



Les bases du système UNIX

Mise à niveau IATIC3

Une famille de systèmes d'exploitation multitâche et multi-utilisateur dérivé du Unix d'origine créé par AT&T à partir de 1970.

Un superviseur (le shell) et des utilitaires actionnables par ligne de commande.

\$ whois philippecolzy@need-data.com

- **Années 90 - Fac d'Orsay (informatique industrielle)**
- **Analystes prog à DSI**
- **25 ans dans la data 'marketing' et ses algo sous toutes ses formes**

Depuis 2011 - Entrepreneur codeur

Tech leader sur apiXpress (open source)

Activité de consulting :

- Architecture web (DAO Decentralized Anonymous Organisation)
- Utilisateurs de : linux, node.js, express, git et techno du web

\$ Whois (are) you

- **Qui sait :**
 - ce que veut dire O.S. ?
 - Ce qu'est la RAM, le CPU ?
 - Le point commun entre fat, ntfs, ext4, reiserfs
- **Qui connaît wsl sous windows 10 ?**
- **Qui utilise linux sur son pc perso ?**

\$ Whois (are) you

Des idées d'avenir proche ?

- **Les systèmes / sécurité IT**
- **Le dev de solutions (data)**
- **La recherche informatique**
- **La gestion de projet IT**

\$ Whois (are) you

Votre objectif plus long terme ?

- **Une équipe IT dans une grosse entreprise 'utilisatrice'**
- **Du consulting dans une ENS**
- **Entrepreneur**
- **Elever des chèvres à la campagne**

Plan du cours en 4 x 3h30

- **OS & UNIX**
- **Prise en main d'un système Unix (term)**
- **Système de fichiers Unix**
- **Un editeur de texte (vi)**
- **Expressions Régulières (regex)**
- **Processus (proc)**
- **Langages de commandes (shell)**
- **Le monde unix (X, net, ssh, ftp, http, mail,...)**

Les TD 8 sur 4x 4h30

- Installation linux
- TD1 à 8, toutes les connaissances utiles se trouvent dans ce cours.
- Tous les TD seront stockés dans des fichiers personnels. A la fin, il faudra rendre un .tgz par étudiant de l'ensemble des TD.
- Une note sera attribuée à chaque élève

- Pour recuperer les td :

```
$ git clone https://gitlab.com/franck.talbart/lectures\_isty.git
```

```
lectures_isty/IATIC3/UNIX/tds/
```

les cours sont dans

```
lectures_isty/IATIC3/UNIX/cours/Cours_unix_pcolzy_CM[1-4]_3h.pdf
```

OS & UNIX

Plan du cours en 4 x 3h30

- OS & UNIX
- Prise en main d'un système Unix (term)
- Système de fichiers Unix
- Un éditeur de texte (vi)
- Expressions Régulières (regex)
- Processus (proc)
- Langages de commandes (shell)
- Le monde unix (X, net, ssh, ftp, http, mail,...)

- **Un OS**
- **Architecture d'Unix**
- **Une grande famille**

Un OS

OS & UNIX

Operating System / Système d'exploitation

Permet à un utilisateur d'utiliser des logiciels

Les logiciels s'appuient sur un OS

L'OS communique avec l'ordinateur pour

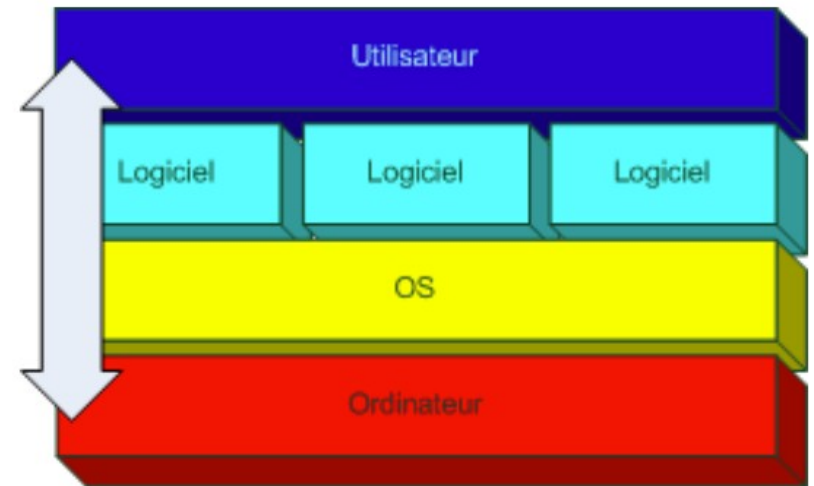
Piloter ses composants :

disques durs, mémoire vive, cpu, ...

