

# ***C2110 Operační systém UNIX a základy programování***

**B02: Proměnné a aritmetické operace**

**PS/2022 Prezenční forma výuky: Rev3**

**Petr Kulhánek**

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta  
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

# Proměnné

---

# Proměnné

V jazyce Bash se proměnnou rozumí **pojmenované umístění** v paměti, které obsahuje hodnotu. Hodnota proměnné v jazyce Bash je vždy **typu řetězec (text)**.

**Nastavení proměnné:** **nesmí** být mezera mezi **jménem proměnné** a =

```
$ JMENO_PROMENNE=hodnota  
$ JMENO_PROMENNE="hodnota s mezerami"
```

**Přístup k hodnotě proměnné:**

```
$ echo ${JMENO_PROMENNE}
```

"TEXT **`\${PROMENNA}**TEXT"

**Zrušení proměnné:**

```
$ unset JMENO_PROMENNE
```

pokud má být hodnota součástí textu, tak se název proměnné uvádí do složených závorek

**Přehled všech definovaných proměnných:**

```
$ set
```

# Nastavení proměnné



```
$ JMENO_PROMENNE="hodnota s mezerami"
```

```
$ JMENO_PROMENNE="hodnota s mezerami"
```

interpretuje se jako název programu

mezera

interpretuje se jako argument programu

```
$ JMENO_PROMENNE="hodnota s mezerami"
```

JMENO\_PROMENNE se nastaví na  
prázdný řetězec,

hodnota proměnná je dostupná  
pouze spuštěnému programu

mezera

interpretuje se jako název programu



```
$ JMENO_PROMENNE="hodnota s mezerami" program [arg1...]
```

lze uvést několik proměnných a jejich  
hodnot (v zápisu se dvojice  
JMÉNO=HODNOTA oddělují mezerou), které  
jsou dostupné jen pro spuštěný program

pokud název programu obsahuje rovnítko,  
musí být název uveden v uvozovkách

# Řetězce

V jazyce Bash lze použít čtyři typy řetězců:

- **bez uvozovek**

A=pokus

B=\*

C=\$A

nedochází k expanzi (nejedná se o argument, ale o hodnotu proměnné)

nahradí se hodnotou proměnné A

- **s uvozovkami**

A="pokus hokus"

B="\* \$A"

hodnota proměnné obsahuje dvě slova oddělené mezerou

nahradí se hodnotou proměnné A, hvězdička se neexpanduje (je uvedena v uvozovkách)

- **s jednoduchými uvozovkami (apostrofy)**

A='pokus hokus'

B='\* \$A'

text je uveden přesně, bez žádné expanze či transformace

- **s obrácenými jednoduchými uvozovkami (obrácený apostrof)**

A="`ls -d`"

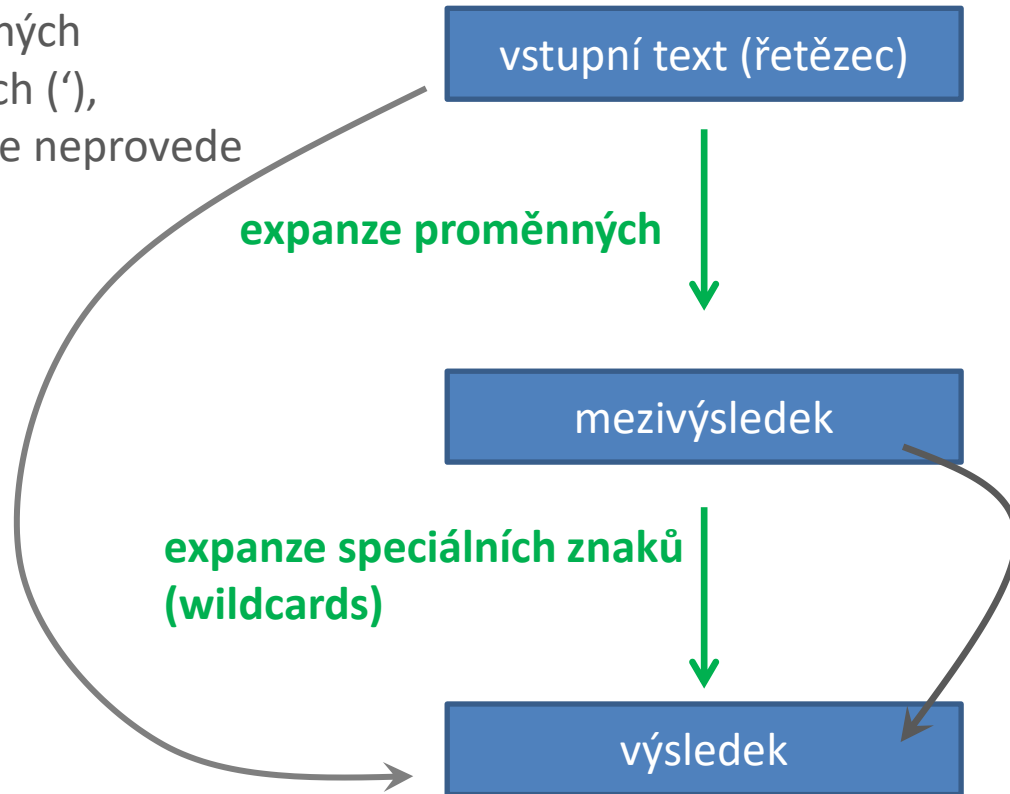
B="pocet : `ls | wc -l`"

