
Chart program

A program két üzemmódban futtatható, működhet feldolgozó illetve adat küldő módban. A küldő fél egy 100+ értéket tartalmazó tömböt továbbít a feldolgozó egységnek (két párhuzamosan futó program) fileok vagy hálózati megoldásokon keresztül (UDP) amit a feldolgozó egység egy BMP file-ba ment grafikonon. File alapú továbbítás esetén a felhasználó home könyvtárába menti [Projekt munka feltétele], egyébként a programot tartalmazó mappába.

Fordítás

A programot gcc segítségével fordítsuk, figyeljünk, hogy minden kapcsoló meg legyen adva.

```
gcc -o chart main.c -lm -fopenmp alprogramok.c
```

Használat

chart [szolgáltatás módja] [továbbítás módja]

Példa: chart -send -socket //A program kliens üzemmódban az információt localhost:3333-ra küldi

Példa: chart -send -file //A program az információt a saját home könyvtárán keresztül küldi feldolgozásra

Kapcsolók

//Információ lekérdezés

--version : Kiíratja a verzió információkat
--help : Kiíratja az alap információkat a programról

//Szolgáltatás módja

-send Adat továbbítás
-receive Adat feldolgozás

//Továbbítás módja

-file File alapú adat továbbítás
-socket Hálózat alapú adat továbbítás

Hibakódok

0 - RUN SUCCESS
1 - USER ERROR
2 - FILE/DIR ACCESS ERROR
3 - NETWORK ERROR
4 - MEMORY ERROR
5 - NO RECEIVER
6 - NETWORK TIMEOUT

9 - COMPILE ERROR

Alprogramok

- void SignalHandler(int sig);
 - SIGINT-re lezárja a server socketet ha fut, szövegesen tájékoztat a bezárásról majd kilép 0-ás hibakóddal
 - SIGUSR1-re ha nem File alapú adatfeldolgozóként indítottuk hibaüzenettel tájékoztat a program.
 - SIGALRM-ra Szerver időtullépés hibával leállítja a hibát és 3-as hibakódot küld. Ez csak az első üzenetküldésnél éles.
- void outHelp();
 - Alapvető információkat írat a konzolra a programról
- int32_t swapEndian((int32_t input);
 - LittleEndian és BigEndian között vált 32 bites int számot, BMP file készítéskor használt alprogram.
- int FindPID();
 - Háttérben futó másik chart programot keres, aminek a PID-jét visszaadja, file alapú adattovábbításnál használt alprogram.
- void BMPcreator(int *Values, int NumValues);
 - Values tömb elemeiből létrehoz egy gráfot és BMP fileba menti a program könyvtárába.
- int Measurement(int **Values);

- Létrehoz egy legközelebbi kisebb negyed óra óta eltelt másodpercek, vagy legalább 100 elemű tömböt amiben az egymást követő számok max 1 értéket térnek el egymástól.
- void SendViaFile(int *Values, int NumValues);
 - NumValues hosszú Values tömböt menti egy Measurements.txt fileba user home könyvtárba “%d\n” szerkezetben, ez után a FindPID segítségével megkeresi a háttérben futó feltételezeten adatfeldolgozó módban futó programot és SIGUSR1 jelet küld neki, hogy dolgozza fel.
- void ReceiveViaFile(int sig);
 - SIGUSR1 jelre beolvassa a user home könyvtárában található Measurements.txt file-t és BMPcreator segítségével létrehoz egy gráfot belőle
- void SendViaSocket(int *Values, int NumValues);
 - localhost:3333-ra továbbítja elsőnek a NumValues értékét, ha ezt visszakapja válaszban a szervertől, akkor továbbítja a Values értékét, erre válaszban a Values tömb méretét kapja vissza.
- void StopSocket(int sig);
 - Leállítja a szerver socketet.
- void ReceiveViaSocket();
 - Létrehoz egy listenert a 3333-as portján és ha beérkező adat van, elsőnek visszaküldi, majd a következő beérkező adatot tárolja egy az első adatnak megfelelő hosszúságú tömbben, visszaküldi a tömb méretét majd a BMPcreator függvény segítségével létrehoz egy gráfot a beérkező adatokból.

Rendszerkövetelmények

Nem tudtam megállapítani különösebb rendszerkövetelményeket, a szoftver Linux alatt fog működni megfelelően.

Én Debian GNU/Linux 12 (bookworm) alatt fejlesztettem.

Alkotó

Molnár Attila

2024.05.01

10. feladat: Én csináltam bibibi :-D