



Spojení dvou setříděných polí

Semestrální práce z KIV/POT

Jméno a příjmení: Xuan Toan Dinh

Osobní číslo: A19B0027P

Email: dinhgos@students.zcu.cz

Contents

1. Zadání.....	3
2. Řešení.....	3
a. Analýza úlohy	3
b. Implementace	3
i. Datová část.....	3
ii. Převod ASCII na HEX.....	3
iii. Uložení hex hodnoty do pole	4
iv. Kontrola přetečení	4
v. Seřazení polí.....	4
vi. Převod HEX na ASCII.....	4
vii. Výpis.....	4
3. Uživatelská dokumentace	5
4. Závěr.....	5

1. Zadání

Spojení dvou setříděných polí. Nejprve se zadají dvě setříděné řady čísel uint16. Program řady spojí tak, aby čísla zůstala setříděná.

2. Řešení

a. Analýza úlohy

Nejprve je zapotřebí převést ASCII vstup do hexadecimální podoby. Poté ji můžeme uložit do pole. To samé je třeba udělat s druhým polem. Poté je možno jednotlivé hodnoty z pole porovnat a uložit si ji do třetího pole. Na konci stačí převést hodnoty z třetího pole na ASCII a vypsát je do konzole.

b. Implementace

i. Datová část

Název	Adresa	Význam
txt1	0x00FF4000	Výstupní text
buffer	0x00FF4010	Vstupní buffer
counter1	0x00FF4204	Čítač prvního pole
counter2	0x00FF4208	Čítač druhého pole
out_buf	0x00FF420C	Výstupní buffer
num_buf	0x00FF4E8C	Buffer pro převod HEX na ASCII
num_size_err	0x00FF4EA0	Chybová hláška
par1	0x00FF4EB0	Pointer na txt1
par2	0x00FF4EB4	Pointer na buffer
out	0x00FF4EB8	Pointer na out_buf
num	0x00FF4EBC	Pointer na num_buf
err1	0x00FF4EC0	Pointer na num_size_err
stack1	0x00FF51E4	První pole
stack2	0x00FF5504	Druhé pole
final_stack	0x00FF5B44	Seřazené pole
stck	0x00FF5BA8	Zásobník

ii. Převod ASCII na HEX

Podprogram převede ASCII hodnotu na hexadecimální hodnotu. Nejprve se v programu vynulují registry ER0 a ER1. Poté se do E1 uloží hodnota 10. Z ER6 se přečte první znak ze vstupního bufferu a otestuje se, jestli to je poslední číslo. Pokud ne, tak se od znaku odečte 30, pak se ER0 vynásobí 10 a nakonec se do ER0 přidá nová hodnota. Pokud je to poslední číslo, tak se zavolá podprogram pro uložení čísla do pole.

iii. Uložení hex hodnoty do pole

Podprogram uloží číslo do pole. Před uložením čísla do pole se nejprve zavolá podprogram pro kontrolu přetečení. Pokud nedošlo k přetečení, tak se do ER5 uloží pole a do ER2 čítač pro pole. Poté se musí inkrementovat ukazatel na vstup, aby po vrácení se z podprogramu mohlo načítat další číslo. Dále se v podprogramu posouvá ukazatel na pole, dokud nebude ukazovat na prázdný místo. Na nové místo se pak uloží nová hodnota, a nakonec se inkrementuje čítač. Pro načítání dalšího čísla se vynuluje R0.

iv. Kontrola přetečení

Podprogram kontroluje, jestli ve vstupu je číslo uint16. Pokud je číslo větší a k přetečení došlo, tak se vypíše chybová hláška a program se ukončí. Podprogram testuje, jestli E0 je prázdný. Pokud E0 není prázdný, tak k přetečení došlo a zavolá se systémové volání pro výpis chybové hlášky.

v. Seřazení polí

Podprogram postupně načítá hodnoty z obou polí a porovná je. Menší číslo uloží do finálního pole. V podprogramu se nejprve načtou všechny pole a jejich čítače. Poté se do ER5 a ER6 načtou čísla z polí pro porovnání. Před porovnáním čísel, se nejprve testuje, jestli se načetla validní čísla z polí, a ne prázdné pole. Pokud se načetlo prázdné číslo z pole, tak se hodnota nastaví na FFFF (65535). Poté se čísla porovnají a menší číslo se uloží a finálního pole. Tohle běží tak dlouho, dokud v obou polí nedojdou čísla.

vi. Převod HEX na ASCII

Podprogram převede číslo z hexadecimální podoby na ASCII hodnotu. Nejprve se uloží buffer pro převod čísla do ER6 a do ER4 se uloží hodnota 10. Poté se vydělí vstupní hodnota v ER3. Výsledek dělení se pak vloží do bufferu. Tahle akce se opakuje tak dlouho, dokud výsledek dělení není nula.

vii. Výpis

Podprogram vypíše výsledek do konzole. Nejprve se do ER3 načte hodnota z pole. Jelikož se pracuje s uint16, tak pro výpočet je zapotřebí jenom R3, tudíž se musí E3 vynulovat. Poté se zavolá podprogram pro převod hexadecimální hodnoty na ASCII hodnotu. ASCII hodnota čísla je uložena v num_buffer. Hodnota z bufferu se pak vypisuje do konzole. Poté co se vypíší všechny ASCII znaky z bufferu, tak se vypíše mezera a celý proces se zopakuje tak dlouho, dokud se nevypíší všechna čísla.

3. Uživatelská dokumentace

Ke spuštění programu je zapotřebí vývojový prostředí HEW. Po stažení programu stačí projekt otevřít a přeložit.

Při spuštění programu se do konzole vypíše text. Poté může uživatel zadat vstup do programu. Vstup musí být seřazený, v rozsahu $\langle 0, 65535 \rangle$, a oddělený “;” (např: 1;3;5;7;9). Pro druhé pole se vstup zadá stejně. Program pak automaticky seřadí zadané pole a vypíše výsledek do konzole.

4. Závěr

Program splňuje zadání. Vstup je ošetřen proti hodnotám větší než 65535 a proti duplikátům. Program by se jistě dal vylepšit, například se opakuje kód pro uložení hodnot do prvního a druhého pole. Smyslem úlohy ovšem nebylo nalézt nejefektivnější řešení, ale získat zkušenosti s programováním v jazyku symbolických adres. Tento cíl byl podle mého názoru splněn.