Introducción a la Línea de Comandos de UNIX/Linux

Benjamin Gutierrez Garcia

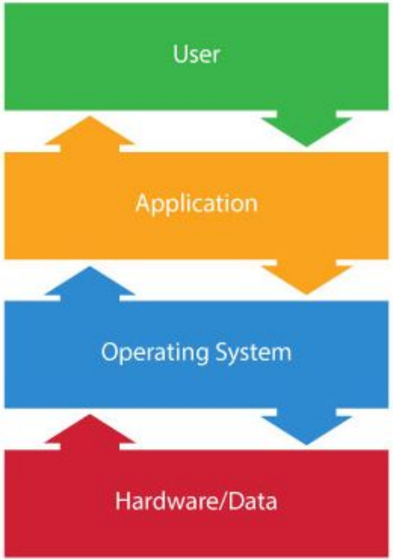
Seminario de Metodos Analiticos en la Empresa

Data carpentry and plumbing

- Command line interface belongs to this type of chores
- Remember the 80/20?
- Get your hands dirty!!!!

Unix/Linux Basics

- Why Unix like operating systems?
  - What Is an Operating System?
    - Interface between Users and the Hardware
    - Take care of Storage Management
    - take care of I/O device management
- Characteristics
  - Multi-user & Multi-tasking
  - Much more stable and reliable than alternatives
  - Leading Distributed OS architecture/platform (datacenter, cloud, containers)
  - Open Source: Kernel, applications, compilers, interpreters, etc.
  - Most applications (including Data analysis tools) ported
  - Less Resource Intensive - (the old family computer that can barely run Windows is more than sufficient to run the latest version of Linux)

