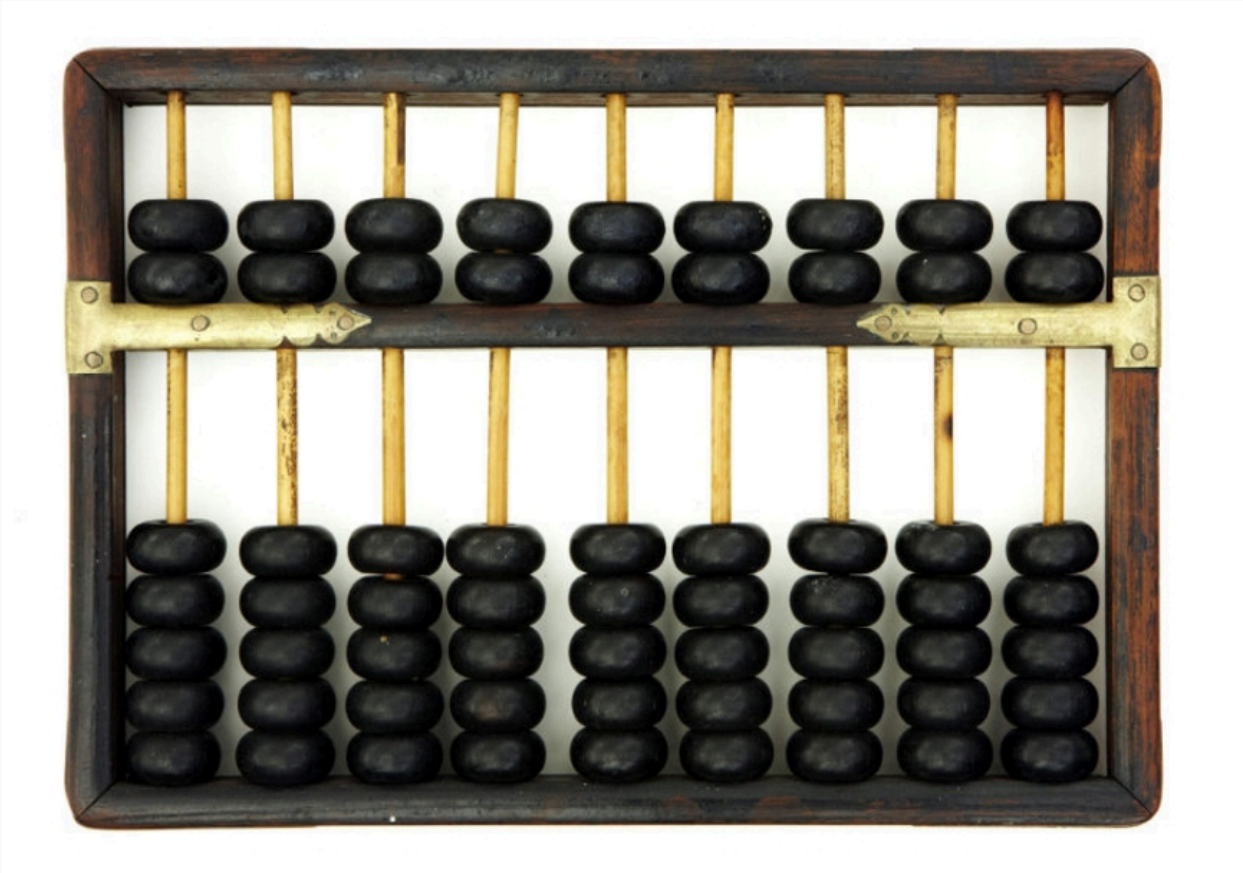
NSI 1ère - Architecture matérielle et systèmes d'exploitation

