

# RENDSZERKÖZELI PROGRAMOZÁS PROJEKTFELADAT – DOKUMENTÁCIÓ

Kohári Ádám

## FORDÍTÁS

A forráskód fordításához Linux rendszeren a **gcc** compiler szükséges. Győződjön meg róla, hogy rendelkezik mindkettő szükséges forrásállománnyal, a *main.c* illetve a *TheProgram.h* fájlal. Navigálja a terminált abba a könyvtárba, ahol a kódokat tárolja, majd gépelje be a következő sort:

```
gcc -o program -fopenmp main.c
```

A parancs sikeres futása esetén létrejön a futtatható állomány *program* néven. Habár a *-o* kapcsoló elhagyható (ez esetben a fordítóprogram egy alapértelmezett nevet fog adni a létrejövő futtatható állománynak) az *-fopenmp* kapcsoló mindenképpen szükséges, mivel a szoftver használja az OpenMP API-t (a több magon futtathatóság érdekében).

## FUTTATÁS

A szoftver a felhasználó által megadott *.bmp* kiterjesztésű képfájlban elrejtett titkos szöveget dekódolja, majd küldi el egy webszerverre.

A program kétféleképp futtatható: az input fájl elérési útjának megadásával parancssori argumentumként, vagy argumentum nélkül indítva a beépített fájlállító alprogramot használva. Ha az előbbi módszert választja, navigálja a terminált a program könyvtárába, majd indítsa el a következőképpen:

```
./program [A_KÉPFÁJL_ELÉRÉSI_ÚTJA]
```

Például: `./program cpu.bmp`, ha a betölteni kívánt *cpu.bmp* képfájl a programmal megegyező könyvtárban található.

Ha a beépített fájlállítót kívánja használni, indítsa el a programot argumentum megadása nélkül (`./program`). A képernyőn megjelenik az aktuális felhasználó alapértelmezett könyvtárának tartalma. A mappák között azok neveinek megadásával tud tallózni. Visszalépni a mappahierarchiában a „`..`” segítségével tud. Ha eljutott abba a mappába, ahol a képet tárolja, csak adja meg annak nevét és a program elkezd működni.

Mindemellett létezik két kapcsoló is: `--version` illetve `--help`. Előbbi némi információt szolgáltat a programról: verziószám, elkészültének dátuma, a fejlesztő neve. Utóbbi az alkalmazás futtatási lehetőségéről ad rövid tájékoztatást.

## A PROGRAM ÁLTAL VISSZAADOTT ÉRTÉKEK

- **Sikeres** futás esetén a **0** értéket adja az operációs rendszernek, valamint a képernyőn is megjelenik egy nyugta
- **Fájlkezelési hiba** esetén az **1** értéket adja
- **Memóriakezelési hiba** esetén a **2** értéket adja

Minden hiba esetén a felhasználó szöveges formában is tájékoztatást kap annak okáról.

# RENDSZERKÖZELI PROGRAMOZÁS PROJEKTFELADAT – DOKUMENTÁCIÓ

Kohári Ádám

## A PROGRAM SZERKEZETE

A *main.c* forrásfájl a főprogramot, a *TheProgram.h* pedig az abban használt alprogramok definícióját tartalmazza.

Az alkalmazás argumentum nélküli indításakor a `BrowseForOpen()` paraméter nélküli függvény kerül meghívásra, mellyel betallózható a kívánt képfájl. Visszatérési értéke egy alacsony szintű integer fájlleíró, mely a betallózott képfájltra „mutat”.

A `ReadPixels(int f, int* NumCh)` függvény első paramétere egy alacsony szintű integer fájlleíró, melyet vagy a főprogram szolgáltat („argumentumos” indítás esetén) vagy az előbb említett `BrowseForOpen()` függvény adja neki. A második paramétere egy integerre mutató pointer, pontosabban annak a változónak a címe, mely a képből beolvasott titkos szöveg karaktereinek számát tárolja (ezt a képfájl headerjéből olvassa ki). A függvény beolvassa a képfájl pixeladatait, majd egy dinamikusan lefoglalt memóriaterületen eltárolja azokat. Ezen memóriaterületre mutató pointer lesz a függvény visszatérési értéke.

Az `Unwrap(char* Pbuff, int NumCh)` függvény első paramétere az előbb tárgyalt függvény által visszaadott memóriaterületre mutató pointer. Második paramétere hasonló a `ReadPixels()` `NumCh` paraméteréhez, azzal a különbséggel, hogy itt nem a változó címe, hanem annak értéke kerül átadásra. A függvény a paraméterként megadott helyen található pixeladatokból kinyeri a titkos szöveget, majd egy dinamikusan lefoglalt memóriaterületen eltárolja. Az erre mutató pointer lesz a függvény visszatérési értéke.

A `Post(char *neptunID, char *message, int NumCh)` függvény első paramétere a NeptunID-t tartalmazó string, mellyel a webszerverre beküldött üzenetek azonosíthatóvá válnak. A második az átküldeni kívánt üzenetre (string) mutató karakter pointer, azaz az `Unwrap()` függvény által visszaadott érték. A harmadik paraméter megegyezik az `Unwrap()` második paraméterével.

A `WhatToDo(int sig)` szignálkezelő eljárás egyetlen paramétere egy egész szám, mely elárulja a kapott szignál típusát. Az eljárás kétféle szignált kezel:

- **Interrupt** szignál esetén elindít egy új PID-val rendelkező folyamatot a pillanatnyi állapotból, majd a jelenlegi folyamat leáll
- **Alarm** szignál esetén megjelenít egy hibaüzenetet, majd a folyamat leáll