計算機システム演習　第三回レポート

17B13541　　細木隆豊

1　説明・工夫

slt を用いた blt の実装 (assigenment1blt.s)

slt を用いない場合

blt $s1, $s2, label

slt を用いた場合

slt $t0, $s1, $s2　　(1)

bne $t0, $zero, label

$s1 < $s2 を満たすとき、label へ jump すればよい。(1) を用いると、条件を満たすとき $t0 = 1 となる。0 と $t0 を比較し、not equal ならば j label より bne を用いた。
比較で$zero を用いると、新たにレジスタに値をいれなくてよいのでコードが短くなる。
assigenment1blt.s は 5 より小さい値を 0 から順に出力するプログラムである。

slt を用いた ble の実装 (assigenment1ble.s)

slt を用いない場合

ble $s1, $s2, label

slt を用いた場合

slt $t0, $s2, $s1　　(2)

beq $t0, $zero, label

blt と同じように考え、$s1 ≤ $s2 を満たすとき label に jump すればよい。こ

の条件は、$s2 < $s1 を満たさない場合、と言い換えることができる。(2) を用いると、$s1 ≤ $s2 のとき $t0 は 0 であるから、beq を用いる。
assigenment1ble.s は 5 以下の数を 0 から順に出力するプログラムである。

長さ 4 の二つの配列の要素の和 (assigenment2.s)

ループした回数 i (< 配列の長さ)、i × 4 (A[i] のアドレスを示すため)、(3 - i) × 4 (B[3-i] のアドレスを示すため) を記録する三つのレジスタを用意して、ループごとにこれらの値を更新していき、syscall を用いて A[i] + B[3-i] を console に出力するようにした。
assembly のコマンドとして、”mul”を用いてよいのかわからなかった (資料のサイトで用いているのを確認していなかった) ためこのようなプログラムになった (assign2.s では mul を用いる)。

任意長の配列に対して (assign2.s)

assigenment2.s に配列の長さが等しいかの確認 (check)、長さが異なった場合にプログラムを終了 (brk) するコードを追加した。実際に動かす場合は data の A,B と、main 内の”#配列 A の長さ”,”#配列 B の長さ”を変えればよい。

2 実行結果

assignment1blt.s

01234

assignment1ble.s

012345

assignment2.s

9999

assign2.s

A = {1, 2, 3, 4}, B = {5, 6, 7, 8} の時

9999

A = {1, 2, 3, 4}, B = {5, 6, 7, 8, 9} の時

Not equal the size of A and B.

A = {1, 2, 3, 4, 5}, B = {5, 6, 7, 8, 9} の時

1010101010

3　感想

アセンブリ言語を扱うのが初めてだったので、最初は慣れていなくてむずかしかったがやっていくうちにわかっていったように感じた。

assign2.s で jal を使って混乱していたが、メモリを配列のように扱うことの良さを実感できたと思う。