

# Aktivity pro výuku základů práce v Bashi



Bc. Jakub Novotný

Texty na následujících stranách byly převzaty z diplomové práce Bc. Jakuba Novotného. Obrázek na úvodní straně byl vytvořen za použití ChatGPT. Obě úpravy provedl Václav Šimandl.

## 8 Sada problémových úloh

### 8.1 Úloha 1

Téma: Přípravné úlohy

Název úkolu

Přípravka

Obsah

V této úloze budeš muset zjistit, jak se pracuje se soubory a složkami. Tedy například vytvořit, zkopírovat či smazat složku, zobrazit si obsah adresáře a textového souboru. Poté také vyzkoušíš změnit heslo uživatele a vypsát si spuštěné procesy.

Zadání

Pomocí terminálu si vytvoř ve svém domovském adresáři novou složku, kterou pojmenuješ svým příjmením bez diakritiky. Následně do této složky vytvoř složku *skripty* a do ní vytvoř soubor *den.txt*, do kterého zapiš aktuální datum a čas. Složku *skripty* dvakrát zkopíruj a následně přejmenuj na *pracovni* a *pracovni2*. Vypiš, jaké složky se aktuálně nachází ve tvém adresáři. Jakmile budeš mít zkopírováno, složku *pracovni2* smaž. Změň si u svého uživatelského účtu heslo a podívej se, jaké jsou v systému spuštěné procesy.

Očekávané cíle

- Žák umí pracovat se soubory (vytvořit, smazat aj.)
- Žák umí pracovat se složkami (vytvořit, smazat aj.)
- Žák umí přecházet mezi adresáři
- Žák umí vypsát obsah adresáře
- Žák umí zobrazit obsah textového souboru
- Žák umí změnit heslo uživatele
- Žák umí vypsát spuštěné procesy

Předpoklady

- Žák umí spustit terminál

## Metoda výuky

- Samostatná práce žáků

## Očekávaná chybná řešení

Žák nebude schopen smazat složku, která obsahuje nějaká data. A to z důvodu, že zpočátku nezadá argument pro rekurzivní mazání dat bez dotazu o neprázdném adresáři.

## 8.2 Úloha 2

Téma: Úvodní problematika zápisu kódu

Název úkolu

Uvítání do programu

Obsah

V této úloze budeš muset zjistit, jak zapsat hlavičku skriptu a vyzkoušíš si textový výpis.

Zadání

Vytvoř si nový skript v Bashi a ulož ho na disk. Jakmile budeš mít vytvořený soubor, zapiš do něj skript tak, aby po jeho spuštění vypsalo následující text:

*„Toto je první řádek textu,*

*zde je druhý řádek výpisu, ale vše bylo původně zapsáno pouze v jednom řádku.“*

Příkaz pro výpis zajistí vypsání obou řádků najednou. Tedy příkaz se v kódu objeví pouze jednou, nikoliv dvakrát. Stejně tak zapišeš text do jednoho řádku bez použití enteru. Pozor také na mezeru na začátku druhého řádku.

Očekávané cíle

- Žák umí zapsat hlavičku skriptu v Bashi
- Žák umí použít příkaz echo
- Žák umí spustit skript
- Žák umí zapsat argument „*new line*“ příkazu echo

Předpoklady

- Žák ovládá základní práci v Linuxu
- Žák umí spustit terminál

## Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák po zápisu prvního řádku nebude vědět, jak zapsat druhý řádek bez použití opětovného použití příkazu echo a bude muset hledat, jak zapsat argument a který použít)

## Očekávaná chybná řešení

U tohoto příkladu se dá očekávat, že žáci nebudou schopni pochopit, že je možné zapsat text mezi písmena, aniž by se zobrazil. Tedy argument `\n` většina žáků napsala až po delší době zkoušení, protože jim nešlo do hlavy, jak to zapsat, aby tam nebylo zobrazené písmeno `n`.

Za další chybné řešení lze brát to, že žák použije mezerník několikrát za sebou, aby se s textem dostal na další řádek. Kontrola toho řešení pak spočívá v jednoduchém roztažení okna terminálu.

## Poznámky

```
# !/bin/bash
echo "Přípravka"
echo "Toto je první řádek, \ntoto je druhý řádek"
```

## 8.3 Úloha 3

Téma: Zápis kódu s použitím argumentů a spojením výrazů

Název úkolu

Kalendář

Obsah

V této úloze budeš muset zajistit, aby se vypsal aktuální čas a datum. Dej si však pozor na to, jak se bude datum a čas zobrazovat.