Toutes les machines ont un OS pour

optimiser et faciliter l'usage des ressources

MS-DOS, VMS, Unix, Multics, MS-windows, android, macOS,...



Architecture d'Unix

OS & UNIX

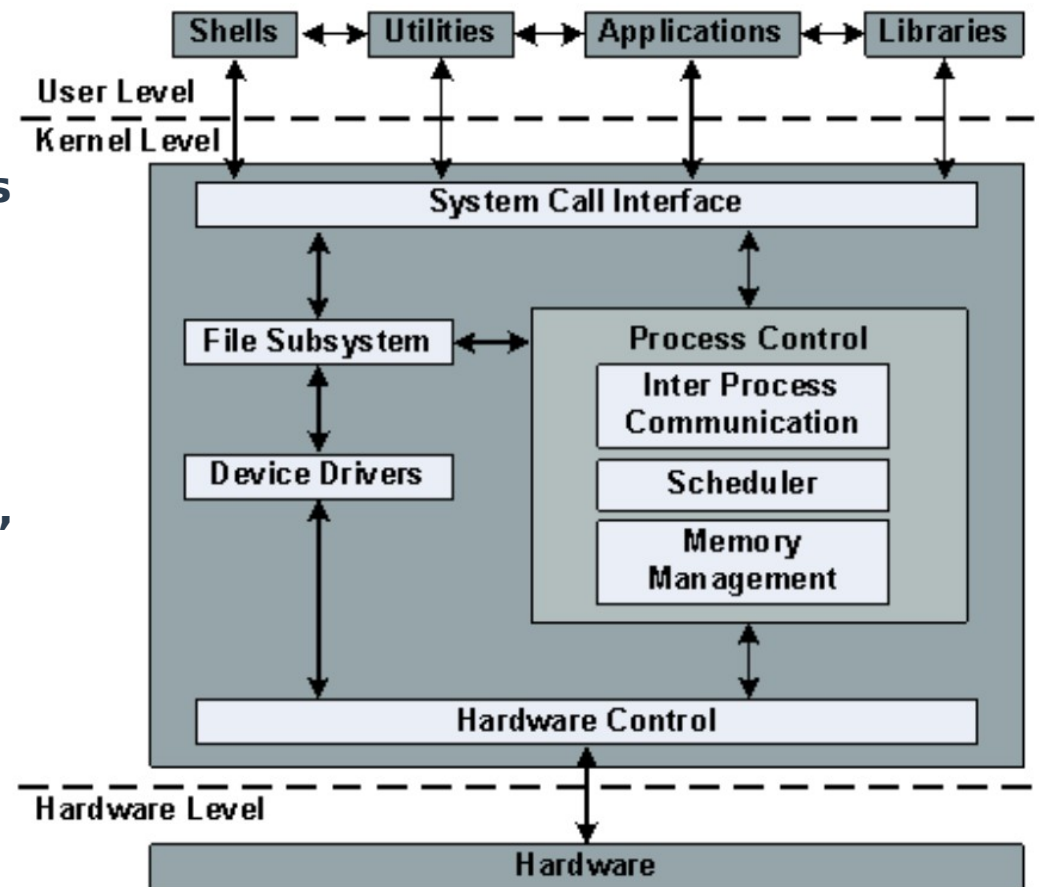
UNIX = Noyau (kernel/device) + Interpreteur + Utilitaires (utilities)