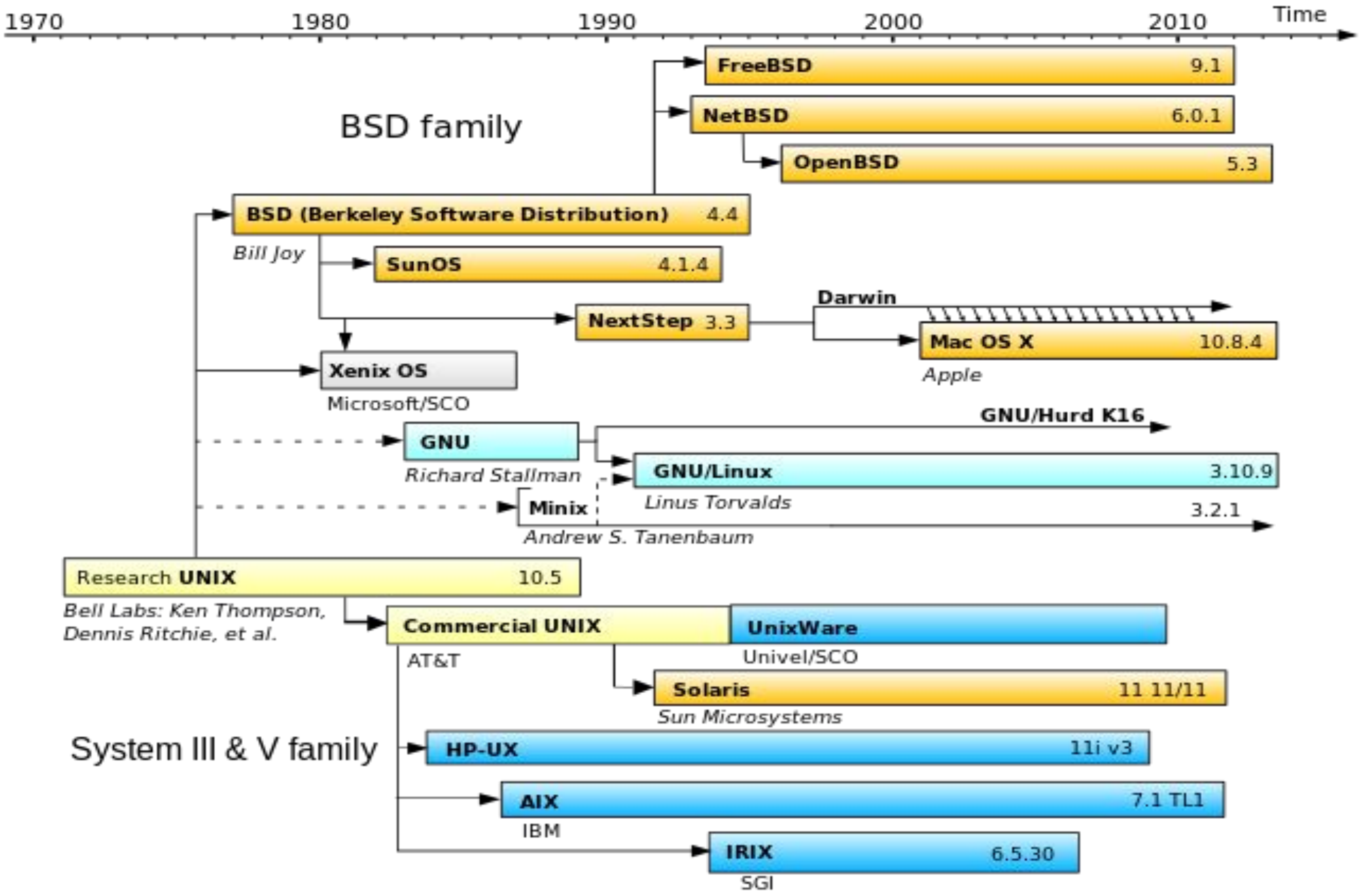


Unix/Linux Basics

- UNIX operating system was born in the late 1960s.
  - Originally began as a one man project led by Ken Thompson of AT&T Bell Labs
  - Has since grown to become the most widely used operating system.
- Over 30 Years Old - It's popularity and use is still high.
- Over these years, many variations have spawned off and many have died off
- It has endured the test of time.
- Windows at best is half as old (Windows 1.0 was released in the mid 80s, but it was not stable or very complete until the 3.x family, which was released in the early 90s).
- More on Unix history: https://www.levenez.com/unix/
- The Portable Operating System Interface (POSIX) is a family of standards specified by the IEEE Computer Society fto ensure compatibility between operating systems and porting apps.

Unix Taxonomy:

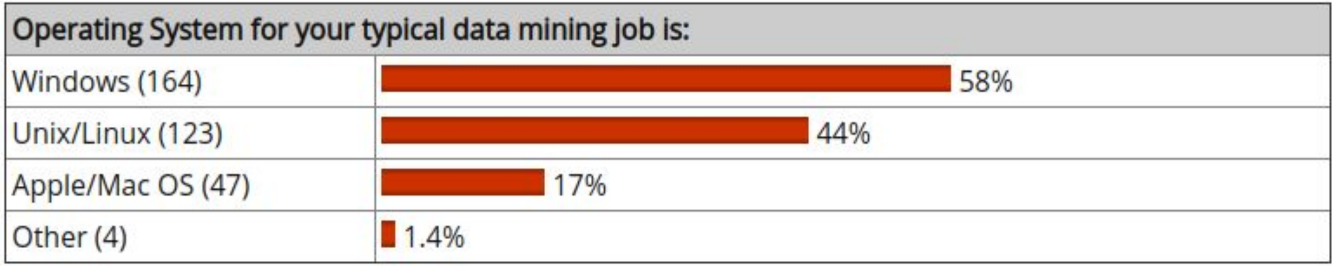
- **Linux Flavors:** RedHat, Fedora, Debian, Mandrake, Ubuntu, CoreOS, CentOS, SuSE, Mint, ClearLinux
- **Cousins**: MacOSX, FreeBSD



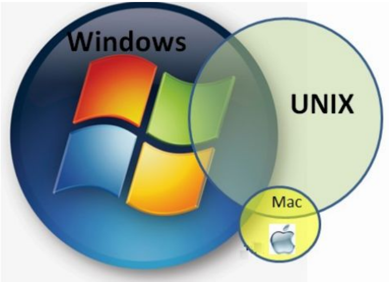
Why do we care about Linux/Unix in Data Science?

