Bezpečnosť informačných a komunikačných systémov

Katedra elektroniky a multimediálnych telekomunikácií

Domáca úloha č.14

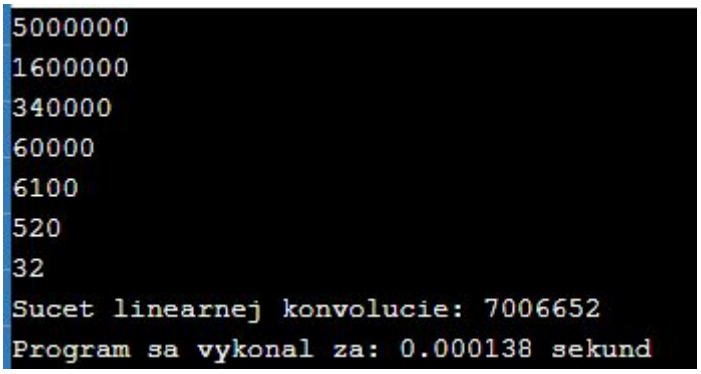
Martin Chlebovec **13. Dec. 2020**

Číslo študenta: 9
Pridelené postupnosti:
a = {0, 1, 1, 2, 1, 1, 2, 3}
b = {1, 2, 2, 3, 3, 2, 1, 2}

Vypracovanie DÚ č. 14:
Ručný výpočet

Pri realizácii zadania - DÚ č. 14 som najprv vytvoril vlastný program v jazyku C, ktorým som bol schopný vypočítať lineárnu a následne aj kruhovú konvolúciu postupností. Začal som s testovacou postupnosťou z dokumentu readme.txt, kde boli konvolúcie vysvetlené aj s ukážkou:
A = (1, 2, 3, 4), B = (5, 6, 7, 8).

