UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA FACULTAD SEDE SOGAMOSO

ESCUELA DE INGENIERÍA ELECTRÓNICA LABORATORIO DE MICROPROCESADORES

# OBJETIVOS

PRÁCTICA Nº 3

MANEJO DE LA PANTALLA, EL TECLADO Y EL MOUSE

# III. ELEMENTOS Y EQUIPOS NECESARIOS

* + Identificar el funcionamiento de las interrupciones software que permiten manejar la pantalla, el teclado y el mouse en los microprocesadores de la familia Intel x86.
  + Identificar las diferencias que implica el manejo de la pantalla en modo gráfico y modo texto.
  + Elaborar un programa en lenguaje ensamblador para el manejo de la pantalla, el teclado y el mouse en los procesadores Intel x86 emulados en el programa Emu8086.

# INTRODUCCIÓN

El lenguaje ensamblador para Intel x86 incluye una serie de interrupciones software para el manejo de la pantalla, el teclado y el mouse emulado mediante el software “Emu8086”. Entre estas encontramos la interrupción 10h función 00h (INT 10h/00h) que permite definir el modo de video, la INT 10h/01h que permite seleccionar el modo texto, la INT 10h/0Ch que permite cambiar el color de un pixel en la pantalla, la INT 33h que permite manejar el mouse, entre otras.

Se recomienda consultar la documentación de Emu8086 para conocer más detalles acerca de las interrupciones que son soportadas por este emulador.

Esta práctica de laboratorio está orientada a que el estudiante comprenda como se hace el manejo de las interfaces de pantalla, teclado y mouse en bajo nivel y los conceptos que posibilitan su uso en los procesadores de la familia Intel x86.

* + Software *Emu8086*.
  + Computador con Sistema Operativo Windows XP o Windows 7.

# DESCRIPCIÓN DE LA PRACTICA

En esta sección describe el proceso para desarrollar una visualización grafica con la cual el usuario puede interactuar usando el mouse y teclado. En la Figura 1 se presenta el esquema de una interfaz gráfica de usuario (GUI) desarrollada.



Ingrese texto aquí

Texto de salida

Pasar texto

Cuadrada

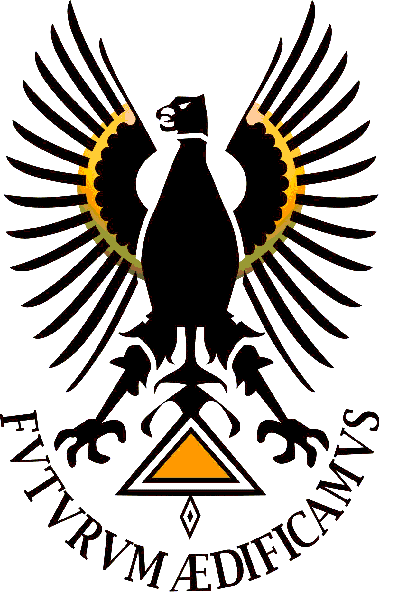
Triangular

Seno

Figura 1. Interfaz gráfica

Como se observa en la Figura 1, se deben pintar varios rectángulos en la pantalla. Por esta razón es necesario crear un procedimiento (llamado BOX) que pinte un rectángulo (pixel a pixel) a partir de un punto o coordenada X,Y en la pantalla, y un tamaño determinado mediante dos valores DX y DY que se deben sumar al punto de inicio (X+DX) y (Y+DY).

Un aspecto importante a tener en cuenta es que la coordenada 0,0 de la pantalla se encuentra en la esquina superior izquierda de la misma.

UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA FACULTAD SEDE SOGAMOSO

ESCUELA DE INGENIERÍA ELECTRÓNICA LABORATORIO DE MICROPROCESADORES

Una vez que se cuenta con el procedimiento BOX se puede proceder a pintar un rectángulo en la pantalla cada vez que sea necesario invocándolo mediante la instrucción CALL y el nombre del procedimiento (CALL BOX). Antes de invocar este procedimiento es necesario suministrar los cuatro parámetros requeridos para su funcionamiento correspondientes al punto de inicio (X, Y) y el tamaño (DX, DY). Usualmente, se emplean los registros de propósito general del procesador (AX, BX, CX, DX) para el paso de parámetros a los procedimientos en lenguaje ensamblador. Por esta razón, se deben realizar una serie de operaciones PUSH para salvar los contenidos de dichos registros en la pila antes invocar el procedimiento y una serie de operaciones POP al final del mismo y antes de la instrucción de retorno (RET) para recuperar sus valores previamente salvados en la pila.

En la parte superior de la interfaz se cuenta con dos rectángulos destinados a ser empleados como contendores de texto. En el primero se escribe un mensaje de texto que se debe pasar al segundo rectángulo cuando se de click en el botón “Pasar texto”. Para la realización de esta tarea se requiere posicionar el cursor dentro del rectángulo dentro del que se esté trabajando.

En la parte inferior de la interfaz se encuentra la zona dedicada a pintar en modo gráfico tres formas de onda (pixel a pixel). Se deben mostrar dos periodos de cada forma de onda dentro del rectángulo ubicado en la parte inferior derecha de la interfaz siempre que se de click dentro del rectángulo respectivo a la forma de onda que se quiere pintar.

El proceso para determinar si se ha dado click sobre un botón requiere que se capture el estado del mouse al momento de producirse un click. Esto se logra usando la interrupción 33h-0003. Esta nos retorna que botón fue presionado y la posición del apuntador al momento de producirse el click.

Con la posición del mouse lo que debe realizase es una comparación entre esta y los limites de los diferentes botones que conforman al interfaz gráfica. Si las coordenadas del mouse está contenida en el área de un botón significa que este se ha presionado.

Cada grupo debe seguir un procedimiento similar al anterior. La interfaz debe estar diseña para permitir a dos personas mantener una partida del clásico juego del triqui. Cada jugador tendrá su propia figura y línea de color que resalte al ganador.



Figura 2: Interfaz gráfica del juego Triqui.

La figura 2 muestra la interfaz que debe desarrollar cada grupo. Los colores, resolución y metodología de juego es desarrollada por cada diseñador así como el sistema de puntaje.

# CONDICIONES DE TRABAJO

El programa debe presentarse identado y con comentarios en los que se explique el funcionamiento de los bloques principales del mismo.

Se debe presentar un informe de laboratorio en el que se consignen los aspectos más importantes que fueron tenidos en cuenta para el desarrollo de la práctica, así como las conclusiones obtenidas de su desarrollo.

# BIBLIOGRAFÍA

[1] Barry B. Bray, the Intel Microprocessors, 8th Edition. Prentice Hall, 2008.

[2] Abel, Peter. Lenguaje ensamblador y programación para PC IBM y compatibles. 3ª Edición, Pearson Education, 1996.

[3] Stallings, William. Organización y arquitectura de computadores. 7ª Edición. Prentice Hall, 2006.

[4] Henessy Jhon & Patterson David. Computer architecture. A quantitative approach. Fourth Edition. Elsevier, 2008.

[5] Rafiquzzaman, M. Fundamental of digital logic and computer design. Wiley Interscience, 2005.

[6] Documentación especializada de Intel: <http://www.intel.la/content/www/xl/es/proce> ssors/architectures-software-developer- manuals.html

[7] Emu8086. Help - Documentation [8] IEEE Transactions on computers.

[9] IEEE Computers Society, Digital library.

[10] Stanford University - Computer Science

[*http://www-cs.stanford.edu/*](http://www-cs.stanford.edu/)

[11] Carnegie Mellon – Schoolof Computer Science

[*http://www.cs.cmu.edu/*](http://www.cs.cmu.edu/)

Columbia University – Computer Science

[*http://www.cs.columbia.edu/*](http://www.cs.columbia.edu/)