

# Introduction to UNIX

## **Libor Forst, SISAL MFF UK**

- General introduction
- History, principles
- File system, organization, tools
- Processes, life cycle, communication
- Shell: conception, commands
- Text processing (ed, grep, sed, vi, awk)

# Literature

- L.Forst: Shell v příkladech aneb aby váš UNIX skvěle shell; Matfyzpress 2010  
`www.yq.cz/SvP`
- The Single UNIX® Specification, Version 3 (POSIX),  
The Open Group Base Specifications Issue 7,  
IEEE Std 1003.1-2008  
`www.opengroup.org/onlinepubs/9699919799`
- manual pages

# Literature (basic)

- G. Todino, J. Strang, J. Peek: Learning the UNIX Operating System; O'Reilly & Associates 2002; ISBN 0-596-00261-0
- A. Robbins: UNIX in a nutshell; O'Reilly & Associates 2006; ISBN 978-0-596-10029-2
- L. Lamb: Learning the vi Editor; O'Reilly & Associates 1990; ISBN 0-937175-67-6

# Literature (programming)

- C. Newham, B. Rosenblatt: Learning the bash Shell; O'Reilly & Associates 2005; ISBN 0-596-00965-8
- D. Dougherty: sed & awk; O'Reilly & Associates 1997; ISBN 978-1-565-92225-9
- A. Robbins, N. Beebe: Classic Shell Scripting; O'Reilly & Associates Inc., 2005; ISBN 978-0-596-00595-5
- C. Albing, J. Vossen, C. Newham: bash Cookbook; O'Reilly & Associates Inc., 2007; ISBN 978-0-596-52678-8
- E. Quigley: UNIX Shells by Example; Pearson Education Inc. (Prentice-Hall), 2005; ISBN 0-13-147572-X
- S. Kochan, P. Wood: Unix Shell Programming; SAMS, 2003; ISBN 0-672-32390-3

# Literature (principles)

- M.J.Bach: The Design of the UNIX Operating System; Prentice-Hall 1986
- E. Raymond: The Art of UNIX Programming; Addison Wesley; 2004; ISBN 0131429019

# Conventions

- Fixed part of command (non-proportional font)
  - used as it is written:

**man** [-k] [section] topic  
.....

- Variable part of command (italics)
  - requested text (word, number etc.) is used:

**man** [-k] [section] topic  
.....

- Optional part of command:

**man** [-k] [section] topic  
.....

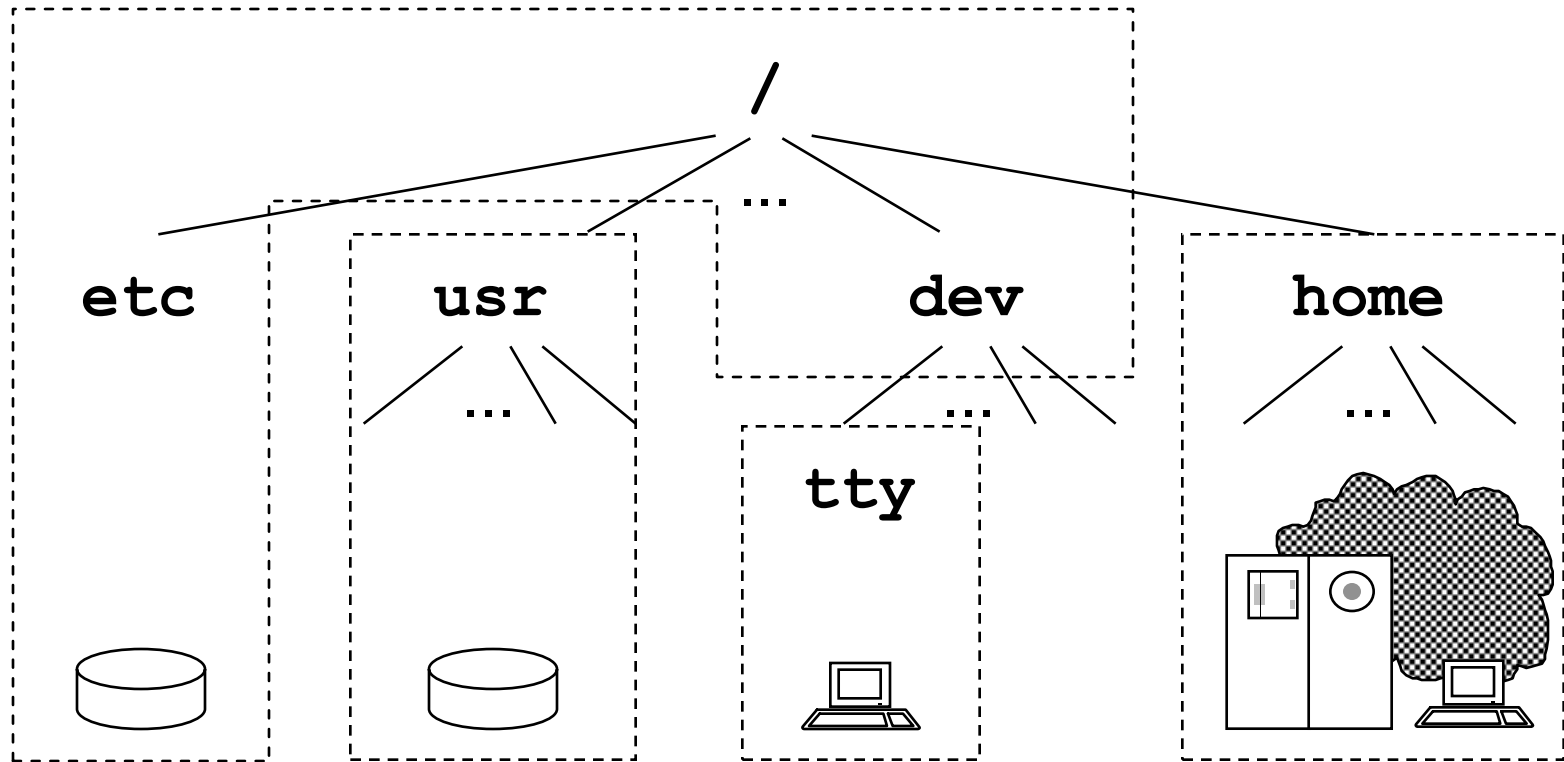
- Selection from more alternatives:

