

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 (assignment1blt.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。

assignment1ble.s も同様だが、assignment1blt.s は二つの値のうち小さい値を出力するプログラムである。

slt を用いた ble の実装 (assignment1ble.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、 $\$s1 \leq \$s2$ 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、 $\$s1 \leq \$s2$ のとき $\$t0$ は 0 であるから、beq を用いる。

長さ 4 の二つの配列の要素の和 (assignment2.s)

ループした回数 i ($<$ 配列の長さ)、 $i \times 4$ ($A[i]$ のアドレスを示すため)、 $(3 - i) \times 4$ ($B[3-i]$ 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)

assignment2.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

3 感想