

## Wprowadzenie do programowania w języku C

grupa RKr, wtorek 16:15-18:00

lista nr 8 (na pracownię 03.12.2019) (wersja 2)

### Zadanie 1. [15p na pracowni lub 10p po pracowni]

Zdefiniuj typ strukturalny **MemDescriptor**, który zawierał będzie wskaźnik do pewnego obszaru pamięci traktowany jako tablica znaków (**char\***) oraz wskaźnik do funkcji indeksującej zadeklarowany w deskrytorze jako „**int (\*indexer)(int x, int y)**”, potrafiącej indeksować tę pamięć jak tablicę dwuwymiarową.

W tym zadaniu nie przejmuj się wyciekami pamięci, alokuj kiedy potrzebujesz i zapominaj kiedy to wygodne.

Napisz funkcję „**int indexerLined(int x, int y)**” oraz funkcję „**int indexerTiled(int x, int y)**”, które będą mapowały kwadratowy obszar rozmiaru 64x64 komórek do liniowego obszaru rozmiaru 64·64 komórek.

Skoro w każdym wymiarze mamy 64 komórki, to do opisanie jednej współrzędnej potrzebujemy maksymalnie 6 bitów, zatem indeksery przekształcają dwie 6-bitowe liczby w jedną 12-bitową. Do implementacji tych indeksów należy użyć operacji bitowych, operacje arytmetyczne są zabronione.

Napisz funkcję o sygnaturze „**MemDescriptor layToLined(MemDescriptor m)**”

oraz funkcję o sygnaturze „**MemDescriptor layToTiled(MemDescriptor m)**”,

które będą konwertowały układ pamięci odpowiednio, z każdego do Lined oraz z każdego do Tiled.

Układ Lined ma działać z wierszami rozmiaru 64, a układ Tiled ma działać z kafelkami rozmiaru 8x8.

W układzie kafelkowym to kafelki są w układzie Lined, czyli w pamięci mają szyk wiersz po wierszu, co oznacza, że jeden kafelek „rozleje się” na cały 64-znakowy wiersz w całej szachownicy.

Napisz funkcję „**MemDescriptor createChequer()**”, która stworzy szachownicę zbudowaną z liter **'B'** i **'W'**, która w pamięci będzie ułożona w szyku wiersz po wierszu. Pola szachownicy mają mieć rozmiar 8x8 znaków, a cała szachownica ma mieć rozmiar 64x64 znaków, a więc logiczny rozmiar alokacji to 64·64·**sizeof(char)**.

Użyj funkcji layToTiled(.) aby zmienić układ pamięci (*ang. memory layout*) na kafelkowy, u nas Tiled\_8x8.

W każdej szachownicy, tej w układzie Lined oraz tej w układzie Tiled, nadpisz pierwszych 8 wierszy ignorując ich układ pamięci, a dokładniej, zawsze pisząc w szyku Lined. Nieparzyste wiersze wypełnij literami **'X'**, parzyste wiersze wypełnij literami **'Q'**. W tym celu stwórz funkcję „**void overwriteLines(MemDescriptor m)**”.

Wypisz na standardowym wyjściu obydwie szachownice, tj. tę w układzie Lined oraz tę w układzie Tiled.

Przed wypisaniem szachownicy mającej układ Tiled, zmień jej układ na Lined lub wypisz indeksorem Tiled.

Napisz funkcję wypisującą szachownicę „**void printChequer(MemDescriptor m, int (\*indexer)(int, int))**”.

```
BBWWBBWW // przykład szachownicy
BBWWBBWW // -----
WWBBWWBB //   rozmiar szachownicy: 8x8 znaków w formie 4x4 pól (każde pole to kwadrat 2x2 znaków)
WWBBWWBB //   rozmiar pola: 2x2 znaków
BBWWBBWW
BBWWBBWW //   indeks znaku podzielony przez rozmiar pola to indeks pola;
WWBBWWBB //   indeks pola modulo 2 pomaga ustawiać kolory pól, pola mające współrzędne sobie równe modulo 2
WWBBWWBB //   pokryte są jednym kolorem, a pozostałe pola (współrzędne różne modulo 2) drugim kolorem
```

### Zadanie 2. [15p] Dostępne w serwisie SKOS.