

COMPUTADORES II

Código ensamblador – Conecta 4

Índice:

Módulo principal	pág.2
modulo_principal.asm	
Interfaz gráfica	pág.3
imprime_color.asm	
imprimir_turno.asm	
comprobar_ganadores.asm	
Tratamiento de la matriz	pág.4
lee_coloca_matriz.asm	
buscar_4_fichas_seguidas	
recorrer_columnas.asm	
recorrer_filas.asm	
recorrer_diagonales.asm	
Módulos complementarios	pág.7
calcula_posicion.asm	
imprimir_cadena.asm	
limpiar_tablero.asm	
Código fuente	pág. 8

Módulo principal:

Este módulo se encuentra en modulo_principal.asm.

Tiene varias funciones:

- o Contiene al tablero:
 - Al comienzo del código se pueden observar las inclusiones de los globl con los módulos necesarios para la ejecución del programa; seguidos de las etiquetas con los textos de error, título... a imprimir; y por último, se puede ver el tablero bajo el nombre mat.
- o Inicializa las pilas:
 - Lo primero que se hace es inicializar la pila de sistema (para poder llamar a las subrutinas durante la ejecución), y la pila de usuario (para poder guardar variables en dicha pila).
- o Menú:
 - Primero se limpia la pantalla y se imprimen el título y el menú.
 - Después se pide la introducción de un número y comprobamos que sea 1,2 o 3.
 - Si se introduce 1 se salta a la etiqueta instrucciones donde se limpia la pantalla, se imprimen las instrucciones y se espera a que el usuario las lea.
 - Si se introduce 2 se salta a la etiqueta iniciar partida que explicaremos más tarde
 - Por último si se introduce 3, se sale del juego;
- o Iniciar partida:
 - Pone todas las variables a sus valores iniciales y llama a "limpiar_tablero" para que vacíe el tablero; ya que, a partir de la segunda partida, el tablero tendrá algunos huecos con fichas.
 - Tras esto comienza el bucle "nueva_partida", el cual se recorre una vez por turno durante toda la partida.
 - Todo lo descrito a continuación es lo que se hace en "nueva_partida".
 - Primero se limpia la pantalla y se imprime el título, tras lo cual se imprime el tablero y el turno del jugador.
 - Luego se llama a "lee_coloca_matriz" (del que hablaremos luego), en donde el jugador decide donde colocar una ficha y se introduce en el tablero.
 - Tras esto comprobamos si hay 4 fichas seguidas y vemos si hay ganador, en cuyo caso se termina la partida y se le notifica al usuario.
 - Por último, se comprueba si se ha llegado al turno 42 lo que indica que el tablero está lleno, en cuyo caso también se finaliza la partida.

Interfaz gráfica:

- o Estos son los módulos dedicados a crear la interfaz gráfica para el jugador. No obstante, como veremos, tienen otras funciones.
- o imprime_color.asm
 - Este módulo imprime el tablero a color según el símbolo
 - Recibe por el registro X la dirección del primer elemento del tablero.
 - Para imprimir el tablero recorre el tablero elemento a elemento (hay 80 elementos).
 - Cada elemento es enviado a “color_selection” donde si es una X se saca en rojo, si es una O se saca en verde, si es un asterisco en azul y en cualquier otro caso en blanco.
- o imprime_turno.asm
 - Este módulo indica el turno del jugador y maneja los incrementos y decrementos de la variable turno.
 - Recibe por el registro a la variable turno.
 - Este módulo actúa según el valor de turno: si es 1 significa que es el turno de J1(X), y si es 2 que es el turno de J2(O);
 - Tras esto:
 - Si es el turno de J1 incrementa turno (en el caso contrario decrementa turno) para que sea 2 y en el siguiente ciclo sea el turno de J2.
 - Muestra por pantalla de quien es el turno
 - Por último, se devuelve el valor de turno por el registro A al módulo principal que es desde donde se llama esta subrutina.
- o comprobar_ganadores.asm
 - Este módulo imprime por pantalla quién ha ganado.
 - Recibe en el registro A el ganador que está codificado de la siguiente forma:
 - o (0): empate
 - o (1): ha ganado J1
 - o (2): ha ganado J2
 - o (3): No ha ganado nadie aun
 - En el caso de recibir 0,1 o 2 imprime un mensaje indicando el ganador.
 - En el caso de recibir 3 se vuelve al módulo llamador sin actuar.

Tratamiento de la matriz:

- o Estos módulos son aquellos destinados a tratar el tablero directamente; y realizan acciones como insertar fichas, buscar 4 fichas seguidas...
- o lee_coloca_matriz.asm
 - Esta subrutina sirve para introducir una ficha en el tablero.
 - Para ello recibe por el registro X la dirección del primer elemento de la matriz y por el registro A el turno del jugador, para poder colocar la ficha correcta.
 - Para explicar este módulo lo dividiremos en partes:
 - o Lectura de la columna:
 - Primero se le pide al jugador que introduzca un número por el teclado entre 1 y 7.
 - Tras esto convertimos el número de ASCII a decimal y lo adaptamos a nuestro tablero (le sumamos 1).
 - Se comprueba que ese número esté en el recién mencionado intervalo.
 - En el caso de que el número verifique lo anterior, se continúa; y en el caso contrario, se muestra un error y se repite la introducción.
 - o Colocación de la ficha y comprobación de columna:
 - En esta parte se presenta un bucle for en el que tenemos como variable a aumentar, el índice de la fila.
 - Recorremos la columna introducida por el usuario de abajo a arriba y vamos comparando el contenido de las celdas.
 - Si el contenido es un espacio, la celda está vacía por lo que se puede introducir en dicha posición la ficha. En caso contrario, continuamos hasta encontrar un lugar libre en la columna para introducir la ficha, para lo cual vamos a "posicion_ocupada", donde realizamos los incrementos necesarios.
 - En el caso de que se llegue a la posición 0, y esté llena, se indica un error y vuelve a pedir la introducción de la columna, porque eso es señal de que está llena.
 - Cabe destacar que según la variable turno (recibida por el registro A), se puede saber qué tipo de ficha (X o O) introducir; pero en este caso turno si es el turno de J1 turno es igual a 2 y si es el turno de J2 turno es igual a 1, ya que en el módulo principal primero se llama a "mostrar_turno", que es la subrutina que maneja el valor de dicha variable.
- o buscar_4_fichas_seguidas
 - Este módulo recopila la llamada a las tres subrutinas que realizan los

