Universidad del Valle de Guatemala

Proyecto 3: HerbKB

Desarrollado por: Carlos López Camey y Byron Morales

19 de Octubre del 2009

Universidad del Valle de Guatemala
Facultad De Ingeniería
Programación en Assembler - CC3005
Proyecto 2 - Gasculator

Tabla de Contenidos

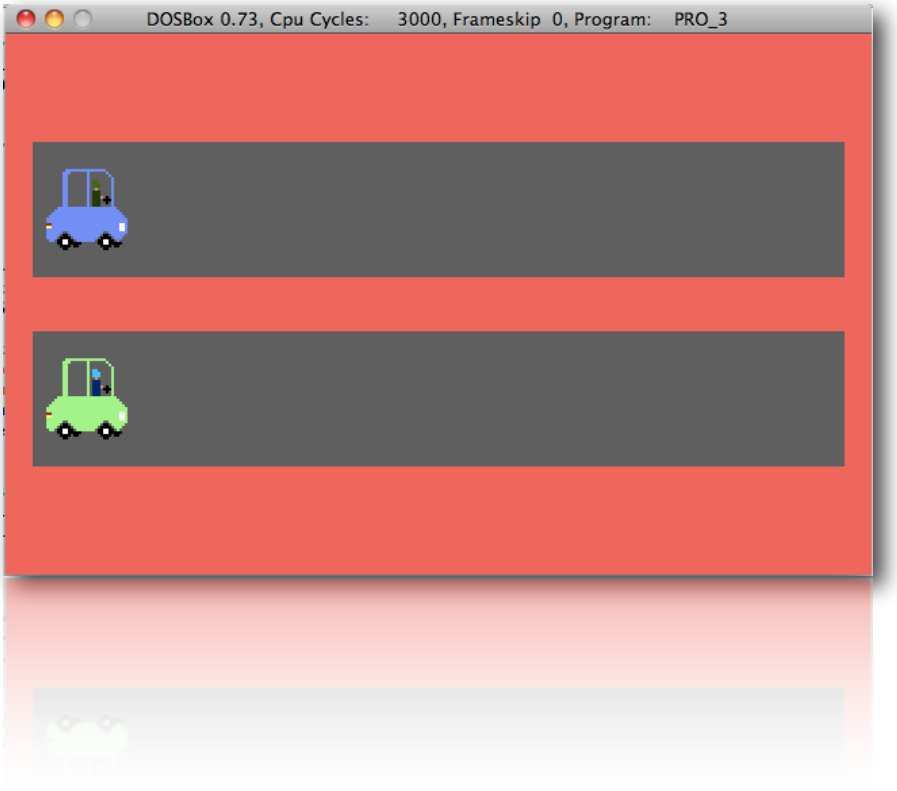
HerbKB

Introducción

HerkKB es un programa que realiza la simulación de 3 procesos en multi-tarea utilizando la interrupción 08H del cronómetro del procesador para el cambio de cada uno.

Existen dos diferentes vehículos, los cuales realizan una competencia caminando en forma horizontal y empezando desde el lado izquierdo. Cada uno de los vehículos tiene dos formas de avanzar, con una tecla diferente cada uno y con la interrupción 08H del cronómetro.

El otro proceso es, una melodía, la cual suena mientras la carrera está en curso, hasta que esta termine.



Diseño del programa

Variables (Etiquetas) a utilizar

Nombre	Descripción
Vehiculo1	Variable que representa el dibujo del vehículo 1. 30 pixeles de ancho y 30 pixeles de alto.
Vehiculo2	Variable que representa el dibujo del vehículo 2. Igual que el vehículo 1, este es de 30x30px.
Estado	Representa el estado del juego. Si este vale 1, el jugador avanza, si vale 2, significa que abandonó el juego, 3 que ganó el primer jugador y 4 que ganó el segundo jugador.
POS1	Representa la posición en el eje X del vehiculo 1 (el de arriba).
POS2	Representa la posición en el eje X del vehiculo 2 (el de abajo).
ABANDONADO, GANO1, GANO2	Mensajes para el usuario.
Freq	Frecuencia en escala de do.
RelojViejo	Guarda direcciones antiguas para poder ser restauradas
TecladoViejo	Guarda direcciones antiguas para poder ser restauradas
VectorBP	Vector o arreglo, de tres direcciones de memoria (tamaño double-word), una posición para cada proceso.
VectorSP	Vector o arreglo de control circular.
Contador	Contador de procesos, cual se está realizando en el momento. Si este es 0 o 1, incrementar el valor y realizar el “próximo procedimiento” si este es 2, el próximo sería el 0.
Tecla	Bandera para terminar el programa

Nombre	Descripción
Hubo_tecla	Bandera que avisa para revisar tecla.

Procedimientos a utilizar

Nombre del Procedimiento	Descripción
Juego	Procedimiento principal. Inicializa el segmento de datos, configura la pantalla (modo gráfico), llama a **CONFIGURAR_JUEGO, PILAS, CAMBIAR_INTERRUPCIONES, CARRO1** y **REGRESO_INTERRUPCIONES.** Luego regresa la configuración de pantalla y muestra el resultado al usuario.
CONFIGURAR_JUEGO	Configura la pantalla de juego, llama a **PINTAR** (que pinta el fondo), **PINTAR_CARRO1** y **PINTAR_CARRO2**.
PINTAR	Pinta el fondo y la pista (carretera) de cada vehiculo.
PINTAR_CARRO1	Dibuja el carro 1, en pixeles expresado por la etiqueta **VEHICULO1**, empezando en la posición X que indica la etiqueta **POS1.**
PINTAR_CARRO2	Análogo a PINTAR_CARRO1.
PILAS	Crea las pilas (stacks) artificiales.
CAMBIAR_INTERRUPCIONES	Metodo que cambia la interrupción del teclado y la del cronómetro.
RELOJ_MODIFICADO	El nuevo reloj.
REGRESO_INTERRUPCIONES	Cambia la interrupción del teclado y cronometro a la configuración inicial utilizando las etiquetas **RelojViejo** y **TecladoViejo.**
SONIDO	Hace sonar la bocina con la melodía.

Melodía

La computadora genera sonido mediante una bocina magnética incrustada en la placa madre, que está conectada a los puertos 42H, 43H y 61H. Para operar la bocina, los pasos a seguir fueron los siguientes.[2]

1. Obtener el estatus del puerto 61H y guardarlo.
2. Mandar a travez del puerto 61H, los bits 0 y 1 encendidos para encender la bocina. El puerto activará el periferico PPI (*Programmable Peripherical Interface)*
3. Para apagar la bocina, mandar los bits 0 y 1 apagados (en 0) a travez del mismo puerto, el 61H.

Multi-tarea

La simulación de multi tarea (tiempo compartido) consistió básicamente en 6 pasos:

1. Inicializar el área de datos
2. Indicar el modo de video a utilizar (en este caso gráfico)
3. Preparar las pilas i.e. procedimientos que queremos ejecutar simultáneamente.
4. Programar las nuevas interrupciones de Cronómetro y teclado.
5. Llamar a proceso al primer proceso con CALL
6. Regresar a las interrupciones normales de cronómetro y teclado.

Bibliografía

[1] *Creative Commons, GNU Lesser General Public Licence,* http://creativecommons.org/licenses/LGPL/

[2] Peter Abel, IBM *Asse11mbly Language and Programming*, Quinta Edición, Prentice Hall.