Zadání

Vytvoř skript v Bashi, který vypíše čas v českém formátu, tedy hodiny:minuty, například 12:58. Skript dále vypíše datum v tzv. krátkém tvaru, tedy například 15.03.2018.

Očekávané cíle

- Žák se naučí používat argumenty pro výpis data a času
- Žák chápe rozdílnost v zápisu velkých a malých písmen při kódování

Předpoklady

- Žák umí vytvořit skript v Bashi
- Žák umí správně použít argumenty v příkazu

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák se dostane do problému ve chvíli, kdy zvolí špatnou variantu zápisu argumentu příkazu a dle toho se mu může stát, že výpis nedopadne dle očekávání)

Očekávaná chybná řešení

Zde lze od žáků očekávat, několik možných chyb. Zejména z důvodu velkého množství zápisu proměnných po hodiny, minuty, dny, měsíce a roky. Změna velkého písmena na malé může způsobit poměrně velkou změnu, kdy se například z měsíce stanou minuty či například rok, který by byl zapsán ve zkráceném tvaru, bude následně zapsat ve tvaru dlouhém.

## Poznámky

```
# !/bin/bash  
echo "Kalendář"  
date +%H:%M  
date +%d.%m.%Y
```



## 8.4 Úloha 4

Téma: Proměnné systému

Název úkolu

Není rovná se jako rovná se

Obsah

V této úloze budeš muset zjistit, jak použít ve skriptu proměnnou, která vypíše předem zadaný obsah.

Zadání

Ve skriptu, který jsi právě obdržel, je chyba. Chceme, aby nám skript při spuštění napsal: „*Jaká barva vlasů je u holek nejhezčí? Možnosti jsou čtyři. Blond. Hnědá. Černá. Zrzavá.*“ Namísto toho skript píše chybu, že příkaz nebylo možné vyhledat a nezobrazí nám odpověď.

Očekávané cíle

- Žák umí použít pevně zadanou proměnnou
- Žák chápe důležitost mezer v přiřazování hodnot proměnným

Předpoklady

- Žák umí kriticky myslet při hledání chyby
- Žák umí vyhledat význam vypsane chyby

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žáci dostanou jeden skript, ve kterém budou proměnné špatně zapsané, protože budou obsahovat mezery a budou muset zjistit, z jakého důvodu skript nezobrazí obsah proměnné, ale namísto toho vypisuje chybu, že příkaz nebylo možné vyhledat. Dále zde v kódu bude i syntaktická chyba.)

Pomůcky

- Skript s možnostmi špatného zápisu

### Očekávaná chybná řešení

U tohoto příkladu lze očekávat, že žáci si budou myslet, že postačí opravit pouze jednu chybu a budou mít vyhráno. Tedy, že v předchozím příkladu zjistili, že zobrazení proměnné je možné pouze s dolarem a tedy žádný dolar v zadání není. Jenže po zapsání dolaru skript stále nefunguje. Žáci tedy začnou hledat další chybu a vyzkouší mezery u znaku rovná se. Jakmile opraví chyby i zde, skript stále nezobrazí obsah proměnných. Ve chvíli, kdy žák opraví všechny chyby, skript začne teprve fungovat.

Pokud by však byl tento úkol pro některé žáky moc složitý, může učitel zvolit jednodušší variantu, kde si žák osahá, jak se s proměnnou pracuje a následně se může vrátit ke složitějšímu příkladu.

### Poznámka

```
#!/bin/bash
echo "Není rovná se jako rovnáse"
Barva1=Blond
Barva2=Hnědá
Barva3=Černá
Barva4=Zrzavá

echo $barva1
echo $barva2
echo $barva3
echo $barva4
```

