

Vistula, IT Faculty, 2015

Operating Systems

Dmitry A. Zaitsev

<http://daze.ho.ua>

Lecture 6:

Programming in Shell Language

Bash

- *Bourne again shell*
- *Arguments*
- *Built-in commands: echo, pwd, read*
- *History*
- *Expansions: ~, \$, *, ?, ;, ", `*
- *Variables: A=5*
- *Environment variables: \$HOME, \$PATH*
- *Sequence control: if, while, until, for, case, select*
- *Execute a script – commands from a file*

Expansion

- Brace expansion: `a{d,c,b}e`
- Tilde expansion: `"~"` - login name
- Parameter expansion: `"$"` - parameters, command, arithmetic expression;
`${parameter}`, `$(command)`, `$((expression))`
- Pathname expansion: `*`, `?`, `[...]`
- Here document: `<<<word ... word`

Variables, arrays, and expressions

- Variable = Value
- Variable[Index] = Value
- A="Hello VU"
- echo \$A

- X=5
- Y=6
- expr \$X + \$Y

Command substitution

- `$(command)`
- ``command``
- `A=$(ls -l | grep di)`
- `echo prefix${A}suffix`

Branching – if

- if list ; then list ; [elif list ; then list ;] ... [else list ;] fi
- if list; then list; else list; fi
- if list; then list; fi

- if /bin/true; then echo true is true; fi
- if ls; then echo listed; fi

Evaluation of expressions

- `[[expression]]`
- test expression
- `>, <, >=, <=, ==, !=`
- Files: `-a, -d, -f, -x`

- `if [[5 < 3]] ; then echo a; else echo b; fi`
- `if [-f abc]; then echo there is abc; fi`
- `if test -f abc ; then echo there is abc; fi`

Loop – while

- while list ; do list ; done

A=1

S=0

while [[\$A < 5]]; do

S=\$((\$S + \$A))

A=\$((\$A + 1))

done

echo \$S

Loop – for

- for name [[in [word ...]];] do list ; done
- for i in 1 2 3 ; do echo \$i ; done
- for n in `ls` ; do echo name is \$n ; done

Shell functions and export

- function name [()] compound-command
- export Variable

- function pl () {ls; ps}
- export pl
- pl

Shell scripts

```
cat > dummy
```

```
#!/bin/bash
```

```
while /bin/true; do $1; done
```

```
Ctrl^C
```

```
chmod +x dummy
```

```
./dummy "I'm dummy"
```

Parameters

- \$1 \$2 ...
- \$*, @\$ - all paremeters
- \$# - number of parameters
- \$? - exit status
- \$\$ - shell PID

Shell variables

- BASH, BASHPID, BASH_VERSION
 - HOSTNAME, PPID, EFS
 - HOME, PATH
 - PS1, PS2, ...
-
- cd \$HOME
 - echo \$PATH

Command line prompt format

- \d - date
- \h - hostname
- \u - username
- \t - time
- PS1="\h:\u>"
- daze-fast:daze>

man bash