Rendszerközeli programozás

Miniprojekt: WrapperBMP

Cél:

Egy TrueColor BMP képfájlba el kell rejteni egy tetszőleges (szöveges) fájlt úgy, hogy a továbbra is szabványos képfájl által nyújtott „vizuális élmény” alig változzon, de az elrejtett tartalom visszafejthető legyen.

Módszer:

A 24bit színmélységű azaz TrueColor BMP fájlok szerkezetét a mellékletként megtalálod. Egy ilyen képfájl esetén minden pixel 24 biten van eltárolva. 8 bit határozza meg a kék szín komponenst, majd 8 bit a zöldet végül szintén 8 a piros komponenst, így összesen több, mint 16 millió szín írható le. Ennyit az emberi szem nem is tud megkülönböztetni, ha a színkomponensek értékét kicsit megváltoztatjuk, nem vesszük észre a különbséget. Ez ad lehetőséget arra, hogy a színkomponensek legkisebb helyiértékű bitjeinek módosításával további információkat tároljunk el a képfájlban.

A képfájl „Pixel Array” részének minden 3 szomszédos bájtja egy másik tetszőleges fájl 1 bájtát fogja elrejteni. A bináris pixel adat első bájtjának utolsó (legkisebb helyi értékű) két bitje felülírandó az elrejtendő bájt két legelső (legnagyobb helyi értékű) bitjével. A pixel második bájtjának utolsó 3 bitje a rejtendő bájt következő 3 bitjére cserélődik. Végül a pixel harmadik színkomponensének utolsó 3 bitje a rejtendő bájt utolsó három bitjét fogja tartalmazni. Tegyük fel, hogy az eredeti képfájl 3 szomszédos bájtja az alábbi biteket tartalmaza (minden betű egy bit, azaz 0 vagy 1): „`abcdefgh ijklmnop qrstuvwx`”, az elrejtendő bájt pedig a következő bitekből áll: „**ABCDEFGH**”. Ekkor a képfájl adott 24 bitjének tartalma az alábbiak szerint módosul: „`abcdef`**AB** `ijklm`**CDE** `qrstu`**FGH**”.

Természetesen a legtöbb esetben a rejtendő bájtok száma nem egyezik meg a pixelek számával, így a dekódoláshoz tudni kell, hogy hány bájt lett ezzel a módszerrel elrejtve. Ennek tárolására kiválóan alkalmas „Vertical pixel/meter” DIB header mező, amelynek tartalmát legtöbb esetben nem használják. A kódolás során ezt a 4 bájtot kell felülírni a rejtendő fájl bájtban megadott méretét tartalmazó egész típusú változó tartalmával (little-endian bájtsorrendben). A dekódolás természetesen a fentiekkel ellentétesen történik.

Feladat:

Írj egy C nyelvű programot, amely tartalmaz egy alprogramot a kódoláshoz (fájltartalom képbe rejtéséhez) és egy másikat a dekódoláshoz (a rejtett tartalom képből történő kinyeréséhez).

A kódolás menete:

- Töltsd be az elrejtendő fájl teljes tartalmát binárisan egy kellően nagy folytonos memóriaterületre!
- Töltsd be az eredeti képfájl teljes tartalmát binárisan egy másik folytonos memóriaterületre!
- A képtartalom 43.-46. bájtját írd felül a rejtendő fájl méretével!
- A képtartalom 55. bájtjától kezdve írj felül minden bájthármast a kódolás logikájának megfelelően.
- Binárisan írd a módosított képtartalmat egy új fájlba!
- Kezeld az esetleges hibalehetőségeket! (Pl. nem megnyitható fájlok, nem TrueColor bmp, a képhez képest túl nagy rejtendő fájl, stb.)

A dekódolás menete:

- Töltsd be a rejtett infót tartalmazó képfájl teljes tartalmát binárisan egy kellően nagy folytonos memóriaterületre!
- A képtartalom 43.-46. bájtjának tartalmát kezeld egy egész számként, ami a rejtett bájtok számát tartalmazza!
- Hozz létre egy megfelelő méretű folytonos memóriaterületet a „kicsomagolt” fájltartalom számára.
- A képtartalom 55. bájtjától kezdve a korábban meghatározott darabszámú bájthármast a dekódolás logikájának megfelelően felhasználva állítsd elő a lefoglalt területre a rejtett tartalmat!
- A „kicsomagolt” bájtokat binárisan írd bele egy új fájlba!
- Kezeld az esetleges hibalehetőségeket! (Pl. nem megnyitható fájlok, nem TrueColor bmp, stb.)

A programban parancssori argumentumok felhasználásával hívd meg a fenti alprogramokat. (Kódolás esetén két fájlnév szükséges, míg dekódolás esetén elegendő egy.)