

# Tema 3. Traducció de Programes

## Estructura de Computadors (EC)

Rubèn Tous

rtous@ac.upc.edu  
Computer Architecture Department  
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# Índex

- 1 3.5 Sentències alternatives if-then-else i switch
- 2 3.4 Sentències iteratives while, for i do-while

## 3.5.3 Sentència switch

Exemple simple (sense fallthrough):

```
1  int month;  
2  int days;  
3  switch(month) {  
4      case 2:  
5          days = 28;  
6          break;  
7      case 4:  
8          days = 30;  
9          break;  
10     case 6:  
11         days = 30;  
12         break;  
13     case 9:  
14         days = 30;  
15         break;  
16     case 11:  
17         days = 30;  
18         break;  
19     default:  
20         days = 31  
21 }
```

## 3.5.3 Sentència switch

Equivaldria a un if:

```
1  int month;  
2  int days;  
3  if (month == 2)  
4      days = 28;  
5  else if (month == 4)  
6      days = 30;  
7  ...  
8  else  
9      days = 31;  
10 }
```

## 3.5.3 Sentència switch

Traducció a MIPS naïf:

```
1      #t1 = month, $t2 = days
2 case2:
3      li $t0, 2
4      bne $t1, $t0, case4
5      li $t2, 28
6      b endswitch
7 case4:
8      li $t0, 4
9      bne $t1, $t0, case6
10     li $t2, 30
11     b endswitch
12     ....
13 default:
14     li $t2, 31
15 endswitch:
```

## 3.5.3 Sentència switch

Traducció a MIPS amb jumtable:

```
1      # $t1 = month, $t2 = days
2      .data
3      jumtable:
4          .word default
5          .word case2
6          .word default
7          .word case4
8          .word default
9      ...
10     subu    $t0, $t1, 1 # month - smaller value
11     sll     $t0, $t0, 2
12     la      $t4, jumtable
13     addu    $t0, $t4, $t0
14     lw      $t0, 0($t0)
15     jr      $t0
16
17 case2:
18     li      $t2, 28
19     b       ends witch
20 case4:
21     li      $t2, 30
22     b       ends witch
23     ....
24 default:
25     li      $t2, 31
26 ends witch:
```

## 3.5.3 Sentència switch

Exemple amb fallthrough (empty cases):

```
1  int month;  
2  int days;  
3  switch(month) {  
4      case 2:  
5          days = 28;  
6          break;  
7      case 4:  
8      case 6:  
9      case 9:  
10     case 11:  
11         days = 30;  
12         break;  
13     default:  
14         days = 31  
15 }
```

Podem traduir amb `if(month == 4 || month == 6...)`.

## 3.5.3 Sentència switch

I si falten 'break'?. Exemple dies total fins a un mes donat:

```
1  int month;  
2  int total_days = 0;  
3  switch(month) {  
4      case 12:  
5          total_days += 31;  
6      case 11:  
7          total_days += 30;  
8      case 10:  
9          total_days += 31;  
10     ...  
11 }
```



## 3.5.3 Sentència switch

Amb ifs.

```
1  if (month == 12)
2      days += 31;
3  if (month == 12) || month == 11)
4      days += 30;
5  ...
```

Amb jump table només hem d'eliminar els 'b' que calgui.

## 3.4.1 Sentència while

En C:

```
1 while (condicio)
2     sentencia_cos_while
```

El patró en MIPS serà:

```
while:
    avaluar condicio
    salta si és falsa a fiwhile
    traducció sentencia_cos_while
    salta a while
fiwhile
```

## 3.4.1 Sentència while

Exemple divisió entera:

```
1 q = 0;  
2 while (dd >= dr) {  
3     dd = dd - dr;  
4     q++;  
5 }
```

En MIPS serà:

```
1  move    $t3, $zero          # q = 0  
2  while:  
3      blt  $t1, $t2, fiwhile   # salta si dd<dr  
4      subu $t1, $t1, $t2       # dd = dd - dr  
5      addiu $t3, $t3, 1        # q++  
6      b   while  
7  fiwhile:
```

## 3.4.1 Sentència while

Optimització: avaluació de la condició al final del bucle:

```
1  move    $t3, $zero          # q = 0
2  blt     $t1, $t2, fiwhile   # salta si dd<dr
3  while:
4  subu    $t1, $t1, $t2       # dd = dd - dr
5  addiu   $t3, $t3, 1         # q++
6  bge     $t1, $t2, while     # salta si dd>=dr
7  fiwhile:
```

## 3.4.1 Sentència while

Si ens volem estalviar escriure dos cops el codi que avalua la condició:

```
1      move    $t3, $zero          # q = 0
2      b       test
3  while:
4      subu    $t1, $t1, $t2       # dd = dd - dr
5      addiu   $t3, $t3, 1         # q++
6  test:  bge   $t1, $t2, while    # salta si dd>=dr
7  fiwhile:
```

## 3.4.2 Sentència for

En C:

```
1 for (s1; condicio; s2)  
2   s3;
```

Equival a:

```
s1;  
while (condicio) {  
    s3;  
    s2;  
}
```

## 3.4.3 Sentència do-while

En C:

```
1 do  
2     sentencia_cos_do  
3 while (condicio);
```

El patró en MIPS serà:

do:

```
traducció sentencia_cos_do  
avaluar condicio  
salta si és certa a do
```