### Wprowadzenie do programowania w języku C

grupa RKr, wtorek 16:15-18:00 lista nr 9 (na pracownię 10.12.2018) (wersja 1)

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

Wykorzystaj typ MemDescriptor z listy zadań 8. Napisz funkcję o sygnaturze "int indexerZTiled(int x, int y)", która utworzy 12-bitowy indeks przeplatając bity 6-bitowych współrzędnych x oraz y w następujący sposób: indexerZTiled(x, y)  $\Rightarrow$  y<sub>5</sub> x<sub>5</sub> y<sub>4</sub> x<sub>4</sub> y<sub>3</sub> x<sub>3</sub> y<sub>2</sub> x<sub>2</sub> y<sub>1</sub> x<sub>1</sub> y<sub>0</sub> x<sub>0</sub>.

Przy pomocy \_\_liftbit(x, a, b) zbuduj makro \_\_chopbits(w), które pomiędzy każde dwa bity liczby w wstawi 0, \_\_chopbits(w) = 0 w<sub>5</sub> 0 w<sub>4</sub> 0 w<sub>3</sub> 0 w<sub>2</sub> 0 w<sub>1</sub> 0 w<sub>0</sub> . To pozwoli zaimplementować indekser oraz funkcję "MemDescriptor layToZTiled(MemDescriptor m)" kopiującą pamięć z deskryptora m konwertując układ pamięci na ZTiled z dowolnego układu w deskryptorze m; asymptotyczny czas indeksera to O(#bits).

Tak jak na poprzedniej liście stwórz szachownicę rozmiaru 64x64 znaków posiadającą pola rozmiaru 8x8. Zastosuj do niej layToZTiled(.) zmieniając układ pamięci na rekurencyjnie/hierarchicznie Z-kafelkowany. W szachownicy nadpisz jedną wybraną 64-znakową linię (jest ich 64) ignorując jej szyk/indekser:

ABCDEFGHIJKLMNOP#######################abcdefghijklmnop

Po zapisaniu tej linii, stworzoną do tego celu funkcją "void overwriteLine(MemDescriptor m)", wypisz szachownicę na standardowym wyjściu używając szyku zgodnego w tej chwili z pamięciowym, czyli ZTiled.

Jeśli indeksy x oraz y składają się z liczby bitów, która jest potęgą dwójki, to można w elegancki sposób napisać funkcję, która wyliczy liniowy indeks w liczbie operacji proporcjonalnej do logarytmu z liczby bitów. W sprzęcie elektronicznym można zapleść ścieżki i taka operacja kosztuje jedną instrukcję niezależnie od liczby bitów, działając w asymptotycznym czasie O(1), nasza wersja będzie działała w czasie O(log<sub>2</sub>(#bits)).

Załóż, że współrzędne mają 8 bitów i napisz funkcję o sygnaturze "short chopbits8(short x)" oraz funkcję o sygnaturze "short zipbitsZ8(short x, short y)", która podobnie jak indexerZTiled wyliczy:

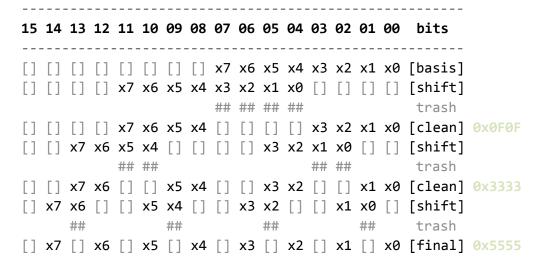
```
zipbitsZ8(x, y) = y_7 x_7 y_6 x_6 y_5 x_5 y_4 x_4 y_3 x_3 y_2 x_2 y_1 x_1 y_0 x_0.
```

Budżet na każdą z operacji bitowych (<< & |) wynosi  $log_2(8 bitów) = 3 sztuki$ . Jeśli nie masz pomysłu na przebieg operacji bitowych, a brak satysfakcji Ci nie przeszkadza, zajrzyj do podpowiedzi na drugiej stronie : ) Można zmienić układ Z na  $\mathcal U$  (rosyjska litera 'i', zamiast niej do oznaczenia używa się czasem podobnego  $\mathcal V$ ). Zastanów się, co trzeba zrobić z bitami aby uzyskać układ o kształcie  $\mathcal V$ ?

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

# post-overwrite 8x8 field:

### bit-flow for chopbits8:



# recursive/nested Z-Tiled pattern:

