

1. Représentation de l'information sous forme de bits

Bonjour à tous, bienvenue dans cette partie "informatique et ses fondements". Cette partie sera constituée de 4 sections : d'abord nous parlerons de codage binaire, ensuite d'algorithmique, de programmation et enfin d'architecture des ordinateurs et des réseaux.

Donc tout d'abord, la section codage binaire. Dans cette section, nous aurons 5 séquences. La première va consister à étudier la représentation de l'information sous forme de bits.

Tout d'abord, pourquoi coder de l'information? On a autour de nous énormément d'informations différentes, que ce soit des images, des sons, des bruits, la température, la quantité d'argent qu'on a dans notre porte-monnaie, notre âge, des dates, etc., énormément de choses différentes. Des choses qui vont être très diverses, plus ou moins structurées et qu'on va capter soit par nos sens humains, soit par des instruments.

La question qui se pose, c'est comment représenter ces informations pour les mettre dans l'ordinateur? Pour faire ça on va les numériser, les coder numériquement, c'est-à-dire les traduire en nombre qu'on va donner ensuite à l'ordinateur.

Au niveau électronique, dans les ordinateurs, ce qui se passe en pratique, c'est qu'on a du courant électrique qui va soit passer, soit ne pas passer. Donc on a des fils qui peuvent soit contenir du courant, soit ne pas en contenir. On a utilisé cette stratégie pour coder binaires en disant "si le courant ne passe pas, ça sera un 0, si le courant passe, ça sera un 1", du codage binaire, et donc c'est de là que vient le nom "bit" pour "Binary Digit", donc un bit va valoir 0 ou 1 selon si le courant passe ou non.

Les ordinateurs actuels sont basés sur l'électronique, donc tout ce que je viens de raconter, c'est basé sur l'électronique. Peut-être qu'à l'avenir ça va changer quand on va faire de la photonique ou de la quantique, mais pour l'instant on va s'intéresser à l'électronique, donc avec des bits qui valent 0 ou 1.

Avec ces bits on va pouvoir faire des opérations. L'opération la plus simple ça sera d'utiliser ce qu'on appelle un transistor. C'est un des composants de base de l'électronique. Le transistor, c'est un robinet qui va soit laisser passer le courant, soit ne pas le laisser passer, et pour commander ce robinet, on va avoir une 3ème borne sur laquelle va arriver du courant. Quand du courant va arriver, le robinet va laisser passer le courant, et quand du courant n'arrivera pas, le robinet va fermer le courant. Il y a différents types de transistors qui existent, en pratique le transistor dessiné ici c'est un transistor habituel, mais dans les microprocesseurs, on aura des transistors beaucoup plus petits, comme on en reparlera un peu plus loin.

Avec les transistors, on va pouvoir faire plein de choses, notamment faire des opérations sur les bits en utilisant ce qu'on appelle des portes logiques. Alors les portes logiques, c'est en fait différentes arrivées de courant qui vont être combinées pour calculer une sortie en fonction des arrivées, donc on va pouvoir faire des portes logiques de type "et", "ou" ou "non", etc. Donc par exemple, une porte "ou" comme dessinée ici, elle va recevoir 2 entrées qui vont contenir du courant ou non, et elle va générer une sortie qui va contenir du courant si une des 2 entrées contient du courant ou si les 2 entrées contiennent du courant. Donc c'est un "ou" logique qu'on effectue entre ces 2 entrées en fait, et donc là on a fait une opération sur les bits,

on les a combinés avec une opération "ou". Avec ça, on pouvoir faire d'autres opérations, on va pouvoir faire des portes "et" qui vont laisser passer uniquement si les 2 entrées sont activées, et d'autres choses. On peut faire des choses encore plus compliquées, comme par exemple des additionneurs de bits qui vont permettre d'ajouter 2 valeurs, 0 ou 1, pour calculer le résultat.

Donc ça, ça nous permet de manipuler des bits, mais les bits c'est assez limité vu qu'on est limités à 2 valeurs, 0 ou 1. Dans la pratique on va vouloir manipuler des nombres plus grands. Pour faire ça, on va assembler les bits pour coder des nombres plus grands. Donc par exemple, un octet c'est en fait un assemblage de 8 bits, donc chacun des bits va pouvoir valoir 0 ou 1, et donc en tout on va voir 256 valeurs possibles, en fait c'est 2 puissance 8, parce que chacun des bits a 2 valeurs possibles. Avec ce genre d'assemblage, on va pouvoir coder plein de choses. Par exemple si on voulait coder un âge, on va pouvoir dire "un âge c'est en gros entre 0 et 127", ça en fait ça peut se coder avec 7 bits. Si on veut coder une température, là c'est un peu plus compliqué parce que la température va pouvoir varier entre beaucoup plus que 0 et 127, elle va pouvoir être négative éventuellement aussi, voire être une température à virgule, si on prend la température humaine quand on regarde si on est malade, elle peut être de 37,6. Donc il va falloir étendre les nombres manipulés par l'ordinateur, pas simplement à avoir des octets qui vont être limités à 256 valeurs, mais avoir une plage de valeurs beaucoup plus grande. Donc en pratique on verra plus loin qu'on a différents grands types de nombres manipulés par l'ordinateur, qui vont être des entiers de différentes tailles, éventuellement des nombres qu'on va appeler "flottants", c'est des nombres décimaux, éventuellement négatifs, etc., et l'ordinateur va savoir manipuler directement ces nombres, il a été programmé pour ça.

