Technologie Sieciowe
Lista 3

Illia Azler
239537

Ramkowanie

Celem pierwszej części ćwiczenia było napisanie programu ramkującego zgodnie z zasadą "rozpychania bitów", oraz weryfikujący poprawność ramki metodą CRC.

Program ma odczytywać pewien źródłowy plik tekstowy 'Z' zawierający dowolny ciąg złożony ze znaków '0' i '1' (symulujący strumień bitów) i zapisywać ramkami odpowiednio sformatowany ciąg do innego pliku tekstowego 'W'. Program powinien obliczać i wstawiać do ramki pola kontrolne CRC formatowane za pomocą ciągów złożonych ze znaków '0' i '1'. Napisz program, realizujący procedurę odwrotną, tzn. który odczytuje plik wynikowy 'W' i dla poprawnych danych CRC przepisuje jego zawartość tak, aby otrzymać kopię oryginalnego pliku źródłowego 'Z'.

Realizowano za pomocą biblioteki zlib(crc32) w Python'ie.

Ramkowanie

Ramki tworzone zgodnie z zasadą rozpychania bitów. Składają się z:

1) bajtowych flag informujących o początku i końcu ramki
2) nagłówka zawierającego informacje o adresacie i źródle
3) fatycznych danych
4) wartości kontrolnej CRC

Kodowanie

1) Czytamy 32 kolejne bity danych wejściowych.
2) Obliczamy kod CRC i dostawiamy go na koniec naszego ciągu.
3) Przeglądamy nasz ciąg i po każdej sekwencji pięciu jedynek dostawiamy zero, żeby pozbyć się z niego fałszywych flag.
4) Dodajemy flagi na początku i końcu.
5) Zapisujemy utworzoną ramkę, a następnie – jeśli nie skończyły się dane wejściowe – tworzymy kolejną, wracając do kroku pierwszego.

Dekodowanie

1) Odczytujemy fragment między dwiema flagami.
2) W otrzymanym ciągu usuwamy zera występujące po sekwencji pięciu jedynek.
3) Dzielimy uzyskany ciąg przez G. Jeśli otrzymamy resztę, ramka uległa uszkodzeniu i jest odrzucana.
4) Jeśli ramka jest poprawna, usuwamy z końca kod CRC (przy naszym G są to trzy bity).
5) Zapisujemy rezultat i, jeśli mamy więcej fragmentów, wracamy do punktu pierwszego.

Przykład z programu:

Plik wejściowy Z ma wartość naszej flagi: 01111110.

Zakodujemy naszym programem i zobaczymy wynik:
01111110011111010110010101011111011011001110110110011111110

Po dekodowaniu otrzymamy z powrotem naszą flagę: 01111110.