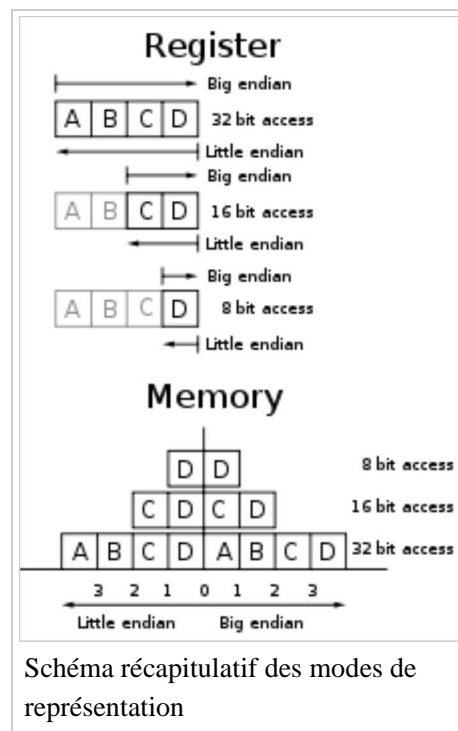


Endianness

En informatique, certaines données telles que les nombres entiers peuvent être représentées sur plusieurs octets. L'ordre dans lequel ces octets sont organisés en mémoire ou dans une communication est appelé ***endianness*** (mot anglais traduit par « **boutisme** »¹ ou par « **endianisme** »).

De la même manière que certains langages humains s'écrivent de gauche à droite, et d'autres s'écrivent de droite à gauche², il existe une alternative majeure à l'organisation des octets représentant une donnée : l'orientation ***big-endian*** et l'orientation ***little-endian***. Ces expressions sont parfois traduites par **gros-boutiste** et **petit-boutiste** ou par **grand-boutien** et **petit-boutien**. Les expressions *byte order*, d'**ordre des octets** ou de *byte sex* sont également utilisées (bien qu'**ordre des octets** fasse référence à l'unité d'une base numérale précise sur 8 bits, que les autres termes plus généraux ne traduisent pas).

L'endianness qualifie aussi bien un fichier (dans lequel ce sont les bytes qui sont ordonnés différemment) qu'un processeur (dans lequel la gestion des bits a aussi un ordre).



Sommaire

- 1 Étymologie
- 2 Dans les ordinateurs
 - 2.1 Big endian
 - 2.2 Little endian
 - 2.3 Bi-endian
 - 2.4 Middle-endian
 - 2.5 Ordre des bits
- 3 Dans les systèmes d'exploitation (par architectures)
- 4 Dans les communications
- 5 Différences pratiques
- 6 Logiciels et portabilité
- 7 Écriture des nombres dans les langues humaines
- 8 Écriture des dates
- 9 Voir aussi
 - 9.1 Notes et références
 - 9.2 Article connexe
 - 9.3 Lien externe

Étymologie

Les termes *big-endian* et *little-endian* ont été popularisés dans le domaine informatique par Dany Cohen³, en référence aux Voyages de Gulliver, roman satirique de Jonathan Swift. En 1721, Swift décrit comment de nombreux habitants de Lilliput refusent d'obéir à un décret obligeant à manger les œufs à la coque par le petit bout. La répression pousse les rebelles, dont la cause est appelée *big-endian*⁴, à se réfugier dans l'empire rival de Blefuscu ce qui entretient une guerre longue et meurtrière entre les deux empires.

En 1980, Cohen publie « une tentative pour arrêter une guerre³ », celle qui oppose les partisans *big-endian* et *little-endian* au sein du groupe qui travaille sur les protocoles réseau qui conduiront à Internet. Dans sa note technique, il indique qu'aucune argumentation logique ne peut montrer la supériorité d'une convention sur l'autre ; cependant, dans le domaine informatique, un choix doit être fait pour éviter l'anarchie.

La première traduction en français du roman de Swift⁵ utilise le terme *gros-boutien*⁶. L'Office québécois de la langue française préconise *gros-boutiste* et *petit-boutiste*, ce qui conduit au nom *boutisme* adopté par au moins un ouvrage d'informatique⁷ pour traduire *endianness*.

En informatique, le suffixe *-iste* est employé de préférence à *-ien*^[réf. nécessaire]⁸.

Dans les ordinateurs

Big endian

Quand certains ordinateurs enregistrent un entier sur 32 bits en mémoire, par exemple 0xA0B70708 en notation hexadécimale, ils l'enregistrent dans des octets dans l'ordre qui suit : A0 B7 07 08, pour une structure de mémoire basée sur une unité atomique de 1 octet et un incrément d'adresse de 1 octet. Ainsi, l'octet de poids le plus fort (ici A0) est enregistré à l'adresse mémoire la plus petite, l'octet de poids inférieur (ici B7) est enregistré à l'adresse mémoire suivante et ainsi de suite.

