

"SO-TP2" - testul de programare



The name and photo associated with your Google Account will be recorded when you upload files and submit this form. Only the email address you enter is part of your response.

***Required**

Problema de programare

Să se scrie trei programe, în trei fișiere sursă (numite în felul următor: prg_P0.c, prg_P1.c și prg_P2.c) și care să colaboreze între ele conform specificațiilor următoare:

Programul P0 va crea, în funcția sa main, două procese fii, iar în fiii creați va starta, prin apeluri exec adecvate, câte unul dintre celelalte două programe, P1 (în primul fiu creat) și respectiv P2 (în al doilea fiu creat). De asemenea, tot în funcția main a lui P0, mai puteți adăuga eventualele inițializări strict necesare pentru comunicațiile descrise în continuare.

Funcționalitatea specifică a programului P0 constă în: P0 primește un parametru în linia de comandă, ce reprezintă calea către un fișier existent pe disc, numit "propozitii.txt". Acest fișier conține un număr variabil de linii de text, fiecare linie reprezentând o propoziție formată doar din litere mici, litere mari și semne de punctuație (deci fără cifre). P0 va citi fiecare linie din acest fișier "propozitii.txt" și le va transmite una câte una, printr-un canal anonim, către procesul P1.

Funcționalitatea specifică a programului P1 constă în: citește din acel canal anonim de la P0, rând pe rând, câte o propoziție (atenție la lungimea mesajului), transformă literele mari în litere mici și trimite rezultatul, urmat de numărul de caractere ce au fost modificate (transmis în format text), printr-un canal fifo, numit "fifo_P1toP2", către procesul P2.



Funcționalitatea specifică a programului P2 constă în: citește din canalul fifo, rând pe rând, fiecare rezultat (i.e. o propoziție + număr de caractere modificate) transmis lui de către P1 și calculează o sumă de control pentru rezultatul respectiv, după următorul algoritm: la numărul de caractere modificate primit, adună valoarea codului ASCII a fiecărui caracter din propoziția primită, iar dacă suma este diferită de suma calculată pentru propoziția precedentă, o va trimite, în format binar, către procesul P0 printr-un alt canal anonim (deci, nu tot prin același canal anonim folosit pentru comunicația de la P0 la P1).

Funcționalitatea specifică a programului P0, continuare: P0 citește sumele de control transmise lui de către P2 prin cel de-al doilea canal anonim și le afișează pe ecran.

Cerință de numire a executabilelor:

Cerință de uniformitate pentru TOȚI studenții (necesară pentru corectarea/testarea semi-automatizată) -- veți compila sursele prg_P1.c și prg_P2.c cu numele de executabile P1 și respectiv P2 ; deci TREBUIE să folosiți aceste nume pentru executabile în apelurile exec pe care le veți scrie!

Recomandare:

Mai întâi, desenați de mână pe o foaie de hârtie o schemă cu "arhitectura aplicației": ierarhia de procese, mijloacele de comunicație între procese și sensul de transmitere a informației prin aceste mijloace de comunicație (precum ați văzut în exemplele de diagrame realizate de mână, atașate la unele exerciții de laborator).

Astfel vă veți clarifica mai bine cerințele specificate în enunțul problemei și, de asemenea, vă va ajuta la partea de implementare!



Baremul de corectare (total 9p) - fiecare criteriu se acordă integral sau deloc:

Criteriul minimal de îndeplinit: fiecare dintre cele trei fișiere sursă trebuie să compileze fără erori (adică trebuie să producă un executabil, chiar și cu eventuale avertismente la compilare), pentru a fi luat în considerare spre corectare!

Altfel, se vor acorda din oficiu 0p (zero!) pentru fiecare dintre criteriile următoare în care se face referire la procesul corespunzător fișierului sursă care are erori de compilare!

Sursa prg_P0.c :

- 1) Implementarea corectă a creării ierarhiei de procese și a apelurilor exec precizate, în funcția main din prg_P0.c : 1p
- 2) Implementarea corectă a funcționalității cerute pentru P0 în (una sau două) funcții separate de main, în prg_P0.c : 1p
- 3) Implementarea corectă a comunicației de la P0 către P1, prin canal anonim : 0.5p
- 4) Implementarea corectă a comunicației de la P2 către P0, prin canal anonim : 0.5p

Sursa prg_P1.c :

- 5) Implementarea corectă a funcționalității cerute pentru P1, într-o sursă separată prg_P1.c : 1p
- 6) Implementarea corectă a comunicației de la P0 către P1, prin canal anonim : 0.5p
- 7) Implementarea corectă a comunicației de la P1 către P2, prin canal fifo : 0.5p

Sursa prg_P2.c :

- 8) Implementarea corectă a funcționalității cerute pentru P2, într-o sursă separată prg_P2.c : 1p
- 9) Implementarea corectă a comunicației de la P1 către P2, prin canal fifo : 0.5p
- 10) Implementarea corectă a comunicației de la P2 către P0, prin canal anonim : 0.5p

//Total MAXIM criteriile 1-10 = 7p

Se va calcula fracția procentuală de îndeplinire a criteriilor de mai sus, iar cele 2 criterii de mai jos vor fi notate prin ponderare cu acest procentaj: $F = \text{total puncte acordate pentru criteriile 1-10} / 7p$ (i.e., total MAXIM criteriile 1-10)

- 11) Tratarea cazurilor de eroare la TOATE apelurile de sistem din toate cele trei programe: $F * 1p$
- 12) Scrierea codului într-o formă indentată, plăcută ochiului (i.e., "readable code"): $F * 1p$ (de exemplu, evitați copy-paste în mcredit)

//Total MAXIM criteriile 11-12 = 2p

Submit *

La finalul testului, creați o arhivă (conform celor precizate anterior pe site) cu cele trei programe sursă pe care le-ați scris în timpul testului, denumiți arhiva "username_TP2.zip" (unde username este numele vostru de cont de pe serverul students) și încărcați-o în formular, folosind butonul "Add file" de mai jos.


 Add File

A copy of your responses will be emailed to the address that you provided.

Back

Submit

Clear form

 Never submit passwords through Google Forms.