

Logiciels libres : introduction

Thierry Dumont

8 Novembre 2022

Quelques évènements, plus ou moins récents, relatés par la presse :

Vous n'avez pas de compte Facebook pour ne pas transmettre de données personnelles à ce réseau social ? **Pas de chance. La firme de Mark Zuckerberg va quand même récolter des données sur vous au travers de votre smartphone Android.** Beaucoup d'applications mobiles intègrent en effet le kit de développement de Facebook Ce kit ***transmet des données à Facebook.***

– Janvier 2019 –

Quelques évènements, plus ou moins récents, relatés par la presse :

(Huawei)... On craint que les services militaires et les services de renseignements chinois n'intègrent des « portes dérobées » logicielles dans l'équipement... qu'ils pourraient exploiter pour dégrader ou désactiver les réseaux sans fil étrangers en cas de crise.

– Janvier 2019 –

Quelques évènements, plus ou moins récents, relatés par la presse :

D'après Bloomberg, de nombreuses firmes informent les agences du renseignement, NSA, CIA et FBI inclus, **dès la découverte de failles de sécurité**, et ce **avant même** de les rendre publiques et de publier des correctifs.

– 2013 –

Quelques évènements, plus ou moins récents, relatés par la presse :

Stuxnet est un ver informatique découvert en 2010 qui aurait été conçu par la NSA pour s'attaquer aux centrifugeuses iraniennes d'enrichissement d'uranium. Le programme a été initié sous l'administration Bush et a continué sous l'administration Obama.

Spécifique au système Windows, il a été décrit par différents experts comme cyber arme, conçue pour attaquer une cible industrielle déterminée. Il s'agirait d'une première dans l'histoire.

Quelques évènements, plus ou moins récents, relatés par la presse :

iOS dispose d'une **porte dérobée** permettant d'effacer à distance une application. Android a aussi une **porte dérobée** encore plus puissante qui permet à Google, non seulement, d'**effacer à distance n'importe quelle application**, mais aussi d'**installer de force** n'importe quelle autre application.

Richard Stallman, 2019.

Quelques évènements, plus ou moins récents, relatés par la presse :

Apple and Samsung are being fined €10m and €5m respectively in Italy for the “planned obsolescence” of their smartphones.

An investigation launched in January by the nation's competition authority found that certain smartphone software updates had a negative effect on the performance of the devices.

The Guardian –2018–

Quelques évènements, plus ou moins récents, relatés par la presse :

Le Dieseldate (2015). VW indique que le logiciel a un fonctionnement qui détecte le roulage sur banc de test et s'adapte donc...

.....

VW reconnaît avoir installé un système qui rend inopérant certains éléments du système antipollution en marche normale.

Quelques évènements, plus ou moins récents, relatés par la presse :

Microsoft : Windows 10+

À chaque utilisateur est associé un identifiant unique, qui est utilisé pour marquer tous ses documents.

Pourquoi ?

Revenus de Google et Facebook.

- ▶ Google : publicité.
 - ▶ *Ads* : mots clés vendus aux enchères ; chaque click fait apparaître une publicité => revenus pour Google et pour l'annonceur.
Ciblage des publicités en fonction des recherches de l'internaute.
 - ▶ *Gmail* : affichage publicitaire en fonction du *contenu du mail*.
- ▶ Facebook : publicité ciblée (cliques d'utilisateurs).
Base de données de toutes les interactions.

- ▶ Comment cela est-il possible ? Comment les logiciels permettent-ils garder de tels **secrets** ?

- ▶ Comment cela est-il possible? Comment les logiciels permettent-ils garder de tels **secrets**?
- ▶ Un peu d'**histoire** pour comprendre.
Prétexte aussi à parler d'histoire de l'informatique.

- ▶ Comment cela est-il possible ? Comment les logiciels permettent-ils garder de tels **secrets** ?
- ▶ Un peu d'**histoire** pour comprendre.
- ▶ Le logiciel **libre**, pour s'affranchir de ces problèmes (et pour d'autres raisons!).

Logiciel ?

Logiciel, n.m.

Journal officiel du [22/09/2000](#).

Définition :

Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données.

Équivalent étranger :

software (en).

Lexique

Programme : Logiciel.

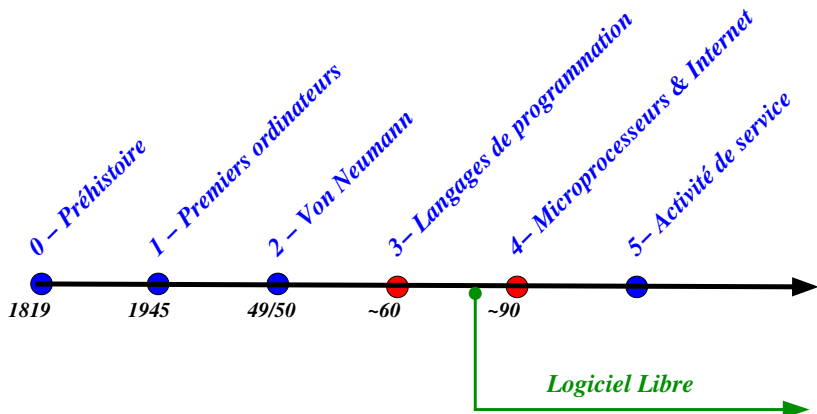
Programmeur : Développeur.

Petite histoire de l'informatique

Cheminement vers l'hégémonie du logiciel.

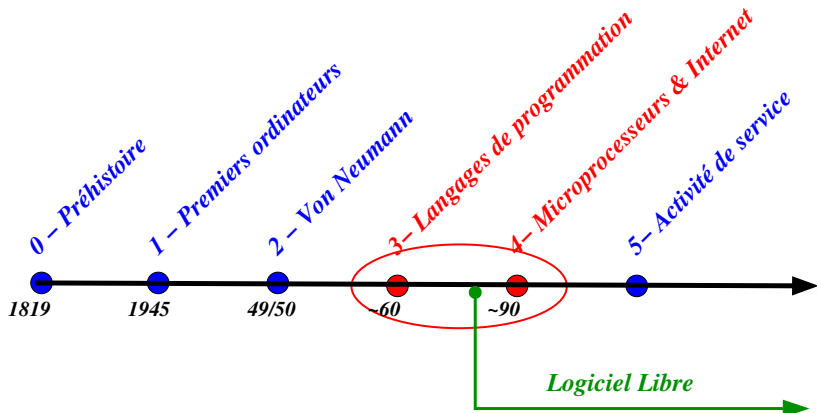
Petite histoire de l'informatique

Cheminement vers l'hégémonie du logiciel.



Petite histoire de l'informatique

Cheminement vers l'hégémonie du logiciel.