	0	1	2	3	
...	A0	B7	07	08	...

Pour une structure de mémoire ou un protocole de communication basé sur une unité atomique de 2 octets, avec un incrément d'adresse de 1 octet, l'enregistrement dans des octets sera : A0B7 0708. L'unité atomique de poids le plus fort (ici A0B7) est enregistré à l'adresse mémoire la plus petite.

	0	1	2	3	
...	A0	B7	07	08	...

Les architectures qui respectent cette règle sont dites *big-endian* ou **gros-boutistes** ou **mot de poids fort en tête**, par exemple les processeurs Motorola 68000, les SPARC (Sun Microsystems) ou encore les System/370 (IBM).

Little endian

Les autres ordinateurs enregistrent 0xA0B70708 dans l'ordre suivant : 08 07 B7 A0 (pour une structure de mémoire basée sur une unité atomique de 1 octet et d'un incrément d'adresse de 1 octet), c'est-à-dire avec l'octet de poids le plus faible en premier. De telles architectures sont dites *little-endian* ou **petit-boutistes** ou **mot de poids faible en tête**. Par exemple, les processeurs x86, qui se trouvent dans les PC ont une architecture petit-boutiste.

	0	1	2	3	
...	08	07	B7	A0	...

...	08	07	B7	A0	...
-----	----	----	----	----	-----

Pour une structure de mémoire ou un protocole de communication basé sur une unité atomique de 2 octets, avec un incrément d'adresse de 1 octet, l'enregistrement dans des octets sera : 0708 A0B7. L'unité atomique de poids le plus faible (ici 0708) est enregistré à l'adresse mémoire la plus petite.

	0	1	2	3	
...	07	08	A0	B7	...

Bi-endian

Certaines architectures supportent les deux règles, par exemple les architectures PowerPC (IBM), ARM, DEC Alpha, MIPS, PA-RISC (HP) et IA-64 (Intel). On les appelle *bytesexual* (jargon), *bi-endian* ou, plus rarement, **biboutistes**. Le choix du mode peut se faire au niveau logiciel, au niveau matériel ou aux deux.

Middle-endian

Certaines autres rares architectures, appelées *middle-endian*, ont un ordonnancement plus complexe: les octets composant les unités atomiques subissent une opération de *swap*. Par exemple 0xA0B70708 est enregistré dans une mémoire middle-endian dont les unités atomiques sont de 2 octets, avec un incrément d'adresse de 1 octet, dans l'ordre : 0807 B7A0 ou bien B7A0 0807.

	0	1	2	3		
...	07	08	A0	B7	...	middle-endian, unité atomique 2-octets, incrément d'adresse 1-octet

ou alternativement

	0	1	2	3		
...	B7	A0	08	07	...	middle-endian, unité atomique 2-octets, incrément d'adresse 1-octet

Il existe une ambiguïté dans la représentation de cette donnée. En effet l'information d'endianness sur la manière d'ordonner les unités atomiques existe toujours bel et bien. Au terme de middle-endian, on utilise donc plutôt les termes de big-endian ou little-endian associé à une caractéristique de *byte-swap*. L'exemple devient dès lors non ambigu :

1. dans une mémoire little-endian avec *byte-swap*, 2 octets d'unité atomique, 1-octet d'incrément d'adresse, 0xA0B70708 est représenté par 0807B7A0, 08 étant à l'adresse 0.

	0	1	2	3		
...	07	08	A0	B7	...	little-endian, <i>byte-swap</i> , unité atomique 2 octets, incrément d'adresse 1 octet

2. dans une mémoire big-endian avec *byte-swap*, 2 octets d'unité atomique, 1 octet d'incrément d'adresse, 0xA0B70708 est représenté par B7A00807, B7 étant à l'adresse 0.

	0	1	2	3		
...	B7	A0	08	07	...	big-endian, <i>byte-swap</i> , unité atomique 2 octets, incrément d'adresse 1 octet

Il est plus difficile de travailler avec de tels processeurs, les PDP-11 par exemple.

Ordre des bits

Dans une architecture *big-endian*, les bits sont conventionnellement numérotés de la gauche vers la droite. Ainsi le bit de poids le plus fort est le bit 0, et le bit de poids le plus faible est le 7 dans un octet. Dans une architecture *little-endian*, c'est le contraire.