Kdnuggets Poll 2015 - The Average number of operating systems used was only 1.2, and 67% of users only used one OS.

Comparing to 2010 poll, Windows share went down from 67% to 58%, while Apple/Mac share more than doubled, from 7.6% to 17%. Unix/Linux share also increased, from 28% to 44%.

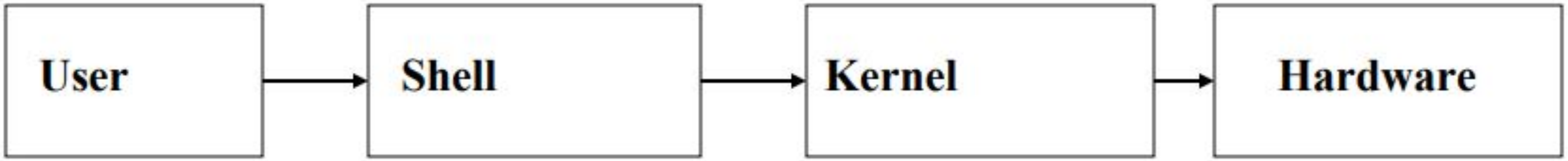


Operating System for your typical data mining job is:	
Windows (164)	58%
Unix/Linux (123)	44%
Apple/Mac OS (47)	17%
Other (4)	1.4%



Unix Architecture

- User space ⟺ Kernel Space



User → Shell → Kernel → Hardware

Basic Commands of a shell/terminal session

- ls, ls -lah , ls -lahrt
- clear
- pwd
- cd   (TAB completion)
- cp, cp -a, cp -v
- mv
- rm, rm -f , rm -i
- man
- mkdir, mkdir -p
- rmdir, rmdir -p
- exit or + will terminate the login session

Basic Commands of a shell/terminal session

- touch
- echo
- cat
- more
- less
- tail
- wc

Basic Commands of a shell/terminal session

- permissions: rwx rwx rwx
- chown, chown -R
- chgrp
- date
- cal
- history
- uname -a
- sudo, su
- Package managers

What is a file System? Data+Meta Data

- Collection of control structures and Data blocks that occupy the space defined by a partition and allow for the storage and management of data. du, df
- tree
- find
- Hard Link: ls -i file
- Soft link ln -s
- Types of Data Files: **Ordinary files**, **Directory files**, **Device files, Symbolic Link**.
- Absolute pathname (start from the /-directory): Eg: /export/home/user1/file1
- Relative pathname (start from the current directory) ./test1 (. = current directory) ../team03/.profile (.. = parent directory)
- tree -d -L 1

Basic Commands of a shell/terminal session

- vi
- nano
- pico
- emacs, emacs -nw
- Types of shells: bash,cshrc, zsh, tcsh,
- echo $0
- Re-enter commands, history and number e.g. !234, search CTRL-R
- alias
- export
- Enviro variables, e.g. $PATH, $TERM
- Setting vars: PATH=$PATH:/usr/ucb

Basic Commands of a shell/terminal session

- Standard I/O Redirection
  - Standard Input File (0)
  - Keyboard Standard Output File (1)
  - Monitor Standard Error File (2)
  - Monitor operators: standard input redirection 0< or <
  - standard output redirection 1> or >
  - standard Error redirection 2>
  - ls > file.txt, >>, program < input_file.txt
  - cat file1 file3 2>err
  - ls | tee ls.save | wc –l
  - command &> out   stderr and stdout both to a file
- Pipes |

More Commands

- Find (name pattern, size, type,-mtime)
- file
- grep (-v, case insensitive, etc)
- tar
- awk
- sed

Exercises

- http://www.matem.unam.mx/~benjamin/practica.pdf