

## Dokumentáció

### 1. A program fordítása

A program fordítását gcc-vel végezzük el. Fontos, hogy a customheader.h saját header állomány ugyanabban a könyvtárban legyen, mint a program forráskódja.

A következő képpen tudjuk lefordítani a programunkat:

```
gcc project.c -o project -fopenmp
```

### 2. Felhasználói útmutató a programhoz

A program futtatásánál választhatunk, hogy adunk-e parancssori argumentumot vagy nem.

Ha a programot a --help parancssori argumentummal futtatjuk, akkor egy rövid segédletet fog kiírni a futtatásról:

```
./project --help
```

Ha a programot a --version parancssori argumentummal futtatjuk, akkor ki fogja írni a program verziószámát, elkészültének dátumát, illetve a készítő nevét:

```
./project --version
```

Megadhatjuk parancssori argumentumként a dekódolandó fájl nevét. Abban az esetben ha ez a fájl létezik a program könyvtárán belül, a program megnyitja olvasásra. Ha nem létezik ilyen fájl vagy ha a fájl megnyitása sikertelen, akkor hibaüzenetet ír ki és 1 értéket ad vissza.

```
./project cpu03.bmp
```

Ha nem adunk meg parancssori argumentumot egy tallózó függvény kerül meghívásra, amely kiírja az aktuális felhasználó alapértelmezett könyvtárának tartalmát. Ezen belül tudunk navigálni oly módon, hogy beírunk egy könyvtár vagy fájlnevet. Amennyiben könyvtárnevet írtunk be, a program belelép és kilistázza a tartalmát. Ez addig ismétlődik ameddig nem adunk meg fájlnevet. A program nem áll le, ha a megadott szöveg se nem könyvtár se nem fájl, hanem új szöveget kér be. A program lehetővé teszi a szülő könyvtárba való lépést, két pont megadásával: ..

Ha sikeres volt a fájlbeolvasás, akkor a program beolvassa egy tömörítetlen, TrueColor bmp fájl tartalmát, dekódolja a benne rejtett szöveget és ezt továbbítja egy webszerverre.

A sikeres feltöltésről megerősítést ad egy a képernyőre kiírt szöveggel.

### 3. A program által visszaadott értékek magyarázata:

- 0: a program sikeresen lefutott, nem történt hiba.
- 1: sikertelen memóiafoglalásra utal vagy sikertelen fájlnyitásra.
- 2: socket létrehozása sikertelen volt.
- 3: csatlakozás a webserverre sikertelen volt.
- 4: az üzenet elküldése sikertelen volt.
- 5: válaszüzenet fogadása sikertelen volt.
- 6: a válaszként kapott üzenet alapján a elküldött üzenet nem, vagy hibásan érkezett meg.

Ha ezek közül bármelyik hiba bekövetkezik, akkor a program a standard hiba felületre hibaüzenetet ír ki és kilép.

### 4. Az elkészített alprogramok rövid leírása(paraméterezés, visszatérési érték):

- `char* Unwrap(char* Pbuff, int NumCh);`

Paraméterként kap egy korábban megadott módon kódolt bmp fájl pixel array részét tartalmazó memóriaterület címét, illetve a kódolt karakterek számát. A függvény lefoglal a dekódolt szöveg számára pontosan elég memóriaterületet, ide menti a dekódolt karaktereket. A dekódolt szöveg címével tér vissza. Ha a memóiafoglalás sikertelen volt, akkor hibaüzenetet ír ki, a program leáll és 1 értéket ad vissza

- `char* ReadPixels(int f, int* NumCh);`

A függvény első paramétere egy binárisan olvasásra megnyitott fájl leírója. A második paraméter output paraméterként működik,erre a címre helyezi el a program a kódolt karakterek számát. A függvény beolvassa a képfájl teljes pixel array tartalmát, és egy megfelelő méretű dinamikusan lefoglalt memóriaterületre helyezi, és ennek a címével tér vissza. Ha a memóiafoglalás sikertelen volt, akkor hibaüzenetet ír ki, a program leáll és 1 értéket ad vissza.

- `int BrowseForOpen();`

A függvény kiírja a kimenetre az aktuális felhasználó alapértelmezett könyvtárának tartalmát. Akkor hívódik meg, ha nem adunk meg parancssori argumentumot. A felhasználó adja meg, hogy melyik könyvtárba lépjen be, illetve melyik fájlt nyissa meg a program olvasásra. A függvény a megnyitás során kapott fájlleíróval tér vissza.

- `int Post(char *neptunID, char *message, int NumCh);`

Ez a függvény továbbítja egy webserverre a dekódolt szöveget. Az első paraméter egy pointer, amely a Neptun azonosítót tartalmazó stringre mutat. A második paraméter egy a dekódolt szövegre mutató pointer. A harmadik paraméter a szöveg mérete bájtokban. A függvény létrehoz egy TCP kapcsolatot, és elküldi az üzenetünket. Ha a szervertől kapott válasz tartalmazza a "The message has been received." szöveget, akkor a függvény bontja a kapcsolatot és 0 értékkel tér vissza. Ha hiba történik, akkor kiír egy hibaüzenetet és a program visszatér egy hibára utaló pozitív értékkel.

Lehetséges hibakódok:

- 2: socket létrehozása sikertelen volt.
- 3: csatlakozás a webserverre sikertelen volt.
- 4: az üzenet elküldése sikertelen volt.
- 5: válaszüzenet fogadása sikertelen volt.
- 6: a válaszként kapott üzenet alapján a elküldött üzenet nem, vagy hibásan érkezett meg.

- `void WhatToDo(int sig);`

Az eljárás képes két szignált is kezelni. Ha a paramétere SIGALRM szignálra utal, akkor hibaüzenetet ír ki, hogy a program túl sokáig futott, majd a program leáll egy hibakóddal. Ha a SIGINT szignál aktiválta a kezelőt, akkor az eljárás létrehoz egy gyerek folyamatot, amely tájékoztatja a felhasználót, hogy a ctrl+c nem állítja meg a programot, majd a gyerek folyamat be is fejeződik a saját magának küldött SIGKILL szignál által. Ez a szignálkezelő eljárás a képfájl megnyitása után érkező szignálok esetén fut le. A fájl megnyitása után elindít egy 1 másodperces időzítőt, amit dekódolás után leállít.