Alors en pratique, l'ordinateur ne sait pas trop ce qu'il manipule. Il sait qu'il y a des nombres, mais il ne sait pas du tout ce qu'ils signifient. Par exemple, c'est le programmeur qui va devoir lui donner les nombres de la bonne façon. Si on veut demander à l'ordinateur par exemple de calculer, d'ajouter 2 distances et ensuite de diviser par le temps pour calculer une vitesse, il faudra que le programmeur ou l'utilisateur fasse attention à donner les distances dans la bonne unité, la durée dans la bonne unité, pour que l'ordinateur fasse le calcul et que le résultat soit cohérent. Si on lui donne en kilomètres en entrée et en heure, il va falloir savoir que l'ordinateur va nous donner le résultat en kilomètres/heure. Donc ça, c'est vraiment le gros problème, c'est que l'ordinateur est bête, il ne va pas pouvoir deviner ce qu'on lui demande de faire, il va falloir que nous on lui demande d'une façon très précise et qu'on interprète le résultat en sortie de manière très précise.

On a vu comment coder rapidement les nombres, maintenant on va regarder d'autres choses, comment par exemple coder du texte. Ici on a une lettre qu'on a représentée en affichant des points à l'écran. Il y a différentes façons de stocker une lettre. On pourrait dire "si je veux stocker ma lettre ici, j'ai des cases qui sont blanches, des cases qui sont noires", ce que je vais faire, c'est que je vais utiliser un bit pour chaque case. Si le bit est à 0, je vais considérer que la case est noire, si le bit est à 1, la case va être blanche. Donc ça, ça me permet de représenter l'affichage de ma lettre A en décrivant chacun des points à l'écran.

Le problème, c'est que ça ne suffit pas parce que quand je vais avoir ma liste de bits qui indique si les cases sont blanches ou noires, il va falloir savoir est-ce que j'ai 5 colonnes et 6 lignes ou est-ce que j'ai une seule colonne avec 30 lignes par exemple? Donc il va falloir ajouter de l'information pour préciser à quoi ressemble ma lettre en pratique. Si on

commence à ajouter de la couleur, ça va être plus compliqué aussi parce qu'un seul bit par case ne va plus suffire.

En pratique il y a plein de problèmes qui se posent dès qu'on commence à parler de l'affichage en fait, donc ça on y reviendra un peu plus loin. Donc en pratique, on ne va pas utiliser ce genre de stockage. Ce qu'on va faire, c'est qu'on va attribuer à chaque lettre un numéro et on va utiliser ce numéro pour représenter les lettres, indépendamment du fait qu'à l'affichage à l'écran, il faudra utiliser différents points, blancs ou noirs, etc.

Donc en fait, il y a quelque chose qui s'appelle le "code ASCII", qui a été défini il y a très longtemps et qui va permettre de donner un numéro à chacune des lettres, donc les lettres minuscules, les lettres majuscules, les chiffres, la ponctuation. Donc en gros il y a une centaine de caractères possibles. En pratique, le code ASCII est codé sur 8 bits, donc 8 bits, 2 puissance 8, ça nous donne 256 valeurs possibles. Et le code ASCII va permettre de coder toutes les lettres de l'alphabet sans problème, et donc on va enchaîner les différents caractères ASCII pour stocker de l'information contenue dans un texte, chacune des lettres ou des points de ponctuation à la suite. Alors pour des langues asiatiques notamment, 256 lettres possibles ça ne suffit pas, donc on va utiliser d'autres systèmes de nos jours qui s'appellent par exemple Unicode, mais le principe reste le même, on a un numéro par caractère possible qu'on va vouloir manipuler dans le texte.

Donc en fait, le dessin, l'affichage de la lettre à l'écran, il est important uniquement quand on veut vraiment afficher, mais quand on veut stocker l'information du texte, on va simplement utiliser un numéro par lettre, pour dire "mon texte est constitué de telles lettres à la suite", mais on n'a pas besoin de savoir comment ça va s'afficher à l'écran, jusqu'à ce qu'on veuille, soit l'afficher, soit l'imprimer. Donc c'est quelque chose qui va être différent, qui va être géré indépendamment.

Alors qu'est-ce qu'il faut retenir de cette première séquence? Tout d'abord le courant électrique est utilisé dans l'informatique actuelle pour coder les bits avec des valeurs de 0 ou 1, et on va assembler ces bits pour créer des nombres plus grands, par exemple en utilisant un octet qui est constitué de 8 bits, qui permet d'avoir des nombres jusqu'à 255. Ensuite on va pouvoir faire des nombres plus grands en ajoutant encore plus de bits, ou coder d'autres choses comme les lettres. En pratique, les circuits électroniques savent faire des opérations avec ces nombres, parce que les ordonnateurs ont été conçus pour. Par contre ils ne savent pas ce qu'ils manipulent, ils savent que c'est des nombres qu'ils ajoutent, mais ils ne savent pas que c'est des nombres qui correspondent à une vitesse ou à un âge ou à autre chose.