RKP Projekt - Rendszerközeli programozás projekt feladat

Készítette: Berecz Zsolt - GN6W3I

Program leírása:

> Hozzunk létre egy olyan Linux alatt futó programot ami egy TrueColor bmp fájlban beágyazott szöveget kicsomagol és elküld egy HTTP POST segítségével egy megadott webszerverre.

Használandó fordító programok és kapcsolók:

A program C nyelven íródott ezért szükséges egy C fordító ez esetben a GNU Compiler Collection-t ajánlott használni.
Linux operációs rendszer esetén a telepítés terminál segítségével:

Debian:
`$ sudo apt-get update`
`$ sudo apt-get install gcc`

Arch:
`$ sudo pacman -S gcc`

CentOS/RedHat:
`$ sudo yum install gcc`

Felhasználói útmutatás:

A program futtatása terminál segítségével történik ha még nem fordítottuk le a programot akkor szükséges ez esetben, ha a **program.c**-ben lévő mappában vagyunk terminál segítségével le tudjuk fordítani:

`$ gcc program.c -o program`

Több magos futattás esetén szükséges egy kapcsoló OpenMP inicializálása

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

`-o`

Kapcsoló esetén a nevét adjuk meg a futattható állománynak.

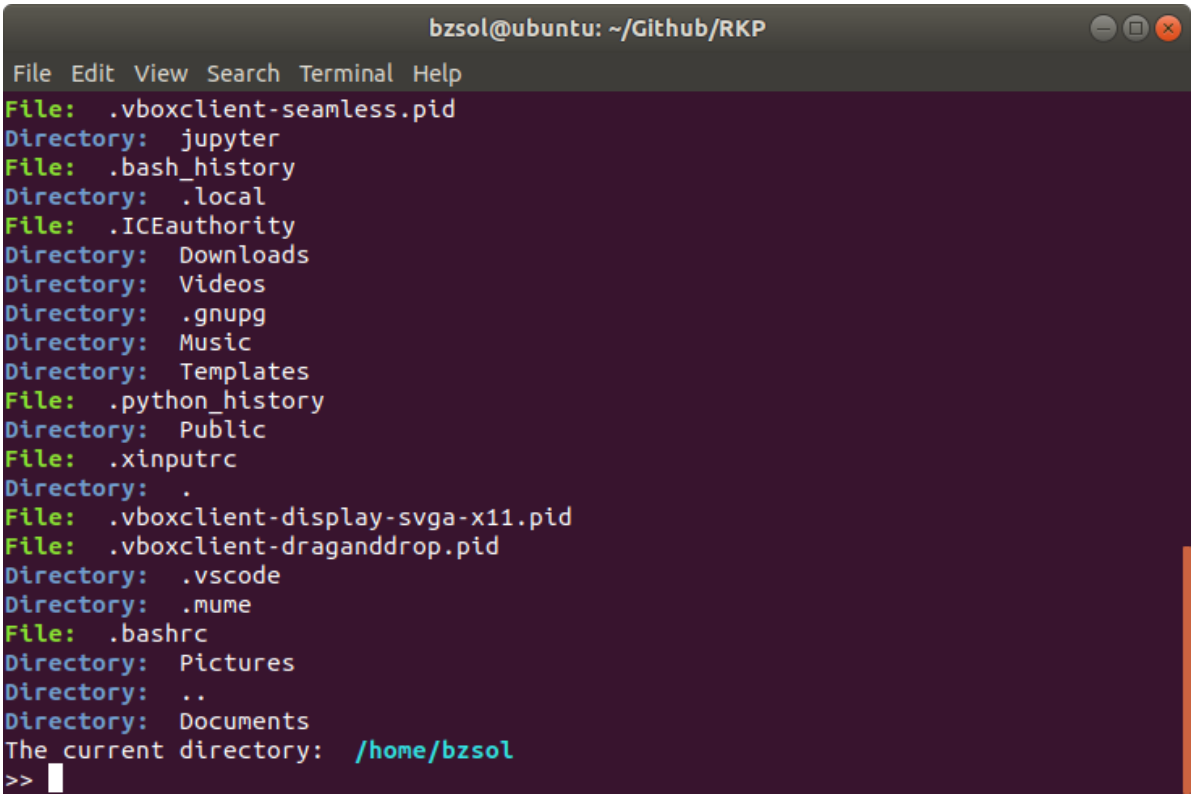
`-fopenmp`

Kapcsoló esetén bővítjük a programot a gcc OpenMP implementációjával.

Lefordítás után kapunk egy **.program** nevezetű futtatható állományt.
Ezt követően a következő parancsot írjuk be:

`$ ./program`

Ezt követően egy file böngészőt láthatunk:



Fájl keresése a file böngésző segítségével:

A file böngésző nagyon hatékonyan dolgozik mivel akár csak a mappa vagy a fájl neve elég hogy keresni tudjunk a mappánkban vagy az adott .bmp fájlt megnyitni.
Ha direkt elérési útvonalat szeretnénk bátran használható ez az opció is.

Ha vissza szeretnénk navigálni az előző mappára `/../` vagy `..` segítségével tudunk!

