Shell Programming in 10 seconds

- So far covered: Single instructions, instruction sequences (piping)
- Still missing: Variables, control structures
- These are provided by the shell
- several shell flavors: sh, ksh, zsh, csh, tcsh, bash
- determine your shell: `echo $SHELL`
Shell Programming II

- make a script executable:
  ```bash
  #!/bin/bash
  ```
  (she-bang line) and set x-bit or
  ```bash
  bash yourScriptfile
  ```

- How to define a variable:
  ```bash
  VariableName=Value
  ```
  (no whitespace around =)
  ```bash
  VariableName: [a-zA-Z_] [a-zA-Z0-9_]*
  ```
  Value: unquoted string without whitespace or
  quoted string if it contains whitespace or
  ```bash
  $VariableName
  ```

Practical: EidPB WS2014/15
Shell Programming III

- simple output: echo
- formatted output: printf
- enable trace mode with set –x, even more verbose with –v
- special variables:
  - $# number of arguments (command line & function)
  - $*,$@ list of all command line arguments
  - “$*” one string, “$@” list of protected strings
Shell Programming IV

- **shift** return first positional parameter (deleting)
- arithmetics within `$(( ))`, operators like in C
- Exit status:
  - Exits status 0 indicates success!
  - `$?` holds the exit status of the last command
  - `exit n` returns immediately exit status `n`
Shell Programming V

- if-elif-else-fi

- **if** pipeline [pipeline...]
  
  **then**

  command-if-true-1

  [**elif** pipeline] [pipeline...]

  **then**

  command-if-true-2

  [**else** commands-in-all-other-cases]

  **fi**

Practical: EidPB WS2014/15
Shell Programming VI

- `test [expression]` or `[[expression]]`
- `-d file` is this a directory?
- `-e file` does file exists?
- `-h file` is this a symbolic link?
- `-s file` is not empty?
- `-x file` is executable?

Practical: EidPB WS2014/15
Shell Programming VII

- Loops: for and while loops
  - `for i in list`
    - `do`
      - commands...
    - `done`
  - `while condition / until condition`
    - `do`
      - commands
    - `done`
Vi I

- very powerful text editor
- available on (nearly) all UNIX machines
- nowadays: vim (vi improved)
- now graphical user interface needed
- needs little resource
- still works when systems responds slowly
Vi – Sources for Help

- [http://www.vim.org](http://www.vim.org)
- man page
- Help system: "`:help`"
- vimtutor
Vi – Modes and Basics

- command mode (normal mode)
- command line mode (ex mode)
- insert mode
- visual mode (vim only)
- command mode -> insert mode: insert/edit command
- insert mode -> command mode: ESCAPE, CTRL-C
- command line mode command mode: RETURN

Practical: EidPB WS2014/15
Vi – navigation

- navigation: cursor keys or h,j,k,l (left, down, up right)
- gg: start of document, G: end of document
- 0 begin of line, w next word, e end of word, $ end of line
- preceded by repeater number like: 5w, 2j
Vi – edit

- Insert: i – after cursor, I – start of line
- Erase: x – erase at cursor, X – erase before cursor
- Append: a – append at cursor, A – append at end of line
- o: append in new line after, O: append in new line before cursor
- r: replace single character, c+positioner: change
- u: undo, “.”: repeat last command
More complex commands

- full command syntax:
  ```
  ["register"
  [counter]command[cursor
  positioner]
  ```

- save and exit: ZZ or :wq

- exit without save: "":q!" (force quit)
Make Life easier

- `:set showcmd` to show your typed commands
- `:set nu` display line numbers
- `:syntax on` to enable syntax highlighting
- `:set hls / nohls` to enable search highlighting
Thank you for your attention