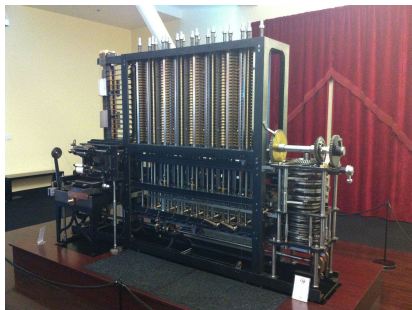


Étape zéro

1819–1944.

Étape **zéro** : préhistoire

- ▶ Les machines à différences de Babage (2) :
Très compliquée à réaliser avec des engrenages. Échec final.



Réalisation moderne : *Difference Engine no 2* (Computer History Museum).

Étape **ZÉRO** : préhistoire

La machine analytique :

- ▶ Programmes : cartes perforées.
- ▶ Mémoire : 1000 nombres de 40 chiffres décimaux.
- ▶ Énergie : vapeur !



Toutes les bases de l'ordinateur contemporain ... sans la technologie.

Étape **ZÉRO** : préhistoire



Ada
Lovelace-Byron
(1815-1852)

- ▶ Premier programme informatique (calcul des nombres de Bernoulli) par Ada Lovelace.



Ada à la Doua

Étape ZÉRO : préhistoire



Ada
Lovelace-Byron
(1815-1852)

- Premier programme informatique (calcul des nombres de Bernoulli) par Ada Lovelace.



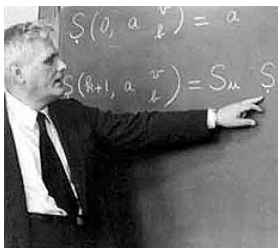
Ada à la Doua

La principale difficulté, c'est la construction d'une machine programmable, pas sa programmation.

Étape **ZÉRO** : juste avant guerre : prémices théoriques



Alan Turing
(1912–1954)



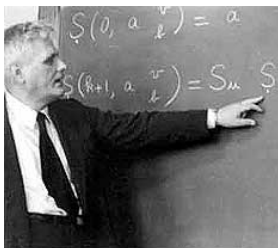
Alonzo Church
(1903–1995)

Qu'est ce qui est *effectivement calculable* ?

Étape **ZÉRO** : juste avant guerre : prémices théoriques



Alan Turing
(1912–1954)



Alonzo Church
(1903–1995)

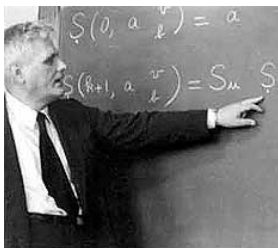
Qu'est ce qui est *effectivement calculable* ?

- ▶ Machine de Turing.

Étape **ZÉRO** : juste avant guerre : prémices théoriques



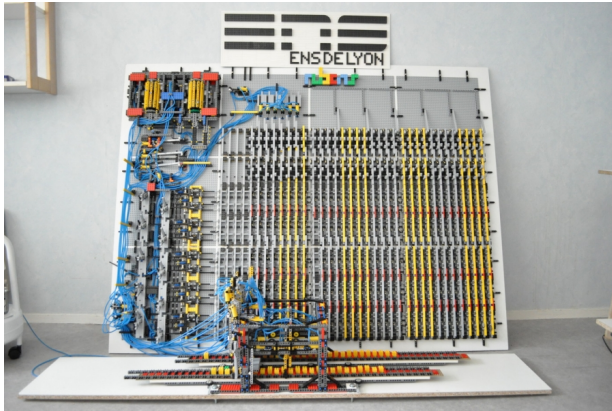
Alan Turing
(1912–1954)



Alonzo Church
(1903–1995)

Qu'est ce qui est *effectivement calculable* ?

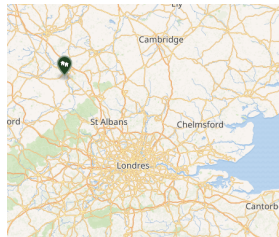
Calculer pour (la machine de) Turing, ce n'est pas seulement calculer avec des nombres.



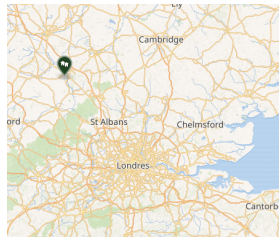
Machine de Turing en Lego, réalisée par les étudiants de l'ENSL, visible à la Maison des Maths et de l'Informatique, 1 Place de l'École, 69007 Lyon.

Étape **ZÉRO**. La guerre : Bletchley Park et la cryptanalyse.

Étape **ZÉRO**. La guerre : Bletchley Park et la cryptanalyse.

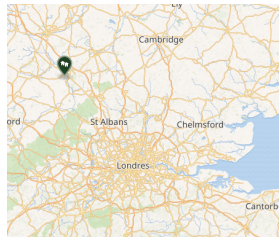


Étape **ZÉRO**. La guerre : Bletchley Park et la cryptanalyse.

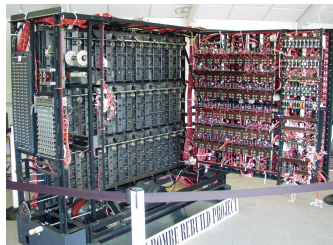


La Machine Enigma

Étape **ZÉRO**. La guerre : Bletchley Park et la cryptanalyse.

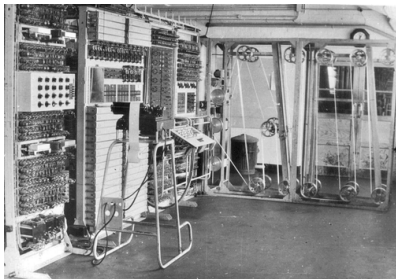


La Machine Enigma



Une « Bombe »

Une étape 1/2? *Colossus* à Bletchley Park, à partir du 1^{er} juin 1944.



Colossus

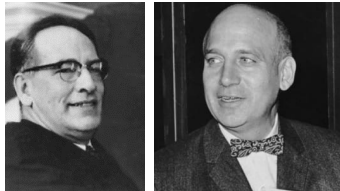
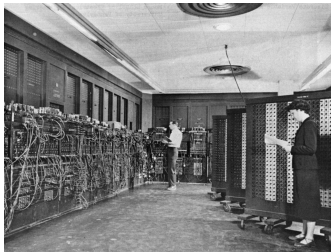
- ▶ Tubes à vide.
- ▶ Spécialisé en cryptanalyse.
- ▶ N'est pas « *équivalent Turing* ».

Étape 1

Premiers ordinateurs

1944, électronique à lampes : l'**ENIAC** (*Electronic Numerical Integrator And Computer*).

Moore School of Electrical Engineering de l'Université de Pennsylvanie.

