

Praznovszky Péter (OUNQE0) Rendszerközeli Programozás Projekt Dokumentáció

Projektem célja egy véletlenszerű értékekből álló („mérési értékek”) adatsor ábrázolása egy függvény képében (.bmp formátumban). Emellett a folyamatok közötti file-on, és socket-en keresztüli kommunikáción volt a hangsúly.

Rendszerkövetelmény:

- Fordító: gcc (11.4.0-es verzió)
- OS: Ubuntu Linux (22.04.4 LTS vagy annál újabb)
- 2 magos processzor
- 4 GB memória

Alkalmazható kapcsolók a program különböző beállításaihoz:

- --version: Alap információkat szolgáltat a felhasználónak az adott verziószámmal, fejlesztővel, dátummal.
- --help: Segít a felhasználónak a program helyes futtatásához, a számára megfelelő kapcsolók megvizsgálásához.
- -send: Egy küldő folyamat indításához szükséges, amely majd a fogadó üzemmódban lévő folyamattal fog kommunikálni.
- -receive: Egy fogadó üzemmódu folyamat indul el, amely az adott beállítás szerint fogja várni a küldőtől az adatsort.
- -file: File-on keresztüli kommunikációt valósít meg.
- -socket: Az adatsort megadott IP cím és Portszaám szerint küldi , vagy fogadja

Fontos, hogy ezen kapcsolók sorrendje tetszés szerint felcserélhető. Alap esetben amennyiben nem adunk meg semmilyen kapcsolót, úgy küldő folyamat indul el file-on keresztüli kommunikációt megvalósítva.

A megfelelő fordítás és futtatás lépései:

1. gcc main.c seged_dolgok.h -o chart
2. ./chart [1.kapcsoló] [2.kapcsoló (amennyiben szükséges)]

Szintén fontos információ, hogy a program csak „chart” néven fut le, ellenkezőleg hibaüzenet értesíti a felhasználót a problémáról! (Ezt Linuxon a -o kapcsolóval tudjuk beállítani.)

Az elkészített alprogramok (seged_dolgok.h-ban találhatóak) rövid magyarázata (elkészítésük sorrendjében):

- **int Measurement(int **Values)** függvény: A vizualizálandó véletlen értékek generálására szolgál , bizonyos feltételek mellett. A függvény lekérdezi a rendszeridőt amely az adott negyedórától eltelt másodpercek és 100 maximuma szerinti darabszámmal állít elő értékeket, amit egy dinamikus memória területen tárol. A számok darabszámmal tér vissza, a tömb-öt pseudocím szerinti paraméteradás segítségével adja át a hívó programegységnek.
- **void BMPcreator(int * Values, int NumValues):** A kapott értékek és azok darabszámának segítségével előállítja a .bmp állományt , megfelelő jogosultságokkal.
- **int FindPID():** File-on keresztüli küldés esetén lényeges. A már éppen futó fogadó üzemmódu „chart” nevű folyamat azonosítóját keresi. Ez az azonosító a küldő folyamat számára lesz elengedhetetlen.

- **void SendViaFile(int* Values, int NumValues) és void RecevieViaFile(int sig):** A -file kapcsoló esetén lépnek érvénybe. Először fogadó folyamatnak kell futnia, ami várni fog egy felhasználó 1-es signált. A küldő folyamat a FindPID függvény által adott értéknek (ami a fogadó üzemmód azonosítója) küldi a signált. Emellett a küldő folyamat a kapott értékeket egy .txt állományban letárolja ahonnan a fogadó pedig kiolvassa. A fogadó szignál érkezése esetén azonnal lefut, és előáll a vizualizált adatsor.
- **void SendViaSocket(int* Values , int NumValues) és void ReceiveViaSocket():** A -socket kapcsoló esetén lépnek érvénybe. Megadott IP cím és Portsám szerint a kliens kommunikál a szerverrel, minden fogadott érték után nyugtázva. A megkapott értékekkel és azok darabszámával előáll a vizualizált adatsor.
- **void SignalHandler(int sig):** Az éppen kapott signál-t kezeli , és értesíti a felhasználót.

Hibakódok magyarázata:

- 11: A küldő folyamat nem talált fogadó üzemmódú folyamatot
- 5: Az elküldött és kapott értékek nem egyeznek meg (socket)
- 6: A kapott és elküldött értékek bájtban megadott mérete nem egyezik meg (socket)
- 29: A szerver nem válaszol megadott időn belül (1 mp) (socket)
- 0: SIGINT szignált kapotunk.