File megnyitása parancssori argumentumként

Ha ezt az opciót szeretnénk használni egyszerűen mellé írjuk a program futtatása előtt a file nevét:

```
$ ./program cpu.bmp
```

További parancssori argumentumok:

```
$ ./program --help
```

Ezzel az argumentummal megtudhatóak a program használatához szükséges tevékenységes és paraméterek.

```
$ ./program --version
```

Ha ezt a parancssori argumentumot írjuk be megtekintehtő lesz a feladat készítője az elkészítés dátuma.

Fontos: A program nem zár be Ctrl+C kombináció esetén mellé ha időtúllépés lépne fel a program bezár.

Program által visszaadott hibakódok:

- **Hibakód: 0**: Ez esetben a program tökéletesen lefutott és egy zöld szöveg segítségével tájékoztat hogy sikeresen megtörtént a kapcsolat a szerverrel és a képből a kód hibátlanul ki lett kódolva.
- **Hibakód: 1**: Memóriafoglalás hiba történt,ilyen esetben érdemes megnézni mennyi szabad memóriával rendelkezünk vagy pedig a program újra fordítása adhat segítséget.
- **Hibakód: 2**: A megnyitott fájl nem valós TrueColor bmp file, ezért a program nem tudja a kívánt feladatot elvégezni.
- **Hibakód 3**: Webszerver ahova csatlakoznánk nem elérhető vagy nem létezik.
- **Hibakód 4**:Hálózati csomag létrehozásakor hiba történt.
- **Hibakód 5**: A szerverhez való csatlakozás meghiúsult.
- **Hibakód 6**: A csomag (dekódolt szöveg) elküldése során hiba történt.
- **Hibakód 7**: A visszaérkező válasz szöveg nem érkezett meg és emiatt hibás a csomag küldés
- **Hibakód 8**: Hibás a csomag elküldése ez esetben egy HTTP hibakód érkezett.
- **Hibakód 9**: A szignál kezelés során hiba történt azaz a kódolás nem történt meg időben (ez esetben 1 másodperc).
- **Hibakód 10**: Ha több parancssori argumentumot adunk meg a programnak ezzel a hibával fogunk találkozni.

RKP.h tartalma:

void mallocFail

Ha memóriafoglalás probléma történik akkor ez az eljárás hívódik meg, gyors hibakezelés.

char* ReadPixels
Ez a függvény azért felel hogy egyrészt megnézi hogy az adott file BMP képfájl ezek után ha ez sikeres,a BMP fej állományban nem használt területen található karakterszámot kiolvasva nem mellé a kezdő pixelt ahol kezdődnek a kép pixelek és beolvassuk a pixeleket egy memóriaterületre.

Paraméterek:

Az f változó a megnyitott file bináris beolvasása után egy int bemeneti paraméter a *NumCh a titkos szöveg karakterszáma ez egy int pointer(mutató).*
Kimenet egy char pixel tömb amiben a titkosított pixelek találhatóak.

int BrowseForOpen
Ez a függvény azért felel hogy a felhasználó számára egy könnyen kezelhető file keresőt tudjon használni, mappa esetén belelép ha fájlt,akkor pedig igyekszik megnyitni.

(Csak is BMP képfájlt tud kezelni).
Paraméterek:

Nincs bemeneti paraméter ellenben a kimeneti paraméter egy int lesz ami a BMP file bináris olvasása(csak olvashat) után a ReadPixels függvény kap meg.

char* Unwrap
Ebben a függvényben a megkapott memóriaterület ahol a titkosított pixelek találhatóak és ezzel a metódussal ki tudjuk kódolni a titkosított szöveget és vissza tudjuk adni egy olyan memóriaterületre ahol a kikódolt szöveg található.

Paraméterek:
*Pbuff ami a lefoglalt pixelekre mutató pointer ez névlegesen a pixel lefoglalt memória terület.

Egy NumCh ami a titkos szöveg karakterszáma.
Kimenet pedig egy str azaz maga a titkos szöveg.

int Post
Egy HTTP POST segítségével elküldjük kikódolt szövegünket, HTTP 200-as kód esetén egy üzenet elküldve üzenet fog érkezni, ellenkező esetben egy hiba üzenetet láthatunk.
Paraméterek:
*NeptunID egy "string" ami a Neptun azonosítót tartalmazza hogy a webszerveren lásd hogy ténylegesen te külted el az üzenetet.

*message maga az üzenetre mutató memóriaterület.
A NumCH pedig maga a titkos üzenet karakterszáma.
Kimenetként két érték jöhet vissza. Ha minden rendben zajlott egy 0 érték tér vissza, ha pedig valami hiba történt és nem tudta rendesen elküldeni egy 8-as érték maga a hiba kódszáma. (Ebben a függvényben több hibakód is van de akkor maga a program is leáll.)

void WhatToDo
Szignálkezelő ami azt szolgálja SIGINT és SIGALRM szignálokat elkapja.
SIGINT esetén egy gyermek process-t ami nem engedélyezi a CTRL+C használatát.
Időtúllépés esetén SIGALRM lép életbe és visszaküld egy hibaüzenetet.
Paraméterek:
Bementként egy sig integer változó érkezik. Ha ez a szignál ami érkezik maga az interruption szignál azaz a SIGINT nem engedi hogy leálljon a program. Ha az Unwrap függvény több mint egy másodperc alatt kódolja ki a pixeleket akkor a SIGALRM egy hibát fog kiírni és bezárja a programot.
Ez egy eljárás nincs kimeneti érték.