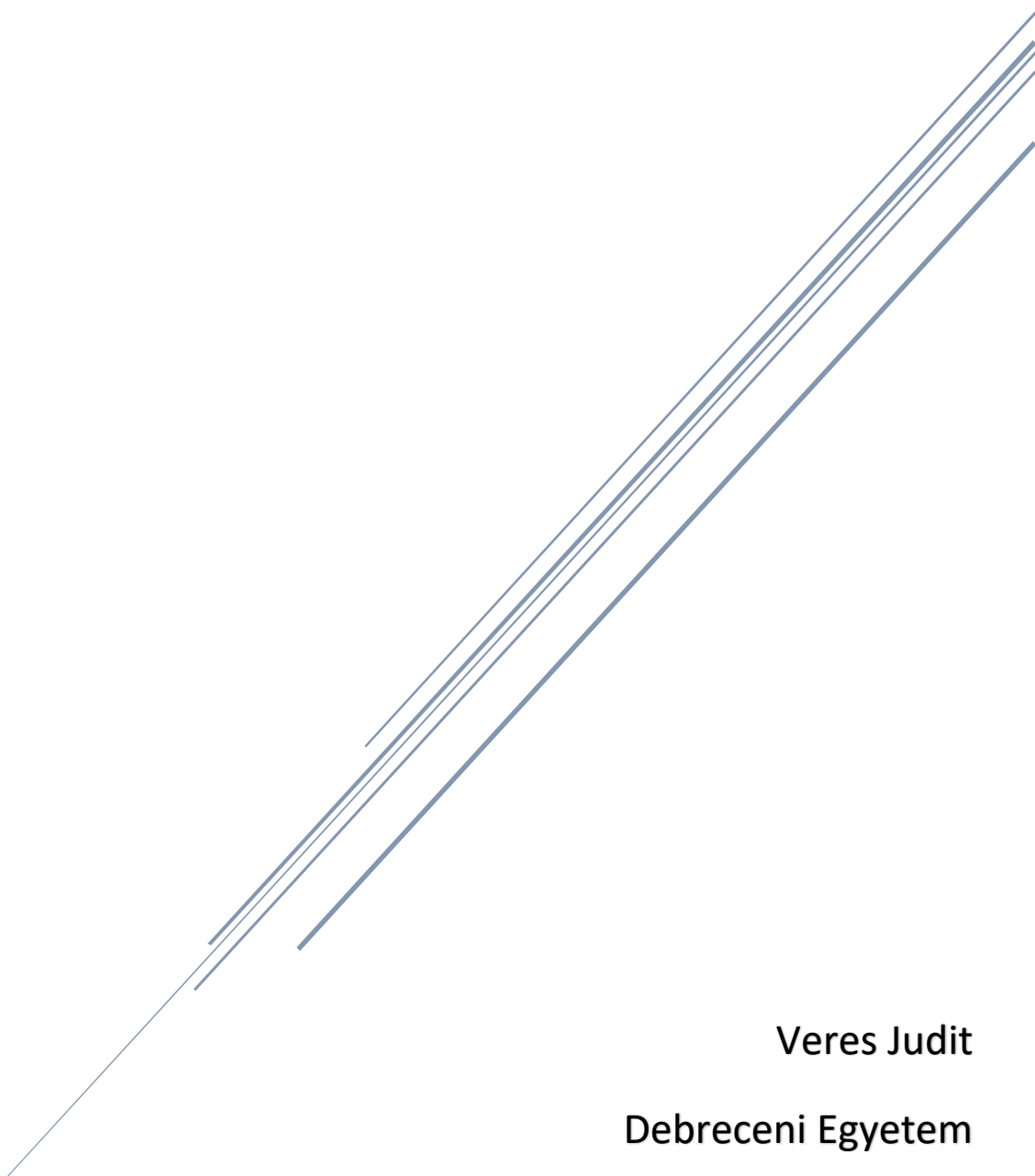


# A PROJEKT

Rendszerközeli programozás



Veres Judit

Debreceni Egyetem

2021.04.20

# Tartalom

A program leírása.....	2
A programkönyvtár tartalma .....	2
Fordítóprogram és annak szükséges kapcsolói .....	2
Felhasználói útmutatás .....	2
A program által visszaadott értékek jelentése .....	3
A programban felhasznált alprogramok leírása .....	3

## A program leírása

A Rendszerközeli programozás című tárgy során kapott projekt feladatban egy C nyelven írt programot kellett elkészíteni, amely egy TrueColor bmp fájlba ágyazott „láthatatlan” szöveget csomagol ki és azt egy HTTP POST metódussal egy megadott webszerverre továbbítja.

## A programkönyvtár tartalma

A program könyvtárában két .c forrásfájl, egy .h header állomány és két (rejtett) .txt fájl található, a megfelelő működéshez ezeknek ugyanabban a könyvtárban kell lenniük.

