Seznam úloh pro ÚPA – řešení na procesoru MIPS v simulátoru SPIM

Obtížnost 1: algoritmus řešení úlohy je velmi jednoduchý.

Obtížnost 2: řešení úlohy vyžaduje též prostudování příslušného algoritmu.

Další požadavky na řešení:

- Všechna vstupní data jsou zadávána z klávesnice, pokud není v zadání uvedeno jinak.
- Před každým vstupem se musí zobrazit „prompt“, ze kterého je zřejmé, co se má zadat.
- Všechny výstupy se zobrazují na displeji v přiměřeném formátu.
- Pro řešení úloh nejsou přípustné velmi nevhodné algoritmy.
- Případné nejasnosti zadání se konzultují se cvičícím.

- Výpočet klouzavého průměru (obtížnost 1)

  Postupně se budou zadávat čísla **int16**, na výstupu se bude průběžně zobrazovat průměr posledních 8 zadaných čísel.

- Mediánový filtr (obtížnost 1)

  Postupně se budou zadávat čísla **int16**. Na výstupu se bude průběžně zobrazovat hodnota, určená jako medián z 5 posledních zadaných hodnot.

- Převod textu podle převodní tabulky (obtížnost 1)

  V paměti bude pevně zadaná převodní tabulka (256 hodnot). Zadávaný text (ascii) bude průběžně kódován a zobrazován.

- Editace řetězce (obtížnost 1)

  Zadá se řetězec a potom povely **i** nebo **d**: např. d7 vymaže znak na pozici 7, i14***x*** vloží na pozici 14 znak ***x***. Potom se vypíše upravený řetězec a očekává další povel.

- Průběžné zatřiďování čísel (obtížnost 1)

  Budou se zadávat čísla **uint16** v libovolném pořadí (maximálně 50 čísel). Po zadání každého čísla se vypíšou dosud zadaná čísla setříděná podle velikosti.

- Test na prvočíslo (obtížnost 2)

  Zadá se číslo **uint16**, program otestuje, zda se jedná o prvočíslo.

- Inverzní čítač (obtížnost 1)

  Program bude postupně generovat hodnoty 00000000, 10000000, 01000000, 11000000, 00100000, 10100000, ... 11111111 . Počet bitů výstupu bude zadán po spuštění programu (např. pro 5 bitů bude výstup 00000, 10000, 01000, 11000, ... , 11111).

- Jednoduchá kalkulačka (obtížnost 1)

  Provádí operace +, -, ×, / s čísly **int16**. Zadá se např. 25+66=, 33-12=, 18×6=, 77/7= . Program zobrazí výsledek a čeká na další zadání.

- Převod zadaného čísla **uint16** do libovolné číselné soustavy (obtížnost 1)

  Napřed se zadá požadovaná soustava, potom číslo které se převede.

- Kruhový zásobník (obtížnost 2)

  Program bude simulovat kruhový zásobník s hodnotami **uint8**. Budou se zadávat povely **w** nebo **r**, např. w20 vloží 20 na aktuální pozici, **r** vybere a zobrazí hodnotu z aktuální pozice. Program bude ošetřovat a indikovat prázdný zásobník a přetečení.

- Simulace činnosti RTOS a algoritmu RMS (obtížnost 2)

  V programu budou zadány 4 úlohy uspořádané podle priority, popsané periodou T_i a časem výpočtu C_i. Program bude v jednotlivých taktech zobrazovat číslo běžící úlohy. Takty se budou zadávat klávesnicí. (Úloha není složitá. Návod k řešení a vhodné hodnoty T a C pro jednotlivé úlohy se domluví se cvičícím).

- Výpočet prvočísla Eratosthenovým sítem (obtížnost 2)

  Program vypočítá nejbližší prvočíslo menší než zadané N (**uint16**). Pro síto se použije bitová mapa, tj. v každém bytu bude informace o 8 číslech.

- Spojení dvou setříděných polí (obtížnost 1)

  Nejprve se zadají dvě setříděné řady čísel **uint16**. Program řady spojí tak, aby čísla zůstala setříděná.

- Práce se spojovým seznamem - vytváření spojového seznamu (obtížnost 2)

  Program bude vytvářet spojový seznam [hodnota, adresa_dalšího_prvku] seřazený podle velikosti. Hodnoty (**uint16**) budou postupně zadávány na vstupu, po každém zadání se projde a vypíše celý seznam (hodnota i adresa). Maximální počet záznamů je 50.

- Práce se spojovým seznamem – vyjímání (obtížnost 2)

  V paměti se (staticky) vytvoří spojový seznam [klíč (**uint8**), hodnota (**uint8**), adresa_dalšího_prvku] (alespoň 15 záznamů). Program vyjme ze seznamu prvek podle klíče zadaného na vstupu. Po každém zadání se projde a vypíše ve vhodném formátu celý seznam (klíč, hodnota i adresa).

- Stavový automat (obtížnost 2)

  V paměti bude uložena tabulka přechodů Mooreova automatu s 8 stavy a 5 výstupními hodnotami. Na vstupu se budou postupně zadávat vstupní hodnoty (4 různé hodnoty). Na výstupu bude zobrazován stav automatu a výstupní hodnota.

