

Bezpieczeństwo systemów informatycznych – laboratorium Zachodniopomorska Szkoła Biznesu			
Grupa/Specialność	Imię i nazwisko	Data wykonania (rrrr.mm.dd)	Tryb studiów
Game Design	Sylwester Dawidowicz	9.05.2020	Z
Nr laboratorium	Temat		
2	Obliczanie CRC		

Na tych laboratoriach zadaniem było przygotowanie programu obliczającego CRC pliku tekstowego. Najważniejsze kroki do wykonania obrazujące działanie programu:

- 1) Odczytać dane z pliku tekstowego oraz podzielić go na bloki

```

/* Deklaracja funkcji dzielącej tekst na bloki */
ReadLines(char* filename)
{
    FILE* f;
    int n;
    char buf[MAXCHARSPERLINE];
    if ((f = fopen(filename, "r")) == NULL)
        return fprintf(stderr, "%: can't open for reading\n"), -1;
    for (nlines = 0; fgets(buf, MAXCHARSPERLINE, f) != NULL; nlines++)
    {
        n = strlen(buf);
        strcpy(line[nlines], buf);
    }
    return nlines;
}

```

Funkcja jako argument pobiera plik tekstowy, następnie wczytuje linię tekstu po linii aż do końca pliku. W moim programie blok ma długość 40 znaków (licząc łącznie ze spacjami).


- 2) Zdefiniować funkcję odpowiedzialną za algorytm CRC. W tym przypadku użyłem algorytmu **CRC-32/JAMCRC** (źródło: <https://stackoverflow.com/questions/21001659/crc32-algorithm-implementation-in-c-without-a-look-up-table-and-with-a-public-li>). Zapisujemy dane do pliku tekstowego (naprzemiennie dane oraz klucz CRC w kolejnej linii).

```

/* Deklaracja algorytmu CRC-32/JAMCRC z zapisywaniem do pliku tekstowego */
unsigned int crc32b(unsigned char* message, FILE* some_file)
{
    int i, j;
    unsigned int byte, crc, mask;
    i = 0;
    crc = 0xFFFFFFFF;
    while (message[i] != 0) {
        byte = message[i];
        crc = crc ^ byte;
        for (j = 7; j >= 0; j--) {
            mask = 0 - (crc & 1);
            crc = (crc >> 1) ^ (0xEDB88320 & mask);
        }
        i = i + 1;
    }
    printf("0x%x \n", crc);
    fprintf(some_file, "%s\n", message); // Zapis bloku do pliku tekstowego
    fprintf(some_file, "0x%x\n", crc); // Zapis CRC do pliku tekstowego
    return ~crc;
}

```

Otrzymany fragment pliku tekstowego prezentuje się w następujący sposób:

 crc_separated — Notatnik

Plik Edycja Format Widok Pomoc

```

Lorem ipsum dolor sit amet, consectetur
0x852a0448
adipiscing elit. Curabitur eget fringill
0x4e1a21e7
a arcu. Sed nunc tortor, fringilla vehic
0x472ef85
ula vulputate non, ornare sit amet urna.
0xc56429c2
Cras vestibulum vel nisi sit amet feugi
0x7073bf97
at. Praesent efficitur, mauris ut suscip
0x11615cbf
it volutpat, augue quam luctus ex, vel l
0xcf57edb9
obortis nisl urna eget purus. Donec sed
0x6a96136b
risus in lacus vulputate condimentum. Na
0x232b5d0b
m nibh dolor, laoreet quis ullamcorper u
0x7a27e134
t, elementum et tellus. Nunc eu leo eget
0x4f336c88
ligula vulputate viverra. Vivamus sagit
0x8e71d2c0

```

- 3) Następnie wyświetlamy blok tekstowy oraz wygenerowane CRC dla tego bloku.

```
/* Wyświetlanie tekstu oraz CRC dla poszczególnych bloków */
for (int i = 0; i < nlines; i++)
{
    printf("Data: %s , CRC:", line[i]);
    crc32b(line[i], crc_file);
}
```

W rezultacie otrzymamy następujące dane:

WybierzKonsola debugowania programu Microsoft Visual Studio

```
Data: Lorem ipsum dolor sit amet, consectetur , CRC:0x852a0448
Data: adipiscing elit. Curabitur eget fringill , CRC:0x4e1a21e7
Data: a arcu. Sed nunc tortor, fringilla vehic , CRC:0x472ef85
Data: ula vulputate non, ornare sit amet urna. , CRC:0xc56429c2
Data: Cras vestibulum vel nisi sit amet feugi , CRC:0x7073bf97
Data: at. Praesent efficitur, mauris ut suscip , CRC:0x11615cbf
Data: it volutpat, augue quam luctus ex, vel l , CRC:0xcf57edb9
Data: obortis nisl urna eget purus. Donec sed , CRC:0x6a96136b
Data: risus in lacus vulputate condimentum. Na , CRC:0x232b5d0b
Data: m nibh dolor, laoreet quis ullamcorper u , CRC:0x7a27e134
Data: t, elementum et tellus. Nunc eu leo eget , CRC:0x4f336c88
Data: ligula vulputate viverra. Vivamus sagit , CRC:0x8e71d2c0
Data: tis, leo ut aliquet condimentum, ex magn , CRC:0x42d16232
Data: a scelerisque enim, a ultricies dolor se , CRC:0xb2bf8211
Data: m vel enim. Duis et arcu at nulla euismo , CRC:0xdd54f6c
Data: d hendrerit vitae ut nisl. Morbi pellent , CRC:0x63839030
Data: esque a justo et varius. Nunc eu est ven , CRC:0xa128d677
Data: enatis, convallis est quis. , CRC:0x5e3a5f64
```

- 4) W kolejnym kroku definiujemy funkcję sprawdzającą spójność danych wygenerowanego pliku tekstowego z wcześniej wygenerowanym CRC, powtarzając proces tworzenia sumy kontrolnej. W tej funkcji znajduje się zduplikowana funkcja algorytmu liczącego CRC (tym razem bez zapisywania danych do pliku). Wynik działania funkcji **crc32b_2** przypisujemy do zmiennej tymczasowej **a** i porównujemy z sumą kontrolną z pliku tekstowego **crc_separated.txt**. Z racji tego, że w jednym pliku znajduje się tekst i suma kontrolna program wczytuje co drugą linię (tą interesującą nas najbardziej). Następnie sprawdzamy spójność danych (sprawdzamy, czy dwa stringi są jednakowe).

```

check()
{
    FILE* fp;
    char buf[bufSize];
    int i = 0;
    int j = 1;
    int a = 0xFFFFFFFF;
    int x = 0xFFFFFFFF;
    if ((fp = fopen("crc_separated.txt", "r")) == NULL)
    {
        perror("fopen source-file");
        return 1;
    }

    while (fgets(buf, sizeof(buf), fp) != NULL)
    {
        buf[strlen(buf) - 1] = '\0';
        if (i % 2 == 0)
        {
            a = crc32b_2(buf); // Liczenie CRC na nowo
        }
        if (i % 2 != 0)
        {
            printf("CRC z pliku 'crc_separated.txt' dla %d wiersza tekstu : %s", j, buf);
            printf(" |      CRC obliczone na nowo: 0x%x\n", a);
            j++;
            /* Sprawdzenie, czy stringi są równe */
            sscanf(buf, "%x", &x);

            if (x == a)
            {
                printf("[DANE POPRAWNE: 0x%x == 0x%x]\n", x, a);
            }
            else
            {
                printf("[DANE NIEPOPRAWNE!!!: 0x%x != 0x%x]\n", x, a);
            }
        }
        i++;
    }
}

