Auxiliar de Assembler CC41C
Rodrigo Canovas
04 Mayo 2006

1. Problema 1

El siguiente es un programa en assembler x86. Escriba el programa equivalente en C.

```
  .globl Q
Q:
  pushl  %ebp
  movl   %esp, %ebp
  pushl  %esi
  pushl  %ebx
  movl   8(%ebp), %edx  # 1er. parámetro
  movl   12(%ebp), %ecx # 2do. parámetro
  movl   16(%ebp), %ebx # 3er. parámetro
  movl   (%edx,%ecx,4), %esi
  movl   (%edx,%ebx,4), %eax
  movl   %eax, (%edx,%ecx,4)
  movl   %esi, (%edx,%ebx,4)
  popl   %ebx
  popl   %esi
  popl   %ebp
  ret
  .globl P
P:
  pushl  %ebp
  movl   %esp, %ebp
  pushl  %edi
  pushl  %esi
  pushl  %ebx
  subl   $12, %esp
  movl   8(%ebp), %edi
  movl   12(%ebp), %ebx
  movl   16(%ebp), %esi
  # si %ebx>=%esi goto .L10
  cmpl   %esi, %ebx
  jge    .L10
.L8:
  # si (%edi,%ebx,4)>=0 goto .L6
  cmpl   $0, (%edi,%ebx,4)
  jns    .L6
  addl   $1, %ebx
  jmp    .L3
.L6:
  subl   $4, %esp
  pushl  %esi # 3er. arg
  pushl  %ebx # 2do. arg
  pushl  %edi # 1er. arg
  call   Q
  subl   $1, %esi
  addl   $16, %esp
.L3:
  # si %ebx<%esi goto .L8
  cmpl   %esi, %ebx
  jl     .L8
.L10:
  movl   %ebx, %eax        # valor de ret.
  leal   -12(%ebp), %esp   # %esp= %ebp-12
  popl   %ebx
  popl   %esi
  popl   %edi
  popl   %ebp
  ret
```

2. Problema 2

Cree un procedimiento que sume números sin signo y de tamaño extendido (Bignums). Estos números se almacenan en arreglos de enteros sin signo y de tamaño variable.

Los números se respresentan mediante la estructura:

```
typedef struct {  int size; unsigned int *words; }
Bignum;
```

El campo size indica el número de palabras contenidas en el arreglo words. El elemento words[0] contiene los 32 bits menos significativos del número presentado, words[1] los siguientes 32 bits , etc.
Se le pide programar el siguiente procedimiento:

```
int addBignum(Bignum *a, Bignum *b, Bignum *res);
```

Este procedimiento suma a y b y deel resultado en res. Además retorna el número de palabrtas que se necesita para almacenar el resultado sin perdida de magnitud.