QK

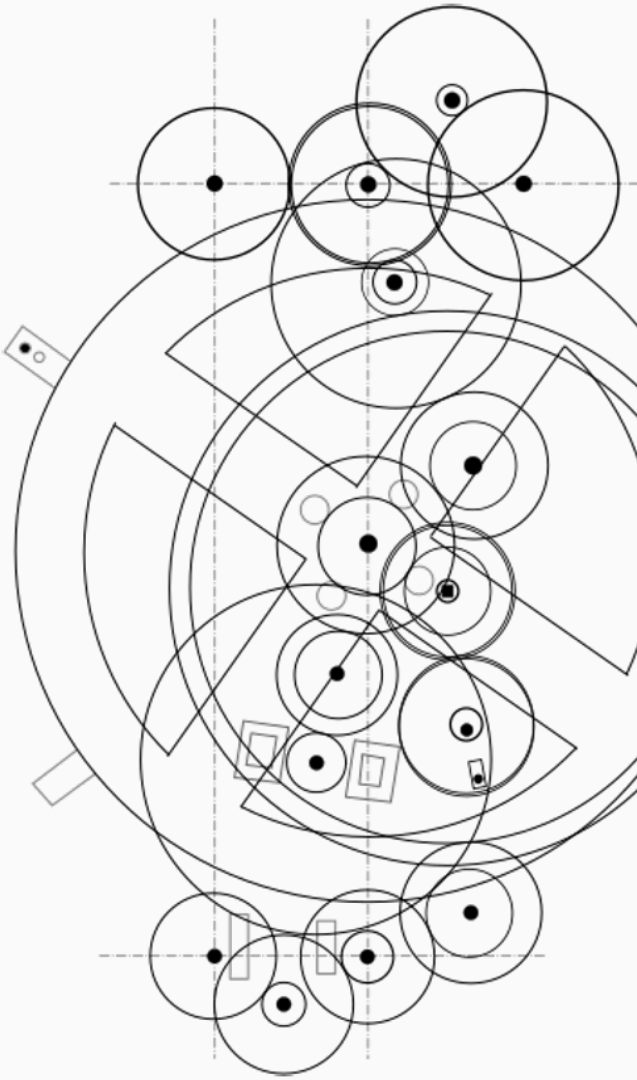
Historique de la machine

Préhistoire informatique

- Compter : les doigts, les orteils et des outils.
- Concevoir des machines pour réaliser des calculs (*calculi* mot latin qui signifie “cailloux”). **Exemple** : Le *boulier*, découvert indépendamment par de nombreuses civilisations



Antiquité : naviguer



La machine d'Anticythère, découverte en 1900 et datant de -87 avant J.-C. servait à calculer les positions astronomiques et donc à **naviguer**

Elle démontre que :

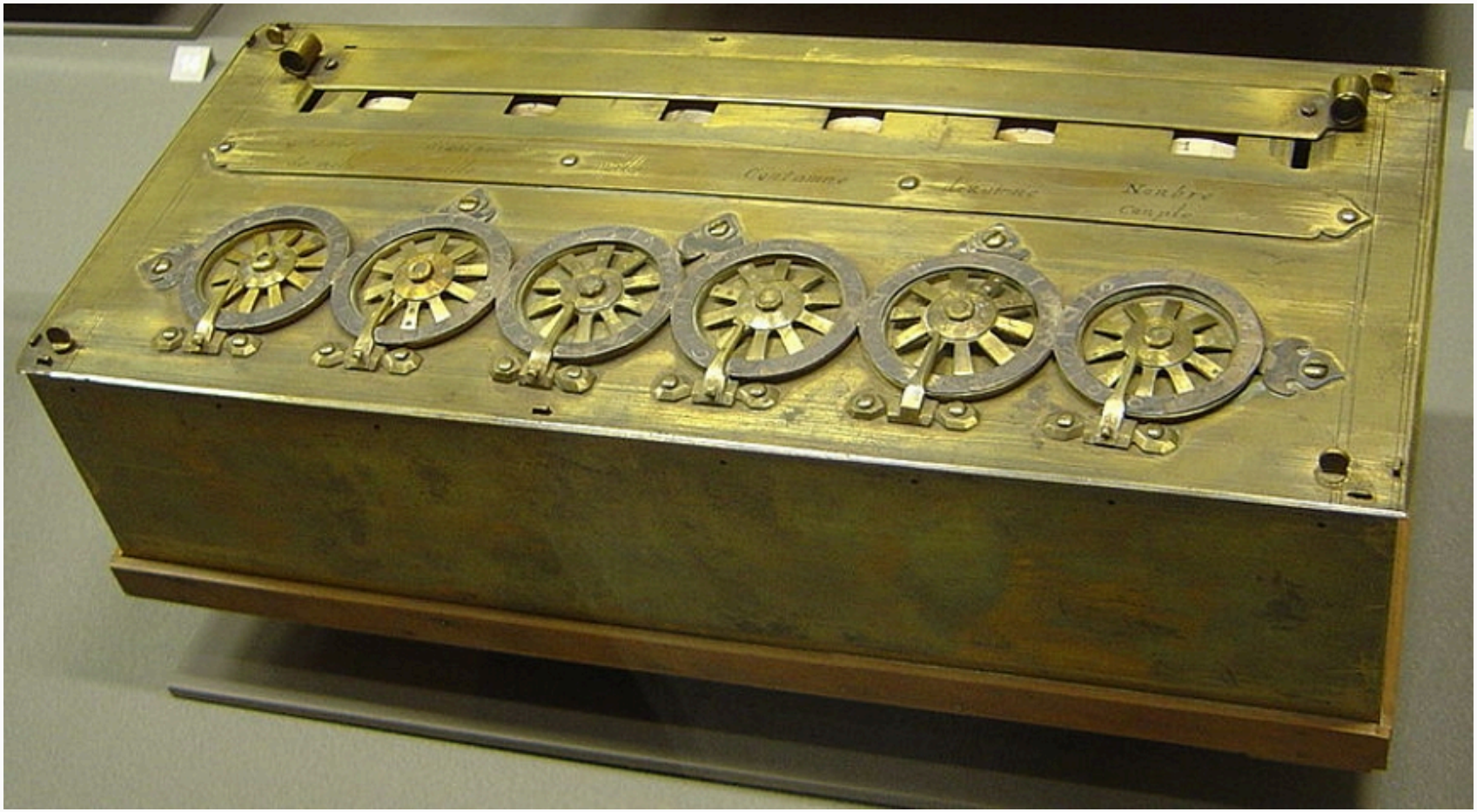
- Calculer a toujours été une entreprise importante
- L'homme s'est toujours montré d'une très grande ingéniosité pour y parvenir

Renaissance : Blaise Pascal

Blaise Pascal. Scientifique et penseur français du XVIIe siècle

- En physique : le *théorème de Pascal* qui exprime les variations de pression dans un fluide
- En mathématiques : *calcul infinitésimal*, *raisonnement par récurrence* etc.
- En philosophie et littérature : les *Pensées* (1669)
- En ingénierie : la *Pascaline*. Première machine à calculer. Inventée à 19 ans pour aider son père devant remettre en ordre les recettes fiscales d'une province.

