

Реферат

До проекту №19

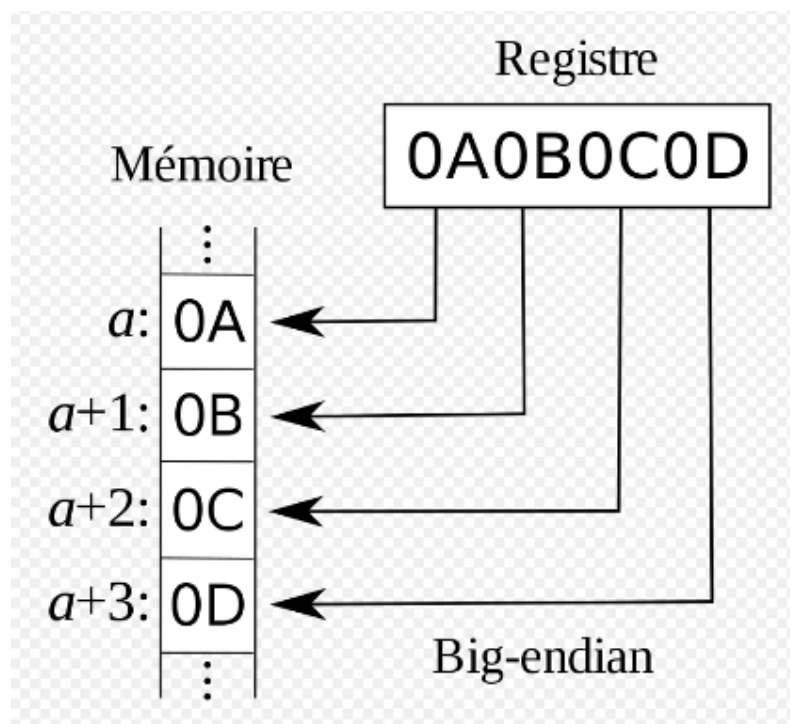
Радченко Ірина
Статистика-2

Порядок байтів — метод запису байтів багатобайтових чисел в інформатиці. У програмуванні, інформація в пам'яті зазвичай зберігається у двійкових даних, розділена на 8 біт.

Взагалі існує 2 варіанти запису порядку байтів big-endian та little-endian.

Ці два терміни запозичено у Джонатана Свіфта з його сатиричного твору «Мандри Гуллівера», де описуються держави Ліліпутія і Блефуску, що вели між собою протягом багатьох років війни через розбіжності з приводу того, з якого кінця слід розбивати варені яйця («гострого» чи «тупого» кінця («little-» and «big-endians»)).

Порядок від старшого до молодшого (big endian) вказує на організацію цифрових даних, яка починається з «великого» кінця слова даних і продовжується в напрямку «маленького» кінця, де «великий» і «маленький» відповідають найбільшому значущому і найменшому значущому бітам відповідно. Цей порядок є стандартним для протоколів TCP/IP, він використовується в заголовках пакетів даних і в багатьох протоколах більш високого рівня, розроблених для використання поверх TCP/IP. Тому, порядок байтів від старшого до молодшого часто називають мережевим порядком байтів (англ. network byte order). Цей порядок байтів використовується процесорами IBM 360/370/390, Motorola 68000, SPARC (звідси третя назва — порядок байтів Motorola, Motorola byte order).



Приклад:

Далі наведено приклад, в якому описується розміщення 4-байтового числа в ОЗП ЕОМ, доступ до якого може проводитися і як до 32-розрядного слова, і побайтно.

Всі числа записані в 16-ковій системі числення.

Число: 0xA1B2C3D4

| | | |
|-----------------------------------|-----------------|--|
| Представлення | | $D4 \cdot 0x01 + C3 \cdot 0x100 + B2 \cdot 0x10000 + A1 \cdot 0x1000000$ |
| Порядок від молодшого до старшого | (little-endian) | 0xD4, 0xC3, 0xB2, 0xA1 |
| Порядок від старшого до молодшого | (big-endian) | 0xA1, 0xB2, 0xC3, 0xD4 |

Незважаючи на сатиричне трактування боротьби «big endians» (прямого порядку, від старшого до молодшого) проти «little endians» (зворотного порядку, від молодшого до старшого), питання порядку байтів насправді дуже важливий для нашої роботи з даними.

Блок цифрової інформації – це послідовність одиниць та нулів. Ці одиниці та нулі починаються з найменшого значущого біта (least significant bit, LSb – зверніть на малу літеру «b») і закінчуються на найбільшому значущому біті (most significant bit, MSb).

Це видається досить простим. Розглянемо наступний гіпотетичний сценарій.

32-розрядний процесор готовий до збереження даних, а отже передає 32 біти даних у відповідні 32 блоки пам'яті. Цим 32 блокам пам'яті призначається адресу, скажімо 0x01. Шина даних в системі спроектована таким чином, що немає можливості змішувати LSb з MSb, і всі операції використовують 32-бітові дані, навіть якщо відповідні числа можуть бути легко представлені 16 або навіть 8 бітами. Коли процесору потрібно отримати доступ до збережених даних, він зчитує 32 біти з адреси пам'яті 0x01. Ця система є надійною, і не потрібно вводити поняття порядку байтів.

Можливо, ви помітили, що слово байт в описі цього гіпотетичного процесора ніде не згадувалося. Все засноване на 32-бітних даних - навіщо потрібно ділити ці дані на 8-бітові частини, якщо все обладнання призначене для обробки 32-бітових даних? Ось тут теорія і реальність розходяться. Реальні цифрові системи, навіть ті, які можуть безпосередньо обробляти 32-бітові або 64-бітові дані, широко використовують 8-бітний сегмент даних, відомий як байт.