### 8.4.1 Zjednodušená varianta

U této úlohy nabízím možnost zvolit jednodušší variantu. Je zde hned několik problémů, které žák musí vyřešit a pro některé z nich to ve výuce bylo velmi komplikované, proto se zvažovala možnost změny zadání. Zadání nakonec zůstalo stejné, ale je možné zadat žákům příklad buďto tento složitější, viz kód výše, či následující kód po tímto textem. Záleží na učiteli, jakou variantu zvolí. Návrhem mohou být následující situace. První příklad dostanou všichni žáci jednodušší a následně dostanou ten složitější. Další variantou může být, že všichni dostanou ihned složitější a po nějaké době neúspěchu učitel rozhodne, komu vyměnit složitější úkol za jednodušší a po jeho vyřešení a vrátí k řešení složitějšího. Eventuálně může při zadávání příkladu učitel rovnou rozhodnout, že někteří žáci dostanou jednodušší variantu, někteří zase tu složitější.

```
#!/bin/bash
echo "Názvy procesorů"
cpu1=Intel
cpu2=Intel

echo $cpu1
echo $cpu2
```

## 8.5 Úloha 5

Téma: aritmetické operace

Název úkolu

Podivuhodná kalkulačka

Obsah

V této úloze budeš muset zjistit, proč kalkulačka počítá nějak divně. Nefunguje tam totiž ani jedna operace tak, jak by správně měla.

Zadání

Ve skriptu *kalkulacka.sh* zjisti, proč ani jedna operace neudělá dle komentáře to, co by dělat měla. Nezapomeň na začátku upravit syntaxi zápisu pro výpočet aritmetických operací. Zjisti, kde je v prvních úlohách problém, že ani jedna operace nefunguje a jakmile to zjistíš, zaměř se na předposlední příklad, na dělení. Vyjde výsledek při tomto zápisu přesně tak, jak očekáváš?

Očekávané cíle

- Žák správně používá ve skriptech aritmetické operace
- Žák rozezná zápis obyčejného textu a příkazu

Předpoklady

- Žák zná základní aritmetické počty
- Žák umí používat proměnné

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žáci obdrží skript, ve kterém budou mít pět příkladů, ale ani jeden nebude fungovat, tedy budou muset zjistit, jaký zápis pro výpočet je správný, opravit ho a následně i objevit, jak zapsat znaménka, aby vyšel výsledek dle zadání)

Pomůcky

- Skript s příklady

### Očekávaná chybná řešení

Opět se setkáváme s problémem, kdy žáci musí naleznou chybu v zápisu proměnných při aritmetickém počítání. Očekává se, že žáci v tomto příkladu budou neustále chybně zkoušet zapisovat znak proměnné, dokud se jim nepodaří odstranit alespoň jednu chybu a zobrazit správný výsledek. Další očekávanou chybou je, že žáci použijí aritmetická znaménka ve špatné formě.

### Poznámka

```
# !/bin/bash
echo "Podivuhodná kalkulačka"

cislo1=10
cislo2=2

# sečti cislo1 a cislo2 (výsledek: 12)
echo "$cislo1 plus $cislo2 = $(( $cislo1+$cislo2 ))"

# Od proměnné cislo1 odečti cislo2 (výsledek: 8)
echo "$cislo1 minus $cislo2 = $(( $cislo1-$cislo2 ))"

# Proměnnou cislo1 vynásob s proměnnou cislo2 (výsledek 20)
echo "$cislo1 krát $cislo2 = $(( $cislo1*$cislo2 ))"

# Proměnnou cislo1 vyděl proměnnou cislo2 (výsledek: 5)
echo "$cislo1 děleno $cislo2 = $(( $cislo1/$cislo2 ))"

# Výsledek je?
echo "$vysledek = $(( $cislo1+$cislo2/$cislo2 ))"
```

## 8.6 Úloha 6

Téma: Podmínky

Název úkolu

Je dopoledne nebo odpoledne?

Obsah

V této úloze budeš muset naprogramovat skript tak, aby po spuštění vypsal, v jaké části dne se zrovna nacházíme.

Zadání

Vytvoř si nový skript *castdne.sh* a vymysli ho tak, aby po spuštění vypsal, zda je před polednem či po poledni.

Očekávané cíle

- Žák umí použít podmínku
- Žák chápe důležitost mezer v zápisu podmínky

Předpoklady

- Žák rozumí pojmu proměnná
- Žák umí proměnnou použít
- Žák rozumí pojmu podmínka
- Žák rozumí pojmu podmíněný příkaz

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák nezná, jak zapsat podmínku a její možnosti)

Očekávaná chybná řešení

Očekává se, že žák udělá chybu ve znaménku větší / menší.

