

Wojciech Myszka

Laboratorium 13: Statystyka znaków

2016-05-08 20:55:20 +0200

1. Zadanie

Program czytający tekst (bez polskich liter!) z konsoli i podający po zakończeniu czytania danych liczbę wystąpień każdego znaku.

Podpowiedź: wszystkich znaków jest nie więcej niż 256 (mają kody w zakresie od zero do 255).

W ostatecznym rozrachunku zadanie sprowadza się do czytania tekstu znak po znaku i aktualizowaniu zapisów w tablicy o długości 256. Na koniec wystarczy wypisać kod ASCII znaku i liczbę wystąpień.

Student zaawansowany może zaryzykować stworzenie prostego wykresu słupkowego pokazującego statystykę.

2. End of File — EOF

Należy się kilka słów wyjaśnienia na temat pojęcia EOF.

Gdy strumień wejściowy związany jest z plikiem zdajemy sobie sprawę, że ma on skończoną długość i że nie można odczytać więcej bajtów niż w pliku jest. NA potrzeby zarządzania operacjami odczytu/zapisy w strukturze **FILE** opisującej otwarty plik jest wskaźnik aktualnej pozycji czytania/pisania. Na początku ustawiony jest ona na zero i mówi z którego miejsca pliku czytany będzie kolejny bajt (lub w które miejsce będzie pisany). Przesuwa się on wraz z odczytywaniem informacji. Gdy dojdzie do końca, a program próbuje odczytać kolejny bajt — następuje sytuacja nazwana „próba czytania poza końcem pliku”. Wszystkie polecenia odczytu starają się poinformować program zwracając wartość zwaną EOF. (Stała o tej nazwie zdefiniowana jest w pliku `stdio.h`.)

Podobnie jest w przypadku czytania z konsoli (z którą normalnie stowarzyszony jest standardowy strumień wejściowy). Normalnie zakłada się, że strumień nie

skończy się: gdy następuje próba czytania, a żadnych danych nie ma w kolejce — program zatrzymuje się i czeka na reakcję użytkownika.

Użytkownik może zasymulować sytuację „próby odczytu poza końcem pliku” naciskając równocześnie klawisze CTRL i D (w systemie Windows — CTRL i Z).

Informacje o napotkaniu końca pliku zwracają wszystkie funkcje czytające ze strumienia: `scanf`, `fgetc`, `fgets`, `getc`, `getchar`,...

```
i = scanf( "%d%d" , &a , &b );
if ( i == EOF )
    printf( "Napotkano koniec zbioru\n" );
else if ( i < 2 )
    printf( "Nie udało się wczytać dwu wartości\n" );
else
    printf( "OK\n" );
```

3. Możliwe rozwiązania

Do czytania tekstu można użyć albo funkcji `fgetc` albo skorzystać z rozwiązania zaproponowanego w Instrukcji Laboratoryjnej nr. 12.

Można też napisać program ogólniejszy, który będzie czytał dane ze standardowego wejścia aż do wystąpienia sytuacji, która nazywa się *koniec pliku* (EOF — *End of File*). Wówczas będzie można wykorzystać program do zadania ogólniejszego — stworzyć statystykę występowania „znaków” w dowolnym pliku (wykorzystując operację przekierowania standardowego wejścia).

Zadanie pomocnicze:

Napisać następujący program (o nazwie, na przykład, `test.c`):

```
#include <stdio.h>
int main()
{
    char z;
    int i;
    while ( ( i = scanf( "%c" , &z ) ) > 0 )
        printf( "z=%c i=%d\n" , z , i );
    printf( "Koniec\n" );
    return 0;
}
```

Następnie należy otworzyć okno terminala (pod windowsem będzie to `cmd`) i przejść do kartoteki w której jest plik wykonywalny. W laboratorium wystarczy

znaleźć kartotekę, w której jest plik źródłowy i wykonywalny i w menedżerze plików kliknąć prawym klawiszem myszy w „puste” miejsce między plikami, z menu wybrać otwórz w terminalu.

W kartotece tej należy utworzyć plik o nazwie, na przykład `a.txt` z dowolną zawartością tekstową (można też wykorzystać plik z kodem źródłowym programu)..

Uruchamiamy program w oknie cmd następującym poleceniem (zakładam, że program wykonywalny nazywa się w Windows¹ `test.exe`):

```
test.exe < a.txt > b.txt
```

Następnie należy obejrzeć plik `b.txt` w notatniku.

Czy ta wiedza może się przydać do realizacji powyższego zadania?

4. Wersja PDF tego dokumentu...

... pod adresem.

Wersja: 52 z **drobnymi modyfikacjami!** data ostatniej modyfikacji 2016-05-08 20:55:20 +0200

¹ Pod linuxem będzie to `test`.