Usage : **multi user / multi tâches**

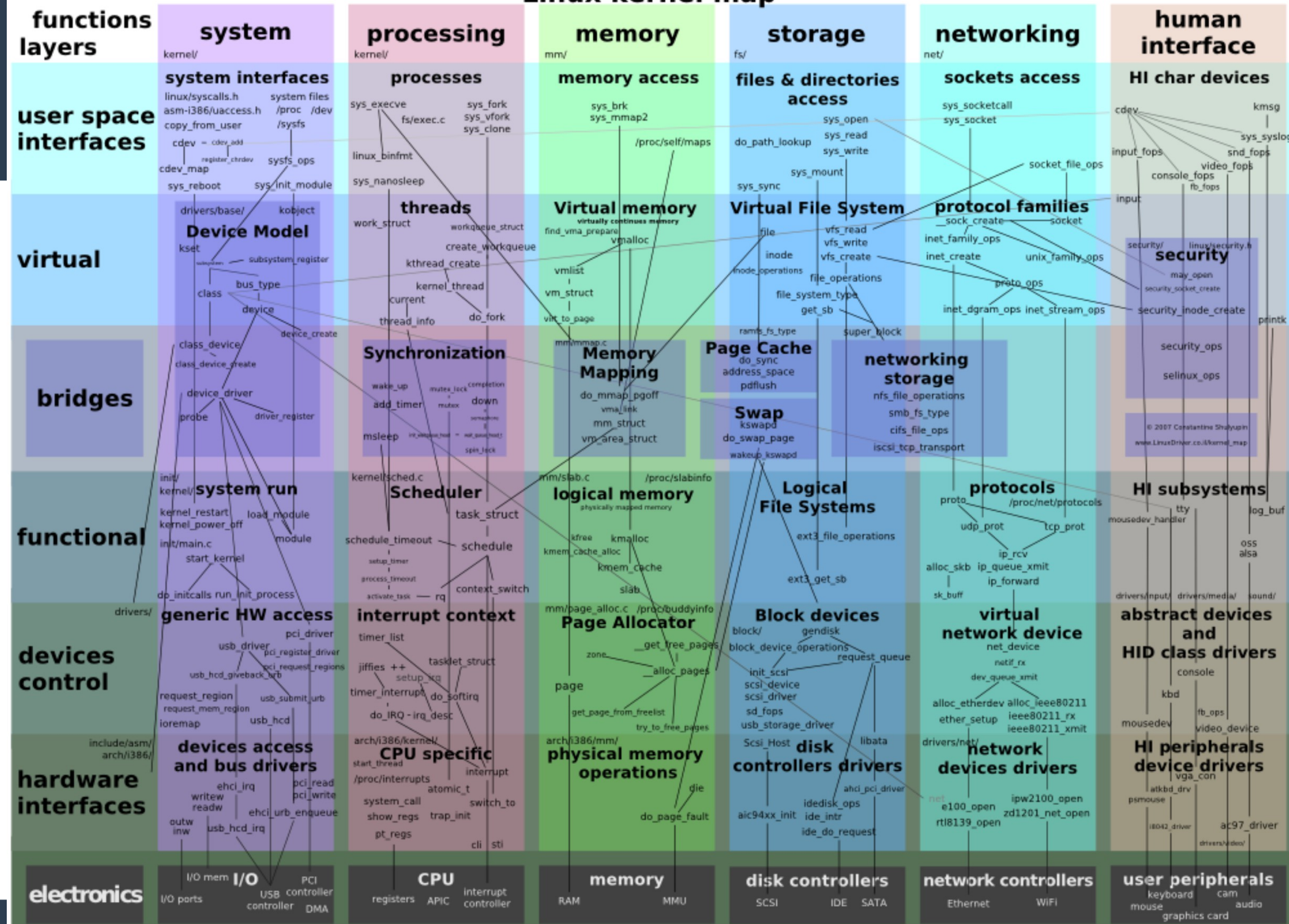
User interactive interface:
terminal / X-windows / applications

Interpreteur de commande :
communication inter processus
entrée/sortie

device drivers : gestion des
ressources (stockage des données,
interface réseaux, wifi, audio,
video, imprimantes...)



