

Evaluación Recuperación (Estructura de Computadores 2018-19)

HISTORIAL DE REVISIONES			
NÚMERO	FECHA	MODIFICACIONES	NOMBRE

Índice

1. 1º Parcial	1
2. 2º Parcial	2
3. Examen Práctico	3

Prueba de Recuperación. 2019 Enero 15.

APELLIDOS:

Grado de Informática 2º curso. Estructura de Computadores. NOMBRE:

Universidad Pública de Navarra. Duración: 100 minutos.

GRUPO:

**importante**

Se han de incluir en las respuestas de los ejercicios todo tipo de desarrollo necesario para obtener el resultado.

1. 1º Parcial

1. (2.0 pts) Realizar la suma $0xAF+0x3B+0x2C+0xEE$ indicando
 - a. respecto de la suma de los dígitos de la "posición primera" de todos los sumandos:
 - I. cuál es el valor del 1º dígito de la suma
 - II. cuál es el valor de la llevada del 1º dígito de la suma
 - III. cuál es el 1º dígito hexadecimal de la suma
 - b. respecto de la suma de los dígitos de la "posición segunda" de todos los sumandos:
 - I. cuál es el valor del 2º dígito de la suma
 - II. cuál es el valor de la llevada del 2º dígito de la suma
 - III. cuál es el 2º dígito hexadecimal de la suma
 - c. Cual es el resultado de la suma, representado en hexadecimal:
 2. (2.0 pts) Una computadora tiene un banco de 8 registros y una unidad de memoria principal de 4MB palabras de 64 bits cada una direccionable byte a byte. En una de las palabras de la memoria se almacena una instrucción que tiene dos operandos, uno referenciado mediante un registro y el otro mediante direccionamiento indexado relativo a memoria. Calcular el formato de dicha instrucción indicando:
 - a. (0.5 pto) ¿Cuales son los campos que completan el formato de instrucción?
 - b. (1.5 pto) ¿Cuántos bits forman cada campo del formato de instrucciones?
 3. (6.0 pts) Desarrollar un programa en código ensamblador GNU i386 que imprima en la pantalla valores de números enteros en el rango $(-127, +127)$ en código binario de tamaño 1 byte .Restricciones:
 - LOS DISTINTOS BLOQUES DEL MÓDULO FUENTE HAN DE ESTAR COMENTADOS PARA SER LEGIBLE Y TRANSPARENTE EL **ALGORITMO** QUE IMPLEMENTA EL PROGRAMA Y SERAN LA BASE DE LA **EVALUACIÓN**
 - NO utilizar ninguna librería de C ni de ningún otro lenguaje.
 - Utilizar solo macros para hacer referencia a los valores constantes.
 - La variable "dato_entrada" para el número de entrada a codificar ha de tener el tamaño de 1 byte.
 - Inicializar el número a representar con el valor -127.
 - Arquitectura i386, Mnemónicos básicos : MOV PUSH POP ADD SUB DIV IDIV MUL IMUL CMP TEST SAL SAR SHL SHR NEG NOT AND OR XOR LEA CALL RET JMP JE JNE JZ JNZ JS JG JL JA JB
-

2. 2º Parcial

1. (2,5 pts) Unidad Central de Proceso de una Computadora
 - a. Cuál es la interfaz entre la CPU y los programas software
 - b. Qué dos bloques básicos completan la microarquitectura de la CPU
 - c. Indica los elementos principales de la Ruta de Datos por orden de importancia.
 - d. Cuál es la función básica de la Unidad de Control.
 - e. Ventajas de la arquitectura RISC desde el punto de vista del programador, compilador, hardware y comercial.
 - f. Indica las Fases principales del ciclo de instrucción.
 - g. Que nombre recibe la técnica de paralelismo a nivel de ciclo de instrucción.
 2. (2,5 pts) Unidad de Memoria
 - a. Qué significa que la memoria principal DRAM sea dinámica
 - b. Qué significa que la memoria SDRAM sea síncrona
 - c. Función básica del controlador de Memoria Principal
 - d. Qué unidades hardware necesita la cpu para conectarse a la Memoria Principal
 - e. Qué nombres reciben los agrupamientos de memoria en que se organiza internamente la Memoria Principal
 3. (2,5 pts) Mecanismos de entrada/salida
 - a. Cuales son los dos tipos de organización del espacio de direcciones de la memoria de entrada/salida
 - b. Qué nombre reciben los registros de memoria de entrada/salida
 - c. Cual es el mecanismo de control de entrada/salida que consigue mayor rendimiento de transferencia de datos.
 - d. Qué es un mecanismo de sincronismo de entrada/salida
 - e. Qué tipo de variable es un vector de interrupción
 4. (2,5 pts) Dos microprocesadores de 32 bits y 16 bits tienen la misma duración del ciclo de bus y la misma latencia de memoria RAM. Teniendo en cuenta que en media, el 20% de los operandos e instrucciones son del tamaño de 32 bits, el 40% de 16 bits y el 40% de 8 bits, responder a las cuestiones siguientes:
 - a. En el caso del microprocesador de 16 bits, si el ciclo de bus es "T", calcular el tiempo de captura medio de datos o instrucciones de la memoria RAM.(0,5 pts)
 - b. En el caso del microprocesador de 32 bits, si el ciclo de bus es "T", calcular el tiempo de captura medio de datos o instrucciones de la memoria RAM.(0,5 pts)
 - c. Calcular la mejora conseguida con el microprocesador de 32 bits en la captura de operandos e instrucciones. (0,5 pts)
 - d. Calcular la mejora en ancho de banda del bus conseguida con el microprocesador de 32 bits.(0,5 pts)
 - e. Si todas las instrucciones tienen un operando, calcular el número de accesos medio a memoria del ciclo de instrucción del microprocesador de 16 bits. (0,5 pts)
-