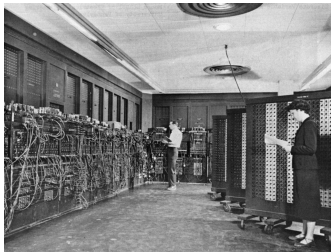


John Mauchly et Prosper Eckert.

Premiers ordinateurs

1944, électronique à lampes : l'**ENIAC** (*Electronic Numerical Integrator And Computer*).

Moore School of Electrical Engineering de l'Université de Pennsylvanie.

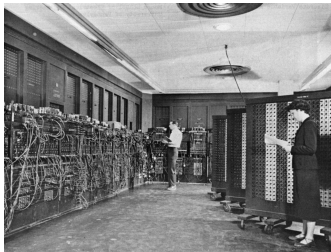


- ▶ Production de tables de tir.
- ▶ 17 468 tubes, 7 200 diodes (max. 116 heures sans panne).
- ▶ Mémoire : 20 nombres de 10 chiffres décimaux.
- ▶ 357 multiplications, 38 divisions par seconde.

Premiers ordinateurs

1944, électronique à lampes : l'**ENIAC** (*Electronic Numerical Integrator And Computer*).

Moore School of Electrical Engineering de l'Université de Pennsylvanie.



- ▶ Production de tables de tir.
- ▶ 17 468 tubes, 7 200 diodes (max. 116 heures sans panne).
- ▶ Mémoire : 20 nombres de 10 chiffres décimaux.
- ▶ 357 multiplications, 38 divisions par seconde.

**Programmation : tableau de connecteurs
=> travail très lent et peu productif.**

Autre machine pionnière

Konrad Zuse (1910–1995)

Recherches solitaires.

Machine électromécanique (le régime nazi ne lui fournit pas de tubes à vide).



Le Z4 (1945) /Deutsches Museum/

Étape DEUX

L'architecture de Von Neumann (1945–1951)

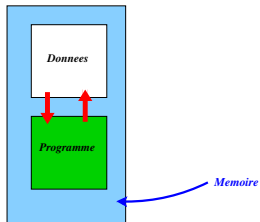


J. v. Neumann
(1903–1957)

L'architecture de Von Neumann (1945–1951)



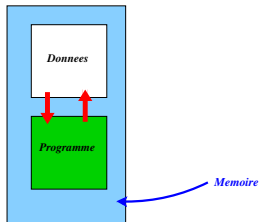
J. v. Neumann
(1903–1957)



L'architecture de Von Neumann (1945–1951)



J. v. Neumann
(1903–1957)



Le métier Jacquard permet de comprendre pourquoi c'est assez naturel.

L'architecture de Von Neumann (1945–1951)

Comment associer un nombre à une ligne (une instruction) d'un programme *Jacquard*?

L'architecture de Von Neumann (1945–1951)

Comment associer un nombre à une ligne (une instruction) d'un programme *Jacquard*?

On code les trous \circ : 1, les pleins $-$: 0.

L'architecture de Von Neumann (1945–1951)

Comment associer un nombre à une ligne (une instruction) d'un programme *Jacquard*?

On code les trous \circ : 1, les pleins $-$: 0.

\circ	$-$	\circ	$-$	\circ	\circ	$-$	\circ	$-$	\circ	$-$
1024	512	256	128	64	32	16	8	4	2	1

L'architecture de Von Neumann (1945–1951)

Comment associer un nombre à une ligne (une instruction) d'un programme *Jacquard*?

On code les trous \circ : 1, les pleins $-$: 0.

\circ	$-$	\circ	$-$	\circ	\circ	$-$	\circ	$-$	\circ	$-$
1024	512	256	128	64	32	16	8	4	2	1
1024	0	256	0	64	32	0	8	0	2	0

L'architecture de Von Neumann (1945–1951)

Comment associer un nombre à une ligne (une instruction) d'un programme *Jacquard*?

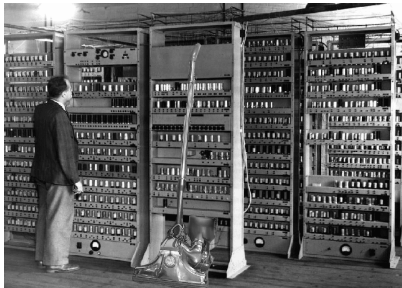
On code les trous \circ : 1, les pleins $-$: 0.

\circ	$-$	\circ	$-$	\circ	\circ	$-$	\circ	$-$	\circ	$-$
1024	512	256	128	64	32	16	8	4	2	1
1024	0	256	0	64	32	0	8	0	2	0

$1024 + 0 + 256 + 0 + 64 + 32 + 0 + 8 + 0 + 2 + 0 = 1386$
soit encore 10101101010 en binaire.

1386, code *unique* de l'*instruction*.

L'architecture de Von Neumann (1945–1951)



EDVAC-Electronic Discrete Variable
Automatic Computer (1949)

- ▶ Productivité énormément améliorée.
- ▶ Toutes les autres approches disparaissent (Exemple : Couffignal en France).
- ▶ Idée que l'ordinateur peut manipuler autre chose que des nombres.

Années 50 : période de grande expansion et d'optimisme.

Années 50 : période de grande expansion et d'optimisme.

- ▶ Mais : progrès très relatifs dus à l'architecture de von Neumann (les programmes dépendent des types de machines, et ils sont très difficiles à écrire).

Années 50 : période de grande expansion et d'optimisme.

- ▶ Mais : progrès très relatifs dus à l'architecture de von Neumann (les programmes dépendent des types de machines, et ils sont très difficiles à écrire).
- ▶ Un échec productif : le fiasco de la traduction automatique.
 - ▶ Développement de l'informatique théorique (grammaires formelles, automates à pile, etc.).
 - ▶ Travaux fondateurs :
Exemple : la collaboration de Noam Chomsky et Marcel-Paul Schützenberger (*Context Free Language*, 1963).

Étape TROIS

Étape **TROIS** : langages de programmation (fin des années 50)

Dès l'origine, on rêve de programmer dans un langage « humain ».

- ▶ Être abstrait par rapport à la machine :

Pouvoir programmer :

$x = 1$

à la place de quelque chose comme :

110011011111101

- ▶ Définir un *alphabet*, un *vocabulaire*, des règles de *grammaire* et leur signification.
- ▶ Traduire depuis ce langage vers le langage de la machine.

- ▶ Apparition du logiciel, au sens actuel du terme.
- ▶ **L'origine du mal...**

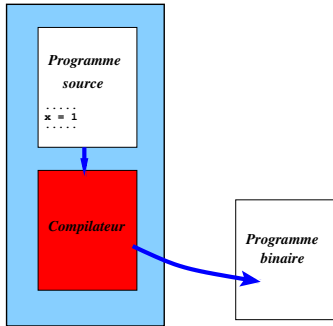
- ▶ Apparition du logiciel, au sens actuel du terme.
- ▶ **L'origine du mal...**

Problème :

Une fois le langage défini, comment le traduire dans le langage *machine (binaire)* de l'ordinateur ?

