

*Instituto Tecnológico de Costa Rica
Centro Académico de Alajuela
Ingeniería en Computación
IC-3101. Arquitectura de Computadoras
Prof. Emmanuel Ramírez Segura
Semestre II – 2022
Valor: 20%*

Fecha de Asignación: 12/10/2022

Fecha de Entrega: 16/11/2022

Proyecto #2:
Programación del Juego de Sudoku en NASM

Objetivo General

Analizar el conocimiento sobre una arquitectura de computadoras actual para la programación desde una perspectiva de bajo nivel, en este caso la Arquitectura Intel utilizando ensamblador NASM.

Objetivos Específicos

1. Analizar diferentes modos de direccionamientos para la solución de un problema particular en ensamblador.
2. Diseñar las macros y los procedimientos como mecanismos para modularizar las instrucciones.
3. Aprender el uso de Interrupciones de linux para el manejo de dispositivos de Entrada/Salida.
4. Programar en ensamblador el juego de “Sudoku” para reforzar la teoría vista en clase.

Sobre el Juego de Sudoku

El Sudoku fue inventado por Howard Garns, un inventor de rompecabezas independiente de Connersville, Indiana, Estados Unidos en 1979, cuando se publicó en "Dell Pencil Puzzles y la revista Word Games". El rompecabezas se conocía como "Ubicar el Número", ya que implicaba colocar números individuales en espacios vacíos en una cuadrícula de 9 x 9.

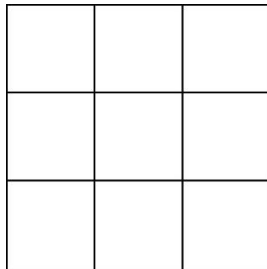
Luego en Japón en 1984, apareció con el nombre de "SUDOKU" que es la abreviatura de una palabra más larga en japonés que es "Sūji wa dokushin ni kagiru", que significa "**los dígitos están limitados a una ocurrencia**".

En la actualidad, es un juego de bastante popularidad y con una gran cantidad de variantes, no solo en la cantidad de cuadrículas que puede contener, sino también en los elementos que pueden ser utilizados en la cuadrícula donde no solo con números sino también pueden ser colores, letras, figuras o símbolos. Evidentemente para estos casos donde no son números que se ingresan a la cuadrícula, las reglas varían.

El presente proyecto programado en ensamblador será desarrollar un SUDOKU, que contemple las reglas básicas del mismo, tal y como se describen a continuación:

Reglas del Sudoku a programar

1. Se utilizará una cuadrícula de 3x3 espacios.



2. Los espacios serán llenados exclusivamente con números, del 1 al 9, es decir el conjunto numérico {1,2,3,4,5,6,7,8,9}.
3. Los números no se repetirán en la cuadrícula, es decir, solo habrá una ocurrencia de un mismo número.
4. Para ganar el SUDOKU se debe tener una sumatoria igual a 15 en las verticales, horizontales y diagonales.

Ejemplo de Formaciones de Gane (entre otras):

4	9	2
3	5	7
8	1	6

2	9	4
7	5	3
6	1	8

6	1	8
7	5	3
2	9	4

5. Cada vez que el usuario juegue, se le presentará una disposición aleatoria de números en la cuadrícula, de tal manera únicamente se presenten cuatro espacios llenos y el resto de espacios (cinco restantes) deberá llenarlos el usuario para completar el SUDOKU.
6. El usuario tendrá un máximo de 2 minutos o su equivalente 120 segundos para ganar, caso contrario, pierde el juego.

Sobre los requerimientos a programar

1. El dibujo de la cuadrícula no será necesario. Idealmente se prefiere.
2. El usuario ingresará los números mediante un sistema de coordenadas (fila, columna).
3. El programa deberá validar:
 - a. Si el número que ingresa el usuario está repetido, en dicho caso despliega un mensaje indicándolo y no permite el ingreso.
 - b. Si la coordenada a ingresar no es válida (por ejemplo 4,3) deberá indicarlo mediante un mensaje.
 - c. Si la coordenada ingresada no está vacía, deberá indicarlo mediante un mensaje y no permitirá sobrescribir el número.
 - d. Si ingresa todos los números en la cuadrícula y la sumatoria no es 15 en las verticales, horizontales y diagonales, deberá desplegar un mensaje indicándolo. Evidentemente siempre que el usuario no haya agotado el tiempo del juego.
 - e. Si ya el tiempo superó los 2 minutos (120 segundos), el programa desplegará el mensaje de fin de juego y el usuario no podrá ingresar más coordenadas. En este caso, el programa dará la opción de “Jugar de Nuevo” o “Salir”.
4. El programa deberá desplegar el tiempo de juego transcurrido en segundos, para esto debe entenderse que una vez iniciado el juego el tiempo transcurrirá sin detenerse. Idealmente el tiempo debería correr en tiempo real y visualizarse en pantalla, pero alternativamente el mismo deberá visualizarle posterior al ingreso de cada coordenada.

Requerimientos de diseño

1. Se trabajará con una arquitectura de procesador Intel x86 (la cual es compatible con procesadores AMD).
2. Se utilizará el ensamblador NASM (Netwide Assembler) sobre Linux.
3. Se trabajará sobre la arquitectura de la máquina virtual en Ubuntu (de preferencia la versión DESKTOP 20.04.2.0 LTS).

