

Solucions Problemes Encarregats (3.34, 3.26, 3.36)

Problema 3.34 de la col.lecció

Donades les següents declaracions de variables globals i de la funció A:

```
char x;  
int z;  
char w[20];  
  
char A(char i, int k, char *v) {  
    int r[10];  
    char c;  
    c = v[3];  
    r[k] = i;  
    return c;  
}
```

a) Traduir a ensamblador la sentència:

```
x = A(x,z,w);
```

b) Dibuixar el bloc d'activació d'una crida a la funció A.

c) Traduir a ensamblador la funció A.

Solució:

```
a) la $t0, x  
lb $a0, 0($t0)  
la $t0, z  
lw $a1, 0($t0)  
la $a2, w  
jal A  
la $t0, x  
sb $v0, 0($t0)
```

```
b) ----- <- sp - 40  
| r[0] |  
| r[1] |  
| .. .. |  
| r[9] |  
----- <- sp
```

```
c) A:  
addiu $sp, $sp, -40  
lbu $v0, 3($a2) # c ocupa $v0  
sll $t1, $a0, 2  
addu $t1, $t1, $sp  
sw $a1, 0($t1) # r[i] = k  
addiu $sp, $sp, 40  
jr $ra
```

Problema 3.26 de la col.lecció

Donades les següents declaracions de variables i funcions en C, tradueix a ensamblador MIPS la subrutina subr3.

```

char V[7];
int s3(char v1[], int tf, int nf, char c);

int subr3(int param)
{
    int ret, tam = 7;

    ret = s3(V, tam, param, 'D');
    return ret + tam + param;
}

```

Solució:

```

subr3:
    # Salvem registres: $ra, $s0 ($a0=param) i $s1 (tam)
    addiu    $sp, $sp, -12
    sw       $s1, 0($sp)
    sw       $s0, 4($sp)
    sw       $ra, 8($sp)

    li       $s1, 7          # tam ocuparà $s1

    # Copia $a0, passa 4 paràmetres, i fa la crida
    move     $s0, $a0        # copia $a0 en $s0
    la       $a0, V
    move     $a1, $s1
    move     $a2, $s0
    li       $a3, 'D'
    jal      s3              # ret queda en $v0

    addu     $v0, $v0, $s1    # ret + tam
    addu     $v0, $v0, $s0    # ret + tam + param

    lw       $s1, 0($sp)
    lw       $s0, 4($sp)
    lw       $ra, 8($sp)
    addiu    $sp, $sp, 12
    jr       $ra

```

Problema 3.36 de la col·lecció

Tradueix a assembleador la següent funció (escrivint punts suspensius per a les sentències desconegudes):

```

void sub(int p1[10], int p2, int p3)
{
    int x1[10], x2;
    ...
    x1[p2] = p1[x2] + p3;
    ...
    sub (x1, p2, p1[3]);
    ...
    x2 = p2 + p3;
    ...
}

```

Solució:

```

sub:
    # Salvem registres: $ra, $s0 ($a1=p2) i $s1 ($a2=p3)
    # Suposem que x2 ocupa $t0
    addiu $sp, $sp, -52
    sw     $s1, 40($sp)
    sw     $s0, 44($sp)
    sw     $ra, 48($sp)
    ...
    sll     $t1, $t0, 2
    addu $t1, $a0, $t1
    lw      $t1, 0($t1)
    addu $t1, $t1, $a2 # p1[x2] + p3
    sll     $t2, $a1, 2
    addu $t2, $sp, $t2
    sw      $t1, 0($t2)    # x1[p2] = ...
    ...
    move $s0, $a1
    move $s1, $a2
    lw      $a2, 12($a0)
    move $a0, $sp          # $a0 ($a1 ja conté p2)
    jal     sub
    ...
    addu $t0, $s0, $s1 # x2 = p2 + p3
    ...
    lw      $s1, 40($sp)
    lw      $s0, 44($sp)
    lw      $ra, 48($sp)
    addiu $sp, $sp, 52
    jr      $ra

```