

# Dokumentáció a Rendszerkövetelmények kurzus önálló projektéhez

## 1. Használat:

A fordításhoz és futtatáshoz szükséges utasítások és kapcsolók:

```
>> gcc main.c myUtils.c -o chart -lm -fopenmp
>> ./chart --help| --version | [-file / -socket] | [-receive / -send]
```

## 2. Rendszerkövetelmény:

```
>> Linux alapú operációs rendszer (Ubuntu 22.04)
>> gcc program meglete (11.4.0)
```

## 3. Felhasználói útmutatás:

Az (1.) pontban található első commanddal egy futtatható chart elnevezésű állomány lesz létrehozva a használt mappában. Ezt a Linux alapú operációs rendszereken a ./[futtatható\_állomány\_neve] paranccsal tudunk futtatni. A ./chart --help() kapcsoló a felhasználónak tud segítséget nyújtani abban, hogy a kapcsolókat használni tudja. A program kipróbálásához két terminál megnyitása is szükséges: mindkettőben a kódokat tartalmazó mappában lévő ./chart --[kapcsolók] kerül futtatásra. Az egyikben a -send [-file/-socket], míg a másikban a -receive [-file/-socket] kapcsolókat szükséges használni.

## 4. A program által visszaadott értékek magyarázata:

```
>> 0 : A program hiba nélkül lefutott.
>> 1 : Ctrl+C-t a felhasználó által megnyomásra került.
>> 2 : Timeout hiba - a szerver nem válaszol.
>> 3 : Fájl által való küldési opció nem végrehajtható.
>> 4 : A bmp fájl nem nyitható meg.
>> 5 : Nem található fogadó üzemmódban működő process.
>> 6 : Socket kreálási error.
>> 7 : Socketen keresztüli küldési hiba.
>> 8 : Socketen keresztüli vételi hiba.
>> 9 : A socketen keresztüli küldésnél a küldött és kapott adat nem egyenlő.
>> 10 : Socketen keresztüli küldési hiba.
>> 11 : Socketen keresztüli vételi hiba.
>> 12: A socketen keresztüli küldésnél a küldött és kapott adat nem egyenlő.
>> 13 : Socket kreálási error.
>> 14 : Kötési (binding) hiba - socket.
>> 15 : Socketen keresztüli vételi hiba.
>> 16 : Socketen keresztüli küldési hiba.
>> 17 : Socketen keresztüli vételi hiba.
>> 18: Socketen keresztüli küldési hiba.
```

## 5. Az alprogramok rövid leírása:

```
>> void SignalHandler(int sig);
    Az eljárás a beérkező signalok elkapására és kezelésére szolgál.

>> void help();
    Az eljárás a ./chart --[kapcsolók] command elírásánál vagy a --help
    kapcsoló használata esetén nyújt segítséget.
```

>> int Measurement(int \*\*Values);  
A függvény egy random bolyongással adatokat generál, melyeket a későbbiekben felhasználunk. Ezeket az adatokat valós bejövő adatoknak tekintjük.

>> void BMPcreator(int \*Values, int NumValues);  
Ez az eljárás a kapott adatokból generál egy bmp fájlt.

>>int FindPID();  
Ez a függvény felel azért, hogy megkeresse a másik futó ./chart -- [kapcsolók] program pidjét (Process ID-ját), amennyiben nem talál ilyen - 1-gyel tér vissza, ellenben a másik program pidjével.

>>void SendViaFile(int \*Values, int NumValues);  
Ez az eljárás létrehoz a felhasználó HOME mappájában egy Measurement.txt elnevezésű fájlt, melyet feltölt a kapott adatokkal.

>> void ReceiveViaFile(int sig);  
Ez az eljárás beolvassa az adatokat a SendViaFile() által létrehozott Measurement.txt fájlból. Ezekre az adatokra kerül meghívásra a BMPcreator() eljárás.

>> void SendViaSocket(int \*Values, int NumValues);  
Ez az eljárás egy egyszerű UDP kliens-szerver kapcsolat kliens oldalát valósítja meg. Adatokat küld át a fogadónak, közben pedig vizsgálja, hogy sikeresen végbe mentek-e ezek a folyamatok.

>> void RecieveViaSocket();  
Ez az eljárás egy egyszerű UDP kliens-szerver kapcsolat szerver oldalát valósítja meg. A kliens módban működő SendViaSocket() eljárástól adatokat kap, melyre a BMPcreator() eljárást hívja meg.