

Rendszerközeli Programozás Projekt

Svec Antal EZJRP1

2023.05.05 Debrecen

Fordítóprogram és annak kapcsolói

A chart program függ az alábbi könyvtárakban található fájloktól:

```
include/helper/helper.c
include/init/init.c
include/info/info.c
include/data/data.c
include/bmp/bmp.c
include/pid/pid.c
include/communication/communication.c
main.c
```

Ezen fájlok mellett az OpenMP könyvtárat is használja.

A program fordításához telepíteni kell a **gcc**-t és az **OpenMP** könyvtárat. Az OpenMP könyvtárat általában a gcc telepítésekor telepítik. Ha mégsem van telepítve, akkor a következő parancsot kell futtatni:

```
sudo apt-get install libomp-dev
```

A program forráskódjának fordítása szükséges a futtatható állomány létrehozásához. Ezt a **make** parancs segítségével történik

A program megírására használt **gcc** verzió:

Program használati dokumentáció

A program fő feladata egy kommunikációs rendszer működtetése, amely a parancssori argumentumok segítségével konfigurálható. A program használatához a következők a lépések szükségesek:

A program futtatása parancssorban az alábbi módon történik:

./chart [OPCIÓK]

Az OPCÍÓK helyére kerülnek az egyes argumentumok.

A program támogatja a következő opciókat:

--version	Kiírja a program verziószámát, elkészültének dátumát és a fejlesztő nevét.
--help	Kiírja a program használati információit, beleértve az elérhető opciókat és a parancssori argumentumok jelentését.
-send	(Alapértelmezett) A program küldő üzemmódban indul el.
-receive	A program fogadó üzemmódban indul el.
-file	(Alapértelmezett) A program fájl alapú kommunikációs módban indul el.
-socket	A program socket alapú kommunikációs módban indul el.

Ha az érvénytelen argumentumot kap a program, akkor ugyanúgy viselkedik, mint a --help opció.

- Ha a program küldőként működik akkor, az generál adat tömböt és ha a fájlban küldés opció volt választva, a Measurement.txt fájlba kiírja azokat.
- Ha a program küldőként működik akkor, az generál adat tömböt és ha a socketen keresztül történő küldés opció volt kiválasztva, akkor az adatokat 127.0.0.1:3333 ip címen fogja továbbküldeni, amit a fogadó üzemmódban lévő programnak fogadnia kell.
- Ha a program fogadóként működik és fájlon keresztüli kommunikáció van kiválasztva, akkor a létrehozott Measurement.txt fájl tartalmát beolvassa és ebből kigenerál egy BMP képet.
- Ha a program fogadóként működik és socketen keresztüli kommunikáció van kiválasztva, akkor várakozás üzemmódba lesz és a küldőtől várja az adatokat amelyekből kigenerál egy BMP képet.

Alprogramok leírása

Config *init(const int argc, const char *argv[])

A fenti kód egy konfigurációs struktúrát inicializál, amely meghatározza a program futásának módját és kommunikációs módszerét. Paraméterként fogadja a parancssori argumentumokat. Itt történik a **chart** program megnevezésének az ellenőrzés is.

void help()

Kiírja a program sugóját.

void version()

Kiírja a program szerzőjét, verziószámát és dátumát, ezeket külön szálakon írja ki.

int Measurement(int **Values)

A függvény létrehoz méréseket egy adatsorban, amelyek értékei véletlenszerűen változnak az idő múlásával. A függvény visszatér a létrehozott mérések számával, és az értékeket a Values paraméteren keresztül adja át, amely egy int ** típusú pointer. A függvénynek nincs más kimenete.

void SendViaFile(int *Values, int NumValues)

A függvény a mért értékek tömbjét írja ki egy **Measurement.txt** fájlba, majd a chart nevű folyamatot értesíti egy szignállal.

Paraméterek: Values: int típusú pointer, amely a mért értékeket tartalmazó tömbre mutat.

NumValues: int típusú változó, amely a mért értékek számát tartalmazza. Nincs visszatérési értéke.

void ReceiveViaFile()

Ez a függvény egy fájlból olvassa be a méréseket, majd elküldi azokat egy BMP készítő függvénynek. A fájl elérési útvonala a felhasználó home könyvtárában található "Measurement.txt" nevű fájl. Nincs paramétere és visszatérési értéke.

void SendViaSocket(int *Values, int NumValues)

A függvény célja, hogy a paraméterként kapott Values tömbben tárolt mérési eredményeket elküldje a 127.0.0.1:3333 címen lévő szerverre, majd fogadja a választ és ellenőrizze annak helyességét. A függvény nem tér vissza semmilyen értékkel. Az alprogram alarm signal segítségével ellenőrzi válaszolt-e a szerver, mert ha nem akkor leáll a futása. A függvény paraméterei a Values tömb és annak hossza, nincs visszatérési értéke.

void ReceiveViaSocket()

Az alprogram fogadja a bejövő adatokat egy UDP socketen keresztül. A függvény folyamatosan fut, amíg le nem állítják. Szerverként működik és várakozik a küldőktől való adatokat, fogadja azokat ellenőrző válaszokat küld a küldőknek. Ha minden adat helyesen érkezett meg akkor generál egy BMP képet azokból. Nincs visszatérési értéke.

void SignalHandler(int sig)

Ez egy signal handler függvény, amelyet a program a jelzések fogadására és kezelésére szolgál. Az alprogram egyetlen paramétere a kapott szignál száma. A függvénynek nincs visszatérési értéke. A függvény a SIGINT, SIGUSR1 és SIGALRM jelzések esetén a feladatban leírt megfelelő műveleteket hajt végre, legtöbb esetben egy üzenetet ír ki és leállítja a programot.

void error_with_exit(const int code, const char *message)

Az eljárás kiírja a paraméterben megadott üzenetet és kilép hibakóddal, amely szintén paraméterben volt megadva.

Hibakódok

1	A futtatott program neve nem 'chart'
2	Nem sikerült helyet foglalni a méréseknek
3	Hiba fájl megnyitásakor
4	Szerver nem válaszol
5	Nem sikerült létrehozni a socketet
6	Nem sikerült elküldeni a mérési eredményt socketen
7	Nem sikerült fogadni a mérési eredményt socketen
8	A fogadott mérési eredmény nem egyezik a küldöttel
9	Nem sikerült a socketet hozzárendelni a címhez
10	Nem sikerült megnyitni a /proc könyvtárat
11	Nem található a Chart processz