

Poznámky k započtovému testu z Operačních systémů

Malinda Kryštof

24. dubna 2024

1 Zadání z předchozích let

1.1 Zadání 1

1. Vytvořte soubor male.txt, kde bude text psaný malými písmeny. Vytvořte script velka.sh, který změní velikosti písmen v souboru male.txt a uloží do souboru velke.txt

```
1 echo "toto je náhody text do souboru male.txt" > male.txt
```

Listing 1: Vytvoříme soubor male.txt

```
1 #!/bin/bash
2
3 # Kontrola, zda existuje soubor male.txt
4 if [ ! -f male.txt ]; then
5     echo "Soubor male.txt neexistuje."
6     exit 1
7 fi
8
9 # Převod malých písmen na velká a uložení do souboru velke.txt
10 tr '[:lower:]' '[:upper:]' < male.txt > velke.txt
11
12 echo "Převod na velká písmena byl dokončen."
```

Listing 2: Jak by měl vypadat velka.sh

```
1 chmod +x velka.sh # Přidáme execute permission, aby se script dal spustit
2 ./velka.sh # Spustíme script
3 cat velke.txt # Zkontrolujeme output scriptu
```

2. Vytvořte soubor ceny v tomto tvaru, kde druhý sloupec je oddělen a třetí platy v jednotlivých pracovníků. Napište skript secti.sh, který sečte všechny platy pro pracovníky oddělení 1.

```
1 echo "
2 Jahoda 1 22010
3 Bruna 2 80000
4 Zuda 1 12000
5 " > ceny.txt
```

```
1 #!/bin/bash
2
3 # Kontrola, zda existuje soubor ceny.txt
4 if [ ! -f ceny.txt ]; then
5     echo "Soubor ceny.txt neexistuje."
6     exit 1
7 fi
8
9 # Sečtení platů pro pracovníky oddělení 1
10 suma=0
11 while read -r jmeno oddeleni plat; do
12     if [ "$oddeleni" -eq 1 ]; then
13         suma=$((suma + plat))
14     fi
15 done < ceny.txt
```

```

16
17 echo "Celková suma platů pro pracovníky oddělení 1 je: $suma"

```

Listing 3: skript secti.sh

3. V Linuxu vytvořte skript s názvem adresare, který z aktuálního adresáře, ve kterém jsou adresáře i soubory, vypíše seznam všech adresářů do souboru

```

1  #!/bin/bash
2
3  # Získání seznamu adresářů v aktuální adresáři
4  find . -type -d > seznam_adresaru.txt
5
6  echo "Seznam adresářů byl uložen do souboru seznam_adresaru.txt"

```

4. Napište skript, kterému se zadají 3 čísla jako parametr. Po zadání těchto čísel se tato čísla sečtou a výsledek se uloží do souboru.

```

1  #!/bin/bash
2
3  number1=$1
4  number2=$2
5  number3=$3
6
7  # Výpočet součtu
8  sum=$((number1 + number2 + number3))
9
10 # Uložení výsledku do souboru
11 echo "Součet čísel $number1, $number2 a $number3 je: $sum" > soucet.txt
12
13 echo "Součet byl uložen do souboru 'soucet.txt'."

```

1.2 Zadání z 23.04.2024

1. V textovém souboru jsou (vytvořte) pod sebou názvy 3 souborů a jejich různé přípony oddělené tečkou. Pomocí skriptu vypíše tyto přípony do externího souboru s názvem externí

```

1  #!/bin/bash
2
3  # Zdrojový soubor
4  zdroj="soubory.txt"
5
6  # Output soubor
7  output="externi"
8
9  pripony=$(awk -F '.' '{print $NF}' $zdroj)
10 echo $pripony > $output

