

Evaluación Continua (Estructura de Computadores 2018-19)

HISTORIAL DE REVISIONES			
NÚMERO	FECHA	MODIFICACIONES	NOMBRE

Índice

1ª Prueba Parcial. 2018 Noviembre 10.
Grado de Informática 2º curso. Estructura de Computadores.
Universidad Pública de Navarra.
Duración: 90 minutos.
Apellidos:
Nombre:

**importante**

Puede utilizarse todo tipo de información escrita como memorias de prácticas, apuntes, hojas de referencia, etc. No puede utilizarse ningún dispositivo electrónico como calculadoras, teléfonos, ordenadores, etc... Se han de incluir en la respuestas todo tipo de desarrollo necesario para llegar al resultado.

1. Computadora Institute Advanced Studies (IAS) de von Neumann:
 - a. (1 pto) Desarrollar un programa que realice la resta 0x00-0xFF y almacene el resultado en la variable denominada "resta". ¿Cuál será el contenido de la posición de memoria de la variable "resta"?
 - b. (1 pto) ¿Qué relación existe entre los tres componentes MAR, MBR y PC?
2. (1 pto) Cuál es el código digital del string de seis caracteres "Hola \n"
3. (1 pto) Los números 0123 y 0777 son números sin signo en base octal. Realizar la suma 0123+0777 directamente en base octal.
4. (1 pto) Los números 0xABC y 0xEFE son números con signo en complemento a dos. Realizar la resta 0xABC-0xEFE directamente. Calcular el valor del resultado.
5. (1 pto) Representar el número decimal 6.25 en formato IEEE-754 de doble precisión.
6. Formato de instrucciones:
 - Una computadora tiene una unidad de memoria de 256K palabras 32 bits cada una direccionable byte a byte. En una de las palabras de la memoria se almacena una instrucción. La instrucción tiene un formato de cuatro campos: un bit de indirección, un código de operación, un campo de operando para direccionar uno de los 64 registros y campo de operando que contiene direcciones de memoria.
 - a. (2 pto) ¿Cuántos bits forman el campo de código de operación? ¿Y del campo de registro? ¿Y del campo de direcciones?
 - b. (2 pto) ¿Cuántos bits forman parte del bus de direcciones y del bus de datos de la unidad de memoria?
7. (2 pto) En una subrutina indicar qué relación existe entre el puntero "frame pointer" del frame de la subrutina y la dirección de memoria donde se guarda la dirección de retorno.
8. (3 pto) Completar el código fuente del programa en lenguaje ensamblador adjunto teniendo en cuenta los comentarios que se adjuntan en el módulo fuente siguiente donde el algoritmo desarrollado realiza la conversión de un número decimal a código binario:

```
### Programa: convert_decbin.s
### Descripción: Convierte el número natural decimal 15 en binario mediante divisiones ←
                 sucesivas por 2
###           El código binario tiene un tamaño de 32 bits
### gcc -m32 -g -nostartfiles -o convert_decbin convert_decbin.s
### Ensamblaje as --32 --gstabs convert_decbin.s -o convert_decbin.o
### linker -> ld -melf_i386 -o convert_decbin convert_decbin.o

## MACROS
```

```

    ## DATOS

dec:      .          15      # decimal (tamaño 4 bytes) a convertir en un código binario de  ←
    32 bits
    ## bin almacena el código en sentido inverso, bin[0] almacena el bit de menor  ←
    peso.
bin:      .space  32      # array de 32 bytes: almacena en cada byte un bit del código  ←
    binario de 32 bits
divisor:      .          # divisor (de tamaño 1 byte)

    ## INSTRUCCIONES

    ## inicializo ECX con el valor del divisor

    ## inicializo el índice del array bin

    ## Cargo el dividendo en EAX
    # eax <-x

    ## extendiendo el bit de signo del dividendo en EDX
    # El dividendo siempre es positivo

    ## Divisiones sucesivas por 2 hasta que el cociente valga 0
bucle:
    ## idivl : [EDX:EAX] / Operando_fuente
    # EAX<-Cociente{x/y} , EDX<-Resto{x/y}
    # guardo el resto (de tamaño 1 byte) en el array bin
    ## extendiendo el bit de signo en edx
    # El dividendo siempre es positivo

    ## actualizo el índice del array

    ## compruebo si el cociente ha llegado a cero para salir del bucle

    ## Devuelvo el número de bits del código binario en EBX

    ## Código de la llamada al sistema operativo

    ## Interrumpo la rutina y llamo al S.O.

```

■ Mediante comandos del depurador GDB

- (2 pto) imprimir el contenido del array "bin" con dos expresiones diferentes utilizando los comandos "examinar" y/o "imprimir".
- (2 pto) imprimir el contenido del primer elemento del array bin
- (2 pto) imprimir el contenido del último elemento del array bin

9. (2 pto) Llamadas al sistema

- Completar el programa "convert_decbin.s" con el código necesario para imprimir en la pantalla un mensaje de bienvenida mediante la llamada directa write.