Bóna Noel

Dokumentáció

1. A használandó fordítóprogram és annak szükséges kapcsolói
2. Rendszerkövetelmények
3. Felhasználói útmutatás a programhoz
4. A program által visszaadott értékek magyarázatát
5. Alprogramok rövid leírása

1. A használandó fordítóprogram és annak szükséges kapcsolói

A programot a GCC azaz GNU Compiler Collection (verzió: 11.3.0) fordítóprogrammal tudjuk lefordítani a következő szükséges kapcsolókkal:

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

2. Rendszerkövetelmények

Operációs rendszer: Linux (ajánlott a legújabb Ubuntu LTS verzió)

Tárhely: Legalább 5~10 MB szabad terület

3. Felhasználói útmutatást a programhoz

A programot a következő paraméterekkel lehet használni:

--version: kiírja a program verzióját

--help: kiírja a program használati utasítását

A program üzemmódjai:

-send: küldő üzemmód (alapértelmezett)

-receive: fogadó üzemmód

-file: fájlt használ a kommunikáció megvalósítására (alapértelmezett)

-socket: socketet használ a kommuniukáció megvalósítására

A programot a saját mappájában futtassuk fordítás után. Például így:

```
./chart --version
./chart -receive -socket
./chart -send -socket
```

Bóna Noel

4. A program által visszaadott értékek

0 – A program megfelelően leállt.

1 – A futtatott állomány neve nem megfelelő.

2 – Sikertelen memória allokálás / foglalás.

3 – Sikertelen fájl megnyitás / írás / olvasás.

4 – Nincs fájl fogadó üzemmódú folyamat.

5 – Nem sikerült a socket létrehozás.

6 – Nem sikerült az adat küldése.

7 – Nem sikerült az adat fogadása.

8 – Checksum érték nem megfelelő.

9 – Nem sikerült kötni a socketet.

10 – A fájlon keresztüli küldés szolgáltatás nem elérhető.

11 – A szerver nem válaszol.

5. Alprogramok

void chart_check(char *str)

Az eljárás paraméterként megkapja a futtatott állomány nevét. A program leáll egy hibaüzenettel és egy hibakóddal, ha ez a név nem egyezik meg *'chart'*-al. Ellenkező esetben nem ad vissza semmit.

void version_argument()

Ez az eljárás végzi el a verziószám, a szerző és a program elkészültének kiíratását, majd leáll a program.

double randfrom(double min, double max)

A függvény vár egy minimum és egy maximum értéket, majd visszatér egy random generált számmal amely ez a két érték között szerepel.

int Measurement(int **Values)

A függvény pszeudo cím szerinti paraméterátadás ként kap egy mutatót amely annak a területnek a helyére mutat amibe egy 3 állapotú 1 dimenziós bolyongás által generált értékeket tárolunk. A függvény majd visszatér ezeknek az értékek számával.

void BMPcreator(int *Values, int NumValues)

Az eljárás paraméterként megkapja a *Measurementben* létrehozott terület címét és a benne szereplő értékek számát. Majd ezeket az értékeket egy 1 bit színmélységű BMP fájlban ábrázoljuk egy grafikonnal. Ez az eljárás írja meg binárisan ezt a képfájlt. A metódus nem tér vissza semmilyen értékkel.

int FindPID()

A függvény nem vár semmilyen paramétert. Megkeresi a *'chart'* nevű folyamat PID-jét (Process ID) amely nem egyezik meg a saját PID-jével majd ezt azt értéket visszaadja. Viszont ha nincs másik ilyen folyamat akkor -1-et ad vissza.

void SendViaFile(int *Values, int NumValues)

Az eljárás paraméterként megkapja a *Measurementben* létrehozott terület címét és a benne szereplő értékek számát. Majd a felhasználó alapértelmezett könyvtárába létrehoz egy Measurements.txt fájlt amit feltölt az értékekkel. Ezek után küld egy szignált és nem tér vissza semmivel.

void ReceiveViaFile(int sig)

Az eljárás paraméterként kap egy egész számot, viszont ezzel nem fogunk semmit se csinálni. A metódus a Measurements.txt fájlt beolvassa egy dinamikusan lefoglalt memória területre, majd erre meghívja a *BMPcreator* eljárást és nem tér vissza semmivel.

void SendViaSocket(int *Values, int NumValues)

Az eljárás paraméterként megkapja a Measurementben létrehozott terület címét és a benne szereplő értékek számát. Az eljárásban egy UDP kliens fut ami először az értékek számát majd maga az értékeket továbbítja a szervernek. Mindeközbe ellenörzéseket végez. Miután sikeresen elküldött mindent akkor leállítja a klienst (socketet) és nem tér vissza semmivel.

void ReceiveViaSocket()

Az eljárás paraméterként nem vár semmit. Az eljárásban megvalósítódik egy folyamatosan futó UDP szerver ami adatokat vár. Először mindig csak egy számot ami a várt értékek száma, majd ezután megkapja maga az értékeket. Ezekre meghívja a *BMPcreator eljárást majd* újra várakozik további beérkező adatokra.

void SignalHandler(int sig)

Az eljárás paraméterként kap egy egész számot (szignált) amelyre különböző kimeneteket produkál. SIGINT szignálra a program leáll helyesen, SIGUSR1-re és SIGALRM-ra hibát ír és kilép egy hibakóddal.