



Universidad Simón Bolívar
Depto. de Computación y T.I.
Organización del Computador (CI-3815)
Enero-Marzo 2020

Proyecto # 1 (13%)

Manejo de interrupciones

Objetivo

Familiarizarse con los principios básicos de interrupciones y de operaciones de E/S (entrada/salida) asignada a memoria.

Objetivos Específicos

1. Implementar manejadores para las interrupciones de teclado (Receiver) y de reloj (Timer).
2. Simular la implementación de una consola mapeada a memoria e
3. Implementar el juego "Snake"

Descripción del juego Snake

Este juego ha existido por mucho tiempo pero es recientemente que ha ganado popularidad al haber sido implementado por Nokia para sus teléfonos celulares.

Se define una ventana o tablero de cierta dimensión, digamos NxM, que estará enmarcada por paredes que determinan el fin del tablero. El juego empieza con una culebra de tamaño fijo posicionada en algún punto de tablero. Por cada unidad de tiempo la culebra avanza siempre en la dirección que apunta su cabeza. Se le puede hacer cambiar su dirección haciéndola girar a su derecha o izquierda (no se puede retroceder).

En el tablero van apareciendo manzanas, una a la vez. Éstas son el objetivo de la culebra, es decir, el jugador debe hacer que la culebra se “coma” las manzanas. Al comer una manzana, la culebra crece en una unidad de longitud. Cuando la cabeza de la culebra choca con una pared o con algún segmento de sí misma, el juego acaba.

Cuando se acumula un número determinado de manzanas comidas, se pasa al siguiente nivel en el cual se aumenta la velocidad de movimiento de la culebra.

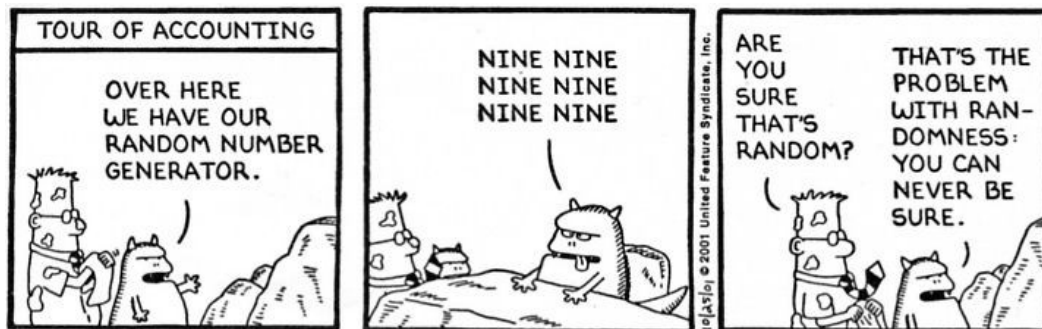
Definiciones

Consola virtual simMars:

Una consola simMars es un dispositivo tipo monitor que puede ser conectado a un computador Mars. La interacción con este dispositivo se realiza a través de la memoria (DMA - Direct Memory Access). La memoria asignada a este dispositivo es un bloque contiguo de bytes que pueden verse como una matriz de dos dimensiones (llamémosla $MAT_{n \times m}$) almacenada por filas a partir de la dirección D de la memoria. Este dispositivo tiene una resolución de N líneas y M columnas y un refrescamiento de S veces por segundo. La fila i de la matriz $MAT_{n \times m}$ corresponderá con la fila i de la consola, de la misma forma sucede con las columnas.

En otras palabras, colocar el carácter X en la posición asociada a la intersección de la fila 4 con columna 20 implica colocar el código ASCII del carácter X en el byte de memoria correspondiente (en la matriz sería $MAT[4,20]$). El mismo procedimiento es válido para determinar el carácter que se encuentra desplegado en un punto cualquiera de la pantalla.

Número pseudoaleatorio:



Un número pseudo-aleatorio es un número generado en un proceso que parece producir números al azar, pero no lo hace realmente. Las secuencias de números pseudo-aleatorios no muestran ningún patrón o regularidad aparente desde un punto de vista estadístico, a pesar de haber sido generadas por un algoritmo, en el que las mismas condiciones iniciales producen siempre el mismo resultado.

Los mecanismos de generación de números aleatorios que se utilizan en la mayoría de los sistemas informáticos son en realidad procesos pseudoaleatorios. Muchos de estos sistemas se valen de elementos del sistema de operación o el hardware, como el reloj. En otras palabras, un generador de números pseudoaleatorios básico puede hacer uso del valor obtenido del reloj del sistema. Y con éste hacer alguna operación como el módulo.

Enunciado del Proyecto

Actividad 1: Simulador de consola

Implemente un simulador del dispositivo simMars previamente definido. Asuma que para esta implementación los siguientes valores:

- $D = 0x50000000$
- $N = 32$ y $M = 42$
- $S = 2$

Cuide que su implementación permita modificar los valores D , N , M y S de forma muy sencilla. Este mismo tratamiento debe seguirse para los valores V y K definidos más adelante.

Nota: modificar más de una línea de código, para cambiar uno de esos valores, no es considerado sencillo.

Actividades 2 y 3: Manejador de Teclado (Receiver) e implementación del juego

Implemente el juego *Snake* y un manejador para la interrupción de teclado que cumpla la definición de las reglas establecidas en el juego. La implementación de la visualización del juego debe hacerse usando el dispositivo consola previamente definido.

