

Zad 6.

piątek, 24 marca 2023 18:47

Zadanie 6. Zaimplementuj w asemblerze x86-64 funkcję liczącą wyrażenie « $x + y$ ». Argumenty i wynik funkcji są 128-bitowymi liczbami całkowitymi ze znakiem i nie mieszczą się w rejestrach maszynowych. Zatem « x » jest przekazywany przez rejestry `%rdi` (starsze 64 bity) i `%rsi` (młodsze 64 bity), analogicznie argument « y » jest przekazywany przez `%rdx` i `%rcx`, a wynik jest zwracany w rejestrach `%rdx` i `%rax`.

Wskazówka! Użyj instrukcji «`adc`». Rozwiązanie wzorcowe składa się z 3 instrukcji bez «`ret`».

```
# x -> %rdi:rsi, y -> %rdx:rcx
addition: addq %rsi, %rcx
          adcq %rdi, %rdx
          movq %rcx, %rax
          ret
```

Adds the destination operand (first operand), the source operand (second operand), and the carry (CF) flag and stores the result in the destination operand.

↑ «`adc`» → add with carry