Linux kernel map



Architecture d'Unix

OS & UNIX

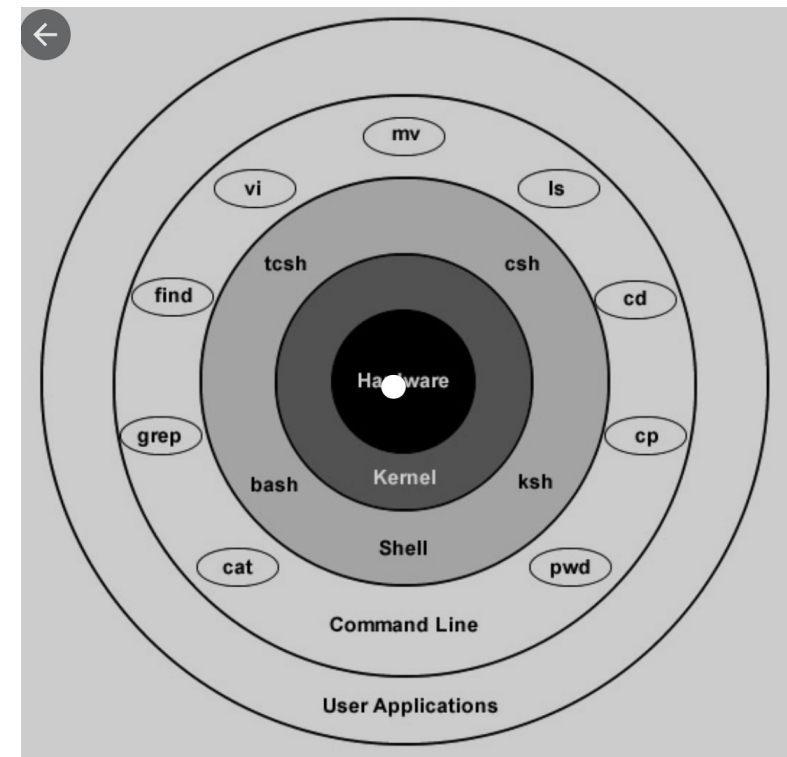
UNIX = Noyau (kernel/device) est basé sur 2 concepts :

Un système de fichier hiérarchisé :

- fichier E/S entre périphériques, fichiers et processus.
- Organisation de la mémoire secondaire (disque)

Gestion de processus

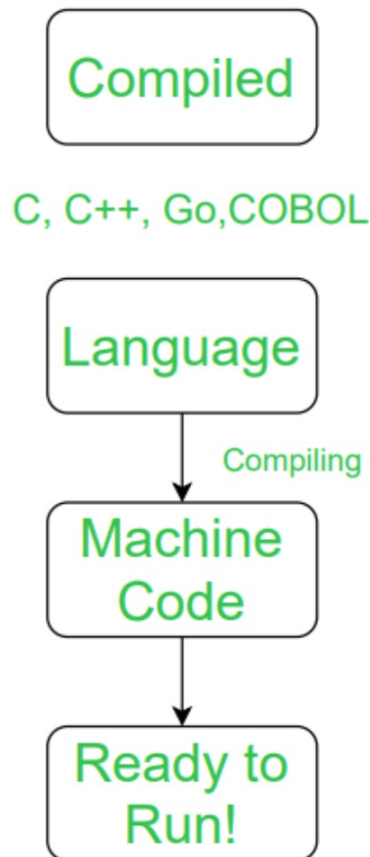
- Processus = exécution d'un programme
- Execution multi-tâches sous contrôle synchrone et asynchrone (gestion d'événements)



Architecture d'Unix

OS & UNIX

Des utilitaires développés en C pour ne pas faire de C



C : un langage compilé, proche de la machine, il nécessite un pré-traitement pour le rendre exécutable par la machine

Source (.h) => prétraitement (gcc *.h) => assemblé adapté à la machine (makefile), installation (un fichier bin exécutable)

- Avantage : performance et optimisation du code
- Inconvénient : faire bon du 1^{er} coup sinon processus itératif très long

Le shell : un langage interprété, permet d'assembler des composants compilés, entre eux avec un système entrée/sortie pour réaliser des tâches complexes

- Avantage : rapidité de creation d'un proccess adaptable facilement dans le temps
- Inconvénient : forcément moins performant qu'un developpement spécifique

