Aluno: Miller Raycell Monteiro Correia

Matricula: 2017009560

LITTLE ENDIAN E BIG ENDIAN

Ordenação dos bytes determina como a data será lida ou escrita na memória e como ela será processada, existem dois tipos mais comuns de ordenação de bytes, sendo big endian e little endian. O big endian se trata de do armazenamento do byte mais significativo na primeira posição da sua respectiva representação, por exemplo de se uma palavra armazenada ocupa os bytes de 600 a 603, o byte mais significativo será armazenado no espaço 600 e o byte menos significativo será armazenado na posição 603, já no little endian a forma de operar é oposta, o byte menos significativo será armazenado na posição baixa da sua respectiva representação. Reaproveitando o exemplo, no modelo little endian o byte menos significativo ficaria armazenado na posição 600 e o mais significativo na posição 603.

Cada arquitetura possui uma forma de classificar e separar os bytes mais e menos significativos. As representações de little e big endian tem as suas diferenças entre si, por exemplo o modelo big endian é muito utilizado na linguagem humana para quando se é realizado os estudos práticos, pois como o byte mais significativo está na primeira posição da sua representação, quando tratamos de números e operações é melhor tratar a representação big endian, que é mais simples para a compreensão humana, mas quando se trata de nível de máquina a melhor forma de expressar algumas operações como de números com ponto flutuante é com o little endian, pois a sua forma de representação auxilia a unidade lógica aritimetica .

Desta forma pode=se perceber que cada forma de representação de bytes ordenados tem suas especifidades e desta forma, alguns projetistas de computadores usam a forma bi-endian, que se trata de o processador e os bus funcionarem tanto com little endian como com big endian, dessa forma pode-se ter uma melhor aproveitamento dessas duas formas cada uma sendo usada na área que vá ajudar nas atividades a serem desenvolvidas pelo computador.