#### Poznámka

- Žák musí zjistit, jak zapsat do lokální proměnné systémový čas
- Žák musí zjistit, jak porovnat podmínkou aktuální čas a dle toho vypsát, o jakou část dne se jedná (vytvořit podmínku, zda je)
- Na základě tohoto prvního příkladu pak žáci budou schopni vytvořit i složitější podmínky pro více částí dne.

```
#!/bin/bash
echo "Je dopoledne nebo odpoledne?"

H=$(date +%H)

if [ $h -lt 12 ]; then
    echo "Je před polednem"
else
    echo "Je po poledni"
fi
```

## 8.7 Úloha 7

Téma: Podmínky

Název úkolu

Je dopoledne, odpoledne nebo poledne?

Obsah

V této úloze budeš muset naprogramovat skript tak, aby po spuštění vypsal, jestli se nacházíme před polednem, po poledni nebo je právě poledne.

Zadání

Vytvoř si nový skript *castdne2.sh* a vymysli ho tak, aby po spuštění vypsal, zda je před polednem, po poledni či je právě poledne. Nezapomeň, že poledne je jen určitou chvíli.

Očekávané cíle

- Žák umí použít podmínku
- Žák chápe důležitost mezer v zápisu podmínky
- Žák umí logicky posoudit, jak podmínku vytvořit, aby určil časové rozmezí

Předpoklady

- Žák rozumí pojmu proměnná
- Žák umí použít proměnnou
- Žák rozumí pojmu podmínka

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jaké argumentu příkazu je možné nastavit, aby podmínka byla rozvětvená, eventuálně byla složená z více částí)



### Očekávaná chybná řešení

Žák musí vymyslet, jak porovnat nejen hodiny, ale i minuty. Před polednem a po poledni může zůstat stále pouze u porovnávání hodin, nicméně u pravého poledne je nutné porovnat i minuty. A to je problém, se kterým se bude žák muset popasovat, protože musí zjistit, jak hodiny zapsat, aby je bylo možné porovnat.

### Poznámka

- Žák musí zjistit, jak porovnat podmínkou aktuální čas a dle toho vypsát, o jakou část dne se jedná (vytvořit podmínku, zda je před polednem, po poledni nebo je právě poledne)
- Žák následně konfliktem zjistí, že není možné porovnávat pouze hodiny, ale je nutné porovnat i minuty, protože by bylo poledne déle než pouze ve 12:00

```
#!/bin/bash
echo "Je dopoledne, odpoledne nebo poledne?"

H=$(date +%H)
M=$(date +%M)

echo $h$m

if [ $h -lt 12 ]; then
    echo "Je před polednem"

    elif [ $h$m -eq 1200 ]; then
echo "Je poledne"

else
    echo "Je po poledni"
fi
```

## 8.8 Úloha 8

Téma: Podmínky

Název úkolu

Jaká je část dne?

Obsah

V této úloze budeš muset naprogramovat skript tak, aby po spuštění vypsal, v jaké části dne se zrovna nacházíme.

Zadání

Vytvoř si nový skript *castdne.sh* a vymysli ho tak, aby po spuštění vypsal, zda je noc, dopoledne, poledne, odpoledne, večer či půlnoc. Nezapomeň, že poledne a půlnoc je jen určitou chvíli. Dopoledne, odpoledne a večer si rozvrhni dle znalostí, v jakých časech se tyto části dne nachází.

Očekávané cíle

- Žák umí použít podmínku
- Žák chápe důležitost mezer v zápisu podmínky
- Žák umí logicky posoudit, jak podmínku vytvořit, aby určil časové rozmezí

Předpoklady

- Žák rozumí pojmu proměnná
- Žák umí použít proměnnou
- Žák rozumí pojmu podmínka

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jaké argumentu příkazu je možné nastavit, aby podmínka byla rozvětvená, eventuálně byla složená z více částí)

### Očekávaná chybná řešení

V tomto příkladu se musí žák zaměřit na kombinaci podmínek. Tedy když žák zadal operand „or“ namísto „and“, výsledek se mu nezobrazil správně, i když měl celý skript správně.

