

Lab 5

Representation: Testing Big Endian Vs. Little Endian

Purpose

To learn how the integer representation used by the underlying hardware affects programming and data layout.

Background Reading And Preparation

Read Chapter 3 to learn about big endian and little endian integer representations and the size of an integer.

Overview

Write a C program that examines data stored in memory to determine whether a computer uses big endian or little endian integer representation.

Procedure And Details (checkmark as each is completed)

1. Write a C program that creates an array of bytes in memory, fills the array with zero, and then stores integer 0x04030201 in the middle of the array.
2. Examine the bytes in the array to determine whether the integer is stored in big endian or little endian order.
3. Compile and run the program (without changes to the source code) on both a big endian and little endian computer, and verify that it correctly announces the integer type.
4. Add code to the program to determine the integer size (hint: start with integer 1 and shift left until the value is zero).
5. Compile and run the program (without changes to the source code) on both a thirty-two bit and a sixty-four bit computer, and verify the program correctly announces the integer size.

Optional Extensions (checkmark as each is completed)

- 6. Find an alternate method of determining the integer size.
 - 7. Implement the alternate method to determine integer size, and verify that the program works correctly.
 - 8. Extend the program to announce the integer format (i.e., one's complement or two's complement).
-

Notes