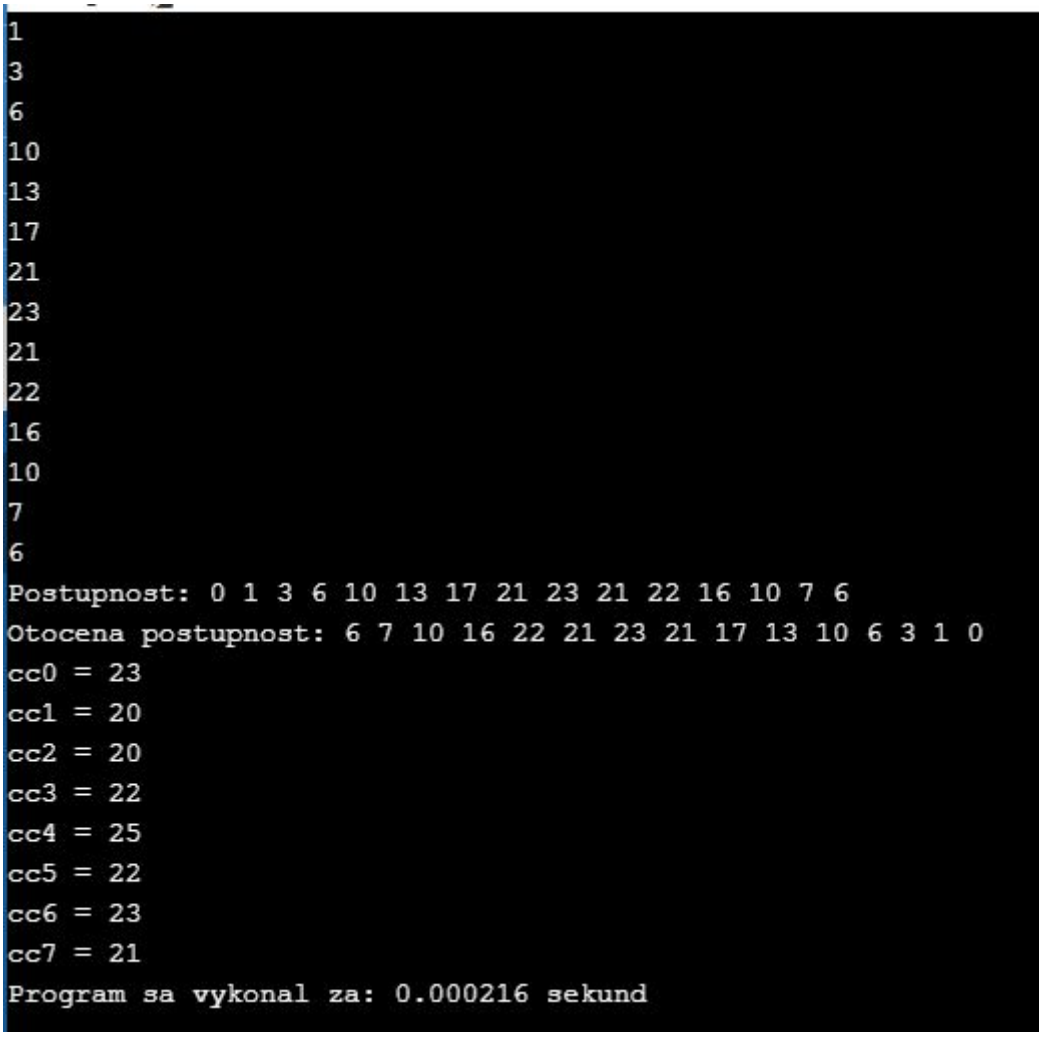
Lin. konvolucia tejto postupnosti



Pri návrhu .c programu pre lineárnu konvolúciu som sa inšpiroval zo stránky https://lloydrochester.com/post/c/convolution/, z ktorej som využil fragment funkcie convolve(). Nakoľko je však navrhnutá pre desatinné (float) čísla, musel som ju poupraviť. Program funguje v dvoch režimoch. **Na začiatku programu je definované makro SUCET**.

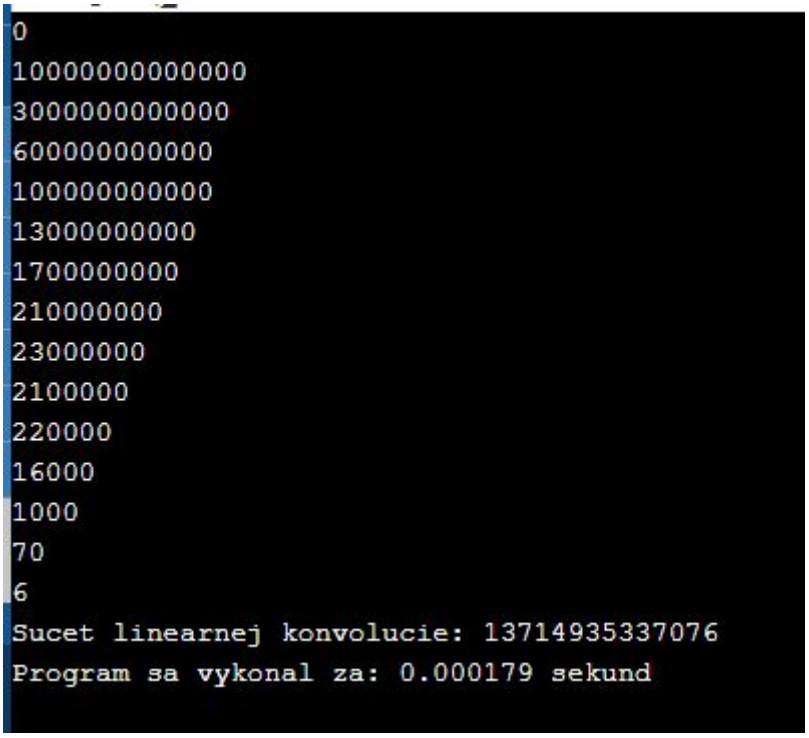
V prípade, že je zakomentované toto makro, nerealizuje sa súčet lineárnej konvolúcie a program pokračuje ďalej pre výpočet kruhovej konvolúcie. V prípade odkomentovaného makra program po vypočítaní súčtu lineárnej konvolúcie skončí a nepokračuje na kruhovú konvolúciu. Pri výpočte kruhovej konvolúcie je najprv postupnosť zarovnaná a výsledky kruhovej konvolúcie sú vypísané do konzole.