do **místa** obrácených uvozovek se vloží standardní výstup **příkazu** uvedeného v uvozovkách

# Expanze řetězce/příkazové řádky

Pořadí expanze řetězce/příkazové řádky:

pokud je text uveden v jednoduchých uvozovkách ('), expanze se neprovede



pokud je text uveden ve standardních uvozovkách ("), nebo není obsažen ve slově, které by bylo možné expandovat, expanze se neprovede

Podrobněji: [man bash](#)

# Cvičení I

Pracujte v interaktivním režimu shellu.

1. Nastavte proměnnou A na hodnotu 55.
2. Vypište hodnotu proměnné A (příkazem echo)
3. Vylistujte všechny proměnné nastavené v daném terminálu. Je mezi nimi proměnná A? Použijte příkaz less nebo more k zřehlednění výpisu.
4. Použijte příkaz grep a vypište pouze řádek obsahující záznam o proměnné A. Vyhledávací vzor zvolte tak, aby byl nezávislý na hodnotě proměnné.
5. Vypište všechny nastavené proměnné, jejichž jména začínají písmenem A (grep ^TEXT).
6. Změňte hodnotu proměnné na "tohle je dlouhy retezec".
7. Vypište hodnotu proměnné A.
8. Zrušte proměnnou A.
9. Ověřte, že jste proměnnou zrušili (postupem řešeným v bodě 4).
10. Postupně nastavujte proměnné A, B a C podle příkladů uvedených na straně 5. Postupně ověřujte jejich hodnotu příkazy set a echo. Analyzujte případné rozpory.

# Aritmetické operace

---



# Aritmetické operace

- Aritmetické operace s celými čísly lze vykonat v bloku `(( ... ))`.
- Znaky se v bloku interpretují jako názvy proměnných. Proto není nutné používat operátor `$` pro získávání jejich hodnoty.
- Hodnoty proměnných se interpretují jako celá čísla. Pokud konverze selže, tak se použije nulová hodnota.

## Možné zápisy:

```
(( I = I + 1 ))  
(( I++ ))
```

## Aritmetická operace se získáním výsledku:

```
I=$(( I + 1 ))  
echo "Hodnota I zvetsena o jedna : $(( I + 1 ))"
```

hodnotu výsledku se vypíše  
do místa zápisu

Další informace: `man bash`

# Aritmetické operace

- Aritmetické operace s celými čísly lze vykonat v bloku `(( ... ))`.
- Znaky se v bloku interpretují jako názvy proměnných. Proto není nutné používat operátor `$` pro získávání jejich hodnoty.
- Hodnoty proměnných se interpretují jako celá čísla. Pokud konverze selže, tak se použije nulová hodnota.

## Možné zápisy:

```
(( I = I + 1 ))  
(( I++ ))
```

vhodnější zápis

## Aritmetická operace se získáním výsledku:

```
I=$(( I + 1 ))
```

```
echo "Hodnota I zvetsena o jedna : =$(( I + 1 ))"
```

hodnotu výsledku se vypíše  
do místa zápisu

Další informace: `man bash`

# Operátory

- = přiřazení
- + sčítání
- odčítání
- \* násobení
- / **celočíselné** dělení
- % zbytek po **celočíselném** dělení (modulo)
- ++ inkrementace (zvýšení hodnoty o 1)
- dekrementace (snížení hodnoty o 1)

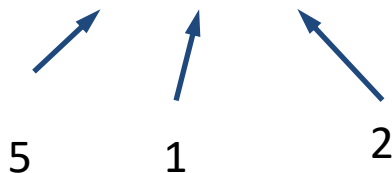
## Příklad:

**A=5**

```
(( B = A / 3 ))
```

```
(( C = A % 3 ))
```

```
echo $A $B $C
```



# Příkaz expr



Příkaz **expr** vyhodnocuje matematické výrazy, výsledky se tisknou do standardního výstupu.