{ **BEGIN** | **END** | /regexp/ | cond | } { cmds }  
.....

# UNIX Characteristics

- inspired but not burdened by the past
- noncommercial environment
- open operating system
- file system
- users, groups
- processes, communication
- command interpreter, GUI
- utilities, C language
- portability, flexibility
- networking support
- public domain SW (e.g. GNU)
- command **man**

# Integrated hierarchical file system





# List of users (/etc/passwd)

```
forst:DxyAF1eG:1004:11:Libor Forst:/u/forst:/bin/sh
```

Field semantics:

- user login name
- encoded password (today e.g. in /etc/shadow...)
- user number (*UID*); superuser (*root*) has UID 0
- number (*GID*) of user's primary group
- full name (optionally with comment)
- home directory
- login-shell

# List of groups (/etc/group)

```
users : : 11 : operator , novak
```

Field semantics:

- group name
- *unused*
- group number (*GID*)
- group members

Users having a group as their primary group are members of the group, too.

# Command Interpreter (*shell*)

- essential program for UNIX operating
- independent system component: more shells exist
- command format:

*command -options operands* e.g. `ls -l /etc`

- metacharacters, e.g.:

`ls *.c > "output *.c"`

- commands:
  - internal: e.g. `echo`, `cd`, `pwd`
  - external: files in file system (path to search: **PATH**)

# Shell language

- shell interprets own programming language
  - control flow statement (e.g. `for`, `if`)
  - variables

```
PATH=/bin:/usr/bin:$HOME/bin
```

- language controls text substitutions (*text processor*)
- programming directly on the command line
- shell-script - file with stored shell program

```
sh test.sh; ./test.sh
```

# Process, communication

- Process
  - general idea: running user or system program
  - created by duplication of parent process
  - process list: command `ps`
- Communication
  - when started, parent prepares data for son;  
no way to share data vice versa
  - pipe - data flow from a producer to a consumer:  
`ls | more`
  - advanced tools (e.g. shared memory)

# man command

- Call:

`man [-k] [section] topic`

- Manual pages sections:

- 1 - general user commands
- 2 - kernel functions (*syscalls*)
- 3 - library (C language) functions
- 4 - devices and device drivers
- 5 - formats of (configuration) files
- 6 - trivial application programs
- 7 - miscellaneous
- 8 - administrator commands and programs

# more command

- Call:  
`more [-n] { +line | +/regexp | } [files]`
- Commands (\* - multiplication prefix *k* accepted):
  - space, **d** ... next page, next half of page (\*)
  - Enter ... next line (\* - *k* will set a default)
  - **s**, **f**, **b** ... skip *k* lines, pages, pages backward (\*)
  - */regexp*, **n** ... search for *k*-th string occurrence (\*)
  - ' ... return to search beginning
  - **!cmd**, **v** ... start shell, editor
  - **=**, **h** ... file position output, display help
  - **:n**, **:p** ... skip to next/previous file