```

2. V linuxu vytvořte skript s názvem zapis.sh, který zjistí zda se dá do souboru zapisovat. Výsledkem bude text ma obrazovce "lze zapisovat" nebo "nelze zapisovat".

```

1  #!/bin/bash
2
3  # Název testovacího souboru
4  test_file=$1
5
6  # Funkce pro kontrolu zápisových práv
7  check_write_permission() {
8      if ls -l $test_file | grep -q -E "\s*w"; then
9          echo "lze zapisovat"
10         else
11             echo "nelze zapisovat"
12         fi
13     }
14
15 # Zobrazení práv souboru
16 echo "Práva souboru $test_file:"

```

```

17 ls -l $test_file
18
19 # Kontrola zápisových práv
20 check_write_permission

```

Listing 4: zapis.sh musel jsem modifikovat pipe aby se zobrazil ale normálně je to jen ta vertikální čárka

- Po ukončení vše odstraňte a použijte příkaz history -c a history -w

```

1 history -c && history -w
2 # Toto smaže historii našeho shellu (bash)

```

2 Věci s přednášek

2.1 AWK

Příkaz	Popis
awk '{print \$1}'	Vytiskne první sloupec z každého řádku
awk '\$3 > 10'	Vytiskne řádky, kde třetí sloupec je větší než 10
awk -F':' '{print \$1, \$3}'	Nastaví oddělovač na dvojtečku a vytiskne první a třetí sloupec
awk 'NR==2{print \$0}'	Vytiskne druhý řádek
awk '\$NF == "pattern\"'	Vytiskne řádky, kde poslední sloupec je "pattern"

Operátor	Popis
/pattern/	Porovnává aktuální řádek s regulárním výrazem
==	Porovnává rovnost
!=	Porovnává nerovnost
^	Porovnává začátek pole
\$	Porovnává konec pole
\$1 /pattern/	Porovnává první pole s regulárním výrazem
\$1 ! /pattern/	Negace porovnání prvního pole s regulárním výrazem

2.2 SED

Příkaz	Popis
s/pattern/replace/	Nahradí první výskyt patternu v každém řádku řetězcem replace
s/pattern/replace/g	Nahradí všechny výskyty patternu v každém řádku řetězcem replace
/pattern/d	Vymaže řádek, který obsahuje pattern
/pattern/p	Vytiskne řádek, který obsahuje pattern
1,10d	Vymaže řádky od prvního do desátého
\$d	Vymaže poslední řádek
\$p	Vytiskne poslední řádek
i\Text before	Vloží text před řádek
a\Text after	Vloží text za řádek

2.3 Jak psát regex

Výraz	Popis
.	Kterýkoli znak
^	Začátek řádku
\$	Konec řádku
*	Nula nebo více opakování
+	Jedno nebo více opakování
?	Nula nebo jedno opakování
[]	Jakýkoli znak uvedený v hranatých závorkách
[a-z]	Jakýkoli znak od 'a' do 'z'
[^]	Jakýkoli znak, který není uveden v hranatých závorkách
\w	Jakýkoli alfanumerický znak
\W	Jakýkoli nealfanumerický znak
\d	Jakékoli číslice
\D	Jakýkoli znak, který není číslicí
\s	Jakýkoli bílý znak
\S	Jakýkoli nebílý znak

2.4 Příkazy v Linuxu

Příkaz	Popis	Příklad
ls	Vypíše obsah aktuálního adresáře.	ls -l
cd	Změní aktuální pracovní adresář.	cd /home/user
mkdir	Vytvoří nový adresář.	mkdir novy_adresar
rm	Smaže soubor nebo adresář.	rm soubor.txt
cp	Zkopíruje soubor nebo adresář.	cp soubor.txt /cesta/
mv	Přesune nebo přejmenuje soubor nebo adresář.	mv soubor.txt novy_soubor.txt
cat	Vypíše obsah souboru.	cat soubor.txt
chmod	Změní oprávnění souboru.	chmod +x skript.sh
pwd	Zobrazí aktuální pracovní adresář.	pwd
rmdir	Smaže prázdný adresář.	rmdir prazdny_adresar
find	Vyhledá soubory a adresáře.	find /home -name "*.txt"
tail	Vypíše posledních n řádků souboru.	tail -n 10 soubor.txt
head	Vypíše prvních n řádků souboru.	head -n 5 soubor.txt
grep	Vyhledá řetězec v souboru.	grep "pattern"soubor.txt
wget	Stáhne soubor z internetu.	wget url
tar	Archivuje nebo rozbaluje soubory v tar archivu.	tar -czf archiv.tar.gz adresar
df	Zobrazí informace o dostupném diskovém prostoru.	df -h