recorridos de la matriz buscando fichas.

- Recibe por el registro D la dirección del primer elemento de la matriz, para poder pasarles este valor a las tres subrutinas que llama.
- Básicamente llama a los recorridos de filas, columnas y diagonales; y si detecta que hay un ganador (registro A j= 3), finaliza la subrutina y se vuelve al módulo llamador.

o recorrer_columnas.asm

- Esta subrutina realiza el recorrido del tablero por columnas, esto es debido a que se desea buscar 4 fichas seguidas del mismo tipo en alguna columna.
- Recibe por el registro D la dirección del primer elemento de la matriz para poder recorrerla elemento a elemento mediante el uso de dos for anidados, con X e Y, usados para guardar los índices de una matriz a la que se podría acceder a un elemento suyo de la forma mat[X][Y].
 - o El for externo va recorriendo las columnas
 - o El for interno va recorriendo todas las posiciones de una fila para encontrar 4 fichas seguidas.
- Este módulo se apoya en “calcula_posicion” para funcionar, ya que en ensamblador es necesario calcular la posición de los elementos de la matriz para poder acceder a ellos.
- Esta subrutina hace uso de contadores de fichas que van seguidas, kX y kO:
 - o se ponen a 0 si se encuentra un espacio (32 en ASCII).
 - o se incrementa kX y se pone a 0 kO si se encuentra una X
 - o se incrementa kO y se pone a 0 kX si se encuentra una O
- En el caso de que uno de los contadores tenga el valor 4 al final del recorrido de una fila; se devuelve por el registro A el ganador; de forma codificada (1=J1 y 2=J2).

o recorrer_filas.asm

- El funcionamiento de este módulo es similar al módulo de “recorrer_columnas”, ya que funcionan igual, cambiando que este recorre filas, por lo que el bucle externo va recorriendo fila por fila; y el bucle interno va recorriendo todos los elementos de una fila.

o recorrer_diagonales.asm

- Esta subrutina comprueba las posibles diagonales del tablero en busca de 4 fichas seguidas del mismo tipo en una misma diagonal.
- Recibe por el registro D la dirección del primer elemento de la matriz para poder recorrerla elemento a elemento de cada una de las posibles diagonales mediante el uso de dos while y un for dentro de cada bucle, con X e Y, usados para guardar los índices de una matriz a la que se podría acceder a un elemento suyo de la forma mat[X][Y].
 - o El while se utiliza con un flag, cont, para asegurarnos que recorre todas las posibles diagonales y ninguna que no sea de menos de cuatro elementos. (Solo hay 6 diagonales en cada

- sentido que verifican estas dos condiciones)
- o El for interno va recorriendo todas las posiciones de una diagonal para encontrar 4 fichas seguidas.
 - Este módulo hace uso, al igual que los anteriores recorridos, de “calcula_posicion” junto a los contadores de fichas, para comprobar si hay cuatro fichas del mismo tipo seguidas, y el mismo sistema para declarar al ganador.

Módulos complementarios:

- o Aquí se tratarán los módulos complementarios; es decir aquellos que realizan pequeñas tareas imprescindibles para el funcionamiento del programa, pero son de carácter auxiliar.

- o calcula_posicion.asm

- Esta subrutina calcula la posición de un elemento de la matriz
- Para ello recibe en el registro D la dirección del primer elemento de la matriz y en los registros X e Y recibe los índices para acceder a una matriz de la forma $mat[X][Y]$.
- Al calcular la posición hace $mat + X * nCol + Y$
- Devuelve la posición obtenida en el registro D
- Cabe destacar que para implementar este módulo nos fijamos en un tipo de matrices que vimos en programación II, las cuales consistían en matrices de un índice y el acceso a los elementos de dichas matrices se realizaba mediante este sistema.

- o imprimir_cadena.asm

- Este módulo imprime cadenas de caracteres
- Es muy parecido al desarrollado en clase
- Recibe una dirección de memoria en X y va avanzando incrementando X, a la vez que presenta por pantalla el contenido de las celdas de memoria a las que se accede con el valor almacenado en X.
- Se detiene cuando alcanza el carácter nulo o $\backslash 0$.

- o limpiar_tablero.asm

- Este módulo, como indica su nombre, limpia el tablero.
- Para ello recibe por el registro D la dirección del primer elemento del tablero.
- Hace un recorrido poniendo espacios en todas las casillas del tablero, excepto a las que pertenecen a la interfaz gráfica como pueden ser los asteriscos, los saltos de línea o los índices de las columnas.
- Este recorrido se realiza mediante dos bucles for anidados. El bucle externo controla el recorrido de las filas y el bucle interno el de las columnas.
- Dentro del bucle interno lo que se hace es calcular la posición de la casilla a resetear, y escribir en ella un espacio.

Código: