

Architektury systemów komputerowych

Lista 6

$x_6 = 6$ (minimum na bdb)

Lista 6 przeznaczona jest na ćwiczenia w tygodniu 13-17 kwietnia (o ile prowadzący ćwiczenia w poszczególnych grupach nie ustalą inaczej). Zadania polegają na napisaniu programów w assemblerze procesora MIPS. Wszystkie programy powinny wykonywać się na symulatorze MARS. Dokładne zasady oddawania zadań zostaną podane do 10 kwietnia. Na wykładzie 8 kwietnia poświęcę jeszcze chwilę czasu na pokazanie przykładowego programu, uruchomienie go w MARSie i krótkie omówienie dyrektyw assemblera i funkcji systemowych, które mogą nam się przydać.

1. Napisz program, który wylicza sumę oraz maksimum z tablicy liczb całkowitych, której adres zadany jest w rejestrze \$a0, a rozmiar w rejestrze \$a1. Wynik umieść w rejestrze \$v0.
2. Napisz program, który sortuje tablicę liczb całkowitych metodą *quicksort* (założenia o tablicy jak wyżej).
3. Napisz program, który zlicza ile jedynek znajduje się w bitowej reprezentacji rejestru \$s0.
4. Załóżmy, że rejestr \$a0 przechowuje **dwie** liczby szesnastobitowe: liczbę a na swoich bardziej znaczących bitach oraz liczbę b na swoich mniej znaczących bitach. Obie zakodowane w reprezentacji uzupełnień do 2. W assemblerze MIPS napisz program, który umieści w rejestrze \$v0 dwie liczby szesnastobitowe: na bardziej znaczących bitach liczbę $a + b$, a na mniej znaczących – liczbę $a - b$. Zakładamy, że błędy przepelnienia nie wystąpią (nie musisz się nimi przejmować).
5. Załóżmy, że rejestr \$a0 przechowuje wartość zmiennopozycyjną zapisaną w standardzie IEEE 754 pojedynczej precyzji. Napisz program, który mnoży tę wartość przez 2^k , gdzie k jest zawartością rejestru \$a1. Wynik ma być zapisany również w tym samym standardzie, w rejestrze \$v0. Nie wolno używać rozkazów koprocatora zmiennopozycyjnego. Zakładamy dla uproszczenia, że wyrzucamy z formatu IEEE 754 wszystkie „sytuacje wyjątkowe” (jak NaN, nieskończoności, czy zdenormalizowane mantysy).
6. (2 pkt.) Załóżmy, że rejestry \$a0 i \$a1 przechowują wartości zmiennie pozycyjne zapisane w standardzie IEEE 754 pojedynczej precyzji. Napisz program, który mnoży te wartości przez siebie i wynik umieści w rejestrze \$v0. Tak jak w poprzednim zadaniu nie wolno używać rozkazów koprocatora zmiennopozycyjnego. I również zakładamy, że wyrzucamy z formatu IEEE 754 wszystkie „sytuacje wyjątkowe”.
7. Przygotuj program sprawdzający czy MARS zachowuje się jak maszyna „little-endian”, czy „big-endian”, tzn. w jakiej kolejności w słowie pamięci przechowuje kolejne bajty 4-bajtowej liczby całkowitej.
8. Napisz funkcję otrzymującą w rejestrze \$a0 adres napisu i zmieniającą w tym napisie duże litery na małe, a małe na duże (zakładamy, że w napisie będą tylko litery alfabetu łacińskiego i spacje). Przetestuj tą funkcję wczytując dane z klawiatury i wyświetlając wynik na ekran (za pomocą wywołań odpowiednich funkcji systemowych).
9. W assemblerze procesora MIPS napisz program wyznaczający rekurencyjnie wartość n -tego wyrazu ciągu Fibonacciego. Liczbę n wczytaj z klawiatury.