La pascaline



Gottfried Wilhelm Leibniz

Leibniz (1646 - 1716) est un penseur allemand fait progresser la philosophie, les mathématiques, la physique et l'ingénierie autant que Pascal.

Il améliore la Pascaline et redécouvre le système **binaire**.

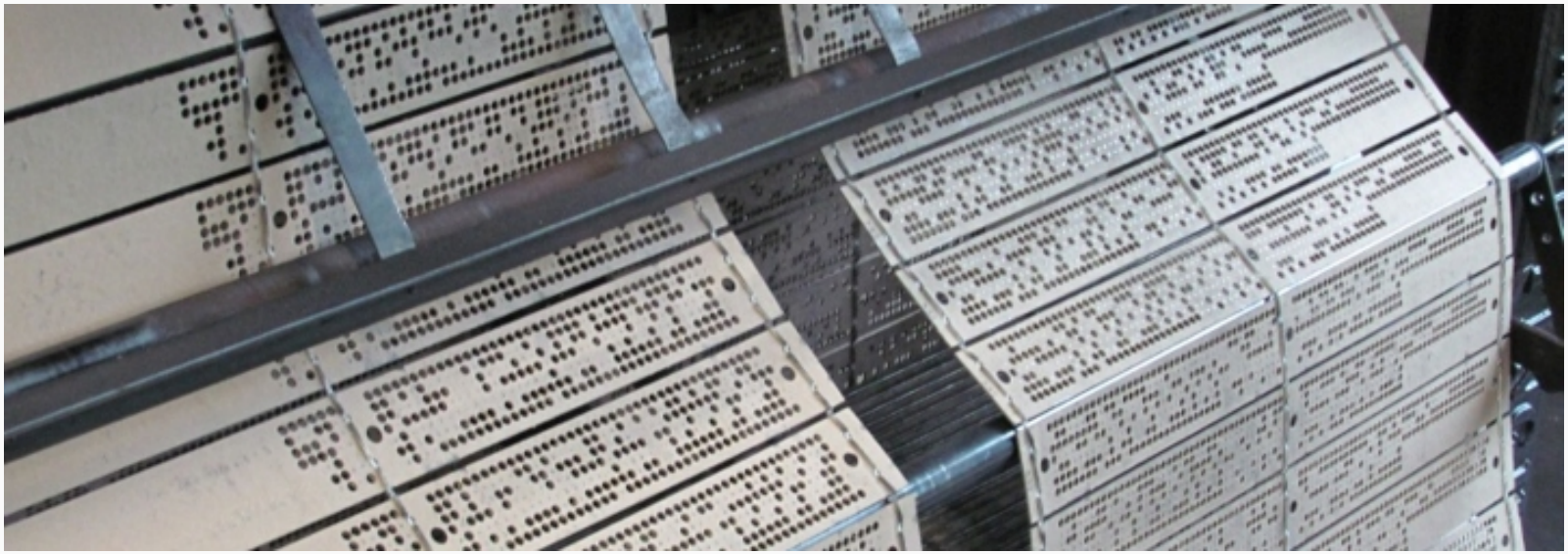
Néanmoins on n'utilisera réellement le binaire qu'après 1945.

Premières machines programmables : métiers à tisser

Joseph Marie Jacquard (1752-1834) améliore des principes déjà existants pour concevoir une machine à tisser utilisant les cartes perforées de **Jean-Baptiste Falcon.**

ON S'EN FOUT C'EST VIEUX !

Les métiers Jacquard sont encore utilisés dans le médical pour réaliser des coudières, genouillères et prothèses d'artères. Et c'est produit en France.



Machine à calculer programmable

Charles Babbage constatant que les erreurs de calculs dans les tables conduisent à de nombreuses catastrophes invente en 1833 le concept de *machine (Difference Engine 1)* permettant d'automatiser le calcul.

Il correspond ensuite avec **Ada Lovelace** (comtesse et fille du poète Lord Byron). Elle conçoit les premiers programmes pour cette machine. Elle est vue comme la première programmeuse du monde.

Trop complexe, trop coûteuse la machine de Babbage ne verra jamais le jour.