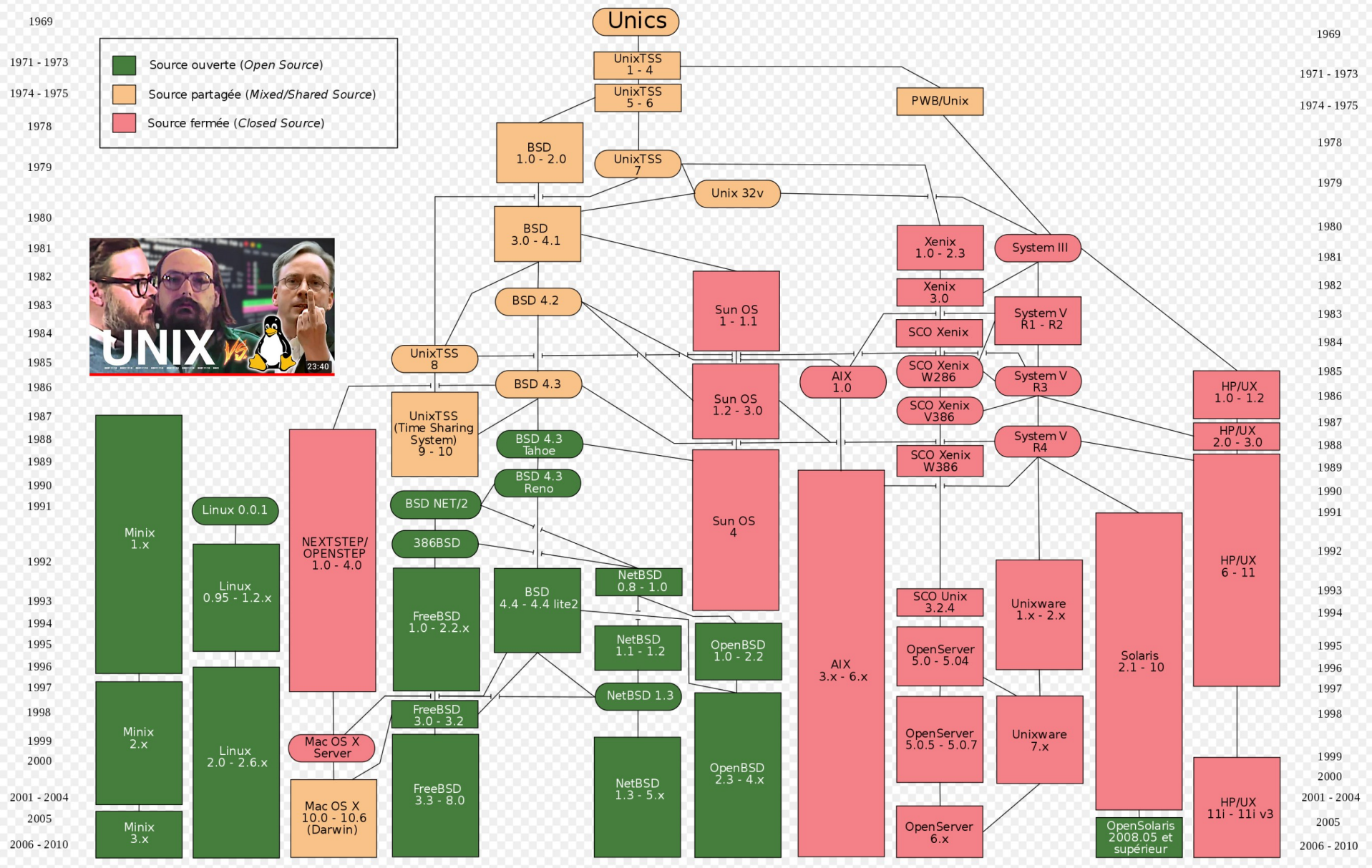
La force d'Unix : l'application du principe KISS (Keep It Simple , Stupid)

Un ensemble de composants simples, bien documenté (man) et respectant des standards d'entrées sorties.

Pour les versions open source => selection naturelle des plus performants



Une grande famille



Prise en main d'un unix

Plan du cours en 4 x 3h30

- OS & UNIX
- Prise en main d'un système Unix (term))
- Système de fichiers Unix
- Un editeur de texte (vi)
- Expressions Régulières (regex)
- Processus (proc)
- Langages de commandes (shell)
- Le monde unix (X, net, ssh, ftp, http, mail,...)

Prise en main d'un Unix

Prise en main d'un Unix

- **Le prompt et la session**
- **Les premiers pas sans se perdre**
- **Naviguer dans les fichiers**
- **Editeur de texte vi**

Le prompt \$ et la session

Prise en main d'un Unix

machine linux : ON

Lancement automatique du bios :

- Initialise le système, les drivers, la ram, le disque, ...
- Attente de session OS (Ubuntu 18.04.1 LTS)
Name(ubuntuBionic) sortie terminal (tty3)

User login

- Par défaut un compte root est installé
- Souvent un compte user ayant un droit root (sudoer) est aussi créé

Le prompt : phil@cloudnd1 : ~/workspace\$

Personnalisable et permet de savoir où on se trouve.

login(phil)@name(nom de la machine)

phil@192.168.1.120 indique qu'on est sur une machine du réseau locale

```
Ubuntu 18.04.1 LTS ubuntuBionic tty3
ubuntuBionic login: _
```

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-125-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Sep  6 10:33:52 CEST 2022

