
Sistemas Operativos (MIEIC @ FEUP)

The Unix way... (2)

- General principles (2)

- Unix & Linux (4)

- Programming specifics' *miscellanea*: (6)

- To always remember (8)

The Unix way...

General principles^{1 2}

- small is beautiful (or KISS)
 - simple but well-done programs
- chaining of (independent) programs
 - use of text streams
 - standard input, output & standard error
 - text format: ease of interoperability & human readability
- all I/O devices handled the same way (as "files")
- if it works well, reuse it; otherwise, rebuild it!
- *continues...*

¹ not necessarily ordered by importance

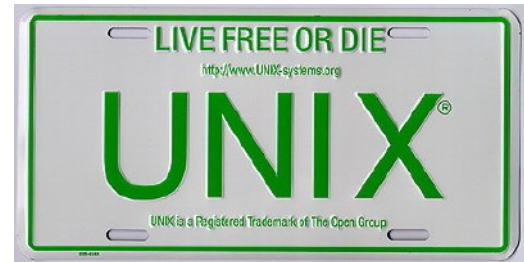
² many apply to both user and programming interfaces

...General principles

- ok to user interaction, but provide for otherwise
 - batch commands (administration!)
- documentation at hand: man pages!
 - `shell> man man`
- no news is good news

Unix & Linux

- many Unix"es", many proprietary
 - proprietary software *versus* free software
 - enters: Minix, Linux and others...
 - Linux distros --> Ubuntu @ FEUP
- user interaction (command interpreter)
 - (text) command line interface (CLI)
 - *shell*
 - also program interpreter: shell scripts¹
 - graphical user interface (GUI)
 - X Window System (XWindows, X11)
 - interconnecting different machines
 - upper layer: KDE Plasma, GNOME Shell, Aqua...
 - *continues...*



¹ system administration!

...*Unix & Linux*

- Unix® ¹ standards
 - Single UNIX Specification
 - POSIX.1 ² - Application Programming Interface to Operating System
 - for portable C source code programs
 - POSIX.2 – Command interpreter and Utility programs
 - many specifications (some extending others), confusing history
 - XPG4 ³, XSI ⁴ ...
- shell> man standards
 - CONFORMING TO manpages' section

1 registered trademark of The Open Group

2 Portable Operating System Interface

3 X/Open Portability Guide, Issue 4

4 X/Open System Interfaces

Programming specifics' *miscellanea*:

- file
 - mostly: regular file
 - persistent storage of data
 - unstructured sequence of bytes
 - remaining: "special" files
 - different characteristics
 - directory, device, symbolic link, ...
 - special files
- (I/O) devices are (special) files
 - similar usage to regular files (normally!)
 - `shell> ls -l /dev`
- *continues...*

...Programming specifics...

- file usage
 - open ; *use* ; close
 - *use*: access (operation upon) through "reference"
 - reference: descriptor, handle...
 - access: read, write, change metadata...
 - position indicator
 - automatic advance with read / write
 - no EOF¹ byte in file!
- *continues...*

1 End Of File

...Programming specifics

- Application Programming Interface¹ (to operating system)
 - system calls vs Standard C Library functions
 - `open()` - `fopen()` ; `read()` - `fread()` ; ...
 - which to choose?
 - portability of source code, *buffering*, ...
 - (almost) all API function calls may fail!
 - always verify RETURN VALUE
 - (usually) in failure: `-1` or `NULL ((void *) 0)`
 - act depending on ERRORS
 - `perror()`, `assert()`, ...

To always remember

- if in trouble: RTFM !

¹ API