

**UNIVERSIDAD INTERAMERICANA DE PUERTO RICO
RECINTO METROPOLITANO
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE CIENCIAS DE COMPUTADORAS Y MATEMÁTICAS**

PRONTUARIO

I. INFORMACIÓN GENERAL

Título del Curso:	ORGANIZACION DE COMPUTADORAS Y LENGUAJE DEL ENSAMBLADOR
Código y Número:	COMP 3200
Créditos:	3
Término Académico:	agosto-octubre 2010 (2011-13)
Profesora:	
Horas de Oficina:	
Teléfono de la Oficina:	787-250-1912 Ext 2326
Correo Electrónico:	

II. DESCRIPCIÓN

Organización y estructura de los componentes principales de un sistema de computadora. Representación y manipulación de datos numéricos y no numéricos en el nivel de máquina. Comparación entre diferentes conjuntos de instrucciones y modos de dirección correspondiente. Carga de registros "Fetching" y ejecución de operaciones dependiendo de la arquitectura. Introducción a conceptos de interrupciones. Técnicas de acceso y manejo de memoria, registros y periféricos. Requiere horas adicionales de laboratorio abierto. Requisito: COMP 2900.

III. OBJETIVOS

Al finalizar las unidades temáticas del curso y mediante las experiencias de aprendizaje, el estudiante podrá:

1. Describir las diferentes representaciones numéricas usadas en assembler: binario, decimal hexadecimal y las representaciones simbólicas: ASCII y BCD.
2. Representar números enteros con y sin signo y los números reales o punto flotante en formato IEEE.
3. Mencionar los diferentes registros de un procesador X86, el uso que se les puede dar y los modos de acceso a memoria.
4. Enumerar los flags y su significado.
5. Calcular una dirección efectiva ("effective address")
6. Describir la sintaxis de las instrucciones en assembler.
7. Mencionar los diferentes componentes de un programa en Assembler.
8. Reconocer y usar las funciones del INT 21H
9. Diseñar un programa simple en assembler que compile sin errores.

10. Explicar la acción tomada y programar haciendo uso de las instrucciones transferencia de datos: MOV XCGH PUSH POP PUSHA POPA
11. Explicar la acción tomada y programar haciendo uso de las Seudo instrucciones para definicion de datos, constantes y reservar espacio: DB DW DD EQU RESB RESW y RESD, el operador DUP (o times)
12. Explicar la acción tomada y programar haciendo uso de las Operaciones aritmeticas: ADD SUB INC DEC ADC SBC NEG MUL IMUL DIV IDIV CBW CWD
13. Explicar la acción tomada y programar haciendo uso de las Instrucciones logicas: AND OR XOR NOT TEST
14. Explicar la acción tomada y programar haciendo uso de las Instrucciones de shift y rotate: SHR SHL SAR SAL ROR ROL RCR RCL
15. Explicar la acción tomada y programar haciendo uso de las Instrucciones sobre flags: CMC CLC CLD CLI STC STD STI
16. Explicar la acción tomada y programar haciendo uso de las instrucciones de transferencia de control: CALL RET INT INTO IRET JMP LOOP LOOPZ LOOPNZ y el grupo de las condicionales con y sin signo
17. Diseñar un programa en assembler que compile sin errores que utilicen estructuras de datos
18. Diseñar e interpretar programas cortos en assembler que usen acceso directo a la pantalla o que usen las funciones del INT n21, instrucciones de entrada y salida IN y OUT.
19. Diseñar e interpretar programas cortos en assembler que hagan conversión entre binario y hexadecimal o binario y decimal al entrar o al escribir datos.
20. Describir el uso del debugger para examinar y/o modificar programas y datos
21. Describir la estructura del stack en el FPU y usar las instrucciones en programas sencillos.
22. Programar usando las instrucciones para "strings" (MOVSB, etc) y estructuras de programación iterativas y condicionales.

IV. CONTENIDO

A. Sistemas numéricos y assemblers

1. Representaciones numéricas usadas en assemblers:
 - a. Binario
 - b. Decimal
 - c. Hexadecimal
 - d. Representaciones simbólicas: ASCII y BCD.
2. Números enteros con y sin signo
3. Números reales o punto flotante en formato IEEE.