Ada Lovelace

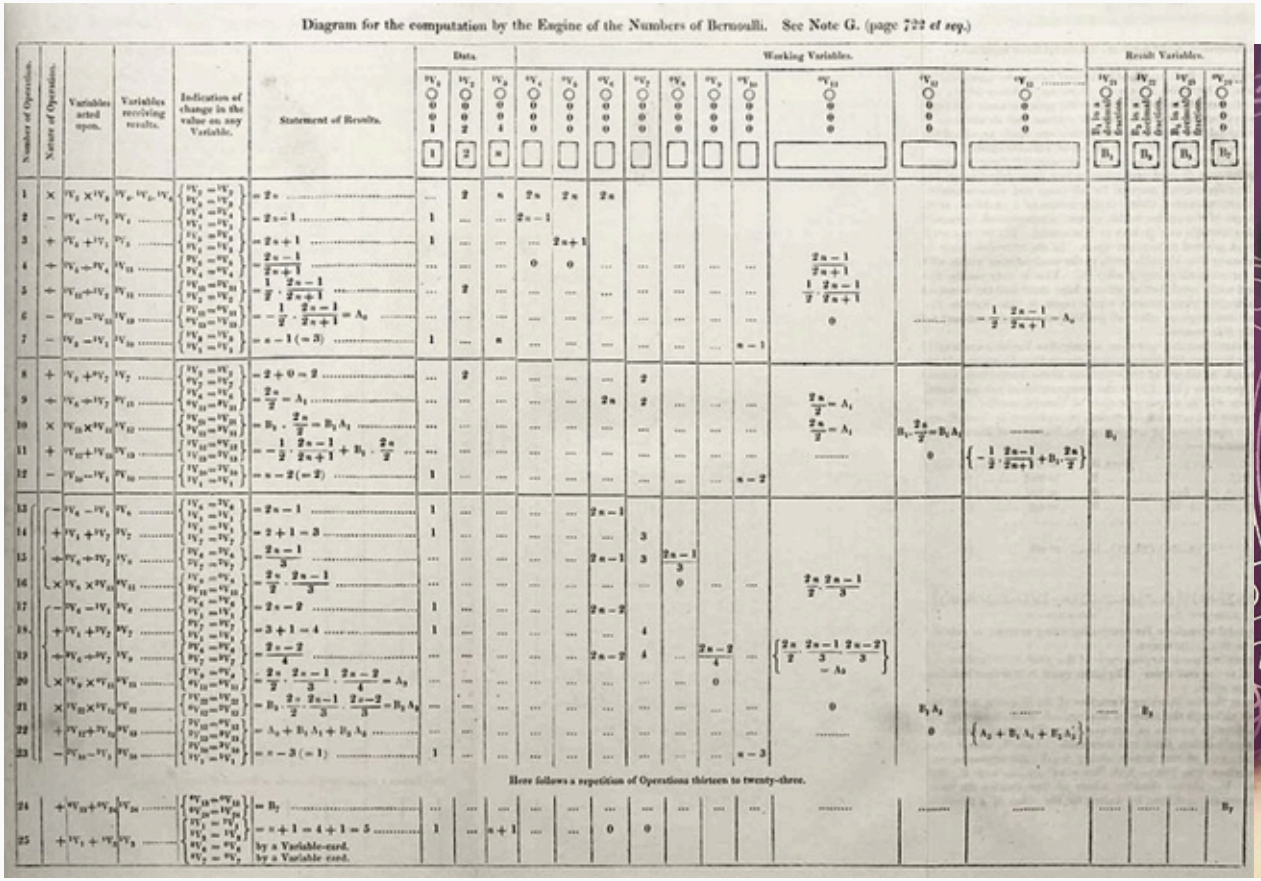


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Électricité

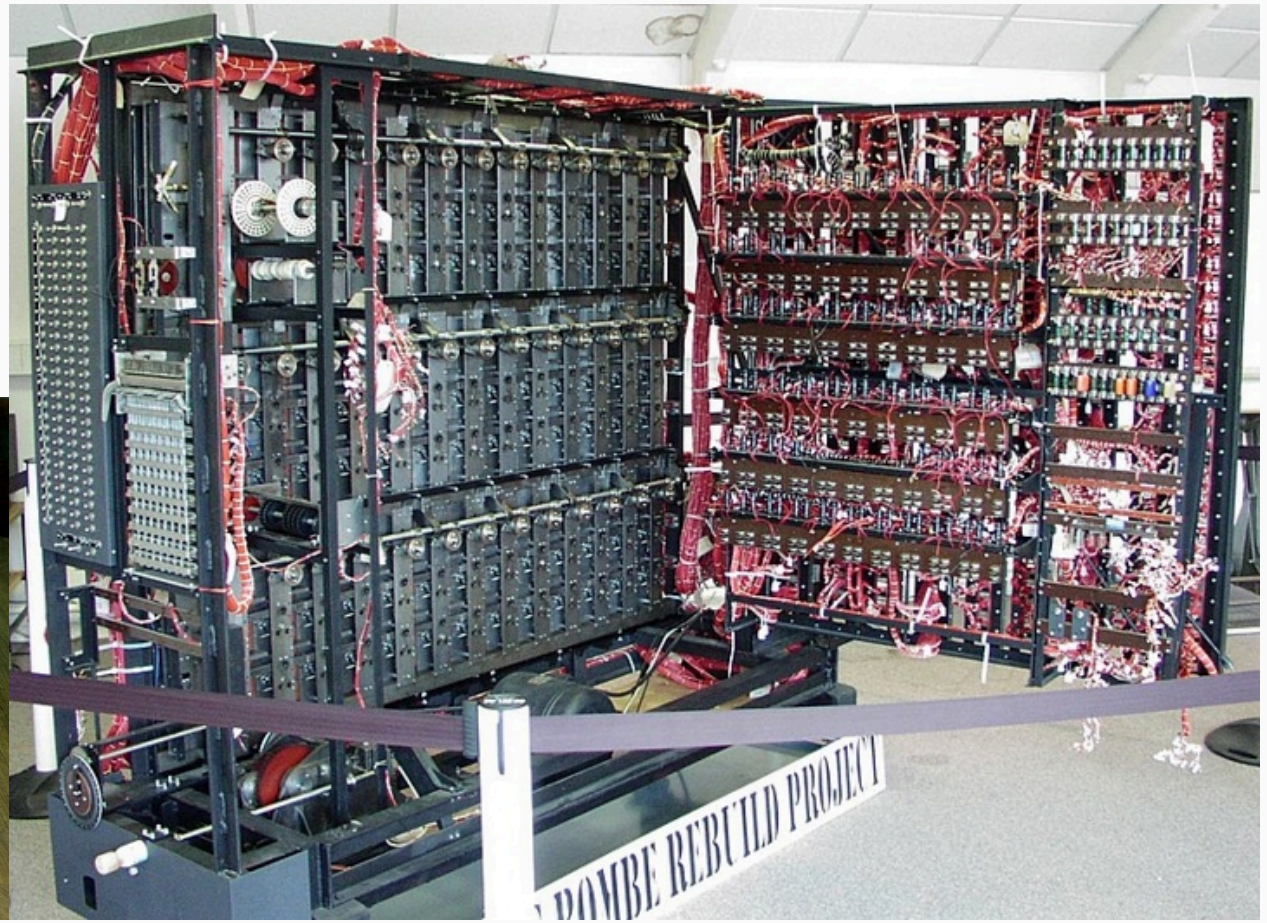
La révolution industrielle de la fin du XIXe siècle conduit à l'apparition de l'électricité et des moteurs qui améliorent les machines à calculer.

Par exemple, aux USA, **Herman Hollerith**, conçoit une machine qui divise par deux le temps nécessaire au recensement de la population. Sa société fusionnera pour devenir IBM.

XXe siècle : un essor fulgurant

- Avant 1936 : première machine capable de *parallélisme*
- 1936 : **Alonzo Church** et **Kurt Gödel** fournissent un cadre théorique et Alan Turing propose un concept théorique de machine qui est encore utilisé.
- Seconde guerre mondiale. L'information est chiffrée et circule alors massivement via les ondes radio et le télégraphe. Le déchiffrement devient un enjeu mondial. Citons Enigma, utilisée par les allemands, déchiffrée par Turing grâce à *la bombe*.