### Poznámka

- Dle složitosti příkladu předpokládám, že i velmi talentovaní žáci tento komplikovaný a vrstvený problém mohou řešit i několik hodin.

```
#!/bin/bash
echo "Jaká je část dne?"

H=$(date +%H)
M=$(date +%M)

echo "$h:$m"

if [ $h$m -eq 0000 ]; then
    echo "Je půlnoc"
elif [ $h$m -eq 1200 ]; then
    echo "Je poledne"
elif [ $h$m -gt 0000 -a $h$m -lt 0800 ]; then
    echo "Je noc"
elif [ $h$m -gt 0800 -a $h$m -lt 1200 ]; then
    echo "Je dopoledne"
elif [ $h$m -gt 1200 -a $h$m -lt 2000 ]; then
    echo "Je odpoledne"
elif [ $h$m -gt 2000 -a $h$m -lt 2359 ]; then
    echo "Je noc"
else
    echo CHYBA
fi
```

## 8.9 Úloha 9

Téma: Výběr dat

Název úkolu

Výběr skladeb oblíbeného interpreta

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript vypsal do terminálu názvy jeho skladeb.

Zadání

Vytvoř si nový skript *interpret.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a následně vypsal veškeré jeho skladby ze seznamu do okna terminálu. Zkontroluj, zda jsi vybral opravdu všechny.

Očekávané cíle

- Žák umí použít příkaz `grep`
- Žák umí vybrat správný argument příkazu `grep`
- Žák chápe důležitost velkých a malých písmen v zápisu výběru dat
- Žák umí použít přesměrování výstupu
- Žák umí zadat vstup z klávesnice do proměnné

Předpoklady

- Žák rozumí pojmu `echo`
- Žák umí použít `echo`

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jak poskládat příkaz pro očekávané zobrazení)

### Očekávaná chybná řešení

Chybové řešení v tomto příkladu se očekává ve chvíli, kdy žák musí správně poskládat příkaz `grep`. Pokud nepoužije uvozovky při vyhledávání interpreta, nedostane přesné výsledky. Jedná se zejména o dvou a víceslovná jména. U jednoslovných jmen interpretů by rozdíl poznat nebyl. Tato chyba přetrvává u všech následujících skriptů s podobnou tematikou.

### Poznámka

- Příprava pro následující příklad s počítáním.
- Větší problém vznikne ve chvíli, kdy žák porovná zdrojový soubor s výsledkem, protože některé řádky nezačínají velkým písmenem, proto bude muset najít, z jakého důvodu a doplnit argument pro ignorování velikosti písmen
- Dávat proměnnou do uvozovek je poměrně nezvyklé a v této sbírce ojedinělé. Důvodem k tomu je ochrana při vyhledávání víceslovného názvu. Pokud by se v seznamu vyhledávaly pouze jednoslovné názvy, pak uvozovky být nemusí.

```
#!/bin/bash
echo "Výběr skladeb oblíbeného interpreta"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" interpreti.txt
```

## 8.10 Úloha 10

### Název úkolu

Výběr skladeb oblíbeného interpreta s pojistkou

### Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript vypsal názvy jeho skladeb.

### Zadání

Vytvoř si nový skript *interpretp.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a následně vypsal veškeré jeho skladby ze seznamu. Zkontroluj, zda jsi vybral opravdu všechny.

### Očekávaná chybná řešení

U tohoto příkladu žáci mohou chybovat v tom, že použijí špatný argument a skript nezobrazí veškeré skladby od interpreta, protože některé řádky nemusí začínat velkým písmenem.

### Poznámka

- Žák umí použít argument `-i`
- Rozšiřující úloha: některé řádky nemusí začínat velkým písmenem, proto bude muset žák zjistit, z jakého důvodu se nemusely vybrat všechny skladby a doplnit argument pro ignorování velikosti písmen

```
#!/bin/bash
echo "Výběr skladeb oblíbeného interpreta s pojistkou"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt
```

## 8.11 Úloha 11

Téma: výběr dat do souboru

Název úkolu

Skladby oblíbeného interpreta do souboru

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript zapsal veškeré skladby do nově vytvořeného textového souboru.

Zadání

Vytvoř si nový skript *interpretzapis.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a následně zapsal všechny skladby do nového textového souboru *skladby.txt*.

Očekávané cíle

- Žák vytvoří nový soubor
- Žák zapíše vybraná data do textového souboru

Předpoklady

- Žák rozumí příkazu `grep`
- Žák umí použít příkaz `grep`

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jak zapsat data do textového souboru)

Očekávaná chybná řešení

Žák bude mít problém se zápisem do nového souboru. Nejprve si soubor ručně vytvoří a až následně do něj bude zapisovat, protože neví, že je možné soubor vytvořit přímo ve skriptu.