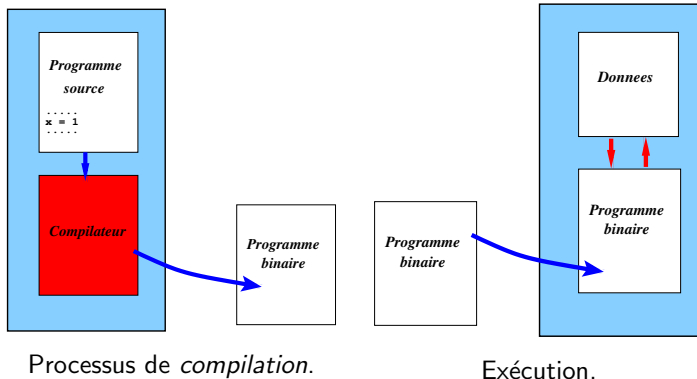
Ou : quand l'ordinateur traite autre chose que des nombres.

Étape **TROIS** : langages de programmation



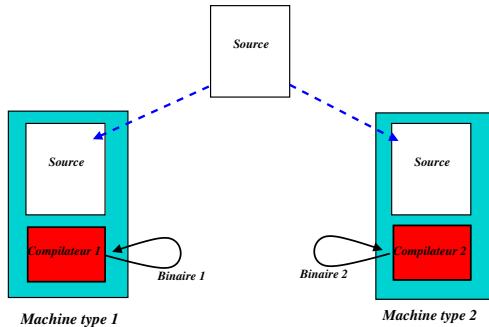
Processus de *compilation*.

Étape **TROIS** : langages de programmation



Avantages :

- ▶ Tout devient plus facile.
- ▶ Sources réutilisables d'un type de machine à l'autre :



Inconvénients ?

Étape QUATRE

Microprocesseurs et internet.



Hollywood, 1964.



Lyon, milieu des années 70, IRIS 80 CII



Lyon, milieu des années 70, IRIS 80 CII

- ▶ Apogée du *Plan Calcul*.
- ▶ Entre 10 et 20 millions de francs de l'époque.
- ▶ Mémoire vive : 1 Méga octets.
- ▶ 10^6 opérations (multiplications, additions,...) par seconde.

Étape **QUATRE** : microprocesseurs et internet

Rupture technologique dans les années 80/90.

	IRIS 80	Votre ordinateur	Rapport
Prix (Euros)	10 millions	1000	10^{-4}
Taille mémoire :	1 MO	8 (16) GO	8000
Ops. /seconde	10^6	$\simeq 10^{10}$	10^4

Étape **QUATRE** : microprocesseurs et internet

Rupture technologique dans les années 80/90.

	IRIS 80	Votre ordinateur	Rapport
Prix (Euros)	10 millions	1000	10^{-4}
Taille mémoire :	1 MO	8 (16) GO	8000
Ops. /seconde	10^6	$\simeq 10^{10}$	10^4

Pas de traitement de texte, pas de bureautique avec la machine à 10 millions de francs (d'euros)!

Retour à l'étape **TROIS** : langages de programmation

Retour à l'étape **TROIS** : langages de programmation

tableau = [1., 2., 3., 5., 10., 100., 120.]. Calculer sa moyenne.

Retour à l'étape **TROIS** : langages de programmation

tableau = [1., 2., 3., 5., 10., 100., 120.]. Calculer sa moyenne.

1) Algorithme :

```
somme = 0.0
```

```
pour x dans tableau faire
```

```
| somme ← somme + x
```

```
fin
```

```
moyenne = somme/taille(tableau)
```

Retour à l'étape **TROIS** : langages de programmation

tableau = [1., 2., 3., 5., 10., 100., 120.]. Calculer sa moyenne.

1) Algorithme :

```
somme = 0.0
pour x dans tableau faire
  | somme ← somme + x
fin
moyenne = somme/taille(tableau)
```

2) Programme en langage évolué (Julia) :

```
function moyenne(tableau)
    somme = 0.0
    for x in tableau
        somme = somme + x
    end
    return somme/length(tableau)
end
```

```
function moyenne(tableau)
    somme =0.0
    for x in tableau
        somme = somme + x
    end
    return somme/length(tableau)
end
```

```
> x=[1., 2., 3., 5., 10., 100., 120.];
> moyenne(x)
34.4285714285714
```

```
function moyenne(tableau)
    somme =0.0
    for x in tableau
        somme = somme + x
    end
    return somme/length(tableau)
end
```

```
> x=[1., 2., 3., 5., 10., 100., 120.];
> moyenne(x)
34.4285714285714
```

```
>Q=[1.,2.,3.,4.,5.,6.,7.,8.,9.,10.]
>moyenne(Q)
5.5
```

On a fabriqué une *boîte* :

```
function moyenne(tableau)
```

```
end
```

qui :

- ▶ *Reçoit* un tableau de nombres.
- ▶ *Renvoie* la moyenne du tableau.

Définit un *contrat*.

On a fabriqué une *boîte* :

```
function moyenne(tableau)  
  
end
```

qui :

- ▶ *Reçoit* un tableau de nombres.
- ▶ *Renvoie* la moyenne du tableau.

Définit un *contrat*.

1. **Réutilisabilité** : on a un moyen réutilisable de calculer la moyenne de n'importe quel tableau.

On a fabriqué une *boîte* :

```
function moyenne(tableau)
```

```
end
```

qui :

- ▶ *Reçoit* un tableau de nombres.
- ▶ *Renvoie* la moyenne du tableau.

Définit un *contrat*.

1. **Réutilisabilité** : on a un moyen réutilisable de calculer la moyenne de n'importe quel tableau.
2. **Mais comment s'assurer que le contrat est bien rempli ?**
Boîte noire ou transparence ?
Faire confiance ou pouvoir vérifier ?
Et en cas de bug ?

On rappelle que le *code source est compilé*, c'est à dire *traduit* dans le *langage* de la machine (suite de 1 et de 0).
Qu'en est-il ?

On rappelle que le *code source est compilé*, c'est à dire *traduit* dans le *langage* de la machine (suite de 1 et de 0).
Qu'en est-il ?

Reprenons notre programme :

```
function moyenne(tableau)
    somme = 0.0
    for x in tableau
        somme = somme + x
    end
    return somme/length(tableau)
end
```

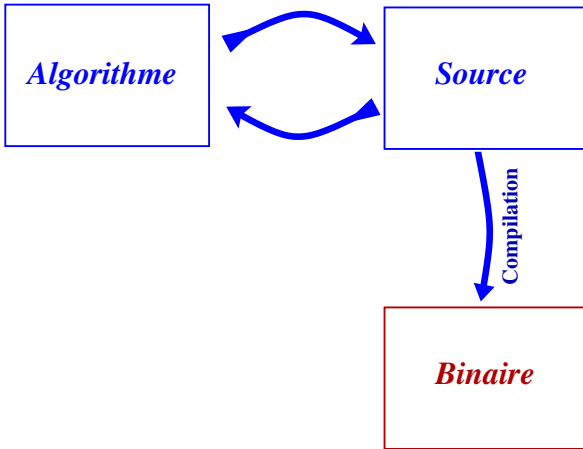
et traduisons (compilons) le.

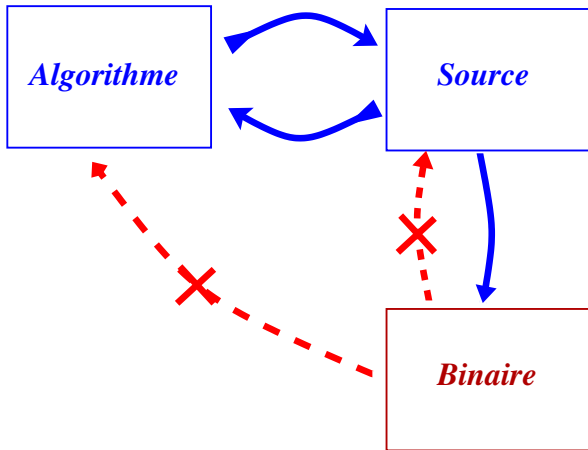
Une sténographie du code binaire engendré :

```
mov rax, qword ptr [rdi + 8]
test rax, rax
jle L56
mov rcx, qword ptr [rdi]
vxorpd xmm0, xmm0, xmm0
vaddsd xmm0, xmm0, qword ptr [rcx]
cmp rax, 1
je L46
mov edx, 1
nop
vaddsd xmm0, xmm0, qword ptr [rcx + 8*rdx]
add rdx, 1
cmp rdx, rax
jb L32
```

```
vcvtsi2sd xmm1, xmm1, rax
vdivsd xmm0, xmm0, xmm1
ret
vxorpd xmm0, xmm0, xmm0
vcvtsi2sd xmm1, xmm1, rax
vdivsd xmm0, xmm0, xmm1
  ret
  nop word ptr cs:[rax + rax]
```

Tous les symboles qu'on voit ici doivent être remplacés par des codes prédéfinis : des suites de 1 et de 0.



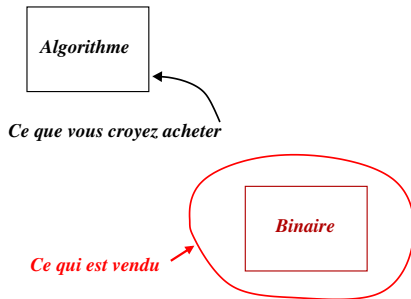


La compilation est un processus quasiment irréversible.

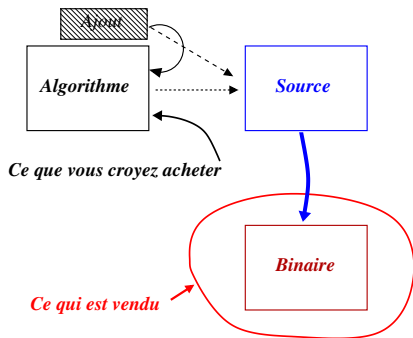
- ▶ Les systèmes d'exploitation : **probablement ≥ 50 millions de lignes de code source pour Windows.**
- ▶ Les navigateurs web : plus de 20 millions de lignes de code source pour firefox.

Conséquences

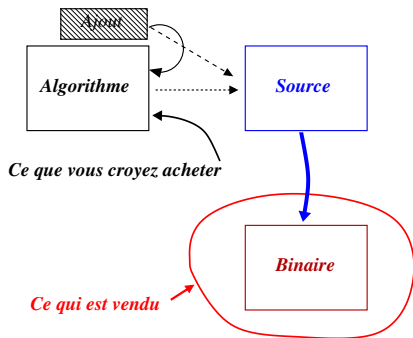
1) Information cachée, logiciels tricheurs



1) Information cachée, logiciels tricheurs



1) Information cachée, logiciels tricheurs

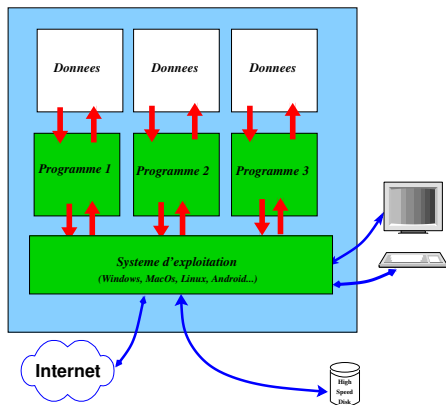


Quelques horreurs...

- ▶ Le *Dieselpgate*.
- ▶ Les portes dérobées.
- ▶ L'obsolescence programmée.

2) La prise du pouvoir par le logiciel

Plus loin avec l'architecture de von Neumann : les **systèmes d'exploitation**.



Celui qui « possède » les systèmes d'exploitation a un pouvoir de vie et de mort sur le matériel (et sur les logiciels).

La prise du pouvoir par le logiciel

- ▶ Début des années 90 : grande variété de microprocesseurs, chaque constructeur ayant le sien, + des indépendants.
- ▶ Fin des années 90 : hégémonie des systèmes d'exploitation Windows (Microsoft) et MacOs (Apple).

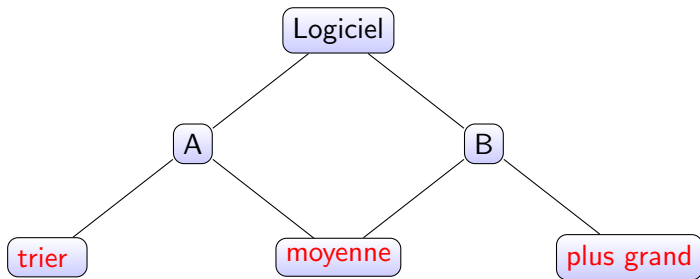
La prise du pouvoir par le logiciel

- ▶ Début des années 90 : grande variété de microprocesseurs, chaque constructeur ayant le sien, + des indépendants.
- ▶ Fin des années 90 : hégémonie des systèmes d'exploitation Windows (Microsoft) et MacOs (Apple).
- ▶ Mais Microsoft et Apple ne portent leur système **que sur une seule gamme de processeurs**. (*WinTel*).
- ▶ Disparition de filières de productions entières, souvent intéressantes.
- ▶ Obsolescence programmée.