2.5 Oprávnění v Linuxu

Číselná forma	Symbolická forma	Popis
0	—	Žádné oprávnění
1	-x	Proveditelné
2	-w-	Zapisovatelné
3	-wx	Zapisovatelné a proveditelné
4	r-	Čitelné
5	r-x	Čitelné a proveditelné
6	rw-	Čitelné a zapisovatelné
7	rwX	Čitelné, zapisovatelné a proveditelné

3 Cvičení 1

3.1 Vytvořte nového uživatele s názvem vašeho loginu a nastavte mu heslo: Password1*

Přepneme se do sudo

```
1 su -i
```

Vytvoříme nového uživatele

```
1 useradd {vas_login}
```

A přidáme požadované heslo Password1*

```
1 passwd {vas_login}
```

3.2 Přihlaste se jako vas-login. V adresari /home/login vytvořte pět souborů s těmito právy

- text1 rwx r-x rwx
- text2 r- - rwx -w
- text3 rwx rwx rwx
- text4 r-x - -x r- -
- text5.. rwx - -x rwx

Přihlásíme se

```
1 su {vas_login}
```

Musíme se přepnout do /home/login

```
1 cd $HOME
```

nebo můžeme použít příkaz

```
1 cd /home/{vas_login}
```

Vytvoříme pět souboru

```
1 chmod 751 text1
2 chmod 704 text2
3 chmod 777 text3
4 chmod 511 text4
5 chmod 762 text5
```

3.3 V domovském adresáři uživatele vas login vytvořte adresář pokus s těmito právy rwxr-r-

Vytvořte adresář

```
1 mkdir pokus
```

Nastavíme práva

```
1 chmod 744 pokus
```

Zkopírujeme soubor text1 do složky pokus a přepneme se do složky pokus

```
1 cp text1 pokus/
2 cd pokus/
```

Nastavíme požadované práva

```
1 chmod 644 ../pokus
```

3.4 Změna vlastníka souboru text5 na root

```
1 sudo chown root text5
```

3.5 Vypsání celé cesty aktuálního adresáře

```
1 pwd
```

3.6 Vytvoření symbolického linku k souboru text3 a ověření, že je symbolický

```
1 ln -s text3 odkaznatext3
2 ls -l odkaznatext3
```

3.7 Výpis souborů v adresáři /etc podle zadaných kritérií

```
1 cd /etc
2 ls a*
3 ls m*f
4 ls m*b*m*c*f
```

3.8 Spuštění programu mc a zjištění čísla jeho procesu

```
1 mc
2 pgrep mc
```

3.9 Zrušení procesu Midnight Commander

```
1 kill PID
2 #(kde PID je cislo procesu ziskane z predchoziho prikazu)
```

3.10 Vytvoření souboru adresy v domovském adresáři a zadání obsahu

```
1 cd ~
```

```
1 cat > adresy
2 Petr Bily 123451
3 Petra Cerna 123213
4 Jan Vlk 151122
5 Jiri Cila 122321
```

A zmáčkne **Ctrl + D**

3.11 Zobrazení obsahu souboru adresy

```
1 cat adresy
```

3.12 Výpis řádků souboru obsahujících slovo Petr

```
1 grep "Petr Bily" adresy
```

3.13 Výpis řádků souboru, které neobsahují slovo Petr

```
1 grep -v Petr adresy
```

3.14 Výpis řádků souboru obsahujících písmeno J a neobsahujících V

```
1 grep "J" adresy | grep -v "V"
```

3.15 Změna malých písmen na velká v souboru adresy a uložení do souboru adresy2

```
1 tr '[:lower:]' '[:upper:]' < adresy > adresy2
```