## Příklady:

```
$ expr 1 + 2  
3
```

`\` zabrání expanzi speciálního znaku `*` na jména souborů a adresářů nacházejících se v aktuálním adresáři

```
$ expr 2 \* 3  
6
```

předáváme hodnotu proměnné, je nutné použít operátor `$`

```
I=`expr $I + 1`
```

Další informace: `man expr`

výsledek vložíme do proměnné `I`

Další možností je použití příkazu **bc**, který umí pracovat i s reálnými čísly.

# Cvičení II

1. Napište skript, ve kterém nastavíte proměnné A a B na hodnoty 5 a 6. Dále postupně vypisujte hodnotu součtu, rozdílu, podílu a násobení hodnot v proměnných A a B pomocí příkazu echo.

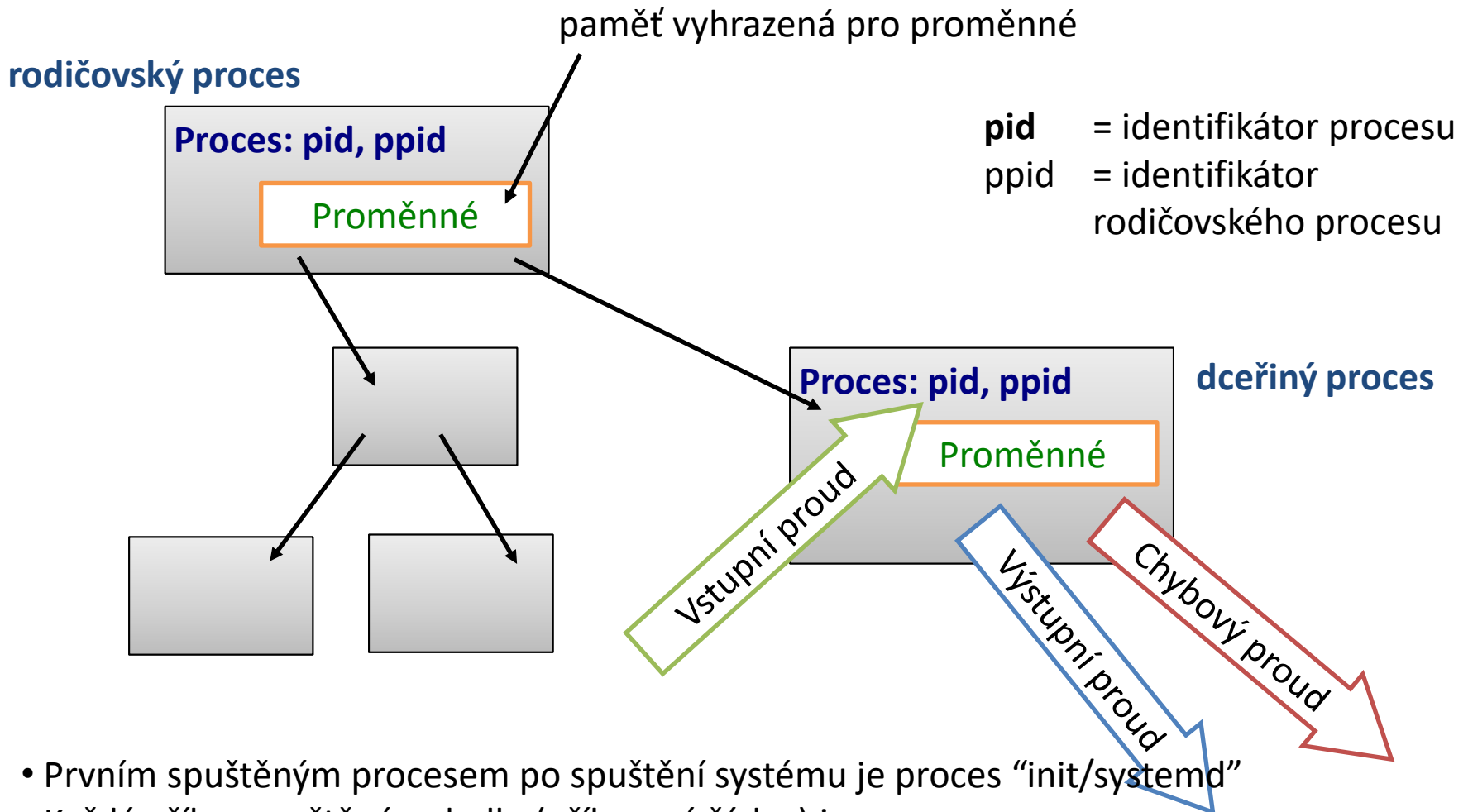
# Proměnné a procesy

---



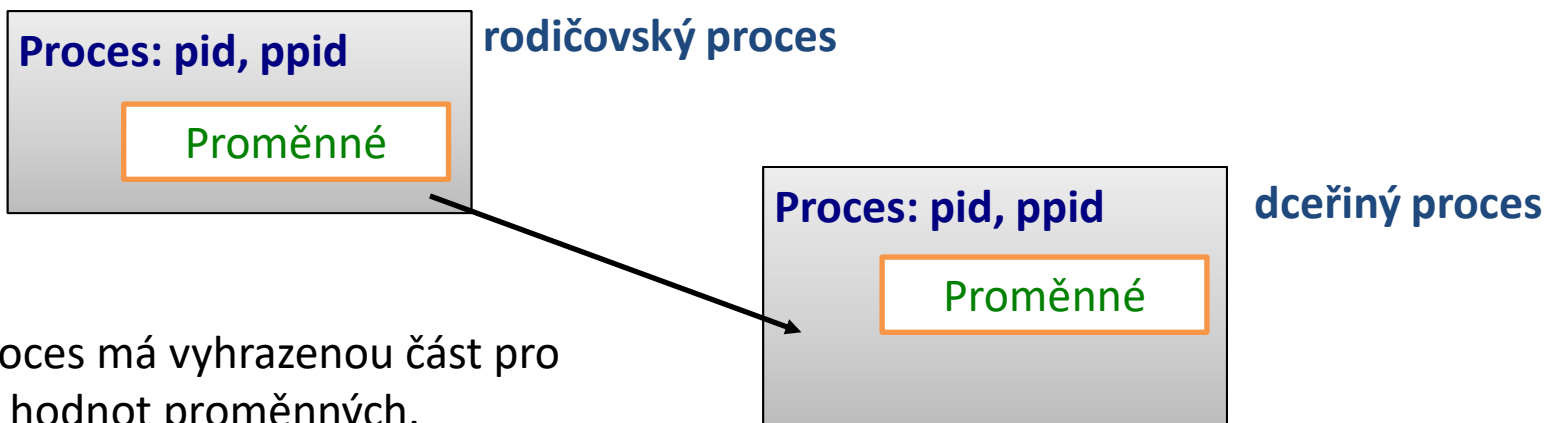
# Procesy

Proces je instance běžícího programu.



- Prvním spuštěným procesem po spuštění systému je proces "init/systemd"
- Každý příkaz spuštěný v shellu (příkazové řádce) je procesem

# Proměnné a procesy



Každý proces má vyhrazenou část pro ukládání hodnot proměnných.

