# -eq 1 ]; then

if [ -d "$1" ]; then

echo "Katalogo \"$1\" turinys:"

ls -1 "$1"

elif [ -f "$1" ] && [ -r "$1" ]; then

read -p "Ar rodyti failo \"$1\" turinį? (T/N) " answer

if [ "$answer" == "T" ] || [ "$answer" == "t" ]; then

cat "$1"

fi

else

echo "Argumentų skaičius: $#"

echo "Argumentų sąrašas: $@"

fi

fi

#!/bin/bash

if [ $# -ne 2 ]; then

echo "Klaida: privalote nurodyti du argumentus."

exit 1

fi

dir="$1"

file="$2"

if [ -d "$dir" ] && [ -f "$file" ]; then

mv "$file" "$dir" 2>/dev/null

if [ $? -ne 0 ]; then

cp "$file" "$dir"

if [ $? -eq 0 ]; then

echo "Failas \"$file\" nukopijuotas į katalogą \"$dir\"."

else

echo "Klaida: nepavyko perkelti arba nukopijuoti failo."

exit 1

fi

else

echo "Failas "$file" perkeliamas į katalogą "$dir"."

fi

else

if [ ! -d "$dir" ]; then

echo "Klaida: pirmasis argumentas nėra katalogas."

fi

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

echo "Klaida: antrasis argumentas nėra failas."

fi

exit 1

fi

5.

#!/bin/bash

declare -A days\_lt\_to\_en=(

["1"]="Monday" ["2"]="Tuesday" ["3"]="Wednesday" ["4"]="Thursday" ["5"]="Friday" ["6"]="Saturday" ["7"]="Sunday"

["pirmadienis"]="Monday" ["antradienis"]="Tuesday" ["trečiadienis"]="Wednesday" ["ketvirtadienis"]="Thursday"

["penktadienis"]="Friday" ["šeštadienis"]="Saturday" ["sekmadienis"]="Sunday"

)

while true; do

read -p "Įveskite skaičių arba lietuvišką savaitės dienos pavadinimą (Q arba q baigti): " input

if [ "$input" == "Q" ] || [ "$input" == "q" ]; then

exit 0

elif [ -n "${days\_lt\_to\_en[$input]}" ]; then

echo "Angliškas savaitės dienos pavadinimas: ${days\_lt\_to\_en[$input]}"

else

echo "Klaida: neteisingas įvesties formatas."

fi

done

6.

#!/bin/bash

counter=1

for arg in "$@"; do

echo "$counter: $arg"

counter=$((counter + 1))

done

#!/bin/bash

for entry in \*; do

if [ -f "$entry" ]; then

file "$entry"

fi

done

#!/bin/bash

read -p "Įveskite skaičių eilutę (skaičiai atskirti tarpais): " input

numbers=($input)

first\_number="${numbers[0]}"

smaller=()

larger\_or\_equal=()

for num in "${numbers[@]}"; do

if [ "$num" -lt "$first\_number" ]; then

smaller+=("$num")

elif [ "$num" -ge "$first\_number" ]; then

larger\_or\_equal+=("$num")

fi

done

echo "Skaičiai mažesni už pirmąjį: ${smaller[\*]}"

echo "Skaičiai didesni arba lygūs pirmąjam: ${larger\_or\_equal[\*]}"

#!/bin/bash

for file in "$@"; do

if [ -f "$file" ]; then

readable="neskaitomas"

executable="nevykdomas"

if [ -r "$file" ]; then

readable="skaitomas"

fi

if [ -x "$file" ]; then

executable="vykdomas"

fi

echo "$file >> $readable/$executable"

else

echo "Klaida: \"$file\" nėra failas."

fi

done

#!/bin/bash

while [ "$#" -gt 0 ]; do

echo "$1"

shift

done

#!/bin/bash

if [ "$#" -ne 1 ]; then

echo "Klaida: privalote nurodyti vieną argumentą."

exit 1

fi

for i in $(seq 0 "$1"); do

echo "$i"

done

#!/bin/bash

if [ "$#" -eq 0 ]; then

echo "Klaida: privalote nurodyti bent vieną naudotoją kaip argumentą."

exit 1

fi

while true; do

for user in "$@"; do

if who | grep -q "^$user"; then

echo "Naudotojas $user yra prisijungęs."

exit 0

fi

done

sleep 60

done

#!/bin/bash

function process\_directory() {

local dir="$1"

local index\_file="$dir/index.txt"

local tmp\_file=$(mktemp)

if [ ! -f "$index\_file" ]; then

touch "$index\_file"

fi

for entry in "$dir"/\*; do

if [ -d "$entry" ]; then

process\_directory "$entry"

elif [ -f "$entry" ] && [ "$entry" != "$index\_file" ]; then

if ! grep -q "^$(basename "$entry")/" "$index\_file"; then

read -p "Įveskite aprašymą failui $entry: " description

echo "$(basename "$entry")/$description" >> "$index\_file"

fi

echo "$(grep "^$(basename "$entry")/" "$index\_file")" >> "$tmp\_file"

fi

done

# Pašalinkite perteklinius failų aprašymus

mv "$tmp\_file" "$index\_file"

}

process\_directory "."