System load:          0.0
Usage of /:            5.1% of 457.38GB
Memory usage:         5%
Swap usage:           0%
Temperature:          50.0 C
Processes:            218
Users logged in:      0
IPv4 address for bond0: 192.168.1.96
IPv4 address for lxdbr0: 10.31.251.1
IPv6 address for lxdbr0: fd42:9492:ef0a:f426::1

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

  https://ubuntu.com/blog/microk8s-memory-optimisation

30 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Aug 30 10:43:40 2022 from 192.168.1.73
phil@cloudnd1:~$
```

Premier pas sans se perdre

Prise en main d'un Unix

Les bases pour rentrer dans un système unix

<https://cheatography.com>

\$ <commande> [< param > ...] ; <commande> [] <cr>

==

<un programme executable> [] optionnel un ensemble de paramètres ; une deuxième commande ; ... Enter Exécute les commandes

Exemples :

```
$ man zcat <CR>
```

execute le bin man avec le parametre zcat

```
$ cp toto titi <CR>
```

copy le fichier toto vers titi

Pour maitriser le monde Unix il faut pratiquer et apprendre :

- Les mécanismes de communication du système (processus E/S)
- Des commandes de bases
- Organiser des commandes dans un script

Conseil pour débuter :

- Avoir sous la main 1 page A4 recto/verso « cheatsheet »
- Approfondir au fil du temps les options d'une commande avec \$ man commande
- Google

Premier pas, man

La commande man \$ man man

\$ man -k expression

Pour une mise en main : faire un man par commande de base : ls cd mkdir rmdir cp mv rm find grep cat

Exemples :

\$ man ls

une vague idée :

\$ man -k ^ls => tout ce qui commence par ls

\$ man -k ls\$ => tout ce qui fini par ls

\$ man -k date => tout ce qui contient date

```
phil@phildata:~$ man -k date
update-alternatives (1) - Maintenance des liens symboliques déterminant les noms par défaut de certaines commandes
update-passwd (8) - met à jour /etc/passwd, /etc/shadow et /etc/group de façon sécurisée
updatedb.conf (5) - a configuration file for updatedb(8)
apport-bug (1) - file a bug report using Apport, or update an existing report
apport-collect (1) - file a bug report using Apport, or update an existing report
archive_entry_perms (3) - functions for manipulating ownership and permissions in archive entry descriptions
archive_entry_paths (3) - functions for manipulating path names in archive entry descriptions
asctime (3) - transform date and time to broken-down time or ASCII
asctime_r (3) - transform date and time to broken-down time or ASCII
autoreconf (1) - Update generated configuration files
autoupdate (1) - Update a configure.in to a newer Autoconf
bundle-lock2.7 (1) - Creates / Updates a lockfile without installing
bundle-outdated2.7 (1) - List installed gems with newer versions available
bundle-update2.7 (1) - Update your gems to the latest available versions
cal (1) - displays a calendar and the date of Easter
catman (8) - create or update the pre-formatted manual pages
chgrp (8) - update group passwords in batch mode
chpasswd (8) - update passwords in batch mode
config.sub (1) - validate and canonicalize a configuration triplet
```

```
LS(1)
NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory or the
    files specified).

Mandatory arguments to long options are mandatory.

-a, --all
    do not ignore entries starting with .

-A, --almost-all
    do not list implied . and ..
```

```
MAN(1)
MAN(1)

NOM
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [options de whatis] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section, following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

    Le tableau ci-dessous indique le numéro des sections de man :

    1 Programmes exécutables ou commandes de l'interpréteur
    2 Appels système (fonctions fournies par le noyau)
    3 Appels de bibliothèque (fonctions fournies par les bibliothèques)
    4 Fichiers spéciaux (situés généralement dans /dev)
    5 File formats and conventions, e.g. /etc/passwd
    6 Jeux
    7 Divers (y compris les macropaquets et les conventions)
    8 Commandes de gestion du système (généralement réservées)
    9 Sous-programmes du noyau [hors standard]

    Une page de manuel est constituée de plusieurs sections.

    Parmi les noms de section conventionnels se trouvent ENVIRONNEMENT, FICHIERS, VERSIONS, CONFORMITÉ, NOTES, BOGUES, EXEMPLE, AUTEURS et VOIR AUSSI.