Dceřiný proces v okamžiku svého spuštění **získá kopii** proměnných (exportovaných) a jejich hodnot od rodičovského procesu. Tyto proměnné může dle potřeby měnit nebo mazat. Dále může nastavovat nebo mazat nové proměnné. **Všechny tyto změny však po skončení dceřiného procesu zaniknou.** Změny se **neprojeví** na hodnotách **původních proměnných** rodičovského procesu.

## Export proměnné:

```
$ export JMENO_PROMENNE
```

export

```
$ export JMENO_PROMENNE="hodnota"
```

export s přiřazením



# Cvičení III

Pracujte v novém terminálu.

1. Zrušte proměnnou PATH. Jak se změna projeví na funkcionalitě příkazové řádky? Zkuste spustit příkaz ls a pwd. Chování vysvětlete.
2. Kdy se provede expanze divokého znaku \* v následujícím příkladu:

```
$ B="Obsah adresare je *"  
$ echo $B
```

3. Napište skript s názvem print\_C, který vypíše hodnotu proměnné C. Vysvětlete chování v následujících příkladech:

```
$ ./print_C  
$ C="hodnota 1" ./print_C  
$ echo $C  
$ C="hodnota 2"  
$ echo $C  
$ ./print_C  
$ export C  
$ ./print_C
```