Si l'octet doit représenter un nombre entier, alors la convention *big-endian* peut porter à confusion car le poids du bit n est alors 2^{7-n} (au lieu de 2^n avec la convention *little-endian*). En revanche, elle convient mieux pour représenter une fraction binaire, le poids du bit n étant alors 2^{-n} (contre 2^{n-7} pour *little-endian*).

Un moyen mnémotechnique pour ne pas confondre les deux notations consiste à remplacer « endian » par « head ». On a alors :

- « big head » pour les bits de poids « fort en tête »,
- « little head » pour les bits de poids « faible en tête ».

Dans les systèmes d'exploitation (par architectures)

Little-endian (OS sur architecture Little Endian):

- DragonFly BSD sur x86, et x86-64
- FreeBSD sur x86, x86-64, MIPS, ARM et Itanium
- Linux sur x86, x86-64, MIPSEL, Alpha, Itanium, S+core, MN103, CRIS, Blackfin, Microblaze**EL**, ARM, M32**REL**, TILE, SH, Xtensa**EL** et UniCore32
- Mac OS X sur x86, x86-64
- iOS sur ARM
- NetBSD sur x86, x86-64, Itanium, etc.
- OpenBSD sur x86, x86-64, Alpha, VAX, Loongson (MIPSEL)
- OpenVMS sur VAX, Alpha et Itanium
- Solaris sur x86, x86-64, PowerPC
- Tru64 UNIX sur Alpha
- ESX sur x86, x86-64
- Windows sur x86, x86-64, Alpha, PowerPC, MIPS et Itanium

Big-endian (OS sur architecture Big Endian):

- AIX sur POWER
- AmigaOS sur PowerPC et 680x0
- FreeBSD sur MIPS, ARM, PowerPC et SPARC
- HP-UX sur Itanium et PA-RISC
- IRIX sur MIPS
- Linux sur MIPS, SPARC, PA-RISC, POWER, PowerPC, 680x0, ESA/390, z/Architecture, H8, FR-V, AVR32, Microblaze, ARME**B**, M32R, SHE**B**, Xtensa et ubicom32.
- Mac OS sur PowerPC and 680x0
- Mac OS X sur PowerPC
- NetBSD sur PowerPC, SPARC, etc.
- OpenBSD sur PowerPC, SPARC, PA-RISC, SGI (MIPSE**B**), Motorola 68k et 88k, Landisk (SuperH-4)
- MVS and DOS/VSE sur ESA/390, and z/VSE et z/OS on z/Architecture

- Solaris sur SPARC

Dans les communications

On appelle cela le problème NUXI, en effet si on veut envoyer la chaîne « UNIX » en regroupant deux octets par mot entier de 16 bits sur une machine de convention différente, alors on obtient NUXI. Ce problème a été découvert en voulant porter une des premières versions d'Unix d'un PDP-11 *middle-endian* sur une architecture IBM *big-endian*.

Le protocole IP définit un standard, le *network byte order* (soit ordre des octets du réseau). Dans ce protocole, les informations binaires sont en général codées en paquets, et envoyées sur le réseau, l'octet de poids le plus fort en premier, c'est-à-dire selon le mode *big-endian* et cela quel que soit l'*endianness* naturel du processeur hôte.

Les périphériques doivent aussi respecter une convention afin d'assurer la cohérence du système. Tout cela est fixé par le protocole de la couche de liaison du modèle OSI.

Différences pratiques

Bien que la différence entre les deux modes *big-endian* et *little-endian* semble aujourd'hui minime et se limite à un problème de convention, on peut signaler des avantages liés à chacun :

Le mode *little-endian* présentait des avantages lorsque les processeurs utilisaient des tailles de registre variables, c'est-à-dire 8, 16 ou 32 bits. À partir d'une adresse mémoire donnée, on pouvait lire le même nombre en lisant 8, 16 ou 32 bits.

Par exemple, le nombre 33 (0x21 en hexadécimal) s'écrit *21 00 00 00* en little endian en 32 bits, ce qui se lit toujours 21 quel que soit le nombre d'octets lus. Ceci est faux en *big-endian* car la première adresse change suivant le nombre d'octets à lire.

Logiciels et portabilité

On a bien compris que ces conventions posent des problèmes dans le portage des logiciels. Par exemple, en lisant des données binaires, selon l'architecture, on ne va pas obtenir la même donnée après lecture si on ne se soucie pas de la convention.

Bien sûr le choix de *big-endian* ou *little-endian* est toujours arbitraire, ce qui soulève des débats intenses, car il y a nombre d'arguments en faveur de l'un et de l'autre. Les langues par exemple, selon le groupe linguistique germanique, anglais ou autre, n'ont pas la même perception.

