

# C2110 UNIX and programming

## 10<sup>th</sup> Lesson

Petr Kulhánek, Jakub Štěpán

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

National Centre for Biomolecular Research, Faculty of Science  
Masaryk University, Kotlářská 2, CZ-61137 Brno



INVESTMENTS IN EDUCATION DEVELOPMENT

CZ.1.07/2.2.00/15.0233

# Contents

- **In-semester test II**
- **Running commands III**
  - Variable PATH
- **Hybrid scripts**
  - Redirection in scripts
- **Commands**
  - type, hash, tr, memcoder, mplayer

# In-semester test II

---

# In-semester test II

## ➤ **Test is questionnaire (ROPOT) in IS**

**Student – ROPOT – e-learning – C2110**

**Length 20 minutes.**

**Only one set of questions.**

**Use ‘Save temporarily’ during work.**

**Evaluation can be done only once.**

**It is allowed and suggested to**

Test commands in terminal.

Search manual pages, lecture notes and lecture presentations.

Call teacher if you have problems.

**It is forbidden to**

Communicate with other person except teacher

# Running commands III

---

- Variable PATH
- Commands
  - tr, mplayer, mencoder

# Running commands and apps, III

For shell to be able to run any command, it needs to have **full path** to file, which contains binary code of program or script.

1. Path to command is first looked up in recently used commands table:

```
$ hash
```

```
hits      command
```

```
1        /bin/rm
```

```
3        /bin/ls
```

Table may be erased by command:


```
$ hash -r
```

2. If no path is found, then directories given in variable **PATH** are searched.

```
$ echo $PATH
```

Search order

```
.../usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```



Directories should be separated by **:** (colon)

3. To find out that path exists use command **type**

```
$ type ls
```

```
/bin/ls
```

# Adjusting variable PATH

## Manual change of value of variable PATH

```
$ export PATH=/my/path/to/my/commands:$PATH
```

Path to directory with, that are to be accessible without need to write full path.

**Path has to be in absolute form!** (relative path is security risk).

separator

Previous value of **PATH** variable (necessary for system commands to be found)

## Automatic PATH value adjustment

Automatic change of variable PATH value (and possibly other system variables) may be done by command **module**.

```
$ module add vmd
```

# Commands for exercise

Command **tr** is may be used to transform or delete symbols from standard input. Result is printed to standard output.

## Examples:

```
$ cat soubor.txt | tr --delete "qwe"
```

From **soubor.txt** contents symbols "q", "w" and "e" are removed.

```
$ cat soubor.txt | tr --delete "[:space:]"
```

From **soubor.txt** file contents all white spaces are removed.

```
$ echo $PATH | tr ":" "\n"
```

All symbols ":" echoed to standard input of tr will be replaced by new line symbol "\n"



# Exercise

1. Print variable **PATH** contents.
2. Print directories from variable **PATH**, each to separate line.
3. What directory contains program **kwrite**?
4. Print contents of recently used commands table.
5. Find out how command **module add vmd** change variable **PATH**.
6. Find directory where is command **vmd**?

# MPlayer

**MPlayer** is useful command to play video. Short usage description is provided by submitting command with no argument.

## Example:

```
$ mplayer movie.avi  
play video movie.avi
```

## Interesting options:

-loop N	play N loops
-fs	play video in full screen mode

<http://www.mplayerhq.hu>

# MEncoder

**MEncoder** is command to code video. It may be used to convert video formats, codec change or creating video from picture frames.

## Creating video from picture frames:

```
$ mencoder "mf://*.png" -mf fps=25 -ovc lavc -o output.avi
```

**Input data.** Use all files with extension png. Pictures sequence is sorted according to file names.

**Output encoder.**

**Output file name.**

**Frame rate per second (FPS – frames per second).**

Overview: <http://mariovalle.name/mencoder/mencoder.html>

# Alternatives

<http://ffmpeg.org/>

FFmpeg is a complete, cross-platform solution to record, convert and stream audio and video. It includes libavcodec - the leading audio/video codec library.

<http://gstreamer.freedesktop.org/>

GStreamer is a library for constructing graphs of media-handling components. The applications it supports range from simple Ogg/Vorbis playback, audio/video streaming to complex audio (mixing) and video (non-linear editing) processing.

# Exercise

1. Copy two files with **avi** extension from directory **/home/kulhanek/Data/Video** to subdirectory of your home **my\_video**, which will be in your home directory.
2. Play both video files in program **mplayer**. Learn basic program functions: pause video, rewind video, switch to full screen.
3. Directory **/home/kulhanek/Data/MovieImages** contains pictures in png format. Create directory **/scratch/your\_login/images** , copy images to that directory.
4. Find size (width, height and bit depth) does have image **e\_0010.png**?
5. Create two videos with FPS=10 and FPS=50 from images.
6. Play created videos.

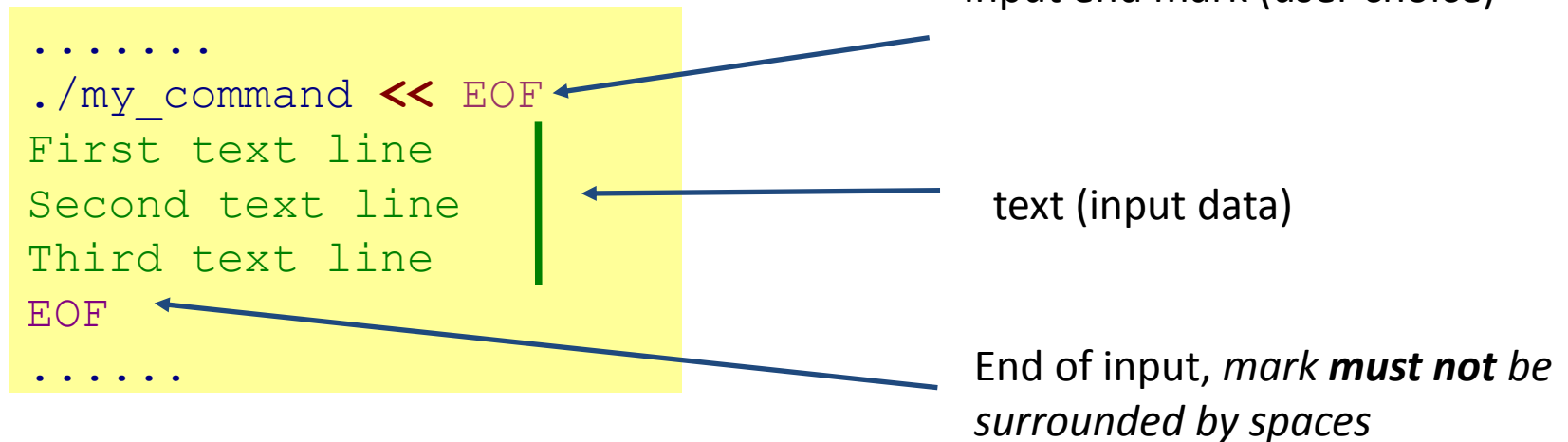
# Hybrid scripts

---

- **Redirection in scripts**

# Redirection in script

Redirection of standard input of program `my_command` from script file.



This redirection form is especially advantageous in scripts, it works in command line as well. Advantage is variables expansion in input text.

# Showcase

```
#!/bin/bash

for ((I=1;$I<=10;I++)); do
    NAME=`printf "%02d.txt" $I`
    cat << EOF > $NAME
    This is file number: $I
EOF
done
```

Result of command in backward apostrophes `` is saved to variable NAME.

Highlighted text is sent to **standard input** of **cat** command, variables are expanded in advance, resulting text is saved to file \$NAME.

```
#!/bin/bash

gnuplot << EOF
plot sin(x)
EOF
```

In this way gnuplot scripts may be automatically generated.



# Exercise

1. Write script, that creates ten files. Each file with name in format XX.txt, where XX is file number. For numbers less than 10 use leading zero to keep file name length. Each file will contain following text (X is file number):

```
Automatically created text file
```

```
File number is: X
```

2. Write script(s), that creates set of images showing moving waves (function sin, or cos in 2D or 3D, up to your choice). Create video from files using command **mencoder**. Play video by command **mplayer**.