Auxiliar 4

Profesor: Pablo Guerrero.
Auxiliar: Ian Yon
Viernes 24 de abril de 2015

Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Problema 1

Traduzca el siguiente programa a C:

```
s:
    pushl %ebp
    movl %esp, %ebp
    pushl %ebx
    movl 8(%ebp), %edx
    movl 12(%ebp), %eax
    movl (%edx), %ecx
    movl (%eax), %ebx
    movl %ebx, (%edx)
    movl %ecx, (%eax)
    popl %ebx
    popl %ebp
    ret
r:
    pushl %ebp
    movl %esp, %ebp
    pushl %edi
    pushl %esi
    pushl %ebx
    subl $8, %esp
    movl 12(%ebp), %edi
    testl %edi, %edi
    jle .L6
    movl 8(%ebp), %ebx
    movl $0, %esi
.L5:
    leal 4(%ebx), %eax
    movl %eax, 4(%esp)
    movl %ebx, (%esp)
    call s
    addl $2, %esi
    addl $8, %ebx
    cmpl %esi, %edi
    jg .L5
.L6:
    addl $8, %esp
    popl %ebx
    popl %esi
    popl %edi
    popl %ebp
    ret
```

Problema 2

Dada la siguiente estructura:

```
struct node{
       int value;
       struct node *left;
       struct node *right;
};
```

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
    pushl %edi
    movl (%ebx), %eax
    pushl %eax
    call %esi
    movl %eax, (%ebx)
right:
    movl 8(%ebx), %eax
    testl %eax, %eax
    je return
```

```
    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:
    movl (%ebx), %ecx
    pushl %ecx
    pushl %esi
    pushl %eax
    call updater
return:
    popl %ebx
    popl %edi
    popl %esi
    leave
    ret
```

Problema 3

Traduzca el siguiente programa a C:

```
t:
    pushl %ebp
    movl %esp, %ebp
    subl $24, %esp
    movl %ebx, -8(%ebp)
    movl %esi, -4(%ebp)
    movl 8(%ebp), %ebx
    movl $0, %eax
    testl %ebx, %ebx
    je .L2
    movl (%ebx), %eax
    movl %eax, (%esp)
    call t
    movl %eax, %esi
    movl 4(%ebx), %eax
    movl %eax, (%esp)
    call t
    leal (%esi,%eax), %eax
    addl 8(%ebx), %eax
.L2:
    movl -8(%ebp), %ebx
    movl -4(%ebp), %esi
    movl %ebp, %esp
    popl %ebp
    ret
```