```

Finalnym wynikiem programu będzie wyświetlenie w oknie konsoli informacji o tym, czy dane są poprawne:

```
CRC z pliku 'crc_separated.txt' dla 1 wiersza tekstu : 0x852a0448 | CRC obliczone na nowo: 0x852a0448
[DANE POPRAWNE: 0x852a0448 == 0x852a0448]
CRC z pliku 'crc_separated.txt' dla 2 wiersza tekstu : 0x4e1a21e7 | CRC obliczone na nowo: 0x4e1a21e7
[DANE POPRAWNE: 0x4e1a21e7 == 0x4e1a21e7]
CRC z pliku 'crc_separated.txt' dla 3 wiersza tekstu : 0x472ef85 | CRC obliczone na nowo: 0x472ef85
[DANE POPRAWNE: 0x472ef85 == 0x472ef85]
CRC z pliku 'crc_separated.txt' dla 4 wiersza tekstu : 0xc56429c2 | CRC obliczone na nowo: 0xc56429c2
[DANE POPRAWNE: 0xc56429c2 == 0xc56429c2]
CRC z pliku 'crc_separated.txt' dla 5 wiersza tekstu : 0x7073bf97 | CRC obliczone na nowo: 0x7073bf97
[DANE POPRAWNE: 0x7073bf97 == 0x7073bf97]
CRC z pliku 'crc_separated.txt' dla 6 wiersza tekstu : 0x11615cbf | CRC obliczone na nowo: 0x11615cbf
[DANE POPRAWNE: 0x11615cbf == 0x11615cbf]
CRC z pliku 'crc_separated.txt' dla 7 wiersza tekstu : 0xcf57edb9 | CRC obliczone na nowo: 0xcf57edb9
[DANE POPRAWNE: 0xcf57edb9 == 0xcf57edb9]
CRC z pliku 'crc_separated.txt' dla 8 wiersza tekstu : 0x6a96136b | CRC obliczone na nowo: 0x6a96136b
[DANE POPRAWNE: 0x6a96136b == 0x6a96136b]
CRC z pliku 'crc_separated.txt' dla 9 wiersza tekstu : 0x232b5d0b | CRC obliczone na nowo: 0x232b5d0b
[DANE POPRAWNE: 0x232b5d0b == 0x232b5d0b]
CRC z pliku 'crc_separated.txt' dla 10 wiersza tekstu : 0x7a27e134 | CRC obliczone na nowo: 0x7a27e134
[DANE POPRAWNE: 0x7a27e134 == 0x7a27e134]
CRC z pliku 'crc_separated.txt' dla 11 wiersza tekstu : 0x4f336c88 | CRC obliczone na nowo: 0x4f336c88
[DANE POPRAWNE: 0x4f336c88 == 0x4f336c88]
CRC z pliku 'crc_separated.txt' dla 12 wiersza tekstu : 0x8e71d2c0 | CRC obliczone na nowo: 0x8e71d2c0
[DANE POPRAWNE: 0x8e71d2c0 == 0x8e71d2c0]
CRC z pliku 'crc_separated.txt' dla 13 wiersza tekstu : 0x42d16232 | CRC obliczone na nowo: 0x42d16232
[DANE POPRAWNE: 0x42d16232 == 0x42d16232]
CRC z pliku 'crc_separated.txt' dla 14 wiersza tekstu : 0xb2bf8211 | CRC obliczone na nowo: 0xb2bf8211
[DANE POPRAWNE: 0xb2bf8211 == 0xb2bf8211]
CRC z pliku 'crc_separated.txt' dla 15 wiersza tekstu : 0xdd54f6c | CRC obliczone na nowo: 0xdd54f6c
[DANE POPRAWNE: 0xdd54f6c == 0xdd54f6c]
CRC z pliku 'crc_separated.txt' dla 16 wiersza tekstu : 0x63839030 | CRC obliczone na nowo: 0x63839030
[DANE POPRAWNE: 0x63839030 == 0x63839030]
CRC z pliku 'crc_separated.txt' dla 17 wiersza tekstu : 0xa128d677 | CRC obliczone na nowo: 0xa128d677
[DANE POPRAWNE: 0xa128d677 == 0xa128d677]
CRC z pliku 'crc_separated.txt' dla 18 wiersza tekstu : 0x5e3a5f64 | CRC obliczone na nowo: 0x5e3a5f64
[DANE POPRAWNE: 0x5e3a5f64 == 0x5e3a5f64]
```

Jeżeli wszystkie dane są ze sobą spójne to oznacza, że dane są identycznie.