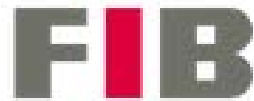


Tema 4. Problemes

Joan Manuel Parcerisa



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Problema 4.1 (e)

- Donades les següents declaracions en C (on NF i NC són constants) tradueix a MIPS la línia del requadre. Assumeix que i, j ocupen els registres \$t0, \$t1

```
int mat[NF][NC];  
int f(){  
    int i, j;  
    . . .  
    return mat[i+5][j-1];  
}
```

Problema 4.1 (f)

- Donades les següents declaracions en C (on NF i NC són constants) tradueix a MIPS la línia del requadre. Assumeix que i, j ocupen els registres \$t0, \$t1

```
int mat[NF][NC];  
int f(){  
    int i, j;  
    . . .  
    return mat[i*10 + 4][6];  
}
```

Problema 4.3

Donada la següent declaració d'una variable global (on N és una constant):

```
int M[N][N];
```

Tradueix a assembler MIPS el següent fragment de codi en C, usant el mínim nombre d'instruccions, i suposant que les variables *i* i *suma* ocupen els registres \$t0, \$t1:

```
suma += M[i][i+1] - M[i+1][i];
```

Problema 4.7 (a)

```
void func(int veci[4]) {  
    int j;  
    for (j=0; j<4; j++)  
        veci[j] = j;  
}
```

- a) Tradueix a MIPS la subrutina *func* utilitzant accés seqüencial al vector *veci* (*j* ocupa \$t0)

Problema 4.7 (b)

Donades les següents declaracions en C:

```
int mati[5][4];  
void main() {  
    int i;  
    for (i=0; i<5; i++)  
        func(&mati[i][0]);  
}
```

- b) Tradueix a MIPS la subrutina *main* utilitzant accés seqüencial a la matriu *mati*. **Fes atenció als registres on guardes el punter i la variable *i***

Problema 4.8 (d)

- Tradueix a MIPS la sentència del requadre. Aplica la tècnica d'**accés seqüencial** per als accessos a la matriu A (les variables *i*, *suma* ocupen \$t0, \$t1)

```
void func(int A[N][N]) {  
    int i, suma=0;  
    for (i=0; i<N; i+=3)  
        suma += A[i][N-1-i];  
}
```

Problema 3.13 (a)

3.13. Donada la següent declaració de la variable global V (on N és una constant):

```
int v[N];
```

Calcula el temps d'execució en cicles de rellotge de cada una de les següents versions del programa que serveix per sumar els elements de V, suposant que:

- Els accessos a memòria tarden 4 cicles
- Els salts que no salten tarden 1 cicle
- Els salts que salten tarden 2 cicles
- La resta d'instruccions tarden 1 cicle

```
a)  move    $t0, $zero      # suma
     move    $t1, $zero      # i
     move    $t2, $zero      # offset
bucle:
     slti    $t3, $t1, N
     beq     $t3, $zero, fi
     la      $t4, v
     addu    $t4, $t4, $t2
     lw      $t4, 0($t4)
     addu    $t0, $t0, $t4
     addiu   $t2, $t2, 4
     addiu   $t1, $t1, 1
     b      bucle
fi:
```

Problema 3.13 (e, f)

3.13. Donada la següent declaració de la variable global V (on N és una constant):

```
int v[N];
```

Calcula el temps d'execució en cicles de rellotge de cada una de les següents versions del programa que serveix per sumar els elements de V, suposant que:

- Els accessos a memòria tarden 4 cicles
- Els salts que no salten tarden 1 cicle
- Els salts que salten tarden 2 cicles
- La resta d'instruccions tarden 1 cicle

e)

```
move    $t0, $zero
li      $t1, N
la      $t2, V
```

bucle:

```
lw      $t3, 0($t2)
addu    $t0, $t0, $t3
addiu   $t2, $t2, 4
addiu   $t1, $t1, -1
bne     $t1, $zero, bucle
```

f) Programeu una nova versió que disminueixi el temps d'execució de l'aparat e).

Examen final 2019/20-Q1

Donada la següent funció en C:

```
short func(short M[][1000], int i) {  
    return M[i][i+3] + M[i+2][i];  
}
```

Completa els requadres del següent fragment de codi en ensamblador MIPS per tal que sigui la traducció correcta de la funció anterior:

```
func:   li      $t0,   
        mult   $t0,   
        mflo  $t0  
        addu  $t0, $t0,   
        lh   $t1, ($t0)  
        lh   $t2, ($t0)  
        addu $v0, $t1, $t2  
        jr   $ra
```

Examen final 2017/18-Q2

Tenim el següent codi en C, on M i V són variables globals:

```
int M[100][100], V[100];  
  
for (i=10;i<90; i++)  
    M[i][i+10]=V[i-10];
```

Completa els quadres de la traducció en assemblador MIPS que hi ha a continuació:

```
la    $t0, M +   
la    $t1, V  
addiu $t2, $t0,   
bucle: lw    $t3,  ($t1)  
sw    $t3, 0($t0)  
addiu $t1, $t1,   
addiu $t0, $t0,   
bne   $t0, $t2, bucle
```

Examen parcial 2017/18-Q1

Donada la següent subrutina en ensamblador MIPS:

```
acces_aleatori:
    li    $t0, 400
    mult  $t0, $a1
    mflo  $t0
    addu  $t0, $t0, $a0
    lw    $v0, 840($t0)
    jr    $ra
```

Sabem que és la traducció de la següent funció en C (de la qual desconeixem els valors dels requadres). Completa els requadres amb les expressions vàlides en C per tal que la traducció sigui correcta:

```
int acces_aleatori (int M[][100], int i)
{
    return M[ ] [ ];
}
```