3. Examen Práctico

Prueba de Recuperación. 2019 Enero 15.

APELLIDOS:

Grado de Informática 2º curso. Estructura de Computadores. NOMBRE:

Universidad Pública de Navarra. Duración: 110 minutos.

GRUPO:

1. Desarrollar el programa **rep_bin.s** en código ensamblador GNU i386 que imprima en la pantalla valores de números enteros en el rango $(-127, +127)$ representados en **código binario** de tamaño 1 byte. Restricciones:

- LOS DISTINTOS BLOQUES DEL MÓDULO FUENTE HAN DE ESTAR COMENTADOS PARA SER LEGIBLE Y TRANSPARENTE EL ALGORITMO QUE IMPLEMENTA EL PROGRAMA Y SERÁN LA BASE DE LA EVALUACIÓN
- NO utilizar ninguna librería de C ni de ningún otro lenguaje
- Utilizar solo macros para hacer referencia a los valores constantes.
- La variable "dato_entrada" para el número entero de entrada a codificar ha de tener el tamaño de 1 byte
- Inicializar el número entero de entrada a representar en código binario con el valor -127.
- Desarrollar el programa partiendo de una versión sencilla e ir extendiendo dicha versión en versiones sucesivas hasta completar el algoritmo final:
- **Versión 1** (1 pto)- **rep_bin_v1.s**: imprimir los 5 mensajes siguientes mediante llamadas directas al sistema operativo:

```
Mensaje 1: "Examen Enero 2019"
Mensaje 2: "El número es Positivo : "
Mensaje 3: "El número es Negativo : "
Mensaje 4: "El número es Par : "
Mensaje 5: "El número es Impar : "
```

- Incluir en el módulo fuente a modo de comentarios: El comando para compilación del programa y el comando de depuración gdb para la visualización del 3º mensaje.
- **Versión 2** (1 pto)- **rep_bin_v2.s**: modificar la versión 1 para imprimir únicamente dos mensajes: el Mensaje 1 se imprime incondicionalmente, el Mensaje 2 se imprimirá únicamente si el número es positivo y el Mensaje 3 si el número es negativo.
 - En el programa fuente a modo de comentario indicar una descripción del algoritmo mediante el lenguaje natural.
 - En el programa fuente a modo de comentario indicar el comando de depuración gdb para visualizar la dirección de memoria de la primera instrucción del programa
- **Versión 3** (2 ptos)- **rep_bin_v3.s**: modificar la versión 2 añadiendo una *subrutina* que imprima si el número es par o impar, por lo tanto se imprimirán 3 mensajes, el Mensaje 1 incondicional y el resto de Mensajes condicionados al signo del dato de entrada.
 - En el programa fuente a modo de comentario indicar una descripción del algoritmo mediante el lenguaje natural.
 - En el programa fuente a modo de comentario indicar el comando de depuración gdb para visualizar de la sección de pila el argumento de la subrutina, la dirección de la pila que almacena el argumento y la dirección de la pila donde se almacena la dirección de retorno.
- **Versión 4** (6 ptos)- **rep_bin.s**: modificar la versión 3 con una subrutina que no imprima si el dato de entrada es par o impar, sino que imprima su correspondiente código binario en complemento a 2. La estructura del programa es la de un solo fichero con la rutina y la subrutina.

nota

Una vez finalizadas las versiones 1, 2 y 3 llamar al profesor para su EVALUACION inmediata. Los programas fuente y el histórico o traza de comandos GDB se archivarán en un único archivo "apellido1_apellido2_N.zip", donde apellido1 y apellido2 son los apellidos de la alumn@ y N el grupo, para su envío a través de la tarea abierta en miaulario para tal efecto.
