

KET/KTL 2020

*Introduction to the Shell Command Language and Standard Tools
Overview with Examples*

Jan Bělohoubek



**FACULTY OF ELECTRICAL
ENGINEERING
UNIVERSITY
OF WEST BOHEMIA**

- 1 Aims of this Lecture
 - Motivation (Anketa)
- 2 Unix Operating Systems
- 3 Command Line Interface – CLI
- 4 (Our) BASH Reference

- provide an **overview** on "classical" engineering SW tools
- no more "What is this?" !
- hands on ... but gently
 - our time is limited
 - speed-up the learning curve as you will face it

Area/Tool	GNU/Linux	CLI	Scripting	VCS	Make	SSH
Embedded SW	maybe	probably	of course	of course	of course	maybe
Application SW	maybe	probably	of course	of course	of course	probably
Digital Design	probably	probably	of course	of course	probably	probably
PCB Design	maybe	maybe	probably	of course	maybe	maybe
Mechanical Design	maybe	maybe	maybe	probably	rather not	rather not
Research ⁺¹	of course	of course				

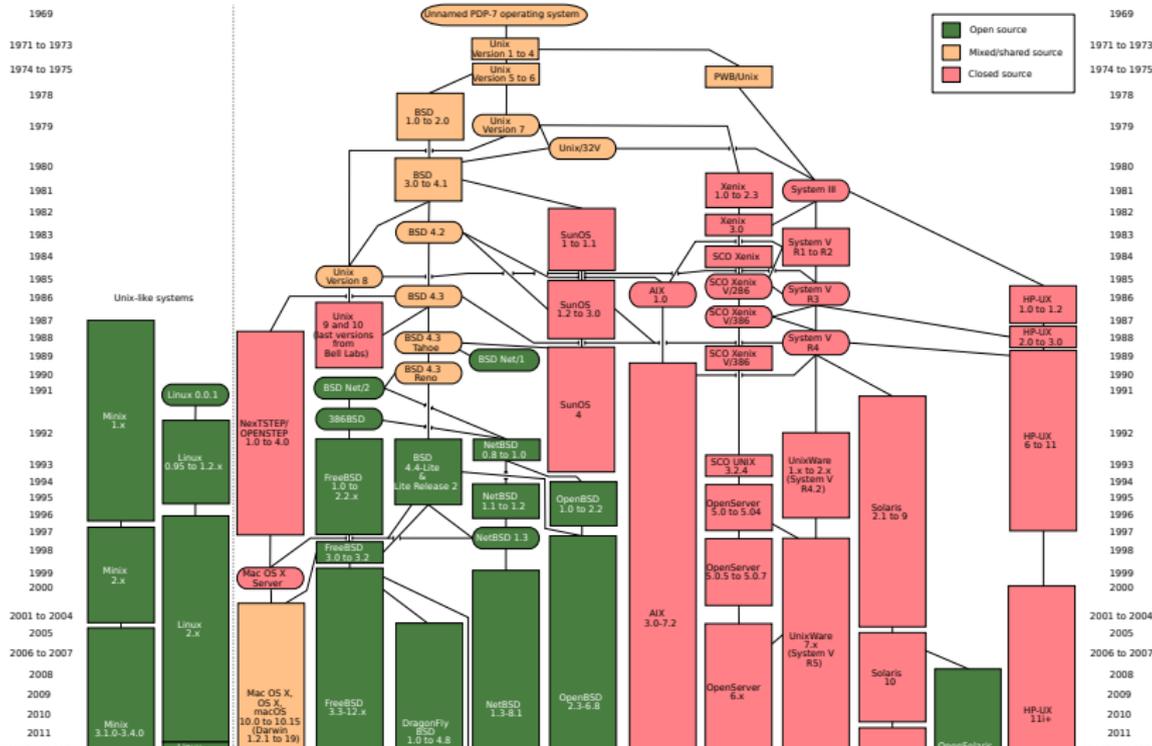
¹Simulations, Supercomputing, Networking, Control, AI ...

- 1 Aims of this Lecture
- 2 Unix Operating Systems
 - Brief History
 - Unix/Linux Surprising/New Features
 - How To Start . . . for Windows Users
 - Linux Distributions Popular in Industry (EDA)
 - Linux Distributions Popular for Home Use
- 3 Command Line Interface – CLI
- 4 (Our) BASH Reference

Unix Operating Systems

Brief History

Image source: https://en.wikipedia.org/wiki/History_of_Unix



- Multi-User
 - multiple users work concurrently (terminals); root user
- Multi-Tasking/Time-Sharing
 - multiple tasks run concurrently
- Hierarchical filesystem – tree structure with root "/"
 - everything is a file (directory, device, ...)
- Software Management Tools and Repositories
 - Linux distributions have software management tools and repositories with ability to install/update software automatically since mid 90s (1997 RPM, 1995 dselect)
 - Android Store comes in ... 2008
 - Windows Store comes in ... 2012
 - How you search for a new software (or updates)?