A fájlok a következők:

- main.c
- my\_tools.c
- my\_tools.h
- .help.txt
- .version.txt

## Fordítóprogram és annak szükséges kapcsolói

A szükséges fordítóprogram: GNU C Compiler 4.2 (vagy későbbi verzió)

A következő paranccsal lehet fordítani a programot terminálból:

```
gcc main.c my_tools.c -o Project -fopenmp -std=gnu99
```

## Felhasználói útmutatás

A program fordítása után lehetséges opciók a futtatásra terminálban:

```
./Project --version
```

- Megjelenik a program verziószáma, a kiadás dátuma és a fejlesztő neve.

```
./Project --help
```

- Rövid leírást ad a programról, valamint útmutatót a lehetséges futtatásokhoz.

```
./Project xxx.bmp
```

- Ahol az "xxx.bmp" a bemeneti fájl neve, amelyet a program fel fog dolgozni.

```
./Project
```

- A program elkezd futni, a bemeneti fájlt karakteres tállózás segítségével lehet kiválasztani.

## A program által visszaadott értékek jelentése

A programnak kilenc különböző visszatérési értéke lehet, az alábbi értelmezésekkel:

- 0: Sikeres végrehajtás
- 1: Fájlmegnyitási / Memóriafoglalási hiba
- 2: Túl hosszú futási idő
- 10: Információk (help, version)
- 11: Socket létrehozási hiba
- 22: TCP kapcsolódási hiba
- 33: Szegmensküldő hiba
- 44: Szegmensfogadó hiba
- 55: A szerver által küldött válasz nem volt megfelelő

## A programban felhasznált alprogramok leírása

A program nyolc alprogramot (5 függvény, 3 eljárás) tartalmaz a main programegységen kívül. Ezek a következők:

### **message txt**

Fejléc: `void message_txt(char *s)`

Feladat: Az eljárás binárisan nyit meg egy fájlt, majd a tartalmát a standard kimenetre írja.

Kapott paraméter(ek):

s – a megnyitandó fájl nevének kezdőcíme

Visszatérési érték: Nincs.

### **open bmp file ronly**

Fejléc: `int open_bmp_file_ronly(char *filename)`

Feladat: A függvény egy .bmp fájlt nyit meg binárisan, kizárólag olvasásra.

Kapott paraméter(ek):

filename - a megnyitandó fájl nevének kezdőcíme

Visszatérési érték: A fájl megnyitása során kapott fájlleíró érték, vagy -1, ha a fájl nem egy .bmp állomány volt.

### **list\_dir**

**Fejléc:** void list\_dir()

**Feladat:** Az eljárás kilistázza az aktuális könyvtár tartalmát.

**Kapott paraméter(ek):** Nincs.

**Visszatérési érték:** Nincs.

### **BrowseForOpen**

**Fejléc:** int BrowseForOpen()

**Feladat:** Lehetővé teszi egy fájl megtalálását karakteres tallózás segítségével, majd az open\_bmp\_file\_only függvény segítségével megnyitja a fájlt bináris olvasásra.

**Kapott paraméter(ek):** Nincs.

**Visszatérési érték:** A fájl megnyitása során kapott fájlleíró érték.

### **ReadPixels**

**Fejléc:** char \*ReadPixels(int file, int \*NumCh)

**Feladat:** Egy binárisan megnyitott BMP fájl ún. PixelArray részét olvassa be a memóriába.

**Kapott paraméter(ek):**

file: A binárisan megnyitott fájl leírója.

NumCh: A beolvasott PixelArray-ben kódolt szöveg karaktereinek számát erre a címre menti el a függvény.

**Visszatérési érték:** A beolvasott PixelArray memóriacíme.

### **Unwrap**

**Fejléc:** char \*Unwrap(char \*Pbuff, int NumCh)

**Feladat:** A 24 bites pixelinformációkba rejtett szöveg karaktereinek kicsomagolását végzi el.

**Kapott paraméter(ek):**

Pbuff: A pixeleket (amelyekben kódolva van szöveg)tartalmazó PixelArray memóriacíme.

NumCh: A beolvasott PixelArray-ben kódolt szöveg karaktereinek száma.

**Visszatérési érték:** A kicsomagolt szöveget tartalmazó memóriaterület címe.

## **Post**

**Fejléc:** `int Post(char *neptunID, char *message, int NumCh)`

**Feladat:** A dekódolt szöveget (egy Neptun azonosítóval együtt) a megadott webszerverre juttatja el egy HTTP POST metódussal.

**Kapott paraméter(ek):**

neptunID: A fejlesztő egyedi azonosítóját tartalmazó memóriaterület címe.

message: A kicsomagolt szöveg memóriacíme.

NumCh: A dekódolt szöveg hossza.

**Visszatérési érték:** A küldés sikerességét jelző érték. (Lásd: visszaadott értékek jelentése.)

## **WhatToDo**

**Fejléc:** `void WhatToDo(int sig)`

**Feladat:** Szignálkezelő eljárás. SIGINT szignál esetén fork művelet után tájékoztatást ad, hogy a CTRL + C billentyűkombinációval nem lehet kilépni a programból, majd befejezteti a gyerekfolyamat saját magát. SIGALRM esetén leállítja a programot egy hibaüzenettel.

**Kapott paraméter(ek):**

sig – a kapott szignál kódja

**Visszatérési érték:** Nincs.