

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

**12. lekce / modul 2**

**PS/2021 Prezenční forma výuky: Rev2**

**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

## ➤ AWK

- **Cykly**
- **Pole**

# Cykly


```
for(inicializace; podminka; zmena) {  
    prikaz1;  
    ...  
}
```

Příklad:

```
for(I=1;I <= 10;I++){  
    sum = sum + $I;  
}
```

Rozdíly vůči jazyku BASH

```
for((inicializace;podminka;zmena)); do  
    prikaz1  
done
```



# Cvičení 1

1. Napište skript, který sečte hodnoty všech čísel uvedených v souboru matice.txt.
2. Napište skript, který vytiskne počet slov, které obsahuje soubor rst.out. Výsledek ověřte pomocí příkazu wc.

**Data jsou v adresáři:**

`/home/kulhanek/Documents/C2110/Lesson12`

# Pole

**AWK** používá asociativní pole. Pole má název, k prvkům pole se přistupuje pomocí klíče. Klíč může mít libovolnou hodnotu a typ. Klíčem může být hodnota proměnné.

## Přiřazení hodnoty:

```
moje_pole[klic] = hodnota;
```

## Získání hodnoty:

```
hodnota = moje_pole[klic];
```

Jako klíče se nedoporučuje používat reálná čísla!

### Proměnná: **A**



proměnná obsahuje **pouze jednu** hodnotu

```
A=5;  
print A;
```

### Asociativní pole: **P**



pole může obsahovat **více hodnot**, pro každý klíč však pouze jednu.

```
P[9]=5;  
P["a"]=10;  
print P[9], P["a"];
```

# Pole - příklady

## Příklady:

```
i = 5;
moje_pole[i] = 15;
print moje_pole[i];

a = "slovo";
moje_pole[a] = "hodnota";
print moje_pole["slovo"],           moje_pole[5];
```

## Praktické použití:

```
BEGIN {
    pocet = 0;
}
{
    data[pocet++] = $1;
}
END {
    print pocet;
    for(i=0; i < pocet; i++){
        print data[i];
    }
}
```

script vypíše počet hodnot ve slouci 1 a poté jejich hodnoty

# Cvičení 2

1. Soubor structure1.dat obsahuje na každém řádku jméno prvku a polohu atomu. Napište skript, který soubor převede do formátu xyz a uloží pod názvem structure1.xyz. Zkonvertovanou strukturu zobrazte v programu VMD.
2. Hromadnost řešení ověřte konverzí souboru structure2.dat.

```
C   -1.8164140   3.6071310   0.6117350
C   -1.8002910   2.2769110   0.4584060
C   -0.6436270   4.3094580  -0.0124580
..   ...           ...           ...
```

první řádek: počet atomů

druhý řádek: libovolný komentář

## Zobrazení molekuly:

```
$ module add vmd
$ vmd structure1.xyz
```

```
20
molekula
C   -1.8164140   3.6071310   0.6117350
C   -1.8002910   2.2769110   0.4584060
C   -0.6436270   4.3094580  -0.0124580
..   ...           ...           ...
```

Data jsou v adresáři:

/home/kulhanek/Documents/C2110/Lesson12

# Samostudium

---





# Pole, ...

## Procházení seznamu klíčů:

```
for( promenna in pole) {  
    print pole[promenna];  
    ...  
}
```

Vykoná tělo cyklu pro každý klíč, který byl použit pro uložení hodnoty do **pole**. Hodnota klíče je uložena do **proměnné**.

**POZOR:** pořadí klíčů není určeno a nemusí tedy odpovídat pořadí zakládání prvků do pole

## Mazání záznamů s klíčem:

```
delete pole[klic];
```