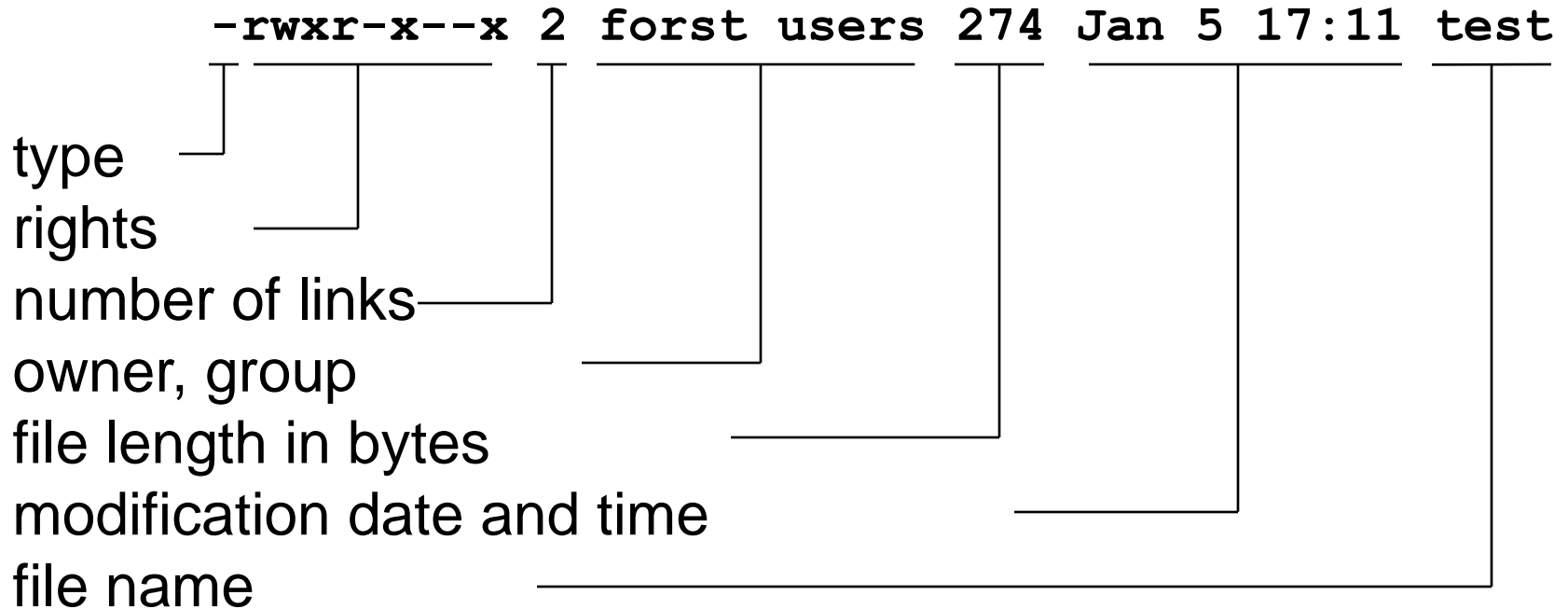
# Regular expressions

Way how to define strings in many utilities. Metachars:

- `.` ... any character
- `[list]`, `[^list]` ... any char from the list, or list complement  
e.g.: `[a-zA-Z0-9_]`, `[^ ]`, `[]^-]`
- `[[:trída:]]` ... any character from the class  
e.g.: `[[:alnum:]]`, `[[:xdigit:]]`
- `^`, `$` ... start, or end of line (used on start or end of regexp)
- `\c` ... metachar used as regular char (e.g.: `\.` means dot)
- `exp*` ... any number of occurrence of the last subexpression  
e.g.: `a*`, `[0-9][0-9]*`



# ls command



options: long (**l**), 1 column (**l**), include hidden (**aA**),  
(don't) print groups (**g**), sort by time (**t**), reverse sort  
(**r**), flag file type (**F**), recursive (**R**), don't follow dirs (**d**),  
follow links (**L**)

