Zadania 1

Zad. 1.1

Napisz program size sprawdzający, ile bajtów zajmują typy: char, short, int, long, long int, long long oraz float, double, long double.

https://en.wikipedia.org/wiki/C_data_types

Zad. 1.2 *

Napisz program size2 sprawdzający, ile bajtów zajmują typy: char, short, int, long, long int i long long bez znaku.

Zad. 1.3

Która z poniższych odpowiedzi jest prawdziwa:

- system 32 bitowy pozwala na uruchamianie programu 32 bitowego

- system 32 bitowy pozwala na uruchamianie programu 64 bitowego

- system 64 bitowy pozwala na uruchamianie programu 32 bitowego

- system 64 bitowy pozwala na uruchamianie programu 64 bitowego

Zad. 1.4

Która z poniższych odpowiedzi jest fałszywa:

- na systemie 32 bitowym można skompilować program do kodu 32 bitowego

- na systemie 32 bitowym można skompilować program do kodu 64 bitowego

- na systemie 64 bitowym można skompilować program do kodu 32 bitowego

- na systemie 64 bitowym można skompilować program do kodu 64 bitowego

Zad. 1.5

- ile bajtów zajmują adresy w kodzie 32 bitowym?

- ile bajtów zajmują adresy w kodzie 64 bitowym?

Zad. 1.6 *

Jaki obszar pamięci można zaadresować przy pomocy adresów 16, 20, 24, 32, 40, 48 i 64 bitowych?

Zad. 1.7

Napisz program bits rozpoznający do ilu bitowego kodu został skompilowany.

Zad. 1.8

Załóżmy, że typ int zajmuje 4 bajty. Na ile sposobów można umieścić w pamięci pod adresem p wartość 1 typu int ? Zadanie rozwiąż w pliku sposoby.txt.

p -> [ ][ ][ ][ ] *p = 1

Zad. 1.9

Procesory w architekturze little-endian czytają młodsze bajty (LSB – least significant byte) od lewej do prawej. Procesory w architekturze big-endian czytają starsze bajty (MSB – most significant byte) od lewej do prawej. Załóżmy, że pod adresem p znajduje się liczba 5 typu int. W pliku endian.txt wypełnij komórki pamięci odpowiednimi wartościami dla obu architektur.

little-endian

p -> [ ][ ][ ][ ] *p = 5

big-endian

p -> [ ][ ][ ][ ] *p = 5

Zad. 1.10

W pliku szereg.txt rozwiń w szereg i wyznacz wartości dziesiętne dla liczb:

1011 – liczba binarna

8732 – liczba dziesiętna

[2][2][1][1] - reprezentacja little-endian

1234 - liczba ósemkowa *

3A5B – liczba szesnastkowa *

Zad. 1.11

Załóżmy, że pod adresem p znajduje się liczba 260 typu int. Wypełnij komórki pamięci odpowiednimi wartościami dla obu architektur. Zadanie rozwiąż w pliku 260.txt.

little-endian

p -> [ ][ ][ ][ ] *p = 260

big-endian

p -> [ ][ ][ ][ ] *p = 260

Zad. 1.12

Napisz program bytes wypisujący reprezentację bajtową dla podanej liczby x typu int. Przykładowa sesja:

value = 260

bytes = 004 001 000 000

Zad. 1.13

Sprawdź na terminalu Linux w jakiej architekturze pracuje twój procesor.

Zad. 1.14

Napisz program endian rozpoznający w jakiej architekturze pracuje procesor.

Zad. 1.15 *

Napisz program endian2 wypisujący little-endian architecture lub big-endian architecture w zależności od architektury, w której pracuje procesor.