endianness

Description

Here's a file that was recovered from a 32-bits system

that organized the bytes a weird way. We're not even

sure what type of file it is.

Download it here and see what you can get out of it

Solution

To represent the word "dopyz" in both **little-endian** and **big-endian** formats, you need to convert each character to its ASCII value, then represent those values in hexadecimal. Finally, arrange the bytes in the respective endian order.

1. **ASCII values of the characters**:
    - 'd' = 100
    - 'o' = 111
    - 'p' = 112
    - 'y' = 121
    - 'z' = 122

Corresponding hexadecimal values:

- 'd' = 64
- 'o' = 6F
- 'p' = 70
- 'y' = 79
- 'z' = 7A

2. **Combine the hexadecimal values into a sequence**:
    - Hex sequence for "dopyz" = 64 6F 70 79 7A
3. **Endian Formats**:
    - **Big-endian**: Store the bytes in the same order as the original sequence:64 6F 70 79 7A
    - **Little-endian**: Reverse the byte order:7A 79 70 6F 64

Final Result:

- **Big-endian**: 64 6F 70 79 7A
- **Little-endian**: 7A 79 70 6F 64