

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

**8. lekce / modul 2**

**PS/2021 Prezenční forma výuky: Rev3**

**Petr Kulhánek**

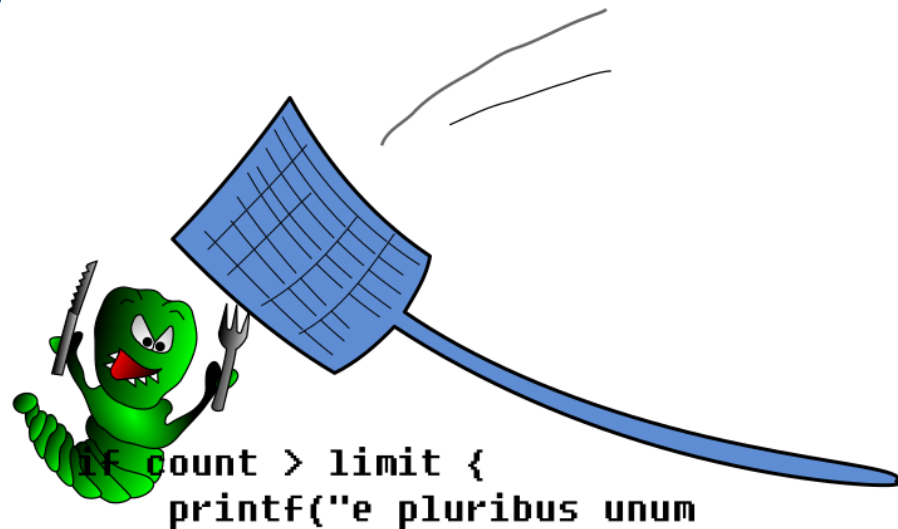
[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

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

# Hledání chyb

---

- Syntaktické chyby
- Logické chyby



# Syntaktické chyby

## Syntaktické chyby

- chyby v zápise neodpovídající specifikaci jazyka
- program není přeložitelný, místo chyby vypíše kompilér (např. C/C++, Fortran)
- skript není možné vůbec spustit (např. javascript)
- interpretér až při běhu skriptu narazí na nesrozumitelný zápis, běh skriptu je ukončen a řádek s chybou je vypsán do chybového výstupu, chyba se může projevit až za specifických podmínek běhu (např. bash, awk, gnuplot)

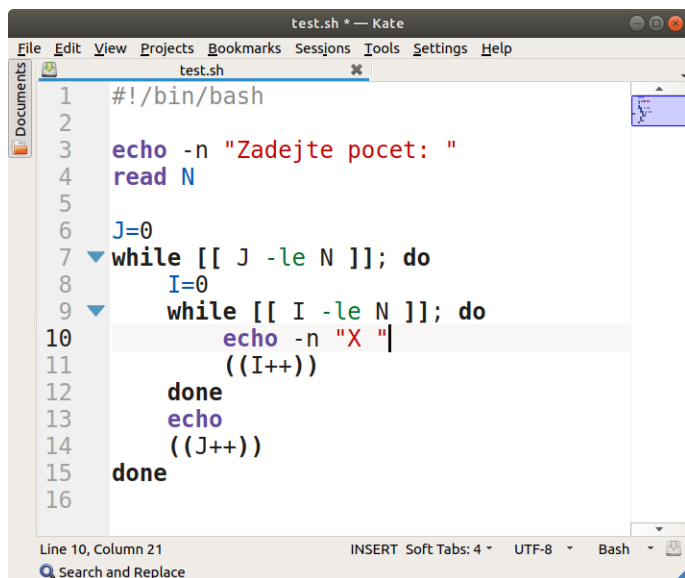
**Pozor! Začátek chyby může být na jiném místě než indikuje chybové hlášení.**

## Hledání chyb:

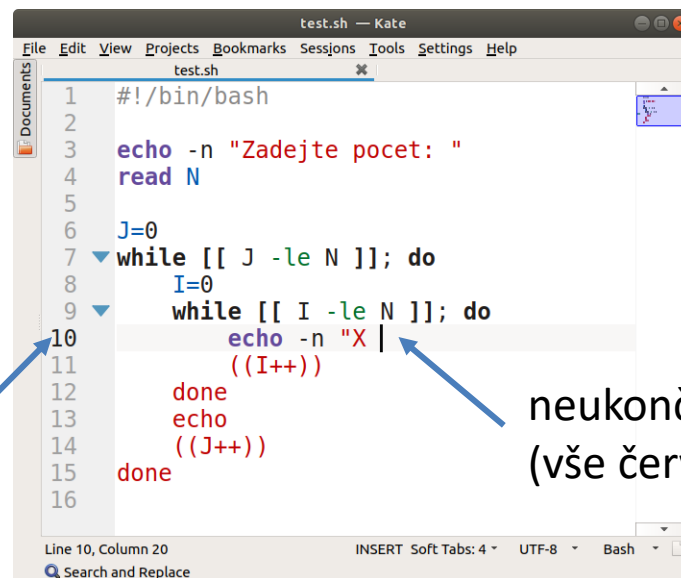
- zvýrazňování syntaxe
- statické analyzátory kódu