4 Cvičení 2

```
1 #!/bin/bash
2
3 awk '{print NR".", $0}' jmena.txt
```

```
1 #!/bin/bash
2
3 # Čekání na zadání 2 čísel
4 read -p "Zadejte první číslo: " cislo1
5 read -p "Zadejte druhé číslo: " cislo2
6
7 # Výpočet a výpis výsledku
8 vysledek=$((cislo1 * cislo2))
9 echo "Výsledek násobení: $vysledek"
```

```
1 #!/bin/bash
2
3 if grep -qE 'Jirka|Petr' jmena.txt; then
4     echo "V souboru je Jirka nebo Petr."
5 else
6     echo "V souboru není Jirka ani Petr."
7 fi
```

```
1 #!/bin/bash
2
3 sh_soubory=$(ls -1 *.sh 2>/dev/null | wc -l)
4 echo "Počet souborů s příponou .sh: $sh_soubory"
```

```
1 #!/bin/bash
2
3 pocet_radku=$(wc -l < jmena.txt)
4 echo "Počet řádků ve souboru: $pocet_radku"
```

5 Cvičení 3

```
1 #!/bin/bash
2
3 # Kontrola argumentu
4 # '$#' je proměnná co kontroluje počet argumentu
5 # Pokud počet argumentu bude cokoliv než 1
6 # Automaticky to vyhodí Chybu a vrátí 'exit 1'
7
8 if [ $# -ne 1 ]; then
9     echo "Chyba: Zadejte název souboru jako argument!"
10    exit 1
11 fi
12
13 # Kontrola existence souboru
14 # '-f' testuje pokud soubor existuje
15 # A když ne vypíše se chyba a exit 1, jako v předchozím 'if'
16
17 if [ ! -f "$1" ]; then
18     echo "Chyba: Soubor $1 neexistuje!"
19     exit 1
20 fi
21
22 # Vypsání čísla řádku následujícího po posledním
23
24 last_line=$(wc -l <"$1")
25 # 'wc' je linux příkaz na získání počtu řádku
26 # argument '-l' vypíše jen počet řádků
27 # Našem případě se to uloží do proměnné 'last_line'
28
29 next_line=$((last_line + 1))
30 # Tady se udělá jen proměnná která přičte 1 k proměnné last_line
31
32 echo "Číslo řádku po posledním: $next_line"
33 # A pak proměnnou $next_line vypíšeme
```

```
1 #!/bin/bash
2
3 if [ $# -ne 1 ]; then
4     echo "Chyba: Zadejte název souboru jako argument!"
5     exit 1
6 fi
7
8 if [ ! -f "$1" ]; then
9     echo "Chyba: Soubor $1 neexistuje!"
10    exit 1
11 fi
12 # Dokumentaci k tomuto kodu najdeme v 'cv3_1.sh'
13
14 line=$(sed -n '5p' "$1")
15 # Tento řádek uloží do proměnné 'line'
16 # Výsledek příkazu $sed -n '5p' "$1"
17 # 5p je pátý řádek
18 # $1 je definovaná proměnná která je náš soubor
19
20 echo "5. řádek souboru $1: $line"
```

```
1 #!/bin/bash
2
3 # ss
4
5 # Zrušení přípony html u všech souborů v aktuálním adresáři
6 # Tento cyklus prohledá v aktuální složce všechny soubory co končí .html
7 # A pak smažeme '.htm' z nalezených html souborů
8
9 for file in *.html; do
10     mv -- "$file" "${file%.html}"
11 done
12
13 echo "Přípona html byla úspěšně zrušena u všech souborů."
```



```

1 #!/bin/bash
2
3 # Čekání na přihlášení nového uživatele
4 until who | grep -q krystofjebuh; do
5     sleep 1
6 done
7
8 echo "*** Právě se přihlásil uživatel krystofjebuh ***"