- connect to a remote machine (our case!)
- select a Linux Distribution and install it:
 - along your Windows Installation (dual-boot)
 - in a virtual environment (e.g. VirtualBox)
 - Windows Subsystem for Linux
- currently, there are more than 500 actively developed Linux Distributions²

²<https://www.tecmint.com/linux-distro-for-power-users/>

- Red Hat Enterprise Linux (RHEL)
- CentOS – RHEL "clone"
- SUSE Linux Enterprise (SLE)
- (Oracle Linux)

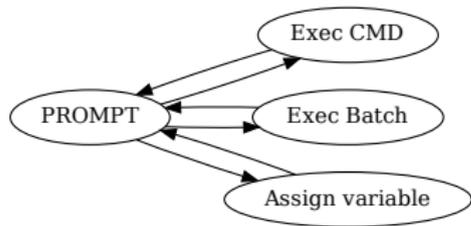
- Fedora – driven by community around RedHat ("test environment" for RHEL)
- OpenSUSE – driven by community around SUSE
- Ubuntu – commercial but free (with a paid support option)
- Debian – greatest community driven distribution
- Many others: Linux Mint, Slackware, Gentoo, Arch, . . .

- 1 Aims of this Lecture
- 2 Unix Operating Systems
- 3 Command Line Interface – CLI
 - It is Used for ...
 - Why to Use ...
 - How to Use ...
 - Shell Command Language – Unix Standard
 - SSH – Secure Shell
- 4 (Our) BASH Reference

- traditional interface for **interactive** access to computers
 - (super)computer **terminals** starting in the mid-1960s
 - PCs in/since 1980s
- nowadays mostly (typical terminal host systems)
 - servers and supercomputers – standardized
 - embedded devices – standardized (e.g. AT) or custom/proprietary (e.g. KETCube)
 - network equipment – routers, probes, . . .
 - **R&Ds personal computers** – often mix TUI & GUI for efficiency & automation

- less resources → remote access, terminal access ...
- "embedded" scripting – automation & **repetitive tasks**
- CLI vs. script language: program execution vs. function call; work with text vs. work with data (objects)
- efficient work with files & text
 - filter certain lines from a huge csv file (Excell will crash!)
 - browse & search log files
 - bulk file rename
- software execution control
 - handle return value
 - access "hidden" features
- ...

- Terminal (emulator)
 - HW device to access a terminal host system
 - SW application (terminal emulator), that mimics HW terminal mostly on PC
- Command Interpreter:



Prompt example – (ba)sh:

```
[jan@host]:/home/jan$ # [user@host]:path$
```

Prompt example – KETCube:

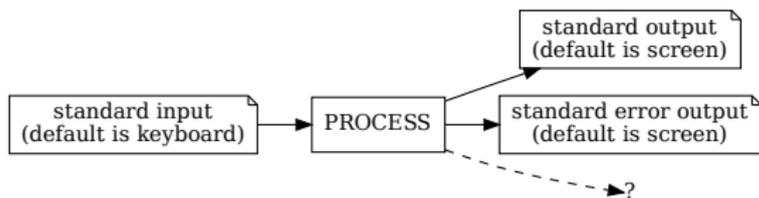
```
>>
```

- shell is a command interpreter
- there are many Shell Command Language implementations with a superset of features
 - sh (Bourne SHell), **bash** (Bourne Again SHell) – most popular
 - ash (Almquist SHell), dash (Debian Almquist SHell) – lightweight, scripts (may be) compatible with bash
 - csh (C shell), tcsh – increased readability (scripting!); C-inspired syntax
 - sh is nowadays a link to (in most cases) bash, tcsh or dash
- We will discover only a small part of the Unix tool universe: **Many details are hidden or simplified intentionally!**

