Korszerű Programozási Technikák

KisZH II

A feladatban a megadott *generateDataComplete* függvény azt szimulálja, ahogy egy szürkeárnyalatos képről hisztogramot készítünk, de véletlenszerű értékeket használ. Az adatokat egy **map**-ben tárolja, melyben a kulcs a szín, az érték a darabszám. A két paraméter a **map** és a darabszám. A harmadik egy **mutex**, amit majd a feladatok során kell használni. A generálást kisebb részekre osztva, azt a *generateDataPart* függvénnyel végezteti el. A feladat a generálás ellőrzése, és a futási idő meghatározása. A generálást egy szál fogja végezni, míg egy másik szál folyamatosan figyeli a folyamat állapotát.

1. Készíts egy lambda függvényt, amely futtatja a *generateDataComplete* függvényt úgy, hogy a *main*-ben megadott histogram **map**-et töltse fel a szintén megadott ***total_generation_count*** mennyiségű adattal. Futtasd a lambda függvényt külön szálon. (1 pont)

2. Hozz létre két **mutex**-et a *main*-ben. Az egyik arra lesz jó, hogy az előbbi lambda függvény jelölje vele a generálás futását. A másikat el kell juttatni a két generáló függvényhez. A generálás több részre oszlik, de amíg a *generateDataPart* függvény generál, addig az adathoz más nem nyúlhat, a másik **mutex**-nek erről kell gondoskodni. (1 pont)

3. Készíts egy másik lambda függvényt, amely egészen addig fut, amíg a generálás tart. Ezt az első **mutex** folyamatos ellenőrzésével éri el. Minden ellenőrzés előtt a függvény várjon 50 millisec-et. Minden ellenőrzésnél várja meg, míg az adat elérhető (vagyis a *generateDataPart* függvény nem fut, amit a második **mutex** segítségével ellenőriz le), majd számolja meg, hogy hol jár a generálás (össze kell adni a map-ben lévő értékeket). Ezt a számot jelenítse meg a függvény, valamint azt is, hogy ez hány százaléka a teljesnek. Ezt a függvényt futtasd külön szálon. (2 pont)

4. A **chrono** névtér segítségével határozd meg, hogy a *main* függvény futása az elejétől a végéig mennyi időt vesz igénybe, és ezt jelenítsd meg (másodpercben). (1 pont)