Anatomie d'un programme

Structure hiérarchique (arbre).

On s'appuie sur des composants réputés fiables (exemple : calcul de la moyenne). Ne pas réinventer la roue !



Nécessité de disposer d'une bibliothèque de **composants réutilisables**.

Mauvais exemple : utilisation de composants de Facebook (espions) par les applications sur smartphone.

Le logiciel libre

La Free Software Foundation (FSF)

La FSF est fondée par Richard M. Stallmann (MIT) en 1985.

La Free Software Foundation (FSF)

La FSF est fondée par Richard M. Stallmann (MIT) en 1985.



RMS,1953 -

La Free Software Foundation (FSF)

La FSF est fondée par Richard M. Stallmann (MIT) en 1985.



RMS, 1953 -



RMS à Lyon

La Free Software Foundation (FSF)

La FSF est fondée par Richard M. Stallmann (MIT) en 1985.



RMS,1953 -



RMS à Lyon

- ▶ *Le logiciel doit être un bien commun.*
- ▶ *Ne rien cacher.*
- ▶ *Soumettre les logiciels à une expertise collective, publique.*

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;

Remarque : ce qui implique l'accès aux sources (en langage évolué).

▶ libre \Rightarrow Open-Source.

▶ Open-Source \nRightarrow libre.

3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.

Logiciel libre, au sens de la Free Software Foundation (FSF)

Quatre libertés :

1. liberté d'exécuter le programme, pour tous les usages ;
2. liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
4. liberté d'améliorer le programme et de distribuer ces améliorations au public, pour le plus grand profit de tous.



Roosevelt / Les quatre libertés (Norman Rockwell)

Cadre juridique : licences

- ▶ Droit d'auteur.
- ▶ Garantissent les quatre libertés.
- ▶ Définissent des devoirs.

1. **Licences avec obligation de réciprocité** : Les œuvres dérivées du logiciel héritent des mêmes libertés et des mêmes devoirs. On parle de « *Copyleft* ».

Exemple fondateur :

GENERAL PUBLIC LICENCE (GPL) : *tout logiciel dérivé doit être diffusé sous la même licence.*

1. **Licences avec obligation de réciprocité** : Les œuvres dérivées du logiciel héritent des mêmes libertés et des mêmes devoirs. On parle de « *Copyleft* ».

Exemple fondateur :

GENERAL PUBLIC LICENCE (GPL) : *tout logiciel dérivé doit être diffusé sous la même licence.*

2. **Licences permissives** : Mention du copyright dans les œuvres dérivées.

Exemples : licence MIT ou BSD.

1. **Licences avec obligation de réciprocité** : Les œuvres dérivées du logiciel héritent des mêmes libertés et des mêmes devoirs. On parle de « *Copyleft* ».
Exemple fondateur :
GENERAL PUBLIC LICENCE (GPL) : *tout logiciel dérivé doit être diffusé sous la même licence.*
2. **Licences permissives** : Mention du copyright dans les œuvres dérivées.
Exemples : licence MIT ou BSD.
3. **Licences pour composants** : N'importe quel programme peut utiliser le composant (avec des obligations de citation).
Exemple : LGPL (*Lesser GPL*).

Licences : validité en France ?

- ▶ Pas de traductions officielles.
- ▶ Concepts du droit anglo-saxon.

Le tribunal de grande instance de Paris a jugé applicable la licence GPL (version 2) en France le 28 mars 2007.

Un développement inséparable de celui d'Internet

Le logiciel libre implique l'échange :

- ▶ Partage.
- ▶ Développement : la plupart du temps mondial.
« Plateformes » comme GitHub <https://github.com/>.

Réalisations *libres* emblématiques.

Un système d'exploitation :

- ▶ commencé sous l'égide de la FSF,
- ▶ achevé (noyau) en 1991 par un étudiant finlandais, [Linus Thorvalds](#).

d'où le nom du système : **LINUX**.



À la différence des systèmes propriétaires, Linux peut être porté sur tous les processeurs.

Réalisations *libres* emblématiques.

Un système d'exploitation :

- ▶ commencé sous l'égide de la FSF,
- ▶ achevé (noyau) en 1991 par un étudiant finlandais, [Linus Thorvalds](#).



d'où le nom du système : **LINUX**.

À la différence des systèmes propriétaires, Linux peut être porté sur tous les processeurs.

Licence la plus restrictive (GPL).

Réalisations *libres* emblématiques.

Un système d'exploitation :

- ▶ commencé sous l'égide de la FSF,
- ▶ achevé (noyau) en 1991 par un étudiant finlandais, [Linus Thorvalds](#).



d'où le nom du système : **LINUX**.

À la différence des systèmes propriétaires, Linux peut être porté sur tous les processeurs.

Licence la plus restrictive (GPL).

Linux fait partie de la famille des systèmes Unix.

Des compilateurs.

Des compilateurs.

et après, beaucoup de **logiciels**
(exemple : LibreOffice/OpenOffice).

Le logiciel libre : une utopie presque réalisée ?

Le logiciel libre : une utopie presque réalisée ?

- ▶ Développement à partir du début des années 90.

Le logiciel libre : une utopie presque réalisée ?

- ▶ Développement à partir du début des années 90.
- ▶ 1998 : fondation de nombreuses associations dont l'*Association Lyonnaise pour de développement de l'informatique libre* (ALDIL).

Le logiciel libre : une utopie presque réalisée ?

- ▶ Développement à partir du début des années 90.
- ▶ 1998 : fondation de nombreuses associations dont l'*Association Lyonnaise pour de développement de l'informatique libre* (ALDIL).
- ▶ 2001 :
Steve Balmer, PDG. de Microsoft :
« Linux is a cancer that attaches itself in an intellectual property sense to everything it touches ».
« Linux is communism ».

Le logiciel libre : une utopie presque réalisée ?

- ▶ Développement à partir du début des années 90.
- ▶ 1998 : fondation de nombreuses associations dont l'*Association Lyonnaise pour de développement de l'informatique libre* (ALDIL).
- ▶ 2001 :
Steve Balmer, PDG. de Microsoft :
« Linux is a cancer that attaches itself in an intellectual property sense to everything it touches ».
« Linux is communism ».
- ▶ 2016 :
« Microsoft vient d'annoncer avoir rejoint la *Linux Foundation*, conçue pour soutenir le développement du système d'exploitation alternatif open source et gratuit » (presse).

Une utopie presque réalisée ?