Enigma et la bombe



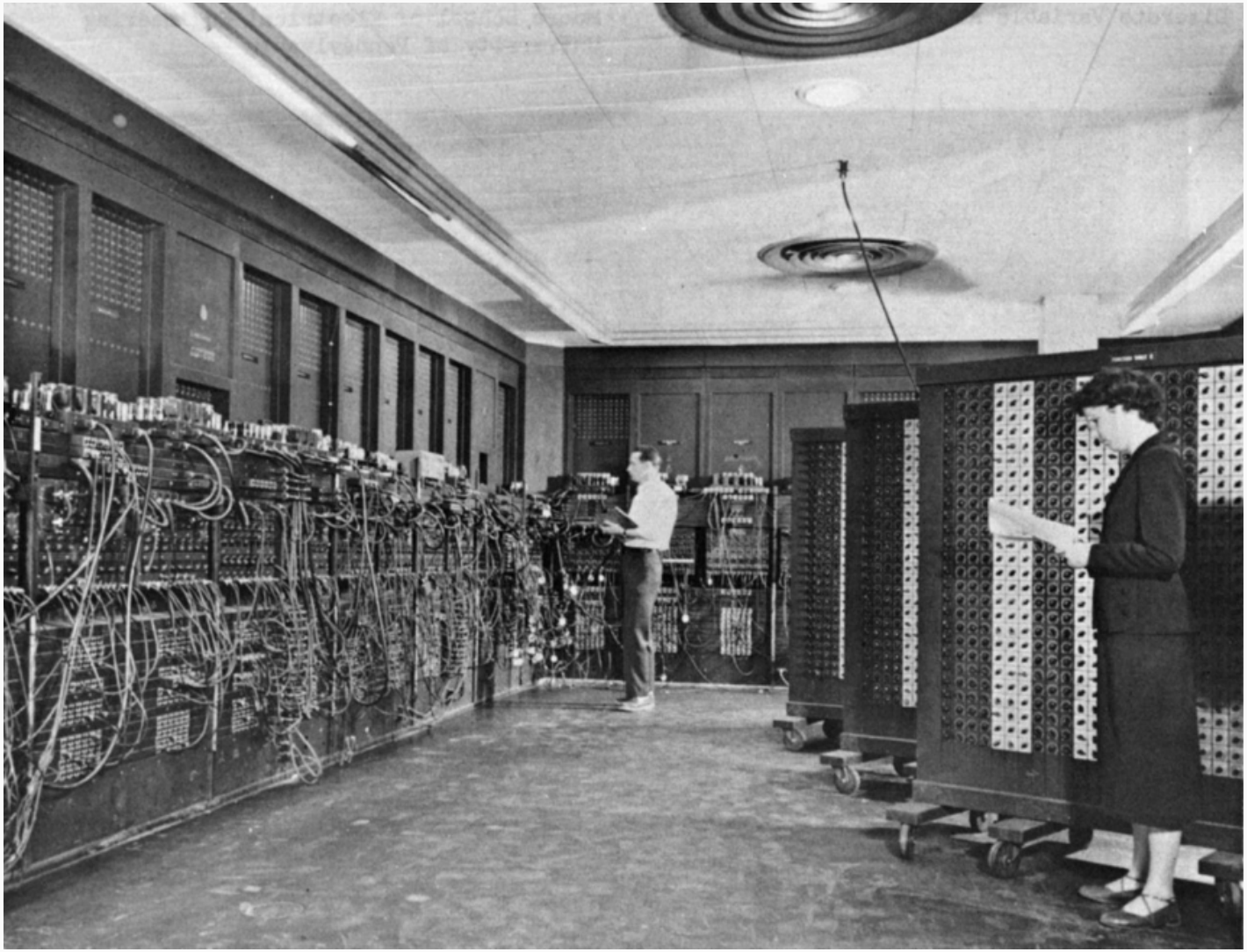
ENIAC

Les machines sont encore colossales !

- ENIAC : 30 tonnes, 167 m^2, 160 kW pour 100 kHz et 100 000 additions par secondes.
- Programme sur cartes perforées, entièrement électronique. Servant au calcul balistique. Programmé par six femmes.
- Calcul d'une trajectoire d'une table de tir. Comparatif :

Moyen	Temps
Homme à la main	2,6 j
Avec une machine à calculer	12 h
Model 5 (concurrent ENIAC)	40 min
ENIAC	3 s
PC ~2000	30 µs

l'ENIAC : un monstre



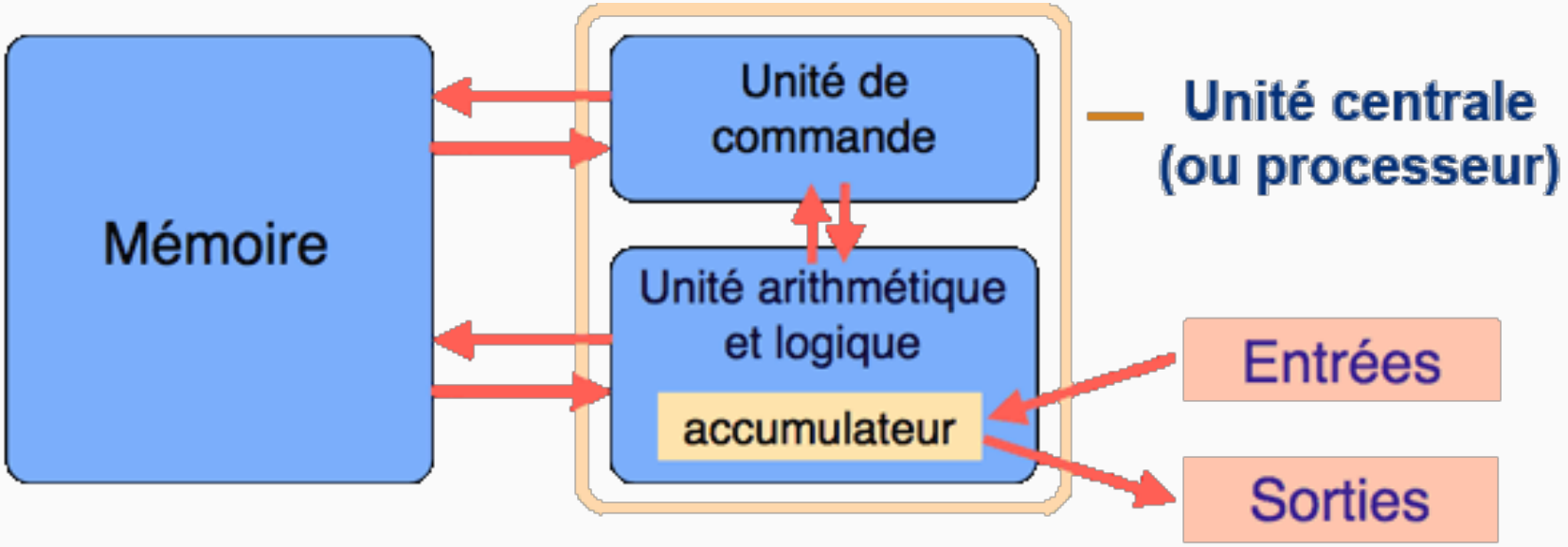
Modèle de Von Neumann

Un esprit universel

John Von Neumann : ingénierie, logique, mathématiques...
Participa au projet Manhattan (première bombe atomique) et à l'ENIAC