Zakomentovane makro SUCET - beh programu



Súčet lineárnej konvolúcie je navrhnutý pre akékoľvek dve postupnosti s rovnakou veľkosťou polí. Vypočítaním som získal hodnoty rovnaké ako pri ručnom výpočte kruhovej konvolúcie. Program som využíval aj pre čiastkové overenie riešenia, výpočet prvkov lineárnej konvolúcie, výsledok kruhovej konvolúcie atď…

Odkomentované makro SUCET - beh programu



Ručný výpočet

Ručný výpočet som realizoval metódou lineárnej konvolúcie s násobením každého prvku druhej postupnosti s každým prvkom prvej postupnosti. Následne som výslednú postupnosť "zarovnal", aby mala postupnosť 16 prvkov - 2^4 - násobok dvojky - párna postupnosť. Kruhovú konvolúciu som realizoval všeobecným vzorcom ---> CC_i= C_i + C_i+N (N = 8). Takto som dokázal vypočítať CC0 až CC7.

Pri výpočte prvkov množiny CC som využil modulo 337, v ktorom bolo aj ukážkové riešenie v readme z ktorého som čerpal. Čiastkové výsledky som si overoval v predmetnom .c programe, ktorý som navrhol. Dosiahol som rovnaké výsledky ako v .c programe, i keď s posunom niektorých prvkov:
CC = [23, 20, 20, 22, 25, 22, 23, 21].

Program NTT + NTT^{-1} v jazyku C
(omega_8 = 85, Modulo = 337)

Pri realizácii programu v jazyku C som v hlavičkovom súbore gf.h upravil definičné hodnoty makier pre omegu_8 = 85,. NTT_SIZE = 8, Modulo = 337, Inv_NTT_SIZE = 295, inv_omega_8 = 226. Najviac zmien som realizoval v programe test.c. Odstránil som všetky kontrolné výpisy, pomalé transformácie "DFT", rovnako tak i generátor postupnosti s dĺžkou NTT_SIZE, meranie cyklov a fragmenty zdrojového kódu, ktoré ich využívali. V programovej implementácii tak ostala iba funkcia pre výpočet priamej a inverznej NTT. Pri realizácii úprav som čerpal aj zo záznamu prednášky, kde bola daná problematika úprav vysvetlená. **Iné parametre pre omega a Modulo som neskúšal, nakoľko som si chcel byť istý s výsledkom riešenia.**

```
C:\Users\Administrator\Desktop\kod>test
c_0=23
c_1=22
c_2=25
c_3=22
c_4=20
c_5=20
c_6=23
c_7=21
Program sa vykonal za: 0.000000 sekund

C:\Users\Administrator\Desktop\kod>
```

Meranie času behu programu

Nakoľko je výstup oboch programov v jazyku .c totožný, experimentálne som overil dĺžku behu jednotlivých programov. Využil som hlavičkový súbor time.h a funkciu clock. Mnou navrhnutý program pre výpočet lineárnej a kruhovej konvolúcie pri meraní dosahoval hodnotu v rozmedzí 0.000200 až 0.000250 sekúnd. Návrh a testovanie programu som realizoval na OS Linux (Omega TUKE server / Online GDB).

Program pre výpočet NTT som spúšťal vo virtuálnom stroji (BIKS-BPS) - OS Windows 7. **Nepodarilo sa mi odmerať dĺžku behu programu pre NTT.** Program som experimentálne spustil aj v prostredí servera OMEGA a dostal som dĺžku behu programu 0.000084 až 0.000130 sekúnd. Program pre NTT je tak preukázateľne rýchlejší ako štandardný výpočet - násobením a sčítavaním prvkov a to až dvojnásobne!

```
Program sa vykonal za: 0.000118 sekund
mc364ve@omega:~/abcd/abc$ ./test
C_0 = 23
C_1 = 22
C_2 = 25
C_3 = 22
C_4 = 20
C_5 = 20
C_6 = 23
C_7 = 21
Program sa vykonal za: 0.000084 sekund
mc364ve@omega:~/abcd/abc$ ./test
C_0 = 23
C_1 = 22
C_2 = 25
C_3 = 22
C_4 = 20
C_5 = 20
C_6 = 23
C_7 = 21
Program sa vykonal za: 0.000128 sekund
```