

Auxiliar 7

Profesor: Pablo Guerrero.

Auxiliar: Ian Yon

Viernes 26 de septiembre de 2014



ARM

Sintaxis general: `< operation > {cond}{flags} Rd, Rn, Operand2`

Si *flags* vale *S*, se actualizan *FLAGS* de condición

Problema 1

Traduzca la siguiente función en Assembler ARM a código C y señale su utilidad:

```
mystery:
    SUB    r1, r0, #'A'
    CMP    r1, #'Z' - 'A'
    ADDLS  r0, r0, #'a' - 'A'
    MOV    pc, r14
    END
```

Problema 2

Implemente la función factorial en ARM Assembler de forma iterativa.

Problema 3

Transcriba el siguiente programa a C y señale su utilidad, luego escriba un programa en ARM Assembler que use la función foo:

```
.globl foo
.equ datum_size,4
foo:
    stmfd sp!, {r4, r5, r6, r7, lr}
    cmp    r1, #0
    ble    last
    mov    r4, r0
    mov    r5, r1
    mov    r6, r2
    mov    r7, r3
    ldr    r1, [r4], #datum_size
    mov    r0, r6
    bl     printf
    subs  r5, r5, #1
    beq    last
foo_loop:
    ldr    r1, [r4], #datum_size
    mov    r0, r7
    bl     printf
```

```

        subs    r5, r5, #1
        bne     foo_loop
last:
        ldmfd   sp!, {r4, r5, r6, r7, pc}

```

Problema 4

Transcriba el siguiente programa a C y señale su utilidad:

```

.globl updater
updater:
    pushl %ebp
    movl %esp, %ebp
    pushl %esi
    pushl %edi
    pushl %ebx
    movl 8(%ebp), %ebx #nod
    movl 12(%ebp), %esi #operation
    movl 16(%ebp), %edi #father
left:
    movl 4(%ebx), %eax
    testl %eax, %eax
    je value
    movl (%ebx), %ecx
    pushl %ecx
    pushl %esi
    pushl %eax
    call updater
value:
    pushl %edi
    movl (%ebx), %eax
    pushl %eax
    call %esi
    movl %eax, (%ebx)
right:
    movl 8(%ebx), %eax
    testl %eax, %eax
    je return
    movl (%ebx), %ecx
    pushl %ecx
    pushl %esi
    pushl %eax
    call updater
return:
    popl %ebx
    popl %edi
    popl %esi
    leave
    ret

```