

Zadania z programowania proceduralnego w języku C do wykładu 11

dr hab. Andrzej Zbrzezny, prof. nadzw. AJD

Operacje na plikach przy użyciu funkcji z biblioteki **stdio.h**

1. Zmodyfikuj program **w11-p01.c** tak, aby badał on możliwość otwarcia każdego z plików podanych w linii wywołania programu.
2. Zmodyfikuj program **w11-p02.c** tak, aby podawała on wielkość każdego z plików podanych w linii wywołania programu.
3. Zmodyfikuj program **w11-p04.c** tak, aby kopiował on (używając funkcji **fgetc** oraz **fputc**) zawartość pliku podanego jako pierwszy argument programu do pliku podanego jako drugi argument programu. Rozszerz obsługę błędów tak, aby uwzględnić przypadek, gdy nie udaje się otworzyć do zapisu pliku docelowego.
4. Zmodyfikuj program **w11-p05.c** tak, aby kopiował on (używając funkcji **fgets** oraz **fputs**) zawartość plik podanego jako pierwszy argument programu do pliku podanego jako drugi argument programu. Rozszerz obsługę błędów tak, aby uwzględnić przypadek, gdy nie udaje się otworzyć do zapisu pliku docelowego.
5. Zmodyfikuj program z poprzedniego zadania tak, aby używał on funkcji **fread** oraz **fwrite** kopiując każdorazowo 512 bajtów (512 bloków o rozmiarze jednego bajta). Oczywiście, ostatnia operacja zapisu może zapisać mniej niż 512 bajtów.
6. Napisz program, który kopiuje zawartość plik podanego jako pierwszy argument programu do pliku podanego jako drugi argument programu, przy czym zamienia małe litery na duże, a pozostałe znaki pozostawia bez zmian.
7. Napisz program, który kopiuje zawartość plik podanego jako pierwszy argument programu do pliku podanego jako drugi argument programu, przy czym zamienia małe litery na duże, duże na małe, a pozostałe znaki pozostawia bez zmian.

Przykładowe rozwiązanie zadania 5:

```
#include <stdio.h>
#include <stdlib.h>

#define BUF_SIZE 512

int main(int argc, char *argv[])
{
    FILE *fin, *fout;

    if (argc != 3) {
        fprintf(stderr, "Sposób użycia: %s plik_1 plik_2\n", argv[0]);
        exit(1);
    }

    if ((fin = fopen(argv[1], "r")) == NULL) {
        fprintf(stderr, "Nie można otworzyć %s\n", argv[1]);
        exit(2);
    }

    if ((fout = fopen(argv[2], "w")) == NULL) {
        fprintf(stderr, "Nie można otworzyć %s\n", argv[2]);
        exit(2);
    }

    size_t size;
    size_t licznik = 0;
    unsigned char bufor[BUF_SIZE];
    while ((size = fread(bufor, 1, BUF_SIZE, fin)) != 0) {
        if (size < BUF_SIZE) {
            fprintf(stderr, "Przeczytano %zu bajtów\n", size);
        }
        fwrite(bufor, 1, size, fout);
        licznik += size;
    }

    if (feof(fin) != 0) {
        fprintf(stderr, "Koniec pliku\n");
        clearerr(fin);
    } else {
        fprintf(stderr, "To nie powinno się wydarzyć\n");
    }

    fclose(fin);
    fclose(fout);
    fprintf(stderr, "Skopiowano %zu bajtów\n", licznik);
    return 0;
}
```