

CC4301 Arquitectura de Computadores

Auxiliar 6

Prof. Aux.: Gaspar Pizarro V.

1 de octubre de 2012

1. Decompilación

Considere el siguiente código en ensamblador para x86, 32 bits, sintaxis AT&T:

```
.globl f
f:
    pushl %ebp
    movl %esp, %ebp
    pushl %edi
    pushl %esi
    pushl %ebx
    movl 8(%ebp), %edi
    cmpl $1, %edi
    je L6
    testl %edi, %edi
    jz L6
    movl $1, %eax
.L5:
    mull %edi
    subl $1, %edi
    testl %edi, %edi
    jg .L5
    jmp L7
L6:
    movl $1, %eax
L7:
    popl %ebx
    popl %esi
    popl %edi
    leave
    ret
```

Transcriba este código a código C, que realicé la misma función.

2. P1 Control 2 año 2006

Transcriba el siguiente programa en Assembler a C:

```
.text
.globl p
p:
    pushl %ebp
    movl %esp, %ebp
    pushl %edi
    pushl %esi
    pushl %ebx
    movl 8(%ebp), %esi
    movl 12(%ebp), %edi
    movl 16(%ebp), %ecx
    decl %ecx
    movl $0, %ebx
.L2:
    leal (%ebx, %ecx), %eax #eax = ebx+ecx
    shrl %eax
    movl (%edi, %eax, 4), %edx
    cmpl %esi, %edx
    je .L5
    cmpl %esi, %edx
    jge .L6
    leal 1(%eax), %ebx
    jmp .L8
.L6:
    leal -1(%eax), %ecx
.L8:
    cmpl %ecx, %ebx
    jbe .L2
    movl $-1, %eax
.L5:
    popl %ebx
    popl %esi
    popl %edi
    popl %ebp
    ret
```

3. P1 Control 2 año 2009

```
.globl p
p:
    pushl %ebp
    movl %esp, %ebp
    pushl %edi
    pushl %esi
    pushl %ebx
    subl $12, %esp
    movl 8(%ebp), %esi
    movl 12(%ebp), %ebx
    movl -4(%esi,%ebx,4), %edi
    leal -2(%ebx), %ecx
    testl %ecx, %ecx
    jle .L2
    leal -8(%esi, %ebx, 4), %edx
.L4:
    movl (%edx), %eax
    movl %eax, 4(%edx)
    subl $4, %edx
    decl %ecx
    jne .L4
.L2:
    movl %edi, 4(%esi)
    movl %ebx, 4(%esp)
    movl %esi, (%esp)
    call q
    addl $12, %esp
    popl %ebx
    popl %esi
    popl %edi
    popl %ebp
    ret
```