

# **„Nagyon kreatív program név”**

## **v1.0**

## *„Nagyon kreatív program név” v1.0*

### **A program célja:**

Linux alatt használandó, C nyelven írt, egy TrueColor BMP fájlba beágyazott „láthatatlan” szöveget kicsomagol és azt egy HTTP POST metódussal egy megadott web szerverre eljuttatja.

A program C nyelven íródott Linux környezetre, C99 language standardet használva.

### **A programhoz szükséges fordítóprogram és a szükséges argumentumok:**

A program Linux alatt a „gcc” compilerrel fordul le, a következő módon:

```
gcc main.c -o rkp -fopenmp -Wall
```

### **A program futtatása:**

A program sikeres fordítás után parancssorból futtatható, a következő módon:

```
./rkp [args]
```

### **Lehetséges argumentumok:**

**--version** : Visszaadja a program nevét, verzióját, készítőjét, compile dátumát.

**--help** : Visszaadja a lehetséges parancssori argumentumokat.

**filename.bmp** : Megnyitja a megadott BMP fájlt feldolgozásra.

Amennyiben nincs megadva parancssori argumentum, a tallózó függvény kerül meghívásra, amellyel kiválaszthatjuk a feldolgozni kívánt BMP fájlt.

## A függvények:

```
int main(int argc, char* argv[])
```

A program main függvénye, itt vizsgáljuk meg az argumentumokat és ennek megfelelően döntünk hogyan fusson tovább a program.

```
int BrowseForOpen() {
```

A tallózó függvény, ez fut le amennyiben nem adtunk meg semmilyen parancssori argumentumot, visszatérési értéke egy fájlleíró.

```
void Decode(int fd)
```

A main-ből hívott függvény, ez intézi a további futást, először a mainből átadott fájl leírót a ReadPixels, majd az abból kinyert adatokat az Unwrap függvények adja tovább, végül pedig a Post függvénnyel elküldi az adatokat.

```
char* ReadPixels (int f, int* numCh)
```

Az átadott BMP fájl header részéből kiolvassa a megfelelő adatokat, visszatérési értéke egy pixel array buffer és a kiolvasandó karakterek száma bájtokban megadva.

```
char* Unwrap(char *Pbuff, int numCh)
```

A ReadPixels függvénytől kapott adatok további feldolgozása, itt történik a kódolt szöveg tényleges dekódolása. Visszatérési értéke a dekódolt szöveg címére egy mutató.

```
int Post(char* neptunID, char* message, int numCh)
```

A http POST küldést megvalósító függvény, itt történik a kibontott szöveg beillesztése egy http requestbe, majd annak elküldése a beégetett webszerver címére. Visszatérési értéke egy kód amely a küldés állapotát jelzi: 0, ha sikeres volt a küldés, 2-5 ha valamilyen hiba lépett fel.

```
void WhatToDo(int sig)
```

Szignálkezelő eljárás, túl hosszú futás (>1mp) esetén leállítja a programot, illetve amennyiben az Unwrap függvény futása közben Ctrl+C (Interrupt signal) utasítás érkezne, azt elkapja és közli a felhasználó felé, hogy jelenleg nem használható az interrupt. Visszatérési értéke nincs.

## *„Nagyon kreatív program név” v1.0*

### **A programhoz szükséges:**

- Linux környezet
- „ileuxt.h” header fájl egy mappában a programmal
- egy BMP fájl a kódolt szöveggel

### **Példa sikeres lefutásra parancssoros Ubuntu Linux környezetben:**

```
je [redacted] rkp$ ./rkp cpu.bmp  
217 bytes have been sent to server.
```