

Fonctionnement de la mémoire

Comment se présente la mémoire dans un ordinateur ?

- La mémoire peut être vue comme une très longue suite de cases appelées « **bits** ».
- Chaque **bit** a deux **états** possibles : 0 ou 1.
- L'emplacement d'une case dans la mémoire est l'**adresse**.
- Le processeur peut écrire ou lire la valeur d'un bit à une certaine adresse.
- Pourquoi la mémoire fonctionne comme ça ?
 - Les systèmes électroniques peuvent facilement détecter ou générer la présence ou l'absence d'un signal électrique, lumineux ou le sens de polarisation d'un aimant.

Comment représenter une information dans la mémoire?

- Avec n bits, on peut représenter 2^n informations différentes.
- Une information sur n bits \approx un mot à n lettres prises dans un alphabet à 2 lettres.
- On peut voir un groupe de k cases comme une lettre, et considérer que l'alphabet est plus grand (2^k lettres).
- Par exemple avec $k = 3$, une information sur $n = 12$ bits représente un mot de longueur 4 dans un alphabet ayant $2^3 = 8$ lettres.

Comment représenter du texte dans la mémoire ?

- Par exemple, si on veut un alphabet d'au moins 26 lettres, on veut regrouper les bits par groupes de taille k avec $2^k \geq 26$. On pourrait prendre $k = 5$ puisque $2^5 = 32$.
- Le codage ASCII code $128 = 2^7$ symboles possibles sur 7 bits et utilise un bit supplémentaire de contrôle. Donc $k = 8$ bits.
- Un symbole ASCII peut représenter une lettre latine majuscule, minuscule, un chiffre ou un symbole basique. ASCII s'est imposé comme la norme mondiale.
- L'unité la plus utilisée pour mesurer la taille de données est un **octet** = 8 bits = un **byte**.
 $1 \text{ o} = 1 \text{ B} = 8 \text{ b}$. Ce choix historique a été motivé par le codage ASCII.
- Avec un octet, on peut représenter caractère ASCII.
- Un **fichier texte** est un fichier dont le contenu est lu ou écrit au format ASCII.
- Un **fichier binaire** est un fichier qui n'est pas un fichier texte