### Poznámka

```
#!/bin/bash
echo "Skladby oblíbeného interpreta do souboru"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt > skladby.txt
```



## 8.12 Úloha 12

Téma: Výběr dat a počet výskytů

Název úkolu

Počet skladeb oblíbeného interpreta

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript zapsal skladby do textového souboru a zároveň spočítal počet jeho skladeb.

Zadání

Vytvoř si nový skript *interpretpocet.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a následně zapsal počet skladeb ze seznamu do textového souboru ze seznamu.

Očekávané cíle

- Žák umí použít příkaz *wc*
- Žák umí použít rouru v Linuxu

Předpoklady

- Žák rozumí příkazu *grep*
- Žák umí použít příkaz *grep*

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jak automaticky spočítat počet výskytů vyhledávaného řetězce)

Očekávaná chybná řešení

Žák se dostane do problému ve chvíli, kdy bude zadávat příkaz pro spočítání řádků. Musí použít *rouru*, jinak se mu počet řádků nezobrazí v jednoduchém tvaru díky příkazu *wc*. Nelze ale považovat za chybné řešení, kdy žáci nepoužijí příkaz *wc*, ale pouze argument *-c* k příkazu *grep*.

### Poznámka

- Žák může počet řádků spočítat i pomocí argumentu `-c` (count) příkazu `grep`

```
#!/bin/bash
echo "Počet skladeb oblíbeného interpreta"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt | wc > skladby.txt
```

## 8.13 Úloha 13

Téma: Výběr dat a počet výsledků

Název úkolu

Počet skladeb oblíbeného interpreta

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript vypsal počet skladeb do okna terminálu.

Zadání

Vytvoř si nový skript *interpretpocetT.sh*, který spočítá počet skladeb. Vymysli ho tak, aby se po spuštění zeptal na jméno interpreta, uložil výsledek do textového souboru a následně obsah souboru vypsal do okna terminálu. Pro spočítání ale nepoužívej k vyhledávacímu příkazu žádný argument, ale kombinaci několika příkazů.

Očekávané cíle

- Žák umí použít příkaz `wc`
- Žák umí použít rouru v Linuxu

Předpoklady

- Žák rozumí příkazu `grep`
- Žák umí použít příkaz `grep`
- Žák umí použít příkaz `cat`

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jak automaticky spočítat počet výskytů vyhledávaného řetězce a následně spojit s příkazem pro zobrazení obsahu souboru)

### Očekávaná chybná řešení

Zde se žáci dostanou do problému ve chvíli, kdy budou muset zobrazit obsah souboru, do které právě uložili získaná data. Sice výsledky uloží, ale již je nezobrazí. Zejména z toho důvodu, že je nenapadne možná kombinace příkazů použitím středníku.

### Poznámka

```
#!/bin/bash
echo "Počet skladeb oblíbeného interpreta"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt | wc > skladby.txt;
cat skladby.txt
```

## 8.14 Úloha 14

Téma: Výběr dat

Název úkolu

Výběr skladeb druhého oblíbeného interpreta

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno druhého oblíbeného interpreta a skript připsal na konec textového souboru názvy skladeb.

Zadání

Vytvoř si nový skript *druhyinterpret.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a následně připsal veškeré jeho skladby ze seznamu na konec textového souboru *skladby.txt*. Soubor by měl po zápisu obsahovat skladby obou interpretů.

Očekávané cíle

- Žák umí použít příkaz `grep`
- Žák umí vybrat správný argument příkazu `grep`
- Žák chápe důležitost velkých a malých písmen v zápisu výběru dat
- Žák dokáže použít přesměrování výstupu

Předpoklady

- Žák rozumí pojmu `echo`
- Žák umí použít `echo`

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák neví, jak poskládat příkaz pro očekávaný zápis)

Očekávaná chybná řešení

U tohoto příkladu žák s největší pravděpodobností přepíše původní obsah. Stane se tak zejména z důvodu, že použije pouze jeden znak špičaté závorky, tedy pro zápis do souboru.

### Poznámka

```
#!/bin/bash
echo "Výběr skladeb druhého oblíbeného interpreta"
echo "Zadej jméno druhého interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt >> skladby.txt
```

## 8.15 Úloha 15

Téma: výběr dat a přesměrování výstupu

Název úkolu

Skladby oblíbeného interpreta do textového dokumentu i do terminálu

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript zapsal všechny skladby do nového textového dokumentu a zároveň je vypsál do terminálu.