```

```

1 #!/bin/bash
2
3 # Výpis informací o argumentech
4 echo "Jméno a příjmení: $1 + $2"
5 echo "Počet argumentů: $# "
6 echo "Název skriptu: $0"
7 echo "Domovský adresář: $HOME"
8 echo "ID procesu skriptu: $$"

```

6 Cvičení 4

6.1 Vytvoření souboru výplaty

```

1 echo "
2 Novak                Jan                12000                Jihlava
3 Novotna              Eva                12521                Brno
4 Jahoda               Petr                22010                Praha
5 Bruna                Vaclav                8000                 Praha
6 Zuda                 David                12000                Jihlava
7 " > platy.txt

```

6.2 Vypsání prvního a třetího sloupce

```

1 awk '{print $1, $3}' platy.txt

```

6.3 Výběr řádků obsahujících slovo Jihlava

```

1 awk '/Jihlava/' platy.txt

```

6.4 Vypsání prvního, druhého a čtvrtého sloupce pro řádky obsahující slovo Jihlava

```

1 awk '/Jihlava/{print $1, $2, $4}' platy.txt

```

6.5 Výběr všech řádků obsahujících v 4. sloupci písmeno P

```

1 awk '$4 ~ /P/' platy.txt

```

6.6 Výběr všech řádků obsahujících v 4. sloupci první písmeno P

```

1 awk '$4 ~ /^P/' platy.txt

```

6.7 Výběr všech pracovníků, kteří mají plat 12000

```
1 awk '$3 == 12000' platy.txt
```

6.8 Výběr všech pracovníků, kteří mají plat menší než 12000

```
1 awk '$3 < 12000' platy.txt
```

6.9 Výběr všech pracovníků, kteří mají plat menší než 20000 a větší než 10000

```
1 awk '$3 < 20000 && $3 > 10000' platy.txt
```

6.10 Výběr všech pracovníků, kteří jsou z Brna nebo z Jihlavy

```
1 awk '$4 == "Brno" || $4 == "Jihlava"' platy.txt
```

6.11 Vypsání čísla řádku, na kterém je ve sloupci jméno Petr

```
1 awk '$2 == "Petr" {print NR}' platy.txt
```

6.12 Zobrazení 2. a 4. řádku

```
1 awk 'NR==2 || NR==4' platy.txt
```

6.13 Vypočtení průměrného platu

```
1 awk '{sum+=$3} END {print "Průměrný plat: ", sum/NR}' platy.txt
```

6.14 Vypočtení průměrného platu pro všechny pracovníky z Jihlavy

```
1 awk '$4 == "Jihlava" {sum+=$3; count++} END {print "Průměrný plat pro pracovníky z Jihlavy: ", sum/count}' platy.txt
```

6.15 Vypsání maximálního platu

```
1 awk 'BEGIN {max=0} $3 > max {max=$3} END {print "Maximální plat: ", max}' platy.txt
```

6.16 Vytvoření programu, který každé 2 minuty zkopíruje výpis obsahu vybraného adresáře do souboru

```
1 #!/bin/bash
2
3 # Adresář, jehož obsah chceme zkopírovat
4 source_dir="/cesta/k/vybranemu/adresari"
5
6 # Soubor, do kterého chceme zkopírovat výpis obsahu adresáře
7 target_file="/cesta/k/cilovemu/souboru.txt"
8
9 # Zkopírování obsahu adresáře do cílového souboru
10 ls -l "$source_dir" > "$target_file"
11
12 # Zpráva o dokončení kopírování
13 echo "Výpis obsahu adresáře byl zkopírován do souboru: $target_file"
```

7 Cvičení 5

7.1 V systému Linux vytvořte skript s názvem adresare, který vypíše počet adresářů v aktuálním adresáři do souboru

```
1 #!/bin/bash
2
3 # Počet adresářů v aktuálním adresáři
4 num_directories=$(ls -l | grep ^d | wc -l)
5
6 # Zápis do souboru
7 echo "Počet adresářů: $num_directories" > adresare.txt
```