Quatre domaines :

1. Sciences et techniques.
2. Web.
3. Vie courante.
4. Postes individuels.

Une utopie presque réalisée : 1) Sciences

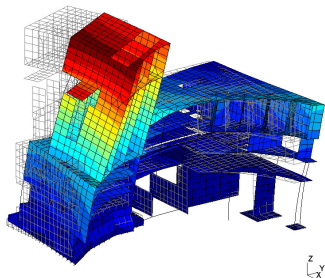
Peut-on faire de la science ou de l'ingénierie avec des boîtes noires ?

Une utopie presque réalisée : 1) Sciences et techniques

Exemple : *Code Aster* (EDF).

- ▶ mécanique du solide,
- ▶ thermique.
- ▶ etc.

1 million de lignes de code source.



Développé depuis 1989, « libéré » en 2001.
Conséquences ?

Cercle vertueux :

Appropriation par un grand nombre d'utilisateurs, travaillant dans des domaines différents d'EDF.

- ▶ découverte et correction de nombreux défauts,
- ▶ nouveaux développements.

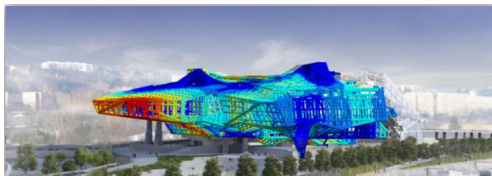
et finalement : *amélioration du produit pour EDF.*

Cercle vertueux :

Appropriation par un grand nombre d'utilisateurs, travaillant dans des domaines différents d'EDF.

- ▶ découverte et correction de nombreux défauts,
- ▶ nouveaux développements.

et finalement : *amélioration du produit pour EDF.*



Une utopie presque réalisée : 2) Web

- ▶ **Serveurs** : 83 % des sites web utilisent les serveurs libres APACHE et NGINX, sous un système LINUX.
- ▶ **Navigateurs** : Firefox (Mozilla), Chromium (Google).
Tous les autres sont soupçonnés d'espionner.
- ▶ **Connexions chiffrées et certifiées** : (https).

Une utopie presque réalisée : 3) Vie courante

Exemples :

- ▶ « Box » internet.
Freebox : 51 logiciels libres (dont Linux).
- ▶ Les « Smartphones »... c'est une autre histoire.

Une utopie presque réalisée : 4) postes individuels

Les machines Linux difficiles à utiliser ? **FAUX !**

- ▶ Environnements simplifiés pour personnes sans culture informatique.
- ▶ Supports linguistiques (exemple : différentes variantes du français (France, Belgique, Canada,... !)).
- ▶ Environnements spéciaux pour déficients visuels.

Les affaires Huawei et les smartphones

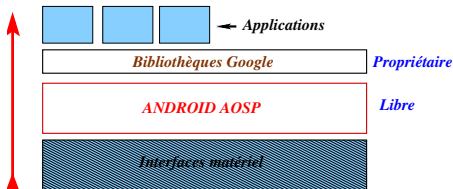
- ▶ Espionnage ?

Les affaires Huawei et les smartphones

- ▶ Espionnage ?
- ▶ **Donald Trump vs. Huawei** : on lit que *Google a retiré à Huawei sa licence Android*. C'est inexact !

Les affaires Huawei et les smartphones

- ▶ Espionnage ?
- ▶ Donald Trump vs. Huawei : on lit que Google a retiré à Huawei sa licence Android. C'est inexact !



Android AOSP (*Android Open Source Project*) :

Android est un dérivé de Linux et Linux est protégé par la licence GPL => **Android est un logiciel libre (GPL)**.

Ce sont les bibliothèques de Google qui sont soumises à l'embargo !

Étape cinq

et « économie » du logiciel libre.

Transformation de l'informatique en une activité de services

L'accès facile, gratuit, à un monde de logiciels *permet*, mais ne rend pas *évidente* la réalisation d'activités de services.

Exemples :

- ▶ Sites web.
- ▶ Sites marchands,

mais aussi :

- ▶ Ingénierie (calcul scientifique).
- ▶ Statistique.
- ▶ *Machine Learning* (IA).

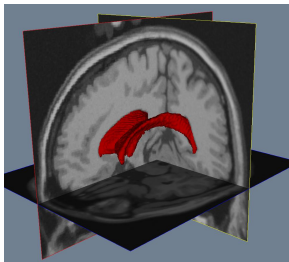
Exemples : PyTorch (origine : Meta), TensorFlow (origine : Google),...

etc.

Exemple d'entreprise produisant du logiciel libre

KITWARE (USA et... Villeurbanne).

Domaines : imagerie, visualisation scientifique.



Logiciels libres, revenus provenant de formations, d'intégration des logiciels à des dispositifs, de grands sponsors.

Produire du logiciel libre est considéré comme un gage de qualité.

L'état et le logiciel libre

Français, encore un effort...

- ▶ Textes législatifs tendant à imposer l'utilisation du logiciel libre dans l'administration ;
- ▶ Obligatoire dans les marchés publics.
- ▶ Faible impact.

L'état et le logiciel libre

Français, encore un effort...

- ▶ Textes législatifs tendant à imposer l'utilisation du logiciel libre dans l'administration ;
- ▶ Obligatoire dans les marchés publics.
- ▶ Faible impact.

Exceptions notables :

- ▶ La Gendarmerie Nationale.
- ▶ Ministère de l'agriculture.
- ▶ Assemblée Nationale.
- ▶ Certaines communes.

Ailleurs...

- ▶ Inde (certains états), Indonésie, Jordanie, Malaisie, Vienne (Autriche), Munich, Schwäbisch Hall, Portugal (régions), Roumanie, Turin, Massachusetts, Maison Blanche, diverses agences américaines, Corée du Sud, Islande, Brésil, Équateur, Iran (?), Pérou, Vénézuéla, Chine (gouvernement, 2019), Cuba, ...

Quel pays est équipé à 100 % en logiciel libre ?

Ailleurs...

- ▶ Inde (certains états), Indonésie, Jordanie, Malaisie, Vienne (Autriche), Munich, Schwäbisch Hall, Portugal (régions), Roumanie, Turin, Massachusetts, Maison Blanche, diverses agences américaines, Corée du Sud, Islande, Brésil, Équateur, Iran (?), Pérou, Vénézuéla, Chine (gouvernement, 2019), Cuba, ...

Quel pays est équipé à 100 % en logiciel libre ?
la Corée du Nord.

L'influence du logiciel libre

À l'origine d'un mouvement « *open* ».

- ▶ Formats ouverts (exemple : pdf).

L'influence du logiciel libre

À l'origine d'un mouvement « *open* ».

- ▶ Formats ouverts (exemple : pdf).
- ▶ Œuvres libres (artistiques, littéraires, etc.). Licences adaptées (*Creative Commons*).

L'influence du logiciel libre

À l'origine d'un mouvement « *open* ».