Reglas modo básico:

- Se define un tablero del tamaño de la consola mapeada a memoria implementada, cada uno de los bordes de este tablero corresponderá a una de las 4 paredes que lo limita. Cada pared es de un carácter de espesor.
- No hay obstáculos salvo las paredes que bordean el tablero.
- Siempre habrá una y sólo una manzana sobre el tablero a la vez. Cuando la culebra se coma una manzana, debe generarse otra en algún punto del tablero que no coincida con el cuerpo de la culebra, este nuevo punto debe ser pseudoaleatorio.
- La culebra empieza con un tamaño de 5 (es decir, que ocupa 5 caracteres). Cada vez que ésta se coma una manzana, debe aumentar su tamaño en 1 unidad.
- La culebra sólo puede girar a su derecha o a su izquierda. Comandos que impliquen que siga de frente o que retroceda deben ser ignorados.
- Si la culebra choca con una pared o consigo misma, termina el juego.
- En cualquier momento puede presionarse la tecla 'p' para pausar el juego, o continuarlo si ya está pausado.
- En cualquier momento puede presionarse la tecla 'q' para terminar el juego.

- Cuando se acabe el juego, ya sea por que se pierde o se abandone, se debe imprimir la cantidad de manzanas que logró comer la culebra y en cuanto tiempo.
- La culebra tiene una velocidad constante de V espacios por segundo, y sólo puede moverse horizontal o verticalmente, nunca en diagonal.
- Comandos:
 - o 'w': Equivalente a la flecha 'arriba', cambia (si puede) la dirección de la culebra para que avance hacia arriba.
 - o 'a': Equivalente a la flecha 'izquierda', cambia (si puede) la dirección de la culebra para que avance hacia la izquierda.
 - o 's': Equivalente a la flecha 'abajo', cambia (si puede) la dirección de la culebra para que avance hacia abajo.
 - o 'd': Equivalente a la flecha 'derecha', cambia (si puede) la dirección de la culebra para que avance hacia la derecha.
 - o 'p': Pausa el juego o lo continua si ya está pausado.
 - o 'q': Abandona el juego.

La visualización de los elementos del juego:

- La culebra:
 - o Cabeza: Carácter O (mayúscula)
 - o Cuerpo: Carácter * (minúscula)
- Las paredes: Carácter #
- La manzana: Carácter \$

Un ejemplo de un “pantallazo” del juego

```
#####
#                                     #
#                                     #
#                                     #
#                                     #
#          $                         #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#          O                         #
#          *                         #
#          *                         #
#          *      *                  #
#          * * * *                   #
#                                     #
#                                     #
#####
```

Reglas modo extendido (Puntos extras) :

- Se define la posibilidad de tener múltiples niveles de dificultad en el juego, por lo menos más de un nivel.
- Se determina que para pasar del nivel i al nivel $i+1$, de debe haber comido un número K de manzanas en el nivel i .
- La dificultad de un nivel (que no sea el nivel básico) vendrán dada por la incorporación de:
 - o obstáculos, específicamente paredes internas.
 - o pasadizos, por ejemplo uno que genere un paso entre la pared izquierda con la derecha.

Este modo extendido tendrá una ponderación adicional al nivel básico, este último debe funcionar perfectamente para poder optar a los puntos adicionales del nivel extendido.

Debe indicar en que consiste el modo extendido implementado en un reporte de **una** página, note que esto no es un informe! Es sólo un reporte.

Ejemplos de “*pantallazos*” del juego en un posible modo extendido

Paredes internas

```
#####
#                                           #
#                                           #
#                                           #
#           #                               #
#           #                               #
#       $   #                               #
#           #                               #
#           #                               #
#           #                               #
#           #                               #
#           #                               #
#       0    #                               #
#       *    #                               #
#       *    #                               #
#       *      *   #                       #
#       * * * *   #                       #
#                                           #
#####
```

La culebra no debe tocar la pared interna o se considera que hubo una colisión y el juego acaba, puede haber otros patrones de paredes internas.

Pasadizos

```
#####
#                                     #
```

```

#                                     #
#                                     #
#                                     #
#          $                          #
#                                     #

**0                                ***
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#####

```

La culebra entra por un extremo del pasadizo ubicado en la pared de la derecha y va apareciendo por el otro extremo del pasadizo ubicado en la pared de la izquierda.

NOTAS IMPORTANTES:

1. Ud. debe usar la versión de Mars publicada con este enunciado (*MarsTimer.jar*), si tiene algún inconveniente con ésta, debe notificarlo a su profesor de inmediato.
2. No seguir las especificaciones aquí dadas podría resultar en la no corrección de su proyecto.
3. Si tiene preguntas, se les recomienda aclararlas con tiempo.
4. Ud debe implementar los manejadores de interrupción en el archivo `exception.s`
5. El simulador del dispositivo definido utilizará la consola de Mars para desplegarse.
6. El programa principal que se debe correr puede ser cualquiera, y la labor que éste lleve a cabo debe ejecutarse “normalmente” (salvo la impresión por consola que será compartida con el dispositivo `simMars`). En particular Ud puede probar su proyecto, por lo menos inicialmente, con un `main` que contenga un ciclo infinito.

Recomendaciones

1. Comience con tiempo.
2. Estructure bien su código.
3. Trabaje en forma ordenada e incremental
4. Pruebe que cada una de sus funciones funciona correctamente
5. **Respete** la especificación dada en el presente enunciado: No cambie el nombre de las rutinas NI las definiciones aquí establecidas.

6. Tenga presente que es mejor tener más funciones pequeñas que menos funciones largas.

Entrega

El proyecto debe ser entregado hasta las 9:00 am del 20 de Marzo. Debe entregar:

1. Entregar el código **impreso** y **apropiadamente** documentado de su implementación, myexception.s
2. Enviar la versión electrónica al correo de su profesor con asunto “Proyecto 2 – Carnet1-Carnet2.
3. Note que las versiones impresa y digital deben corresponder.