Listing 5: Script adresare.sh

7.2 V Linuxu vytvořte adresář cv5 a v něm vytvořte 3 soubory první, druhý, třetí. Dále v něm vytvořte skript s názvem prejmenuj, který dá všem souborům příponu html

```
1 mkdir cv5 #Vytvoří složku
2 touch první druhý třetí #vytvoří soubory
```

```
1 #!/bin/bash
2
3 # Přejmenuj soubory v adresáři cv5
4 for file in cv5/*; do
5     mv "$file" "${file}.html"
6 done
```

Listing 6: Script prejmenuj.sh

7.3 Přihlaste se do Linuxu pod účtem root. Vytvořte skript, který vytvoří nového uživatele a nastaví mu heslo

```
1 su -i
```

Listing 7: Přihlásíme se jako root

```
1 whoami
```

Listing 8: Můžeme zkontrolovat za koho jsme přihlášení

```

1 #!/bin/bash
2
3 # Vytvoření nového uživatele
4 username="krystofjebuh"
5 useradd $username
6
7 # Nastavení hesla pro nového uživatele
8 echo "Password!*" passwd --stdin $username

```

Listing 9: Script novy uzivatel.sh

- 7.4** V Linuxu vytvořte skript s názvem balicky, který po spuštění vypíše všechny nainstalované balíčky, které začínají na php. Pokud tam žádný balíček není, nainstalujte ho

```

1 #!/bin/bash
2
3 # Zjištění nainstalovaných balíčků začínajících na "php"
4 php_packages=$(rpm -qa | grep '^php')
5
6 # Pokud žádný balíček nezačíná na "php", nainstalujte ho
7 if [ -z "$php_packages" ]; then
8     yum install php
9 else
10     echo "Nainstalované balíčky začínající na 'php':"
11     echo "$php_packages"
12 fi

```

Listing 10: Script balicky.sh

- 7.5** V Linuxu vytvořte skript s názvem existuje, který zjistí, zda je nějaký soubor existuje nebo ne. Výsledkem bude text: „existuje“ nebo „neexistuje“.

```

1 #!/bin/bash
2
3 # Zjištění existence souboru
4 if [ -e "$1" ]; then
5     echo "existuje"
6 else
7     echo "neexistuje"
8 fi

```

Listing 11: Script existuje.sh

8 Cvičení 6

- 8.1** Napište příkaz, který vypíše vaše aktuální UID.

```

1 id -u

```

- 8.2** Napište příkaz, který vypíše seznam skupin, v nichž se nacházíte.

```

1 groups

```

- 8.3 Vytvořte soubor jmena.txt s 21 jmény. Z výstupu jmena.txt vypište jen sudé řádky a na začátky lichých řádek souboru jmena.txt vložte znak „l”, na začátky sudých řádek vložte znak „s”.

```
1 #!/bin/bash
2
3 # Vytvoření pomocného souboru pro liché a sudé řádky
4 awk 'NR%2==0' jmena.txt > suda.jmena.txt
5 awk 'NR%2!=0' jmena.txt > licha.jmena.txt
6
7 # Přidání značek "s" pro sudé řádky
8 sed 's/^/s /' suda.jmena.txt > temp.txt
9
10 # Přidání značek "l" pro liché řádky
11 sed 's/^/l /' licha.jmena.txt >> temp.txt
12
13 # Zobrazení výsledného souboru
14 cat temp.txt
15
16 # Smazání pomocných souborů
17 rm suda.jmena.txt licha.jmena.txt temp.txt
```

- 8.4 b za každý řádek přidá volný řádek

```
1 sed G jmena.txt
```

- 8.5 Na začátky desáté až dvacáté řádky souboru jmena.txt přidejte znak #.

```
1 sed '10,20 s/^/#/' jmena.txt
```

- 8.6 Pomocí příkazu najděte soubor jmena.txt v adresáři /home i v podadresářích.

```
1 find /home -name "jmena.txt"
```

- 8.7 Pomocí příkazu najděte v adresáři /home i v podadresářích všechny soubory začínající j s příponou .txt

```
1 find /home -type f -name "j*.txt"
```

- 8.8 Pomocí příkazu najděte v adresáři /home i v podadresářích vše, co bylo změněno před 7 dny.

```
1 find /home -type f -mtime +7
```

- 8.9 Zjištění rodičovského procesu

- 8.9.1 Spusťte zobrazení procesů

```
1 ps -l
```

8.9.2 Jaký je rodičovský proces procesu ps?

Měl by to být uživatelský shell.

