Rendszerközeli programozás

Projekt feladat

Dobos Martin József – GH6ZAS

A program egy 1bit szín mélységű BMP fájlt hot létre, amely egy virtuális szenzor által rendszeres időközönként szimulált értékeket grafikonon ábrázol. A program két különböző üzemmódra képes, melyek küldő és fogadó, ezek folyamatok egymás közötti kommunikációval valósítják meg az adatcserét.

A használandó fordítóprogram:

- gcc (Ubuntu 11.2.0-7ubuntu2) 11.2.0 verzió

A program futtatása:

- gcc main.c myfunc.h -o chart -lm -fopenmp

Rendszer követelmények:

- Linux Ubuntu operációs rendszer

Felhasználói útmutató:

A program különböző üzemmódjait indításkor parancssori argumentummal lehet beállítani, melyek a következők:”--version”,”--help”,”-send”,”-receive”,”-file”,”-socket”.

Először válasszuk ki a kommunikáció módját, mely lehet fájl vagy sokcet.
Ezután indítsuk el a program fogadó folyamatát a kiválasztott kommunikációs móddal
Például:”./chart -socket -receive”.
Fogadó program a „Ctrl+c” billentyűkombinációval terminálható.
Következő lépésben egy másik terminálban indítsuk el a küldő folyamatot.
Például:”./chart -socket -send”.

A program a küldés befejeztével automatikusan leáll.

A program indítható argumentum megadása nélkül:

„./chart” a program alapértelmezés szerint fájlon keresztüli küldő üzemmódban indul.

A program indítható 1 argumentum megadásával:

„./chart -send” vagy „./chart -receive” ekkor a program alapértelmezetten fájlon keresztüli kommunikációval indul.

„--version” kapcsolóval indítva a program verzióval és fejlesztéssel kapcsolatos információkat jelenít meg.

„--help” kapcsoló használata esetén a program tájékoztatja a felhasználót a futtatás lehetséges opcióiról és az argumentumok jelentéséről.

A program indítható 2 argumentum megadásával:

Ebben az esetben az argumentumok sorrendje tetszőleges az alábbiak közül, de egy csak egyszer szerepelhet.

„-send”,”-receive”,”-file”,”-socket”

A program által vissza adott értékek:

- exit(-1): Nem létezik fogadó folyamat.
- exit(0): A program hiba nélkül be fejezte a futást
- exit(1): A memória foglalás sikertelen.
- exit(2): Socket készítési hiba.
- exit(3): Hálózaton való küldés közben hiba lépett fel.
- exit(4): Hálózaton küldött és fogadott adatok méretei különböznek.
- exit(5): Üzenet fogadási hiba.
- exit(6): A szerver nem válaszolt időben.
- exit(7): A program neve nem megfelelő.
- exit(8): Nem megfelelő parancssori argumentum.

Alprogramok:

- int Measurement(int **Values)

A függvény egy 3 állapotú 1 dimenziós „bolyongást” implementál, azaz random állítja elő egész számok egy véges sorozatát. A két szomszédos elem különbségének abszolút értéke 1.
A kezdő érték 0. A mért értékek megegyeznek az adott negyedórából eltelt másodpercek számának és a 100-nak a maximumával.
A függvény az éppen aktuális adatot a **Values mutatón keresztül adja át majd a megadott tömbnek és a tömb méretét adja vissza int-ként.

- void BMPcreator(int *Values, int NumValues)

A BMPcreator függvény létrehoz és ír ki egy BMP képfájlt a megadott értékek alapján. A függvénynek két paramétere van: Values, ami a pixel értékek tömbjét tartalmazza, és NumValues, ami a kép szélességét és magasságát határozza meg
A függvény először megnyitja a "mybmp.BMP" fájlt írásra, majd generálja a pixel adatokat a tomb tömbben. Ezután kiírja a BMP fájl fejlécét és metaadatait, majd a pixel adatokat a fájlba. Végül felszabadítja a memóriát.

- int FindPID()

Megvizsgálja a Linux fájlrendszer gyökerében lévő „/proc” könyvtárnak az alkönyvtáraiban található „status” nevű fájlok tartalmát. A fájl első sorának a formátuma: "Name:\tchart\n". Ha a tabulátor és az új sor karakter között a „chart” karaktersorozat megtalálható, akkor keres egy olyan sort, amely „Pid:\t” sztringgel kezdődik és egy egész szám követi.
A függvény ezzel az egész számmal tér vissza, ha pedig nem talál ilyen fájlt, akkor -1-et ad vissza

- void ReceiveViaFile(int sig)

Ez az eljárás megnyit a felhasználó alapértelmezett könyvtárában egy „Measurement.txt” nevű szöveges állományt, tartalmát beolvassa és eltárolja egy memóriaterületen, majd a BMPcreator eljárást meghívja és átadja neki az értékeket és azok darabszámát.

- void SendViaFile(int *Values, int NumValues)

A *Values mutató egy egészeket tartalmazó tömb kezdőcímét kapja meg. A NumValues változó fogja tárolni a tömbben lévő egészek darabszámát. Az eljárás létrehoz egy „Measurement.txt” nevű szöveges fájlt a felhasználó alapértelmezett könyvtárában, majd soronként beleírja a tömbben lévő értékeket. Miután bezárta meghívja az FindPID nevű függvényt. Ha nem talál fogadó üzemmódban működő folyamatot, leáll -1-es hibakóddal.

- void SendViaSocket(int *Values, int NumValues)

Az eljárás UDP protokoll segítségével a localhost, 3333-as portját figyelő fogadó folyamattal kommunikál. Első paramétere a szimulált adatokat tartalmazó tömb kezdőcíme, második a tömb mérete.

- void ReceiveViaSocket()

Az eljárás egy végtelen ciklusban UDP szegmenseket vár a 3333-as porton. Dinamikusan lefoglal memóriaterületet a küldő eljárás által átadott tömbméretnek megfelelően. A tömb létrehozása után meghívja a BMPcreatort.

- void SignalHandler(int sig)

Ez egy szignálkezelő eljárás. Három féle szignált képes kezelni:
SIGINT: A program elköszön és leáll, visszaad egy 0 értéket.
SIGUSR1: A fájlon keresztüli küldés szolgáltatás nem elérhető.
SIGALRM: A szerver nem válaszol időkereten belül.

- void Help()

Az eljárás a képernyőre írja a felhasználót segítő információkat. Például a --help kapcsoló vagy rossz argumentum esetén.