

# Obliczanie i weryfikacja sum kontrolnych CRC

w. 1

## Wprowadzenie

Celem ćwiczenia jest opracowanie programu do równoległego obliczania i weryfikacji sum kontrolnych CRC (ang. *Cyclic Redundancy Check*): w wersjach CRC-12, CRC-16 i CRC-32.

**Definicja:** n-bitowy cykliczny kod nadmiarowy (CRC) obliczony dla bitowego ciągu danych jest resztą z dzielenia tego ciągu przez (n+1)-bitowy dzielnik CRC, zwany również wielomianem CRC.

UWAGA: Zarejestruj wszystkie polecenia wykonywane w trakcie realizacji zadania oraz odpowiedzi systemu; jeżeli nie jest to możliwe w sposób automatyczny – wykorzystaj mechanizm kroniki terminala SSH. Zarejestrowany materiał będzie dowodem wykonania ćwiczenia i podstawą sprawozdania. Sposób włączenia rejestracji kroniki konwersacji podany jest w materiałach kursu.

Algorytmy obliczeniowe dla różnych wersji CRC dostępne są online, np. [1, 2,3].

## Wymagania dotyczące programu

- a) Program ma być opracowany w języku C i przedstawiony (w sprawozdaniu i w odrębnym pliku) w wersji nadającej się do bezpośredniej kompilacji kompilatorem `gcc`.
- b) Program ma sprawdzać liczbę i poprawność parametrów wywołania. W przypadku stwierdzenia błędu, ma być wyprowadzony komunikat diagnostyczny (co jest źle?), a wykonanie programu zakończone z kodem 1.
- c) Poprawne wykonanie funkcji programu oznaczane jest kodem powrotu równym 0.
- d) Program ma realizować dwie funkcje; ich wybór następuje przez pierwszy parametr wywołania programu:
  - a. `OBLICZ` oznacza wyznaczanie wartości CRC,
  - b. `TESTUJ` – badanie poprawności wprowadzonego CRC.
- e) Każdy z tych parametrów może być wprowadzony literami dowolnej wielkości, np. `tEstUJ`, lub `obLIcZ`.
- f) Dla funkcji `OBLICZ`
  - a. drugim parametrem wywołania programu jest bitowy ciąg danych, dla którego wyznacza się wartość CRC, zakodowany w postaci szesnastkowej,
  - b. trzecim parametrem jest liczba ze zbioru {12, 16, 32} określająca wersję obliczanego CRC.
- g) Dla funkcji `TESTUJ` drugim parametrem wywołania programu jest badany CRC, podany jako 32-bitowa liczba zakodowana szesnastkowo.  
Program winien sam określić, według którego algorytmu (CRC-12, CRC-16 i CRC-32) został obliczony podany CRC. Sprawdzenie to ma odbywać się równoległe dla wszystkich trzech algorytmów. W przypadku CRC niepasującego do żadnej z podanych wersji, program ma wyprowadzić odpowiedni komunikat zawierający jego wartość.

## Zadania

- a) Zapoznaj się z algorytmem obliczania CRC i wykonaj ręczną kalkulację dla 16-bitowych danych i algorytmu CRC-3 (GSM).
- b) Opracuj program realizujący określoną wyżej specyfikację. Jego działanie ma być zrównoleglone przy wykorzystaniu API OpenMP.
- c) Zbadaj poprawność działania na kilku przykładach.
- d) Wykonaj obliczenie CRC-12

## Wskazówki

Wykorzystaj dostępne poradniki programowania w języku C, w tym opis funkcji wejścia/wyjścia. Dodatkowe informacje zawarte są w materiałach kursu.

## Literatura

- [1] <http://www.algorytm.org/sumy-kontrolne/cykliczna-kontrola-nadmiarowa.html>
- [2] [https://4programmers.net/Algorytmy/Obliczanie\\_sum\\_kontrolnych\\_CRC-32](https://4programmers.net/Algorytmy/Obliczanie_sum_kontrolnych_CRC-32)
- [3] [http://wmii.uwm.edu.pl/~bnowak/userfiles/downloads/dydaktyka/Sieci\\_Komputerowe/CRC.rtf](http://wmii.uwm.edu.pl/~bnowak/userfiles/downloads/dydaktyka/Sieci_Komputerowe/CRC.rtf)

## Opracowanie wyników

Sprawozdanie z realizacji ćwiczenia winno być wykonane na właściwym formularzu oraz zawierać m.in.:

- Ogólny opis idei algorytmu obliczania CRC, bez wskazywania konkretnej wersji (wielomianu),
- Przebieg ręcznego obliczania CRC-3 z 16-bitowych danych,
- Pełny kod opracowanego programu.

Do przesłania są 2 pliki niespakowane:

- kompletne sprawozdanie,
- plik źródłowy z programem w języku C.