8.10 Zobrazte celou stromovou strukturu všech procesů operačního systému.

```
1 pstree
```

8.11 Procesy na pozadí, popředí gedit

8.11.1 Spustíme Textový editor GEdit

```
1 gedit &
```

8.11.2 Dáme process do pozadí

```
1 bg %1
```

8.11.3 Přeneseme proces do popředí

```
1 fg %1
```

8.12 Sledujte procesy v reálném čase top

```
1 top
```

8.13 Pojmenovaná roura

8.13.1 Vytvoříme routu

```
1 mkfifo roura
```

8.13.2 Načtení datumu do routy

```
1 date > roura &
```

8.13.3 Zobrazení dat

```
1 cat roura
```

9 Cvičení 7

9.1 Pomocí příkazu vytvořte soubor se jménem „datum“. Do tohoto souboru vložte pomocí příkazu aktuální datum.

```
1 date > datum
```

9.2 Vypište deset největších souborů a pak deset nejmenších souborů v adresáři /etc

9.2.1 Největší soubory

```
1 ls -lS /etc head -n 11
```

9.2.2 Nejmenší soubory

```
1 ls -lSr /etc head -n 11
```

9.3 V adresáři /usr/bin najděte soubor, který byl modifikován naposledy.

```
1 ls -lt /usr/bin head -n 2
```

9.4 Ze souboru studenti.csv vypište jen první sloupec, tj. sloupec s příjmeními (použijte find, případně cut).

```
1 cut -d ',' -f 1 studenti.csv
```

9.5 Udělejte kopii souboru studenti.csv, nahraďte pomocí příkazu středníky pomlčkou a pomocí diff se podívejte, které řádky byly změněny.

```
1 # Uděláme kopii
2 cp studenti.csv studenti_copy.csv
3
4 # Nahradíme ;, -
5 sed -i 's/;/-/g' studenti_copy.csv
6
7 # Ukážeme rozdíl
8 diff studenti.csv studenti_copy.csv
```

9.6 Rozdělte příkazem studenti.csv soubor na kusy po pěti řádcích (split).

```
1 split -l 5 studenti.csv studenti_chunk_
```

9.7 Napište příkaz, který vypíše počet všech (pro vás viditelných) adresářů v podstromu /etc

```
1 find /etc -type d wc -l
```

9.8 V podstromu adresáře /tmp najděte všechny soubory, které jsou větší než sto kilobyte (find).

```
1 find /tmp -type f -size +100k
```

- 9.9 Do své složky (home) zkopírujte soubor `/etc/passwd`, který přejmenujete například na `passwd2`. V souboru `passwd2` odstraňte nějaké řádky. Vytvořte skript, který projde vytvořené uživatele (`/etc/passwd`) a zjistí, rozdíl se souborem `passwd2` (`diff`). Rozdíl se uloží do souboru `rozdil.txt`. Tedy zjistí, kteří uživatelé jsou v souboru `/etc/passwd` navíc.

```
1 # Zkopírování souboru /etc/passwd
2 cp /etc/passwd ~/passwd2
3
4 # Odstranímě nějaké řádky (vybral jsem 2 a 5)
5 sed -i '2d;5d' passwd2
6
7 # Spustíme script
```

```
1 #!/bin/bash
2
3 diff /etc/passwd ~/passwd2 > rozdil.txt
```

Listing 12: Script který spustíme

- 9.10 Pomocí příkazu vypište verzi OS linux (číslo jádra a další informace).

```
1 uname -a
```