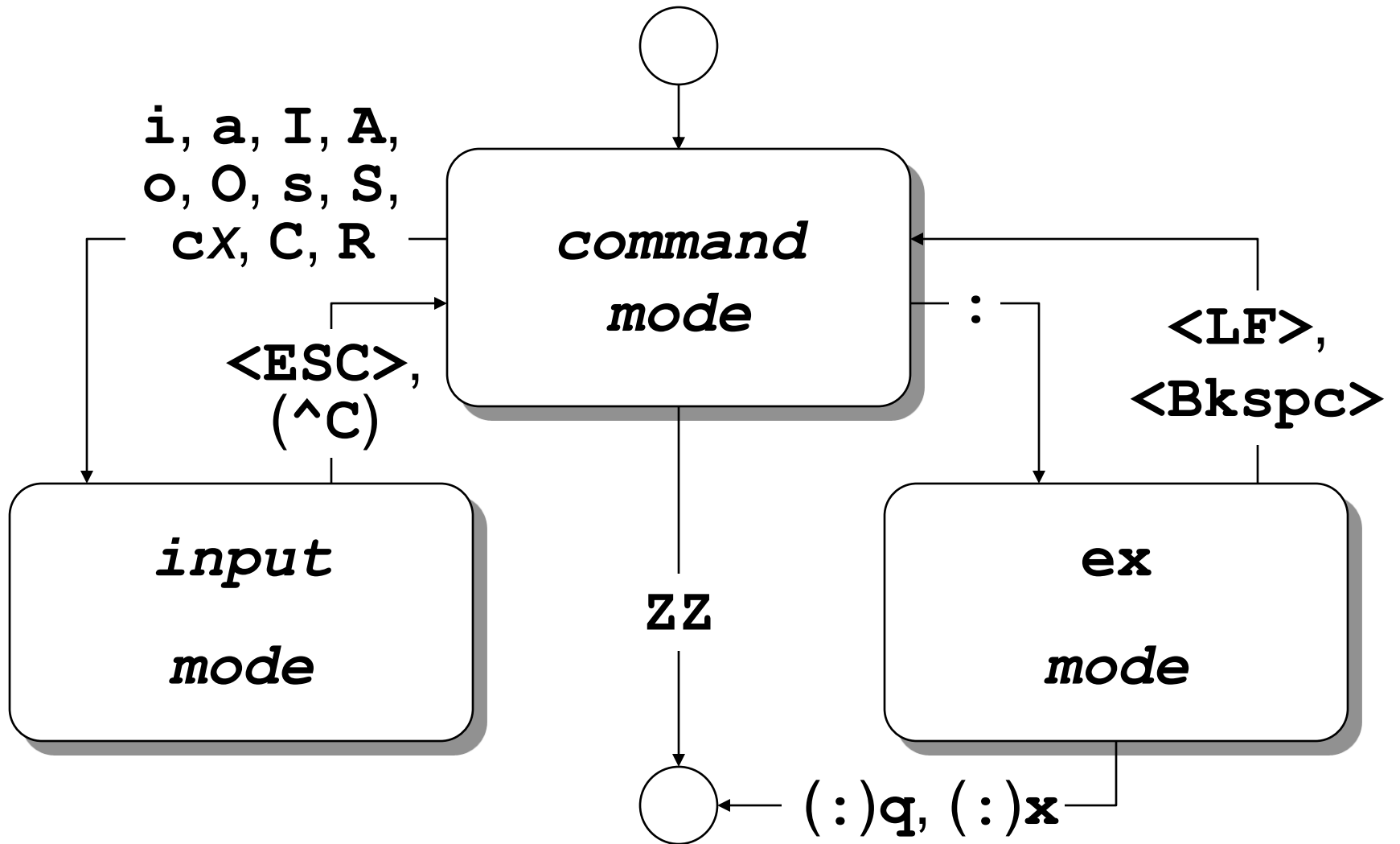
# General commands

- copy file: **cp** [-prR]
- move (rename) file: **mv**
- remove file: **rm** [-rfi]
- change date + time: **touch** [{ -t`time` | -r`file` }]
- change current directory: **cd**
- print working directory path: **pwd** [-P]
- make directory: **mkdir** [-p] [-m`mode`]
- remove directory: **rmdir**
  
- no undelete command!

# Work with text files

- output (concatenate) files: **cat** [*files*]
- file beginning output: **head** [-n *n*] [*files*]
- file end output: **tail** [{-n|-c} [+]*n*] [-f] [*files*]
- count bytes, words and/or lines: **wc** [-cwl]
- duplication to output and file: **tee** [-a] *file*
- cutting parts of lines (cannot change order of parts):  
**cut** [-s] { -clist | -flist -dchar } [*files*]
- pasting “columns” of files; pasting all lines of one file:  
**paste** [[ -s ] -dchars ] [*files*]
- character conversion:  
**tr** [-c ds] *table1* [*table2*] př.: `tr 'A-Z\n' 'a-z:'`

# Modes of vi



# Essential commands of **vi**

- **vi** *file* ... editor call
- **i** ... text inserting mode
- *text being inserted*
- **<ESC>** ... finishing input mode
- **h, j, k, l** ... cursor movements
- **/pattern** ... string pattern searching
- **x, dd** ... deleting a char, a line
- **A** ... appending to the end of line
- **J** ... joining lines
- **ZZ, :x** ... closing editor
- **:q!** ... cancelling editor