Écriture des nombres dans les langues humaines

Les langues humaines n'ont pas toutes la même convention pour la transcription des nombres.

Le produit de trois par sept a la même valeur en France et en Allemagne. Ce nombre se prononce « vingt-et-un » en français (ce qui correspond à une convention *big-endian*) ; en allemand, il se dit « einundzwanzig », c'est-à-dire « un et vingt ». Dans chacune de ces langues, la convention d'écriture en toutes lettres reflète la prononciation orale.

En arabe, quand on écrit un nombre en chiffres en suivant le sens habituel d'écriture de l'arabe, on commence par écrire le chiffre des unités, puis celui des dizaines, pour finir avec le chiffre de poids le plus fort : ce qui est analogue à une convention *little-endian*. Comme l'arabe s'écrit de droite à gauche, le chiffre des unités est à droite de la page. À l'opposé, pour écrire un nombre en chiffres dans un texte en français, on commence par le chiffre de poids le plus fort, pour terminer par le chiffre des unités : c'est donc une convention *big-endian*, par rapport au sens d'écriture usuel du français. Néanmoins, comme le français et l'arabe s'opposent à la fois sur l'*endianness* et sur leur sens d'écriture usuel, dans les deux cas, le chiffre des unités est à droite de la page, et le chiffre de poids le plus fort se trouve à gauche.

Écriture des dates

Certains pays ont des standards concernant l'écriture des dates. La notion d'endianness y est présente comme le montrent les exemples suivants :

- Europe : JJ/MM/AAAA (little endian)
- Japon : AAAA/MM/JJ (big endian)
- États-Unis : MM/JJ/AAAA (middle endian), AAAA-MM-JJ (big endian)

Voir aussi

Notes et références

- ↑ Le terme **boutisme** est encore peu utilisé, bien que la référence étymologique est établie et les traductions de Gulliver connues en français depuis le XVIII^e siècle, et la dérivation du mot « bout » est aussi très utilisée dans les termes « jusqu’au-boutisme » et « jusqu’au-boutiste », par extension naturelle du mot français « bout » ; de plus le mot français « bout » admet ces extensions par préfixe et/ou suffixe comme dans les termes « rabouter », « rebouteux », « bouton », etc.
- ↑ Les langages qui s'écrivent de haut en bas ne représentent pas une catégorie différente, du point de vue logique. L'organisation des données par rapport à la page qui les contient est identique, à une rotation près. (C'est-à-dire que les données de poids forts par rapport à l'orientation de la page sont toujours dans l'une ou l'autre de ces catégories)
- ↑ Danny Cohen: *On Holy Wars and a Plea for Peace*. (http://www.ietf.org/rfc/ien/ien137.txt) IETF, avril 1980.
- ↑ texte original (http://en.wikisource.org/wiki/Gulliver%27s_Travels/Part_I/Chapter_IV) sur wikisource.
- ↑ traduction par Pierre-François Guyot Desfontaines, publiée en 1727.
- ↑ Le voyage de Gulliver à Lilliput (http://fr.wikisource.org/wiki/Les_Voyages_de_Gulliver_:_Voyage_%C3%A0_Lilliput) traduit sur wikisource.
- ↑ Le terme est utilisé dans le livre Unicode 5.0 en pratique paru chez Dunod en 2008. Voir l'index du livre (http://hapax.qc.ca/dunod/Unicode_Index.pdf).
- ↑ Le débat n’est pas encore tranché, la guerre entre les boutistes et les boutiens, sur les forums et listes de discussions Internet, n’étant pas moins féroce qu’à Lilliput, et les partisans de chaque camp prenant référence de chaque côté à une traduction française publiée ou une autre des mêmes Voyages de Gulliver !

Article connexe

- Alignement de données

Lien externe

- (en) [**PDF**] White Paper: Endianness or Where is Byte 0? (http://3bc.bertrand-blanc.com/endianness05.pdf)

Ce document provient de « <http://fr.wikipedia.org/w/index.php?title=Endianness&oldid=92608112> ».

Dernière modification de cette page le 30 avril 2013 à 08:39.

Droit d'auteur : les textes sont disponibles sous licence Creative Commons paternité partage à l’identique ; d’autres conditions peuvent s’appliquer. Voyez les conditions d’utilisation pour plus de détails, ainsi que les crédits graphiques. En cas de réutilisation des textes de cette page, voyez comment citer les auteurs et mentionner la licence.

Wikipedia® est une marque déposée de la Wikimedia Foundation, Inc., organisation de bienfaisance régie par le paragraphe 501(c)(3) du code fiscal des États-Unis.