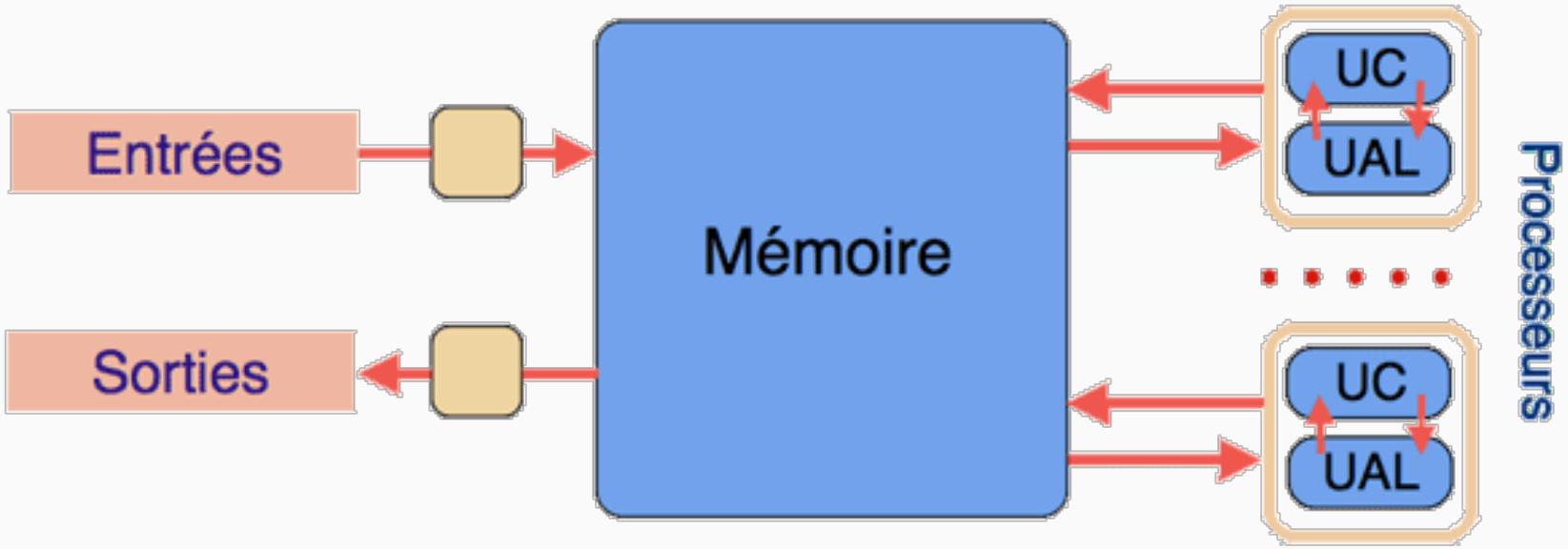
Il propose en 1944 un modèle *d'architecture* novateur qui sert toujours de base à nos architectures actuelles.

Présentation du modèle de Von Neumann



- **Unité de commande** : contrôle la séquence d'instruction
- **Unité arithmétique** : exécution de ces instruction
- **Processeur** : réalise les calculs
- **Mémoire** : contient les données et **les programmes**
- **Entrées** : clavier, cartes perforées, etc.
- **Sorties** : affichages, imprimantes, écran

Aujourd'hui : évolutions du modèle de Von Neumann



- **Entrée-sorties** : contrôlées par des processeurs autonomes
- **Multiprocesseur** : unités séparés ou multiples coeurs dans une même puce. Augmentation de la puissance sans augmenter la vitesse des processeurs, limitée par les capacités d'évacuation de chaleur
- **La mémoire au coeur du système** et de plus en plus de **parallélisme**.

L'assembleur, le C, Python

- 1954 : premier langage **assembleur** par IBM. Il sert d'intermédiaire entre le programme (*pour i allant de ...*) et le langage machine (*...01011001...*). Cela reste une langage très bas niveau (proche du métal)
- 1972 : Dennis Ritchie et Ken Thompson développent **le langage C** pour programmer le système d'exploitation **UNIX**. Rapide, à la syntaxe lisible, il est rapidement adopté.
- 1989 : Guido van Rossum crée Python qui deviendra le langage de script de référence.

Séquencement d'une instruction : Python, C, assembleur

Partons d'un programme simple et regardons sa traduction en langage machine

Les premiers nombres de la suite de Fibonacci en Python

```
x = 0
y = 1
while x < 255:
  print(x)
  x, y = y, y + x
```

