Eötvös Loránd Tudományegyetem

Informatikai Kar

Programozási Nyelvek és Fordítóprogramok Tanszék

Osztott rendszerek specifikációja és implementációja

IP-08bORSIG

Dokumentáció az *3.* beadandóhoz

`Szalóki Sándor`
`H8L59S`

2017. december 29.

1. Kitűzött feladat

Egy előre megadott fájlban képek szöveges reprezentációját találhatjuk (pixelenként RGB módon).

A program parancssori paraméterként kapja meg az alábbiakat: A képek átméretezési arányát %-ban. (pl. 50 - ekkor 50%-ra, azaz felére kell csökkenteni az összes képet. 25 esetén negyed-méretet kapnánk, etc.) Annak a fájlnak a neve, a képek primitív leírását tartalmazza. (pl. 'pictures.txt') A kimeneti fájl neve (pl. 'picross.quiz') A fájlból (első param.) beolvasott képeket először a megadott arányban át kell méretezni.

Ezek után az így kapott kisebb képek színeit kell leképezni az előre megadott 8 szín valamelyikére.

Ezt követően az így kapott ábrákban minden sorra és oszlopra ki kell számolni, hogy egymás után hány azonos színű pixelt láthatunk (de nem szimplán azt, hogy az adott sorban/oszlopban hány különböző szín található).

A kapott eredményeket (a méretezett és megfelelő színre konvertált képeket és a hozzájuk tartozó címkéket) írjuk ki a kimeneti fájlba (3-ik paraméter)!

2. Felhasználói dokumentáció

A program futtatásához parancssoros indítás szükséges. Az első paraméter a képek átméretezésének százaléka, majd a képeket tartalmazó fájl, majd a kimeneti fájl neve.

2.1. Rendszer-követelmények, telepítés

A programunk több platformon is futtatható, dinamikus függősége nincsen, bármelyik, manapság használt PC-n működik. Külön telepíteni nem szükséges, elég a futtatható állományt elhelyezni a számítógépen.

2.2. A program használata

A program futtatásához parancssoros indítás szükséges. Az első paraméter a képek átméretezésének százaléka, majd a képeket tartalmazó fájl, majd a kimeneti fájl neve.

3. Fejlesztői dokumentáció

A feladatot 4 részre osztjuk. 1 master folyamat, ami a 3 child folyamatot kezeli, mint részfolyamat. Az implementáció egy Adatcsatorna-feladat, melynek a függvényei (first, second, third child) a feladatban megfogalmazott lépéseket tartalmazzák, majd a master (mint F függvény) egyesíti az egész feladatot. Az egyes részfolyamatokat taskfarmok, illetve D&C módon oldjuk meg.

3.1. Megoldási mód

A master folyamat elindítja a 3 child folyamatot. A master először beolvassa a képeket (ez a tételben használt D), majd egyesével elküldi az első gyerek folyamatnak (f_1), majd várja, hogy az utolsó folyamat is végezzen a kép feldolgozásával (f_3), majd a kapott eredményt kiírja egy fájlba. Az első folyamat felelős a kép átméretezéséhez, amit egy DC megoldással oldottunk meg. A képet több kisebb részre osztva számoljuk ki az átméretezett új képet. Az így kapott képet küldi tovább a második gyerek folyamatnak, ami egy taskfarm segítségével soronként kiszámolja az új kép színeit, majd ezt küldi tovább. Az utolsó, harmadik gyerek folyamat szintén taskfarmok segítségével soronként, majd oszloponként kiszámolja a végső kép címkéit. Ezt a címkézett képet küldi tovább a master folyamatnak.

3.2. Implementáció

A színek ábrázolására bevezettünk egy Color osztályt, a kép ábrázolására egy Image osztályt. Az Image osztályban található egy 2 dimenziós Color vector a kép színeit tárolva, illetve kettő int vector a címkék tárolására. PVM adatküldéshez a vector-okból C array-t kell létrehozni, így az Image osztályhoz tartozik egy pack, illetve unpack függvény a konvertáláshoz. A D&C, illetve taskfarm implementálásához std::future-t használunk.

3.3. Fordítás menete

A program 5 forráskódból áll: master.cpp, first.cpp, second.cpp, third.cpp, ezek tartalmazzák a fő, illetve gyerek folyamatok kódjait, illetve egy model.hpp header-ből, amely tartalmazza a Color, illetve Image osztályok implementációját.

3.4. Tesztelés

A program helyes futásának teszteléséhez az összes megszokott bemenetre készült 1-1 teszteset. A méretezés 100, 50, illetve 25 értékkekkel tesztelése. A bemeneti fájl 0, 1, sok elemű változatára is került teszt fájl. A sebesség teszteléséhez a párhuzamosításba bevont gépek számát lehetett állítani, ahol a több gép gyorsabb számolási időhöz vezetett.