B. Arquitectura X86

1. Registros de un procesador X86
 - a. uso que se les puede dar
 - b. modos de acceso a memoria.
2. Flags y su significado
3. Direccion efectiva ("effective address")

C. Assembler

1. Sintaxis de las instrucciones en assembler.
2. Componentes de un programa en Assembler.
3. Compilación
4. Funciones del INT 21H

D. Conjunto de Instrucciones

1. Transferencia de datos
 - a. MOV
 - b. XCGH
 - c. PUSH
 - d. POP
 - e. PUSHA
 - f. POPA
2. Seudo instrucciones para definición de datos, constantes y reservar espacio:
 - a. DB, DW , DD
 - b. EQU , RESB , RESW, RESD
 - c. Operador DUP (o times)
3. Operaciones aritméticas:
 - a. ADD, SUB, MUL, DIV
 - b. INC, DEC, ADC, SBC
 - c. NEG, IMUL, IDIV, CBW, CWD
4. Instrucciones lógicas:
 - a. AND
 - b. OR
 - c. XOR
 - d. NOT
 - e. TEST
5. Instrucciones de shift y rotate:
 - a. SHR, SHL,SAR, SAL
 - b. ROR, ROL, RCR, RCL
6. Instrucciones sobre flags:
 - a. CMC, CLC, CLD, CLI
 - b. STC, STD, STI
7. Transferencia de control:
 - a. CALL RET
 - b. INT INTO IRET
 - c. JMP LOOP LOOPZ LOOPNZ
 - d. Grupo de condicionales con y sin signo

E. Estructuras de datos y programación

1. Implantación en programas de:
 - a. Acceso directo a la pantalla
 - b. Que usen las funciones del INT n21.
 - c. Que usen instrucciones de entrada y salida IN y OUT
2. Conversión
 - a. entre binario y hexadecimal

- b. binario y decimal
- c. (al entrar o al escribir datos)
- 3. Uso del debugger para examinar y/o modificar programas y datos
- 4. Estructura del stack en el FPU y las instrucciones del FPU
- 5. Instrucciones para "strings" (MOVSB, etc)
- 6. Estructuras de programación iterativa y condicional.

V. ESTRATEGIAS DE ENSEÑANZA

- A. Conferencias
- B. Presentaciones Computadorizadas
- C. Demostración de programas
- D. Solución de problemas de programación
- E. Tareas

VI. EVALUACIÓN

Tareas	20%
2 Exámenes parciales y Examen final	80%

VIII. BIBLIOGRAFÍA

Libro de Texto

The Art of Assembly Language Programming – (versión para DOS). Randall Hyde - No Starch Press, San Francisco CA, 2003.

Materiales

- 1. Ambiente de programación DosBox - disponible libre de costo en <http://www.dosbox.com>
- 2. <HTTP://SISAB.LCE.ORG/COMP3200>

IX. NOTAS ESPECIALES

A. Servicios Auxiliares o Necesidades Especiales

Todo estudiante que requiera servicios auxiliares o asistencia especial deberá solicitar los mismos al inicio del curso o tan pronto como adquiera conocimiento de que los necesita, mediante el registro correspondiente en la oficina del Consejero Profesional José Rodríguez, Coordinador de Servicios a los estudiantes con Impedimentos, ubicada en el Programa de Orientación Universitaria.

B. Honradez, Fraude y Plagio

La falta de honradez, el fraude, el plagio y cualquier otro comportamiento inadecuado con relación a la labor académica constituyen infracciones mayores sancionadas por el [Reglamento General de Estudiantes](#). Las infracciones mayores, según dispone [Reglamento General de Estudiantes](#), pueden tener como consecuencia la suspensión del estudiante de la Universidad por un tiempo definido mayor de un año o la expulsión permanente de la Universidad, entre otras sanciones.

Recuerde que cualquier tarea del curso debe cumplir con el Reglamento de estudiante Capítulo V, Artículo 1, Sección B.2 que establece "El plagio, la falta de honradez, el fraude, la manipulación o falsificación de datos y cualquier otro comportamiento inapropiado relacionado con la labor académica son contrarios a los principios y normas institucionales y están sujetos a sanciones disciplinarias."

C. Uso de Dispositivos Electrónicos

Se desactivarán los teléfonos celulares y cualquier otro dispositivo electrónico que pudiese interrumpir los procesos de enseñanza y aprendizaje o alterar el ambiente conducente a la excelencia académica. Las situaciones apremiantes serán atendidas, según corresponda. Se prohíbe el manejo de dispositivos electrónicos que permitan acceder, almacenar o enviar datos durante evaluaciones o exámenes.

X. REFERENCIAS

Peter Abel, IBM PC Assembly Language and Programming 3rd Edition, Prentice Hall, 1995

Randall Hyde, The Art of Assembly Language, No Starch Press, San Francisco, CA, 2003

William C. Runnion, Structured programming in assembly language for the IBM PC, PWS Publishing Co., Boston, MA, 1988