Et en C

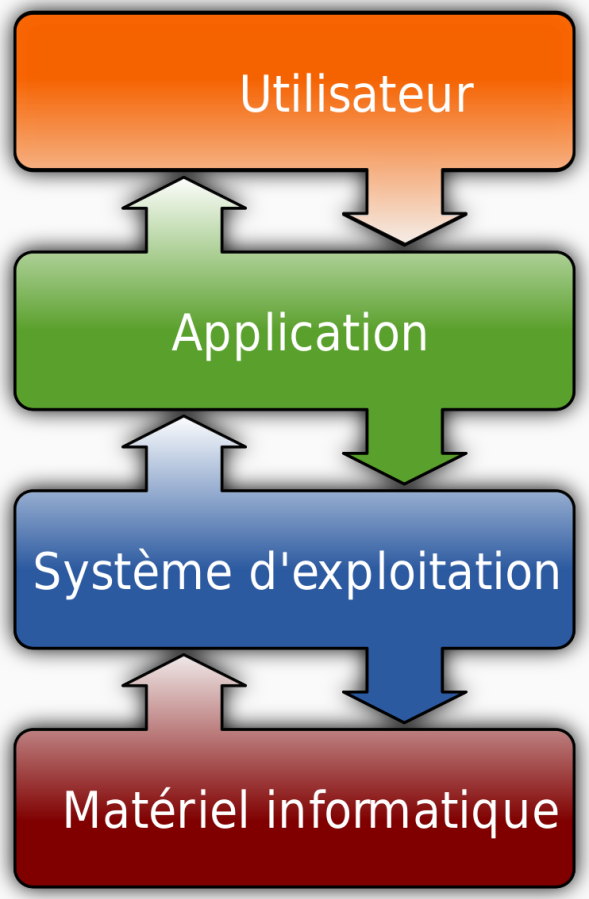
```c
# include <stdio.h>
int main(void){
  int x, y, z;
  while(1){
    x = 0;
    y = 1;
    do{
      printf("%d\n", x);
      z = y + x;
      x = y;
      y = z;
    } while (x < 255);
  }
}
```

En assembleur

- google drive : /dev/nsi/architecture matérielle : Vidéo avec soustitres fr
- La video de Bean Eater

Systèmes d'exploitation

Présentation



Le système d'exploitation est un intermédiaire entre les logiciels d'application et le matériel

Fonctionnalités

l'OS permet l'utilisation des applications utilisant les ressources de la machine.

- **Exécuter des programmes**
- **Partager les ressources** entre usagers, entre programmes
- Ressources :
  - **Périphériques** : chaque périph. a ses propres instructions
    Évite au programme de devoir écrire les siennes
  - **Fichiers** : tient compte du format de chaque fichier, protège les fichiers (droits d'utilisateurs)
  - **Détection, récupération d'erreurs**
  - **Contrôle** : surveillance des performances, des ressources etc.

UNIX vs Windows

On distingue deux grandes familles de systèmes d'exploitation : UNIX et windows.

Parmi UNIX on trouve, en particulier :

- macOS,
- GNU/Linux
- Android
- iOS
- FreeBSD, NetBSD etc.

Répartition

- **Ordinateurs personnels** : Windows (90% de part depuis 15 ans)
- **Mainframes et serveurs** : Linux, UNIX. (beaucoup de virtualisation depuis 10 ans)
- **Mobile** : Android (90% des smartphones)

UNIX et le Libre

Développé au Bell Labs dans les années 1960, UNIX s'est rapidement séparé en de nombreuses variantes (BSD, Solaris etc.) et de nombreux procès de propriété intellectuelle ont compliqué son développement dans les années 80/90.

Dès les années 80 **Richard Stallman** commence le développement de GNU.

En 1991 **Linus Torvald** crée le premier noyau linux à partir de GNU. Le succès est immédiat.

- GNU/Linux est un logiciel **open source** : le code source est consultable, éditable, utilisable etc. Des dizaines de milliers de développeurs ont participé à ce projet via internet

Cela s'oppose aux **sources fermées**, généralement propriété des entreprises.

Linux : Kernel et distribution

- Le noyau (**kernel**) d'un système UNIX comporte les fonctions de base du système d'exploitation. Il fournit aux logiciels l'interface de programmation pour utiliser le matériel
- le kernel est multitâche et multi-utilisateur. Il respecte les normes POSIX et ses programmes peuvent (ou devraient) fonctionner sur tous les systèmes UNIX
- La **distribution** est l'ensemble des logiciels qui s'ajoutent au noyau et rendent l'ordinateur utilisable. Par exemple : Ubuntu (PC), Raspbian (Raspberry Pi) ou Android (mobile)

Utilisation courante : CLI vs GUI

Command Line Interface

L'utilisateur tape dans un terminal ses commandes directement. Courante en informatique embarquée, administration de serveurs ou à distance

Graphical User Interface

Les fenêtre auxquelles nous sommes habitués. Le gestionnaire de fenêtre n'est qu'un programme qu'on peut changer rendant l'expérience utilisateur totalement différente.

Quelques préjugés

- *Linux c'est galère.*
  Au début. Ensuite c'est difficile de faire machine arrière. Chez moi personne le remarque. Essayez une machine virtuelle !
- *C'est impossible de jouer ou d'utiliser tel logiciel sous linux.*
  Impossible non, compliqué oui. Certains programmes (la suite adobe, solidworks) n'existent simplement pas sous linux mais ont des alternatives. Mon conseil est de garder un dualboot avec windows.
- *Linux c'est moche.*
  Là c'est faux, vous en faites ce que vous voulez.
- *Ça sert à rien.*
  Faux encore, ça sert à apprendre et pour programmer c'est bien mieux. Ensuite, et plus important, ça vous rend le contrôle de **votre** machine.

Réseaux

Préhistoire des réseaux : Arpanet

TCP / IP

Internet

Wifi, bluetooth

Objets embarqués

I/O

Périphériques d'entrée et de sortie

Interface Homme Machine