- ▶ Formats ouverts (exemple : pdf).
- ▶ Œuvres libres (artistiques, littéraires, etc.). Licences adaptées (*Creative Commons*).
- ▶ Données libres. Exemples :
 - ▶ Lyon.
 - ▶ Paris.

Licences *Open Database*.

L'influence du logiciel libre

À l'origine d'un mouvement « *open* ».

- ▶ Formats ouverts (exemple : pdf).
- ▶ Œuvres libres (artistiques, littéraires, etc.). Licences adaptées (*Creative Commons*).
- ▶ Données libres. Exemples :
 - ▶ Lyon.
 - ▶ Paris.

Licences *Open Database*.

- ▶ Wikipédia.
- ▶ OpenStreetMap.
- ▶ etc.

Débats actuels sur la *Science ouverte*.

Dangers

- ▶ *Le Cloud.*

Dangers

- ▶ Le *Cloud*.
- ▶ Les *brevets logiciels*.
En Europe :
 - ▶ Les programmes sont protégés par le droit d'auteur.
 - ▶ Les logiciels sont exclus du champ de la brevetabilité depuis 1973, tout comme les jeux et les mathématiques.

Mais...débat au parlement européen en 2000.

Non-brevetabilité maintenue en 2005.

(voir les textes de F. Pellegrini).

Beaux projets

Software Heritage.

<https://www.softwareheritage.org/>

Projet initié par Roberto Di Cosmo (professeur, Paris VII) en 2014.

- ▶ Archiver *tout* le code source créé par l'humanité.
- ▶ Ouverture en 2016 (soutien de l'Inria).
- ▶ Début 2017, l'UNESCO soutient le projet.
- ▶ 2021 : \simeq 11 milliards de fichiers sources pour \simeq 160 millions de projets sont archivés.

On y trouve par exemple le code source de la mission Apollo 11 !
Mais bien sûr, on n'y trouve que du logiciel libre.

Lutte contre la « fracture numérique ».

Lutte contre la « fracture numérique ».

- ▶ **Collectif Emmabuntüs** : reconditionnement d'ordinateurs avec un système Linux (Emmaüs).

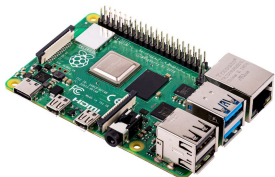
Lutte contre la « fracture numérique ».

- ▶ **Collectif Emmabuntüs** : reconditionnement d'ordinateurs avec un système Linux (Emmaüs).
- ▶ **Lalis** (Lyon Association Libre Informatique Solidaire) : reconditionnement, formation.

Linux

Pour préparer l'Install Party.

Du plus petit au plus puissant...

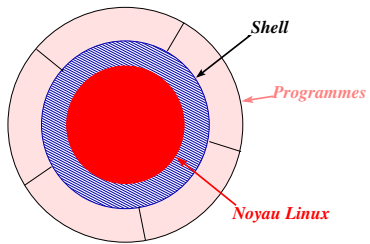


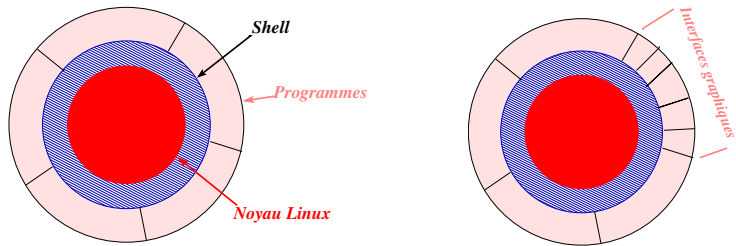
Raspberry Pi



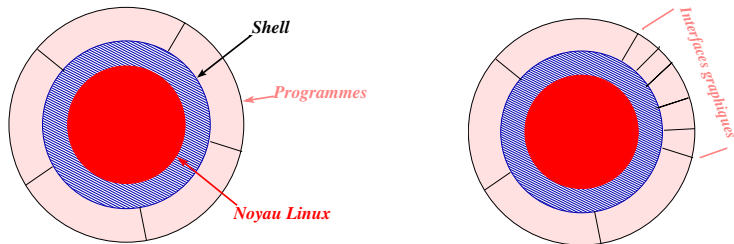
Taihu

- ▶ Sur tous les types de processeurs (Intel, AMD, ARM,..., Sunway,...), y compris des processeurs « anciens ».
- ▶ (Port sur Apple M1/M2 en cours).





- ▶ Différentes interface graphiques (xfce, mate, gnome, kde, ukui,...).



- ▶ Différentes interface graphiques (xfce, mate, gnome, kde, ukui,...).
- ▶ Énorme collection de logiciels (libres).
- ▶ Nécessité de gérer les mises à jour.

Distributions Linux

But : offrir un système simple à utiliser pour :

- ▶ Installer facilement le système de base et des programmes jugés indispensables (libreoffice, navigateur, etc.)
- ▶ Les mettre à jour facilement (bugs, sécurité, nouvelles versions).
- ▶ Installer les nouvelles versions du système.

Distributions Linux

But : offrir un système simple à utiliser pour :

- ▶ Installer facilement le système de base et des programmes jugés indispensables (libreoffice, navigateur, etc.)
- ▶ Les mettre à jour facilement (bugs, sécurité, nouvelles versions).
- ▶ Installer les nouvelles versions du système.

Exemples de distributions :

- ▶ [Debian](#) – > [Ubuntu](#) – > [Mint](#) – >...
- ▶ [RedHat](#), [Fedora](#) – > [Scientific Linux](#) ...
- ▶ [ArchLinux](#)...

Bibliographie I



La Bataille du Libre.

Film.

ARTE, 2019.



Pierre-Éric Mounier-Kuhn.

L' informatique en france de la seconde guerre mondiale au
plan calcul.

*L'émergence d'une science, Presses de l'Université
Paris-Sorbonne, 2010.*

Bibliographie II



Dominique Foray and Jean-Benoît Zimmermann.
L'économie du logiciel libre.
Revue économique, 52(7) :77–93, 2001.



François Pellegrini and Sébastien Canevet.
Droit des logiciels (livre).
Presses Universitaires de France, 2013.



François Pellegrini.
Droit des logiciels (transparents).
http://dept-info.labri.fr/ENSEIGNEMENT/droitlog/m1/cours/droit-log_stu_fr1_20180314.pdf.

- ▶ Ces diapos sont disponibles sur
<https://thierry-dumont.fr/Exposes>
- ▶ ALDIL : <https://aldil.org>
- ▶ Journées du Logiciel Libre (JDLL, 01-02 Avril 2023, 10h-18h.)
<https://www.jd11.org/>
- ▶ Contact : thierry@thierry-dumont.fr