# bash - kontrola syntaxe

1. Základní kontrolu syntaxe je možné provést vizuálně **v textovém editoru**, který umožňuje **zvýraznit syntaxi** programovacího jazyka.



```
1 #!/bin/bash
2
3 echo -n "Zadejte pocet: "
4 read N
5
6 J=0
7 while [[ J -le N ]]; do
8     I=0
9     while [[ I -le N ]]; do
10         echo -n "X |"
11         ((I++))
12     done
13     echo
14     ((J++))
15 done
16
```



```
1 #!/bin/bash
2
3 echo -n "Zadejte pocet: "
4 read N
5
6 J=0
7 while [[ J -le N ]]; do
8     I=0
9     while [[ I -le N ]]; do
10         echo -n "X |"
11         ((I++))
12     done
13     echo
14     ((J++))
15 done
16
```

neukončený řetězec  
(vše červeně)

2. Chybové hlášení:

```
$ bash test.sh
```

```
Zadejte pocet: 5
```

```
test.sh: line 10: unexpected EOF while looking for matching `"'
```

```
test.sh: line 16: syntax error: unexpected end of file
```

**zapněte si zobrazování čísla řádků**

# bash - statická analýza

## 3. Statický analyzátor kódu:

- online  
<https://www.shellcheck.net/>
- příkazová řádka

```
test.sh — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
test.sh
1 #!/bin/bash
2
3 echo -n "Zadejte pocet: "
4 read N
5
6 J=0
7 while [[ J -le N ]]; do
8     I=0
9     while [[ I -le N ]]; do
10        echo -n "X |
11            ((I++))
12        done
13        echo
14        ((J++))
15    done
16
```

```
tests : bash — Konsole
File Edit View Bookmarks Settings Help
[kulhanek@pes tests]$ shellcheck test.sh

In test.sh line 10:
    echo -n "X
    ^-- SC1009: The mentioned parser error was in this simple command.
    ^-- SC1073: Couldn't parse this double quoted string.

In test.sh line 16:
^-- SC1072: Expected end of double quoted string. Fix any mentioned problems and try again.

[kulhanek@pes tests]$
```

# Logické chyby

## Logické chyby

- program/skript je možné spustit, ale výsledek neodpovídá očekávání a nebo není reprodukovatelný
- tyto chyby se **VELMI špatně hledají/opravují** - je vhodné se jim vyvarovat důkladným návrhem algoritmu

## Příčiny logických chyb:

- špatný návrh algoritmu (všechny jazyky)
- práce s neinicializovanými proměnnými (všechny jazyky)
- práce s nealokovanou/uvolněnou pamětí, zápis do nepřidělené paměti (C/C++, Fortran)
- souběh (race condition) u paralelních úloh (OpenMP, MPI, threads)

## Hledání chyb:

- dynamické analyzátoři běhu (C/C++, Fortran: valgrind)
- ladění běhu programu pomocí debuggeru

# debugger - bashdb + visual code

## 1. Spuštění editoru:

```
$ module add vscode  
$ code
```

## 2. Instalace rozšíření (pouze jednou):

<https://marketplace.visualstudio.com/items?itemName=rogalmic.bash-debug>

## 3. Konfigurace běhového prostředí (launch.json, pouze jednou):

```
{  
  // Use IntelliSense to learn about possible attributes.  
  // Hover to view descriptions of existing attributes.  
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387  
  "version": "0.2.0",  
  "configurations": [  
    {  
      "type": "bashdb",  
      "request": "launch",  
      "name": "Bash-Debug",  
      "cwd": "${workspaceFolder}",  
      "program": "${file}",  
      "args": [],  
      "terminalKind": "integrated"  
    },  
  ],  
}
```

Praktická ukázka konfigurace a použití.

# debugger - bashdb + visual code

Debugger

Hodnoty proměnných (operátor: \$)

zarážky (breakpoints)

Krokování

Aktuální pozice

Terminál: vstup/výstup

The screenshot shows the Visual Studio Code interface with the following components:

- Debugger:** Located in the top-left sidebar, indicated by a blue arrow.
- VARIABLES:** A section in the sidebar showing local variables like `$PWD` and `$?`.
- WATCH:** A section in the sidebar showing the values of variables `$J`, `$I`, and `$N`, indicated by a blue arrow.
- Breakpoints:** Red dots on the left margin of the script, with a label "zarážky (breakpoints)" and a blue arrow pointing to them.
- Script Editor:** The central area showing a bash script with a yellow highlight on line 10, labeled "Aktuální pozice".
- Stepping Controls:** A toolbar at the top right of the editor with icons for stepping through code, labeled "Krokování".
- Terminal:** The bottom window showing the execution of the script, with input and output, indicated by a blue arrow.



# Cvičení I

1. Krokujte běh skriptů z L08.M01.C01.U03.

# Instalace

---

Instalace shellcheck a bashdb do OS Ubuntu



# Poznámky k instalaci

1. **shellcheck** je součástí standardních balíčků:

```
$ sudo apt-get install shellcheck
```

2. **bashdb** je možné instalovat ze zdrojových kódu (viz L13.M02) nebo do Ubuntu 18.04 LTS z NCBR repositáře balíčků, jméno balíčku: ncb-bashdb

<https://wolf.ncbr.muni.cz/whitezone/packages/public/18.04/>

3. **Visual Studio Code** je možné instalovat z balíčku (.deb) nebo spouštět z binárního archívu (.tar.gz). Návody viz dokumentace.

<https://code.visualstudio.com/>

Pro uživatele Infinity (C2115): bashdb a rozšíření bash-debug vyžaduje systémový příkaz pkill, který je v prostředí Infinity předdefinován.

Kolizi vyřešíte pomocí příkazu unset zadaného před spuštěním editoru.

```
$ unset pkill
```