Zadání

Vytvoř si nový skript *interpretTerminal.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a vytvořil textový dokument, do kterého zapíše názvy všech jeho skladeb a zároveň se nalezené skladby zobrazí na obrazovce terminálu.

Očekávané cíle

- Žák umí použít & pro kombinaci příkazů

Předpoklady

- Žák rozumí příkazu `grep`
- Žák umí použít příkaz `grep`

Metoda výuky

- Samostatná práce žáků

#### Poznámka

- Rozšíření předchozích úloh
- Žáci si vyzkouší, jak je možné zobrazit výpis do terminálu a zároveň ho zapsat do textového souboru

```
#!/bin/bash
echo "Skladby oblíbeného interpreta do textového dokumentu
i do terminálu"
echo "Zadej jméno interpreta"
read "interpret"
grep "$interpret" -I interpreti.txt > skladby.txt;
cat skladby.txt
```



## 8.16 Úloha 16

Téma: Výběr dat a přesměrování výstupu

Název úkolu

Skladby oblíbeného interpreta do textového dokumentu i do terminálu

Obsah

V této úloze budeš muset naprogramovat skript tak, aby bylo možné zadat jméno oblíbeného interpreta a skript zapsal všechny skladby do nového textového dokumentu a zároveň je vypsals do terminálu.

Zadání

Vytvoř si nový skript *interpretTerminal2.sh* a vymysli ho tak, aby se po spuštění zeptal na jméno interpreta a vytvořil textový dokument, do kterého запиše názvy všech jeho skladeb a zároveň se nalezené skladby zobrazí na obrazovce terminálu. Nepoužívej však kombinaci příkazů jako v předešlé úloze.

Očekávané cíle

- Žák umí použít příkaz tee

Předpoklady

- Žák rozumí příkazu grep
- Žák umí použít příkaz grep
- Žák umí použít rouru

Metoda výuky

- Samostatná práce žáků

Očekávaná chybná řešení

U tohoto příkladu se u žáků předpokládá, že se pokusí zadat název souboru, do které se mají data zapsat a následně název souboru, ze kterého se mají data přečíst. Tak tomu bylo v předchozích příkladech. Nejprve bylo nutné data načíst, uložit a následně zobrazit. Příkaz *tee* udělá uložení a zobrazení najednou.

#### Poznámka

- Rozšíření předchozích úloh pro zopakování
- Obdobná úloha pro použití roury, pokud by žáci při počítání skladeb obešli příkaz `wc` argumentem `-c` příkazu `grep`

```
#!/bin/bash
echo "Skladby oblíbeného interpreta do textového doku-
mentu i do terminálu"
echo "Zadej jméno interpreta"
read "interpret"
grep -i "$interpret" interpreti.txt | tee skladby.txt
```

## 8.17 Úloha 17

Téma: Vyhledání a filtrování souborů

Název úkolu

Kdo hledá, najde

Obsah

V této úloze budeš muset naprogramovat skript, díky kterému ve směsi textových souborů najdeš pouze ty smysluplné.

Zadání

Vytvoř si nový skript *hledejmysl.sh* a vymysli ho tak, aby ze všech souborů ze složky *smes* vybral pouze ty soubory, které obsahují v názvu souboru text *vtip*. Jakmile je najdeš, zobraz si jejich obsah.

Očekávané cíle

- Žák rozumí příkazu `find`
- Žák umí použít příkaz `find`

Předpoklady

- Žák rozumí příkazu `grep`
- Žák umí použít příkaz `grep`
- Žák umí použít rouru

Metoda výuky

- Samostatná práce žáků
- Pedagogický konflikt (žák nezná příkaz pro vyhledávání v řetězcích)

Očekávaná chybná řešení

Zde je velmi pravděpodobně, že žák zadá špatně název vyhledávaného souboru. Buď to nezadá hvězdičky, ale jiné znaky nebo nevyplní hvězdičky z obou stran u slova *vtip*. Kvůli tomu pak nezíská všechny soubory, které klíčové slovo obsahují. Dále také mohou žáci chybovat v zadávání místa, ve kterém bude skript prohledávat soubory s daným názvem.

#### Poznámka

```
#!/bin/bash  
echo "Kdo hledá, najde"  
find . -name "*vtip*"
```