

Solucions Problemes Encarregats (4.2, 4.9, 4.11)

Problema 4.2 de la col·lecció

Tradueix a llenguatge ensamblador MIPS la següent funció, on N és una constant:

```
void func(int mat[][N], int i, int j) {
    if (i>j)
        mat[i][j] = mat[j][i];
}
```

```
func:    ble    $a1, $a2, fi_if
        #@mat[j][i] = @mat + (j*N + i)*4
        li     $t0, N
        mult   $a2, $t0
        mflo   $t1
        addu   $t1, $t1, $a1
        sll    $t1, $t1, 2
        addu   $t1, $t1, $a0
        lw     $t1, 0($t1)
        #@mat[i][j] = @mat + (i*N + j)*4
        mult   $a1, $t0
        mflo   $t3
        addu   $t3, $t3, $a2
        sll    $t3, $t3, 2
        addu   $t3, $t3, $a0
        sw     $t1, 0($t3)
fi_if:   jr     $ra
```

Problema 4.9 de la col·lecció

Donades les següents declaracions:

```
#define N 100
char vecchar[N];
int matint[N][N];
long long matlong[N][N];

void func() {
    int i;
    ... /* aquí va la sentència de cada apartat */
}
```

Tradueix a MIPS les següents sentències utilitzant accés seqüencial, suposant que pertanyen a la funció func, i que la variable i ocupa el registre \$t0:

a) for (i=1; i<N; i+=2) /* Atenció, la i va de 2 en 2 */
 vecchar[i] = N-i;

Solució:

```
        li     $t0, 1           #i
        la     $t1, vecchar + 1 #Punter inicial
        li     $t2, N
for:     bge    $t0, $t2, fi_for
        subu   $t3, $t2, $t0
        sb     $t3, 0($t1)
        addiu  $t1, $t1, 2       #stride
        addiu  $t0, $t0, 2
```

```

        b      for
fi_for:
b) for (i=0; i<N; i++)
        matint[i][0] = 0;

```

Solució:

```

        li      $t0, 0   #i
        la      $t1, matint          #Punter inicial
        li      $t2, N
for:      bge    $t0, $t2, fi_for
        sw      $zero, 0($t1)
        addiu   $t1, $t1, N*4        #stride
        addiu   $s0, $t0, 1
        b      for
fi_for:
c) i=4;
while (matint[3][i] != 0) {
    matint[3][i]--;
    i+=3;
}

```

Solució:

```

        la      $t1, matint + 3*N*4 + 4*4
while: lw    $t2, 0($t1)
        beq     $t2, $zero, fi_while
        addiu   $t2, $t2, -1
        sw      $t2, 0($t1)
        addiu   $t1, $t1, 12        #stride
        b      while
fi_while:
d) for (i=N-1; i>=0; i--)
        matlong[i][4] = i;   /* Atenció a la part alta! */

```

Solució:

```

        li      $t0, N-1   #i
        la      $t1, matlong + (N-1)*N*8 + 4*8
for:      blt    $t0, $zero, fi_for
        sw      $t0, 0($t1)
        sw      $zero, 4($t1)
        addiu   $t1, $t1, -N*8      #stride
        addiu   $t0, $t0, -1
        b      for
fi_for:

```

Problema 4.11 de la collecció

Donades les següents declaracions en C:

```

#define N 10
int mat[N][N];

```

Tradueix a ensamblador MIPS el codi següent fent servir la tècnica d'accés seqüencial, usant un sol punter.

```

i=0;
do {

```

```
    mat[i][i] = mat[i][i+1] - mat[i+1][i];  
    i += 2;  
} while (i<N);
```

SOL:

```
    li    $t0, 0  
    li    $t1, N  
    la    $t2, mat  
bucle:addu $t3, $t2, 4  
    lw    $t3, 0($t3)  
    addu   $t4, $t2, N*4  
    lw    $t4, 0($t4)  
    subu   $t4, $t3, $t4  
    sw     $t4, 0($t2)  
    addiu  $t0, $t0, 2  
    addiu  $t2, $t2, (N+1)*8  
    blt    $t0, $t1, bucle
```