- Funkce pro dělení 32 bitů / 32 bitů metodou se zpětným přičítáním (obtížnost 2)

  Na vstupu bude zadán dělenec a dělitel (**uint32**). Program vypočítá a zobrazí podíl a zbytek (**uint32**) metodou se zpětným přičítáním.

- Vyhledávání podřetězce Krap-Rabinovým algoritmem (obtížnost 2)

  Na vstupu bude zadán znakový řetězec a hledaný podřetězec (délka maximálně 16 znaků). Program najde pozici podřetězce s využitím Karp-Rabinova algoritmu.

- Vkládání doplňkových nul do vysílaných dat (obtížnost 1)

  Na vstupu budou postupně zadávána vysílaná data v hexadecimálním formátu (**uint8**). Program bude průběžně vypisovat data převedená do bitové podoby a v případě výskytu 5 bitů s hodnotou 1 vloží automaticky bit s hodnotou 0. Např. data 0xA7, 0xE5 vyšle jako 1010011111<u>0</u>100101.

- Vyjímání vložených nul v přijímaných datech (obtížnost 2)

  Program zpracovává jednotlivé bity zadávané na vstupu. Zadaná data jsou propouštěna (nejsou zpracovávána) dokud není přijata posloupnost 01111110. Potom jsou další přijatá data formátována do 8bitových bytů a zobrazována na výstupu. Vždy po přijetí 5 bezprostředně následujících bitů s hodnotou 1 je následující 0 z dat vypuštěna. Např. 01010100111111001011010001111101110000 11 je dekódováno jako 0x5A, 0x3F, 0xC3. Pokud bude za sebou následovat více než 5 bitů s hodnotou 1 (kromě úvodní sekvence), bude indikována chyba.

- Histogram výskytu znaků v textu (obtížnost 1)

  Z klávesnice se zadá libovolný text. Program vypočítá histogram výskytu jednotlivých znaků. Výpis bude omezen jen na znaky malé a velké abecedy, ostatní znaky se budou počítat jako jediná „nealfanumerická“ kategorie.

- Násobení matic A a B velikosti **n** × **n** (obtížnost 1)

  Nejprve se zadá parametr **n**, potom postupně po řádkách obě matice (**int16**). Výstupem bude matice C = A × B. Paměť musí být dimenzována pro matice 6 × 6, matice musí být v paměti uložena v souvislém prostoru (tj. za posledním prvkem určitého řádku následuje v paměti bezprostředně první prvek následujícího řádku).

- Transpozice matice velikosti **n** × **n** (obtížnost 1)

  Nejprve se zadá parametr **n**, potom postupně po řádkách matice A (**int16**). Výstupem bude matice A^T. Transpozice musí proběhnout rozsahu v paměti, ve které je uložena matice A. Paměť musí být dimenzována pro matice 7 × 7 matice musí být v paměti uložena v souvislém prostoru (tj. za posledním prvkem určitého řádku následuje v paměti bezprostředně první prvek následujícího řádku).

- Jednoduchá kalkulačka se zadáváním RPN (obtížnost 2)

  Úkolem je naprogramovat kalkulačku, která provádí základní operace +, - , ×, /. Zadávání dat (**int16**) a operací se bude provádět způsobem RPN (Reverse Polish Notation). Délka zásobníku pro operandy a operace je maximálně 5 položek.

- Regrese metodou nejmenších čtverců (obtížnost 2)

  Postupně se budou zadávat dvojice dat ***x*** a ***y*** (**int16**). Při zadání jiného zvoleného znaku vypočte program regresi zadaných dat metodou nejmenších čtverců ($y = a \cdot x + b$) a zobrazí hodnoty ***a*** a ***b***.

- Výpočet faktoriálu (obtížnost 2)

  Po zadání ***n*** vypočítá program ***n!*** rekurentní metodou (tj. $n! = n \cdot (n-1)!$). Důležitou částí programu bude funkce fakt(n), která bude volat sama sebe s parametrem n-1, tj. fakt(n-1). Funkce tedy musí být napsána jako reentrantní.

- Výpočet nejmenšího společného násobku (obtížnost 1)

  Program vypočte ke dvěma zadaným číslům (**uint8**) jejich nejmenší společný násobek (**uint16**). Prvočísla potřebná pro výpočet mohou být v programu pevně zadána.

- Výpočet největšího společného dělitele (obtížnost 1)

  Program vypočte ke dvěma zadaným číslům (**uint16**) jejich největší společný dělitel (**uint16**) Eukleidovým algoritmem.

- Výpočet řady rekurentní metodou (obtížnost 2)

  Po zadání $\boldsymbol{m}$ a $\boldsymbol{k}$ (**int16**) vypočte program $\boldsymbol{m}$-tý člen řady $a_n = a_{n-1} \cdot 2 + k$, $a_1 = 0$, rekurentní metodou. Důležitou částí programu bude funkce clen(m), která bude volat sama sebe s parametrem m-1, tj. clen(m-1). Funkce tedy musí být napsána jako reentrantní.

- Převod čísla int32 na float

  Program převede zadané číslo (**int32**) do zobrazení v pohyblivé řádové čárce a výsledek zobrazí v hexadecimálním zobrazení.