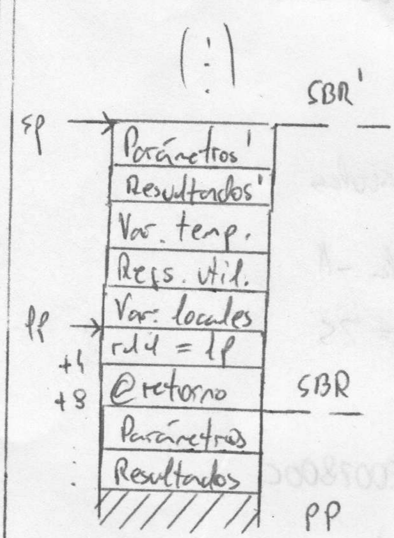
SUBRUTINAS RECURSIVAS

```
    PUSH {r0..r3}   (tachado)
    sub sp, sp, #4n        <-
    PUSH {rn}
    bl  etiq
- - - - - - - - - - - - - - -
etiq PUSH {lr, r11}
    mov r11, sp
    PUSH {rn}
    ldr rn, [r11, #4n]
    ;Tratamiento        NO TRIVIAL
                      ↓ TRIVIAL/REGRESO
    str rn, [r11, #4n]
    POP {rn, r11, pc}
- - - - - - - - - - - - - - -
    add sp, sp, #4n
    POP {rn}
```

PILA, BLOQUE DE ACTIV.

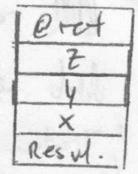


ASM - C

char n → ldrb/strb

int n → ldr/str

f(x, y, z) →



Valor → Datos

Referencia → Dirección

Parametros → r0-r3, resto en pila

Resultados → r0-r3, resto en mem (r11)

RUTINAS DE SERVICIO

```
    LDR r0, =VICVectAddr0    ; =@FFFFF100
    LDR r1, =perif_so
    mov r2, #n               ; n = IRQn
    ldr r3, [r0, r2, LSL #2]
    str r3, [r1]
- - - - - - - - - - - - - - -
    LDR r1, =RSI_perif
    str r1, [r0, r2, LSL #2]
-----------------------------
    LDR r0, =VICIntEnable    ; =@FFFFF010
    mov r1, #2_1000...0   (n-1)
    str r1, [r0]
-----------------------------
    ;Bucle principal
-----------------------------
    LDR r0, =VICIntEnClr     ;=@FFFFF014
    mov r1, #2_1000...0   (n-1)
    str r1, [r0]
-----------------------------
    LDR r0, =VICVectAddr0
    LDR r1, =perif_so
    ldr r1, [r1]
    mov r2, #n
    str r1, [r0, r2, LSL #2]
fin b fin
```

```
RSI_perif  sub lr, lr, #4
           PUSH {lr}
           mrs r14, spsr
           PUSH {r14}
           msr cpsr_c, #2_01010010 / 0x52
           PUSH {rn}
- - - - - - - - - - - - - - -
;Transferencia de información
;Si es necesario, señales de control y/o bajar prioridad
;Si conveniente, tratar información
- - - - - - - - - - - - - - -
           POP {rn}
           msr cpsr_c, #2_11010010  0xD2
           POP {r14}
           msr spsr_fsxc, r14
           LDR r14, =VICVectAddr    ; @FFFFF030
           str r14, [r14]
           POP {pc}^
```

PERIFÉRICOS

TECLADO (UART 1) (IRQ 7)

```
; Prólogo
    LDR r0, =RDAT_UART1     ; 0xE0001000
    ldrb r1, [r0]
; Tratamiento de la letra
    bic r1, r1, #0x20       ; Mayúsculas
    sub r1, r1, #65         ; Tecla - A
    cmp r1, #25             ; >= 25
    bhi etiq
    LDR r0, =IOCLR          ; 0xE002800C
    mvn r2, #0              ; = -1
    str r2, [r0]
    LDR r0, =IOSET          ; 0xE0028004
    mov r2, #1
    mov r2, r2, LSL r1      ; r2 [tecla] = 1
    str r2, [r0]
; Epílogo
```

TIMER (TIMER0) (IRQ4)

```
; Prólogo
    LDR r0, =T0_IR          ; 0xE0004000
    mov r1, #1              ; = 1
    str r1, [r0]            ; T0_IR [0] = 1
    ; Tratamiento del tick
; Epílogo
```