

Rendszerközeli programozás dokumentáció

Tarr Imre – IGXWVJ

Projekt feladat

C nyelven megírni egy programot, amely egy 1 bit színmélységű bmp fájlt hoz létre.

A bmp fájl egy véletlenszerűen változó mennyiség időbeli változását szemléltető grafikont ábrázol.

A program kétféle üzemmódban képes működni:

- Küldő: előállítja az adatokat

- Fogadó: ábrázolja az adatokat

Egyetlen program, de két folyamat. Folyamatok közötti kommunikáció megvalósítása, vagy fájlkezeléssel vagy socket (hálózaton) keresztül.

Fordítóprogram és kapcsolói

Fordítóprogram: GCC Compiler 11.2.0

Kapcsoló: -o (gcc controller.c main.c -o chart)

Rendszerkövetelmények

Linux operációs rendszer, Ubuntu

GNU Compiler 4.3

Felhasználói útmutató

A program egy .bmp kiterjesztésű képet fog generálni.

A programot chart néven kell elindítani, másképp nem fog működni.

```
gcc controller.c main.c -o chart
```

Miután helyes nevet adtunk meg, a felhasználó az alábbi parancssori argumentumokat használhatja:

`--version`

`--help`

`-send`

`-receive`

`-file`

`-socket`

A `--version` paranccsal lekérdezheti a program verzióját, készültjét és fejlesztőjét. A `--help` paranccsal további segítségeket kérhet.

Válaszhat, mely kommunikációs módot szeretné használni, *file* vagy *socket*.

Az elérhető üzemmódokról, *send* vagy *receive*, azaz küldő vagy fogadó. Az alapértelmezett indítási paraméterek `-send` és `-file`

Két terminálra lesz szükség, az egyik a küldő, a másik a fogadó.

Először elindítjuk a fogadót, vagy fájlkezeléssel vagy hálózaton keresztül.

	File	Socket
Fogadó	<code>./chart -file -receive</code>	<code>./chart -socket -receive</code>
Küldő	<code>./chart -file -send</code>	<code>./chart -socket -send</code>

Visszaadott értékek magyarázata

<code>exit(-1)</code>	Nem találtam fogadó üzemmódban működő folyamatot!
<code>exit(0)</code>	A program leállt!
<code>exit(1)</code>	A fájlon keresztüli küldés szolgáltatás nem elérhető!
<code>exit(2)</code>	Socket creation error.
<code>exit(3)</code>	Binding error.
<code>exit(4)</code>	Server timeout!
<code>exit(5)</code>	Sending error.
<code>exit(7)</code>	Error size!

Alprogramok rövid leírása

```
int commands(int* send_mode, int* file_mode, int argc, char *argv[]){  
/*A commands függvényen belül, lehet beállítani program üzemmódját a "-send" vagy  
a "-receive" parancssori argumentummal, adhatjuk meg. Az előbbi az alapértelmezett, azaz  
küldő.  
Ha send_mode == 0, akkor send,  
Ha send_mode == 1, akkor receive  
Ha file_mode == 0, akkor file,  
Ha file_mode == 1, akkor socket*/  
}
```

```
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 x0=0  
    (*Values + 0) = 0;  
//42,8571% eséllyel xi+1=xi+1  
    (*Values + i) = (*Values + (i - 1)) + 1;  
//11/31-ed részben xi+1=xi-1  
    (*Values + i) = (*Values + (i - 1)) - 1;  
//Néha lehetnek azonosak xi+1=xi  
    (*Values + i) = (*Values + (i - 1));  
/*A mért értékek megegyeznek az adott negyedórából eltelt másodpercek számának és a  
100 maximumával.*/  
return meres_szam;  
}
```

```
void BMPcreator(int *Values, int NumValues){  
/*Az adatsort egy BMP képfájl segítségével jelenítjük meg.  
Az első paraméter az ábrázolandó értékeket tartalmazó tömb kezdőcíme, a második pedig  
az előbbi tömbben eltárolt egész értékek száma.*/  
}
```

```
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 „bash” karaktersorozat megtalálható, akkor keres egy olyan sort, amely „Pid:\t” sztringgel  
kezdődik és egy egész szám követi, a függvény visszatér ezzel az egész számmal, ha pedig  
nem talál ilyen fájlt, akkor kilép -1-es hibakóddal */  
    return atoi(tomb);  
    return -1;  
}
```

```
void ReceiveViaFile(int sig){
```

```
/*Egy eljárás, megnyit a felhasználó alapértelmezett könyvtárában egy „Measurement.txt”  
nevű szöveges állományt, tartalmát beolvassa és eltárolja egy memóriaterületen. Majd a  
fentebb megírt BMPcreator eljárást meghívjuk és átadjuk 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){
```

```
/*Egy eljárás amely létrehoz egy localhost-ot.
```

```
Az első paraméterként kapott memóriacím egy tömb kezdőcíme.
```

```
A második pedig a tömbben lévő egész típusú értékek száma.
```

```
Az eljárás a socketen keresztül küldi el a fogadónak a NumValues értékét.
```

```
Ha a küldött és kapott értékek eltérőek, akkor a program leáll.
```

```
Ha az értékek megegyeznek, akkor az eljárás a Values címen kezdődő tömb NumValues  
darab int típusú értékét küldi át. Egyetlen üzenetben a fogadónak. */
```

```
}
```

```
void ReceiveViaSocket(){
```

```
/*Az eljárás visszaküldi a kapott értéket a küldő üzemmódú kliens folyamatnak.
```

```
Dinamikusan lefoglalásra kerül ennyi darab egész szám számára memóriaterület.
```

```
A kapott adatokkal meghívásra kerül a BMPcreator eljárás, végül a lefoglalt memóriacímek  
felszabadulnak és várják a következő üzenetet a küldőtől.*/
```

```
}
```

```
void SignalHandler(int sig){
```

```
/*Egy szignálkezelő eljárás.
```

```
Három féle szignált képes kezelni:*/
```

```
    SIGINT /*A program elköszön és leáll, visszaad egy 0 értéket.*/
```

```
    exit(0);
```

```
    SIGUSR1 /*A fájlön keresztüli küldés szolgáltatás nem elérhető.*/
```

```
    exit(1);
```

```
    SIGALRM /*A szerver nem válaszol.*/
```

```
    exit(4);
```

```
}
```