- the cryptographic network protocol – establishes a secured client-server connection/channel
- capabilities (openSSH, SSHv2):
 - remote SHell & tunneling
 - user authentication by password/private-public key pairs
 - Secure Copy (SCP) and File Transfer (SFTP) protocols
- Software support:
 - Un*x: openSSH client/server (ssh, sshfs, scp)
 - Windows: Putty, WinSCP

```
$ ssh username@eryx.zcu.cz -i .ssh/id_rsa # access remote with
$                                     # pub/priv. key pair
$ ssh -L8080:localhost:80 ws.net # access ws.net:80
$                                     # from localhost:8080
$ ssh -X username@eryx.zcu.cz xeyes # display xeyes locally
```

- 1 Aims of this Lecture
- 2 Unix Operating Systems
- 3 Command Line Interface – CLI
- 4 (Our) BASH Reference**
 - Command Execution
 - Builtin Commands
 - Shell Variables
 - Filesystem Tree – Logic Filesystem
 - Standard Unix Tools – Filesystem
 - Standard Unix Tools – Text (Files) and Filters
 - Standard Unix Tools – Control Flow
 - Standard Unix Tools – Get Help
 - Bash Tricks
 - Midnight Commander Tricks



CMD [parameters] > output < input

- inputs may be redirected from file: {< , <<}
- outputs may be redirected to file: {> , >>}
- commands may be concatenated: {; , &&, ||, ...}
 - cmd1 ; cmd2 – run cmd1, then run cmd2
 - cmd1 | cmd2 – filter
 - cmd1 && cmd2 – run cmd2 only if cmd1 OK
 - cmd1 || cmd2 – run cmd2 only if cmd1 ERROR



- `source` – source file
- `echo` – print args to stdout
- `pwd` – get absolute path of the working directory
- `exit` – exit shell + return value

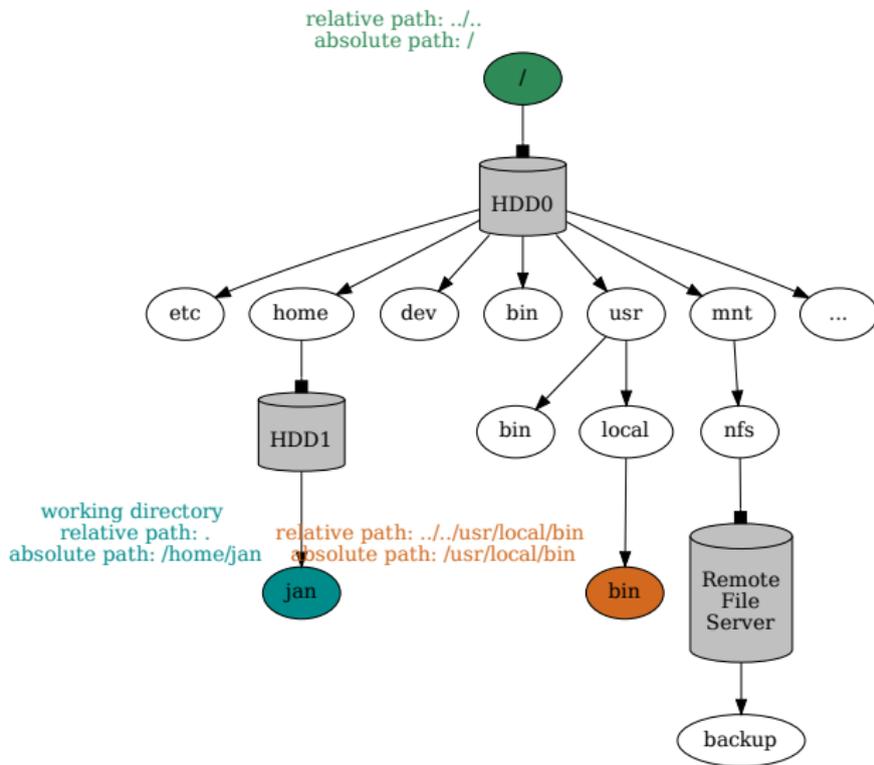
```
$ source .bashrc
$
$pwd
/home/jan
$
$ echo Hello World!
Hello World!
$ d
bash: d: Command not found...
$ alias d="date"
$ d
Po 2. listopadu 2020, 15:39:00 CET
$
$ exit 0          # no error
```

Important variables:

- HOME – The current user's home directory
- PATH – list of directories in which the shell looks for commands
- PS1 – prompt definition
- ? – last command status

```
$ echo $HOME
/home/jan
$
$ echo $PATH
/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin
$
$ PATH=${PATH}:/sbin # add new directory to PATH
$ echo $PATH
/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/sbin
$
$ echo $PS1
[\u@\h \W]\$ # special characters are evaluated by BASH
$
```

