

Wstęp do programowania w języku C

Grupa MSz we wtorek

Lista 9 na zajęcia 11.12.2018

1. *(15 punktów w trakcie pierwszej pracowni, 10 punktów później)*

Napisz program, który wypisuje ile razy został już uruchomiony od skompilowania, zapisując tę informację wyłącznie w swoim pliku wykonywalnym.

Test (w konsoli):

```
./9.1
./9.1
./9.1
cp 9.1 9.1-backup
./9.1
rm 9.1 && mv 9.1-backup 9.1
./9.1
```

Wynik:

```
Liczba uruchomien: 1
Liczba uruchomien: 2
Liczba uruchomien: 3
Liczba uruchomien: 4
Liczba uruchomien: 4
```

Wskazówka: Najłatwiej zapisać licznik (`int`) na końcu pliku, razem z drugą (jakaś ustalona i unikalna) liczbą weryfikującą. Druga liczba pozwoli nam sprawdzić czy nie jest to pierwsze uruchomienie programu, po kompilacji.

To jest bazowy program, który kopiuje nasz program do tymczasowego pliku, co pozwala go zmodyfikować, a na końcu podmienia z oryginałem.

```

#include <stdio.h>
#include <sys/stat.h>
#include <string.h>
int main(int argc, char *argv[]) {
    // Otwieramy nasz plik wykonywalny
    FILE *fmain = fopen(argv[0], "rb");

    // Tworzymy tymczasowy plik
    char tmpname[strlen(argv[0])+5];
    strcpy(tmpname, argv[0]);
    strcat(tmpname, ".tmp");
    FILE *ftmp = fopen(tmpname, "wb+");
    chmod(tmpname, 0777);

    // Kopiujemy bajt po bajcie
    int ch;
    while ((ch = fgetc(fmain)) != EOF) fputc(ch, ftmp);
    // Tutaj można zmodyfikować ftmp

    // Zamykamy i podmieniamy
    fclose(fmain);
    fclose(ftmp);
    rename(tmpname, argv[0]);
}

```

Przydatne funkcje:

```

// Ustawia pozycję na jeden int przed końcem pliku
fseek(ftmp, -sizeof(int), SEEK_END);

// Odczytuje jeden int z pliku
int x = getw(ftmp);

// Zapisuje jeden int do pliku
putw(x, ftmp);

```

Uwaga: Nie polecam robienia tego na Windowsie, gdyż nie można tam (łatwo) podmienić pliku z działającym programem. Dla tych którzy koniecznie chcą na Windowsie, proponuję żeby program po prostu generował swoją zmodyfikowaną kopię.

Projekty

- Program powinien być przenośny, a przynajmniej działać na Linuxie. Do programu należy dołączyć makefile z poleceniem kompilacji.
- Można używać popularnych bibliotek, np. Allegro, SFML, SDL, GTK+. Można założyć, że pakiety bibliotek są zainstalowane w systemie.

Sugestie tematów (szczegóły do ustalenia):

1. Gra w trybie tekstowym. Np. text-based RPG, gra planszowa z komputerem (coś z https://en.wikipedia.org/wiki/List_of_abstract_strategy_games, albo <https://pl.wikipedia.org/wiki/Superfarmer>), gra logiczna (np. wariant Sokobana). Polecam zadbać o oprawę wizualną, wykorzystując różne znaczki i kolory w konsoli.
2. Gra w trybie graficznym. Np. jakaś platformówka, RPG w 2D, coś z budowaniem (proste SimCity, Tycoon), albo gra jak wyżej.
3. Przeglądarka fraktali (w trybie graficznym): zbiór znanych fraktali z możliwością interaktywnej zmiany ich parametrów; do tego powiększanie, przesuwanie itp.
4. Biblioteka i program testujący wydajność wybranych tricków bitowych ze strony: <http://graphics.stanford.edu/~seander/bithacks.html>.
5. Biblioteka do obliczania liniowo-algebraicznych własności automatów skończonych (rozkład stacjonarny, wymiar macierzy, należy efektywnie zimplementować eliminację Gaussa).
6. Generowanie losowej melodii do podanej progresji akordowej. Należy zaimplementować jakieś różne reguły, żeby wynik był ciekawy. Program powinien umieć zapisać wynik w jakimś istniejącym formacie, np. https://en.wikipedia.org/wiki/ABC_notation. Drugi wariant jest odwrotny: generowanie progresji akordowej do podanej melodii.