    Les conventions suivantes s'appliquent à la section SYNOPSIS :

    texte gras      à taper exactement comme indiqué ;
    texte italique   à remplacer par l'argument approprié ;
    [-abc]           tous les arguments entre [ ] sont facultatifs ;
    -a|-b            les options séparées par | ne peuvent pas être combinées ;
    argument ...     argument peut être répété ;
    [expression] ... toute l'expression située à l'intérieur des crochets est optionnelle.

    Le rendu exact dépend du dispositif d'affichage. Par exemple, dans un terminal, on verra typiquement le soulignement ou la coloration du texte à la place.

    Le synopsis de la commande ou de la fonction est un modèle à copier et à coller. Il est demandé d'illustrer plusieurs utilisations exclusives comme dans la partie SYNOPSIS de cette page de manuel.

EXEMPLES
    man ls
    Affiche la page de manuel de l'élément (du programme) ls.

    man man.7
    Display the manual page for macro package man from section 7.

    man 'man(Z)'
    Display the manual page for macro package man from section Z. Note that the parentheses must normally be quoted to protect them from being interpreted as part of a shell command.

    man -a intro
    Affiche, successivement, toutes les pages de manuel de la section intro.

    Manual page man(1) line 1 (press h for help or q to quit)
```

Naviguer dans les fichiers

Prise en main d'un Unix

Unix tout est fichier : la mémoire, les textes, les binaires, ...

Les commandes de base à maîtriser:

ls cd mkdir rmdir cp mv rm find grep cat

Les concepts à comprendre :

- **Arborescence de fichiers**

un repertoire est un « noeud » qui contient des fichiers

Les Codes couleurs differents en fonction des propriétés

\$ mkdir toto (créer un repertoire toto)

\$ rmdir toto (supprime le repertoire toto si existe et si le droit)

- **Path absolu relatif (Chemin absolu et relatif)**

\$ cd ~ => se retrouve dans *home* du user (/home/martin)

\$ cd / => retourne à la racine

\$ cd .. => remonte d'un cran

\$ cd ../../.. => remonte de 3 crans

\$ cd ../home => remonte au home si je suis dans un /martin ou /lisa

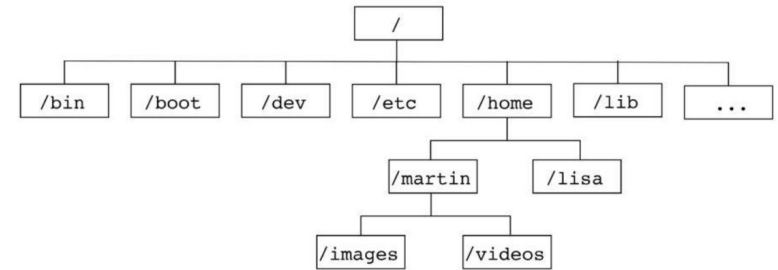
\$ pwd => indique l'endroit où on se trouve

- **Quelques pièges**

Les fichiers/repertoires qui commence par un . sont cachés par défaut

\$ ls -alhS

permet de lister tous les fichiers de voir leur droits trié par taille décroissantes (S) avec un format humain (h)



```
phil@cloudnd1:/$ cd /
phil@cloudnd1:/$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found
ia mnt opt proc root run sbin snap srv swap.img sys tmp usr
phil@cloudnd1:/$ cd ~/workspace/apixpress/
phil@cloudnd1:~/workspace/apixpress$ ls
README.md data node_modules tmp
apixpress.js incron detection.sh package.json utils
config.js middlewares routes xipacoin.sh
config.jsold models setup yarn.lock
phil@cloudnd1:~/workspace/apixpress$ cat incron detection.sh
#!/usr/bin/env bash

# Detection changes in ../apixpress/data/domain
# rerun maildigitcreator if
# params variables $@ $# $% $& dev|prod
# folder / file / textaction / idaction / mode dev or prod
# IN_CLOSE_WRITE = idaction 8

FOLDER=$1
FILE=$2
```


Editeur de texte vi (vim)

Prise en main d'un Unix



Créer, visualiser, modifier un fichier texte : vi (ou vim)

- **Editeur de texte dans un terminal**

Indispensable de le maîtriser en production (rarement window)