(Our) BASH Reference Filesystem Tree – Logic Filesystem



- `cd` – Change Directory
- `cp` – CoPy file
- `mv` – MoVe file
- `rm` – ReMove file
- `mkdir` – Make DIRectory
- `touch` – change modify time of file, create empty file if the file does not exist
- (`pwd` – get absolute path)
- + `mc` – Midnight Commander :-)

```
$ ls -l /
...
drwxr-xr-x    2 root root    4096 Mar 14  2020 bin
dr-x-----   3 root root    4096 Oct 23  03:51 boot
...
drwxr-xr-x    2 root root          6 Sep 23  2013 home
...
$
$ pwd
/users/b/belohoub/home
$ mkdir KTL
$ cd KTL
$ ls
$ touch test
$ ls
test
$ rm test
$ ls
$
```

- `cat` – CATenate
- `head` and `tail`
- `more` and `less`
- `wget`
- `sort` and `uniq`
- `grep` – like `ed` command `'g/re/p'`
- `sed` – Stream EDitor (`ed` – 1973)
- `awk` – Aho–Weinberger–Kernighan
- `wc` – Word Count
- `diff` – DIFFerences
- `vi` – VIsual (learn how to exit :-))

Dataset description:

<http://opendata.zcu.cz/Energeticky-dispecink.html>

```
$ wget http://openstore.zcu.cz/OD_ZCU_09_2020/\
OD_ZCU_POCASI_09_2020_CSV.zip
$
$ unzip OD_ZCU_POCASI_09_2020_CSV.zip
Archive:  OD_ZCU_POCASI_09_2020_CSV.zip
  inflating: OD_ZCU_POCASI_09_2020.CSV
$
$ cat OD_ZCU_POCASI_09_2020.CSV | wc -l
$
$ less OD_ZCU_POCASI_09_2020.CSV
...
$
$ tail -n 10 OD_ZCU_POCASI_09_2020.CSV
$
$ # get data for 1.9.2020
$ grep \"01.09 OD_ZCU_POCASI_09_2020.CSV
$
```

Dataset description:

<http://opendata.zcu.cz/Energeticky-dispecink.html>

```
$  
$ # get first 10 temperatures  
$ cat OD_ZCU_POCASI_09_2020.CSV | awk -F";" '{print $2}' \  
| head -n 10  
$  
$ # print date&times when temperature was 14 - 15 C  
$ awk -F";" '{if (($2 > 14) && ($2 < 15)) print $1}' \  
OD_ZCU_POCASI_09_2020.CSV  
$  
$ # now compute in how many days in October  
$ # the temperature approached 25 C  
$ ...  
12  
$
```

Conditional branch:

```
$ touch /file
$ if [ "$?" = "0" ]
> then
>   echo "Last␣command␣execution␣OK!"
> else
>   echo "Last␣command␣execution␣FAILED!"
> fi
```

For loop:

```
$ cd KTL    # $( cmd ) -> execute cmd and use its std output
$ for a in $( ls /etc | head -n 10 )
> do
>   echo $a ...
>   touch ${a}_$( date +%s )
> done
```

+ while, case (switch)

- `-h` | `-help` command parameters
- man pages with 9 sections:
 - 1 – Executable programs or shell commands
 - 3 – Library calls (functions within program libraries)
- info pages (more complex)
- Example:

```
$ man printf
```

```
...
```

```
$ man 1 printf
```

```
...
```

```
$ man 3 printf
```

```
...
```

- TAB – (auto)complete command
- ↑ ↓ – browse command history
- CTRL+R – search command history
- CTRL+C – interrupt the running process
- ...

- TAB – switch focus between panels
- INSERT – mark/remove mark
- ALT+S – search
- CTRL+O – switch to terminal
- ALT+O – in inactive panel display parent directory (for file) or directory content (for directory)
- ALT+ENTER – insert file name to cmdline
- ...

■ Featured Reading and Resources:

- https://pubs.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html
- <https://www.gnu.org/doc/doc.html>
- <https://www.gnu.org/software/bash/manual>
- <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- <https://www.linuxexpres.cz/praxe/serial-o-bashi> (cs)
- <https://learnxinyminutes.com/docs/awk/>
- <https://learnxinyminutes.com/docs/bash/>

Thank you for your attention!

Jan Bělohoubek
UWB, Czech Republic
belohoub@fel.zcu.cz
+420 377 63 4514



**FACULTY OF ELECTRICAL
ENGINEERING
UNIVERSITY
OF WEST BOHEMIA**