

KIV/POT – Počítačová technika

Samostatná práce

Referát

Jan Hinterholzinger (A19B0052P)

5.6.2020

LS 2019/2020

Rámcové zadání

Zadá se řetězec a potom povel y i nebo d: např. d7 vymaže znak na pozici 7, i14x vloží na pozici 14 znak x. Potom se vypíše upravený řetězec a očekává další povel.

Popis programu

Program při startu očekává vstup počátečního řetězce, který po zadání se přesune do určené části paměti, která je určena pro uložení a úpravu řetězce. Je vypočítána délka řetězce, která je uložena v registru R2 po celou dobu běhu programu.

Poté se program přesune na zadávání pokynů. Program akceptuje dva druhy pokynů. Vložení a odebrání. Program při vložení pokynu vypočítá počet znaků v příkazu a ukládá jej do registru R3, kde je uložena po celou dobu výkonu pokynu. Index zadávaný v pokynu je obdobný jako v poli v programovacích jazycích. První znak je na indexu 0, druhý znak na indexu 1, atd. Pro přidání znaku x, tak aby byl třetí (na pozici 2), je potřeba zadat příkaz **i2x**.

Celý program setrvává v cyklu zadávání pokynů, nemá tedy konec.

Vkládání znaku

Pokyn pro vkládání znaků má syntaxi iXY, kde X je celočíselný index pozice znaků, kam se má znak vložit, a kde Y je znak, který se bude na danou pozici vkládat. Pokyn tedy musí obsahovat minimálně 3 znaky. Tato podmínka je kontrolována a v případě porušení ukončí obsluhu pokynu s chybovou hláškou. Hodnota X je převedena z ASCII do číselné hodnoty. Obsluha pokynu kontroluje, zda zadáný index je v závislosti na délku řetězce smysluplný, případně pokyn ukončí s chybovou hláškou. Poslední znak pokynu se uloží do pomocného registru. Tento znak se poté uloží do paměti na vypočítanou adresu. Všechny následující znaky řetězce se posunou o byte dále a inkrementuje se informace o velikosti řetězce. Obsluha pokynu se ukončí.

Odebírání znaku

Pokyn pro odebírání jednotlivých znaků má syntaxi dX, kde X je celočíselný index pozice odebíraného znaku. Pokyn tedy musí obsahovat minimálně 2 znaky. Tato podmínka je kontrolována a v případě porušení ukončí obsluhu pokynu s chybovou hláškou. Hodnota X je převedena z ASCII do číselné hodnoty. Obsluha pokynu kontroluje, zda zadáný index je v závislosti na délku řetězce smysluplný, případně pokyn ukončí s chybovou hláškou. Znak se odstraňuje tak, že se znaky řetězce od posledního znaku přesouvají o jeden byte před. Přesun se zastaví na adrese odstraňovaného prvku. Také se provede dekrementace informace o velikosti řetězce a obsluha pokynu se ukončí.

Ukončení obsluhy pokynu s případnou chybovou hláškou

Jako ukončení obsluhy pokynu je bráno případné vypsání chybové hlášky, vypsání aktuálního stavu řetězce a skoku na zadávání nového pokynu.

Převod ASCII znaků na číselnou hodnotu.

Je použita standardní podprogram použitý na cvičení. Podprogram je modifikován, aby vstupní ASCII znaky četl z paměti od adresy vstupního bufferu (kde je očekáváno uložení pokynu) + 1 (znak pokynu i nebo d). Také je z délky pokynu vypočteno kolik znaků pro převod se v pokynu nachází. Výsledek převodu se uloží do registru R5.

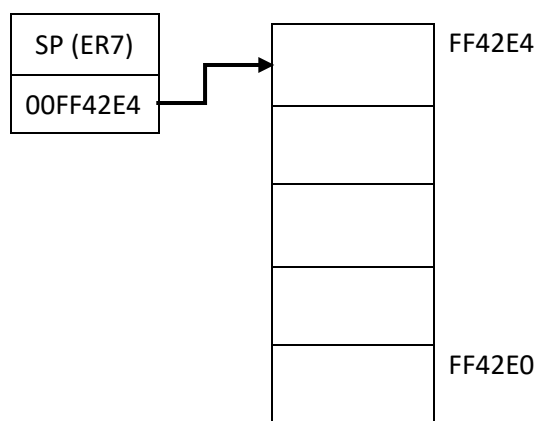
Tabulka definovaných proměnných a jejich umístění v paměti

| Definované proměnné | Umístění v paměti |
|---------------------|-------------------|
| txtStart | FF4000 |
| txtInput | FF401C |
| txtChyba | FF4030 |
| txtNewLine | FF405D |
| buffer | FF4063 |
| retezec | FF4162 |

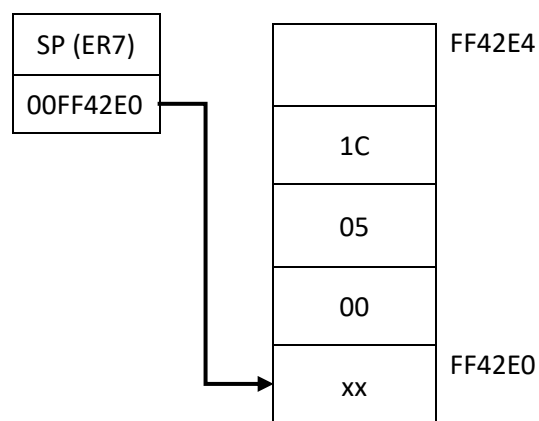
Obsah obsazené části zásobníku v situaci, kdy je výpočet v těle podprogramu se zvýrazněnou návratovou adresou z podprogramu.

Zvolil jsem situaci po zavolání podprogramu vypsání aktuálního stavu řetězce při ukončování povelu (000518, další příkaz: 00051C).

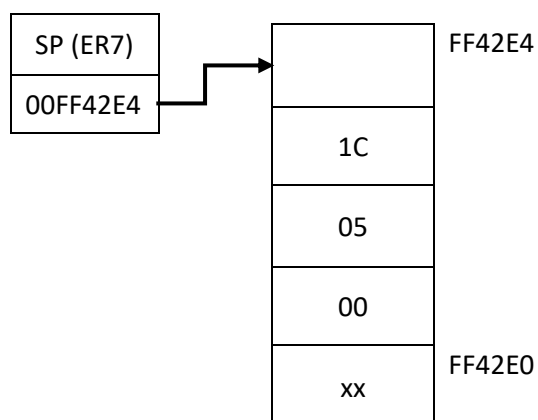
Před voláním JSR vypis (JSR 00049A)



Po volání JSR vypis (JSR 00049A)



Návrat z podprogramu



Aby zásobník neobsahoval pouze návratovou hodnotu, rozhodl jsem o přidání dalšího příkladu u podprogramu prevod, který převádí ASCII znaky na číselnou hodnotu. Součástí implementace je také vkládání a vybírání hodnot zásobníku, proto si myslím, že to vhodný příklad. Žlutou barvou je vyznačena návratová adresa (adresa na kterou program skočí po skončení podprogramu).