```
$ vim testphil.txt
```

Commande de base :

<escape> i passe en mode INSERT

<escape> : passe en mode de commande

- **Les commandes de base**

:q quitte l'éditeur

:q ! quitte l'éditeur malgré des modifications non sauvegardées

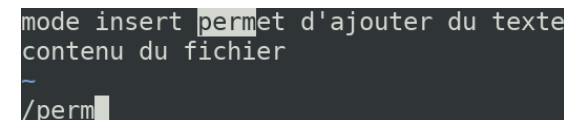
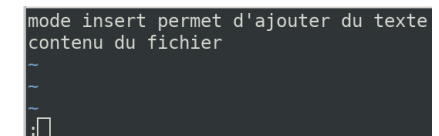
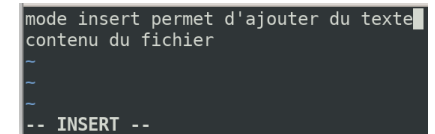
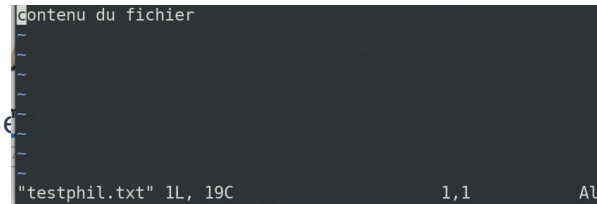
:w écrit (write) le contenu dans le fichier

:wq écrit (write) et quitte l'éditeur

:d ou dd supprime la ligne du curseur

/pattern cherche dans le texte un pattern puis taper n pour la recherche suivante ou N pour la précédente

<https://cheatography.com/> rechercher VIM



Comprehensive VIM Cheat Sheet Cheat Sheet by [typo209](#)

Comprehensive VIM Cheat Sheet information sourced from: <http://www.tuxfiles.org/linuxhelp/vimcheat.htm>

Working With Files	Editing Blocks of Text	Inserting and Overwriting Text
o Open a new file. You can use the Tab key for automatic file name completion, just like at the shell command prompt.	~ Change the case of characters. This works both in visual and command mode. In visual mode, change the case of highlighted characters. In command mode, change the case of the character under cursor.	I Insert before cursor.
w Save changes to a file. If you don't specify a file name, Vim saves as the file name you were editing. For saving the file under a different name, specify the file name.	> [V] Shift right (indent).	I Insert at the start of the current line.
:q Quit Vim. If you have unsaved changes, Vim refuses to exit.	< [V] Shift left (de-indent).	A Append after cursor.
:wq Write file without saving changes.	c [V] Change the highlighted text.	O Append to the end of the current line.
:wq Exit the file and exit.	y [V] Yank the highlighted text. In Windows terms, "copy the selected text to clipboard."	O Open a new line below and insert.
x Almost the same as wq , write the file and exit if you've made changes to the file. If you haven't made any changes to the file, Vim exits without writing the file.	d [V] Delete the highlighted text. In Windows terms, "cut the selected text to clipboard."	C Change the rest of the current line.
	y or y Yank the current line. You don't need to highlight it first.	r Overwrite one character. After overwriting the single character, go back to command mode.
	D Delete the current line. Again, you don't need to highlight it first.	R Enter insert mode but replace characters rather than inserting.
	p Put the text you yanked or deleted. In Windows terms, "paste the contents of the clipboard." Put characters after the cursor. Put lines below the current line.	ESC Exit insert/overwrite mode and go back to command mode.
	P Put characters before the cursor. Put lines above the current line.	
	Note: the Vim commands marked with (V) work in visual mode, when you've selected some text. The other commands work in the command mode, when you haven't selected any text.	
Moving Around in the File	Deleting Text	Search
j or Up Move the cursor up one line.	x Delete characters under the cursor.	/pattern Search the file for pattern.
k or Down Move one line.	X Delete characters before the cursor.	n Scan for next search match in the same direction.
h or Left Left one character.	d or d Delete the current line.	N Scan for next search match but opposite direction.
l or Right Right one character.		
e To the end of a word.		
E To the end of a whitespace-delimited word.		
b To the beginning of a word.		
B To the beginning of a whitespace-delimited word.		
o To the beginning of a line.		
A To the first non-whitespace character of a line.		
\$ To the end of a line.		
H To the first line of the screen.		
M To the middle line of the screen.		
L To the last line of the screen.		
:n Jump to line number <i>n</i> . For example, to jump to line 42, you'd type :42 .		
		Undo and Redo
		u Undo the last action.
		U Undo all the latest changes that were made to the current line.
		Ctrl+R Redo.
		Visual Mode
		v Start highlighting characters. Use the normal movement keys and commands to select text for highlighting.
		V Start highlighting lines.
		ESC Exit visual mode and return to command mode.