La versión de UBUNTU se puede descargar del siguiente enlace:

<https://ubuntu.com/download/desktop/thank-you?version=20.04.2.0&architecture=amd64>

4. Se debe de documentar cada una de las rutinas programadas como parte del proyecto, específicamente el comentario previo a cada subrutina o macro debe ir como sigue:

Objetivo de la subrutina/macro:

5. El juego resultante será un archivo ejecutable de Linux.
6. El juego debe ser programado y desarrollado por cada grupo, cualquier copia de código se considerará fraude y se descalificará la entrega, adicionalmente se aplicará lo que establece el reglamento.
7. El juego se manejará mediante el siguiente flujo (ver página siguiente):

La Pantalla Principal, presenta al usuario las dos opciones de juego, donde con la opción 1 se puede iniciar un juego y la opción 2 permite salir del programa.

Pantalla Principal

Bienvenido al Juego de SUDOKU

Seleccione una opción:

1. Iniciar Juego
2. Salir

Si se elige la opción: 2 – Salir

Si se elige la opción: 1 – Iniciar Juego

Se deberá desplegar una matriz de 3 filas por 3 columnas con cuatro espacios llenos, tal como se aprecia a continuación:

Bienvenido al Juego de SUDOKU

	0	1	2
0	[]	[]	[]
1	[]	[]	[]
2	[]	[]	[]

Coordenadas: fila, columna, numero

Tiempo Restante: 120s

Mensaje: XXXXXXXXXXXXXXXXXXXXXXXX

Salir ESC

De la pantalla anterior se desprenderá la siguiente información:

Las letras resaltadas en color amarillo indican posibles valores del juego.

Por ejemplo (iniciando de arriba hacia abajo):

Coordenadas: Este campo es para ingresar la fila, la columna y el valor a ingresar. Pueden ser separados por comas o espacios. Por ejemplo, si el usuario ingresa 0,1,4, significará que en la fila 0 y columna 1 ingresará el número 4.

Tiempo Restante: Es un contador que inicia el conteo en 120 segundos (2 minutos) y no se detiene hasta llegar a 0. Si llega a 0 y el usuario no completa el SUDOKU entonces automáticamente pierde el juego.

Mensaje: Mensaje es un campo utilizado para desplegar errores que comete el usuario, acá se desplegarán las validaciones a realizar como parte de este juego y que se mencionan en los requerimientos a programar.

Salir ESC: Es la tecla para salirse hacia el menú principal, se puede ingresar en cualquier momento.

➤ Aspectos Evaluativos

La evaluación de este trabajo consta de tres aspectos:

1. Un informe o documento digital, cuyo porcentaje será de un 10%.
Importante: Se utilizará letra Arial tamaño 12 como parte del formato del informe.
2. El código fuente del programa, cuyo porcentaje será de un 70%.
Importante: El programa principal se llamará: **sudoku.asm** y el archivo externo para macros se llamará **macros.asm**.
3. Una defensa del proyecto ante el profesor el día de la entrega: 16/11/2022, cuyo porcentaje será de un 20%.
Importante: En la defensa deberán estar todos los integrantes del proyecto, caso contrario, no se calificará este rubro.

Con respecto a la evaluación del informe o documento digital se solicita:

- A. PORTADA (la misma tendrá el formato del Anexo 1 (observarlo al final del documento)).
- B. INDICE (debe tener enlaces referenciados a las secciones siguientes con hipervínculos).
- C. OBJETIVOS (los objetivos General y Específicos del presente trabajo).
- D. FLUJO DEL PROGRAMA (en esta sección deberán colocarse pantallazos que evidencien la ejecución del juego. Para tal fin, debe mostrar el paso a paso del flujo del programa (incluso a nivel de consola o terminal desde que ensambla el programa), imagine que está haciendo el manual de un juego para que contemple el detalle necesario). Tome los screenshots que considere necesarios para complementar su documentación. Importante: Muestre acá (con los screenshots) que su proyecto cumple con todas las validaciones solicitadas como parte de los requerimientos.
- E. RESUMEN DE LOGRADO: Colocando el siguiente recuadro en el documento digital y colocando una X en la casilla correspondiente autoevalúense. Colocando si el requerimiento ha sido **Logrado**, **Logrado Parcialmente** o **No Logrado**. Este cuadro determinará la nota del código fuente de su programa.

Requerimientos	***Logrado (100%)	**Logrado Parcialmente (40%)	*No Logrado (0%)
1. Se implementa el flujo del programa. solicitado (menús y opciones) (5%)			
2. Se documenta el código fuente (5%)			
3. Se cumple con la siguiente validación (7%): Si el número que ingresa el usuario está repetido, en dicho caso despliega un mensaje indicándolo.			
4. Se cumple con la siguiente validación (7%): Si la coordenada a ingresar no			

es válida (por ejemplo 4,3) deberá indicarlo mediante un mensaje.			
5. Se cumple con la siguiente validación (7%): Si la coordenada ingresada no está vacía, deberá indicarlo mediante un mensaje.			
6. Se cumple con la siguiente validación (7%): Si ingresa todos los números en la cuadrícula y la sumatoria no es 15 en las verticales, horizontales y diagonales, deberá desplegar un mensaje indicándolo.			
7. Se cumple con la siguiente validación (7%): Si ya el tiempo superó los 2 minutos (120 segundos), el programa desplegará el mensaje de fin de juego y el usuario no podrá ingresar más coordenadas. En este caso, el programa dará la opción de “Jugar de Nuevo” o “Salir”.			
8. Se utiliza(n) archivo(s) adicionales para almacenar Macros en archivos externos (5%).			
9. El juego se ejecuta sin complicaciones, errores, problemas u observaciones detectadas (50%).			
10. Extras (entrega anticipada 5%-10%)			
TOTAL (campo llenado por el profesor):			

Notas:

***No Logrado:**

- La ejecución del programa o característica solicitada no funciona o funciona pero no hace nada relativo a lo solicitado.

- Si no se logró, colocar en el mismo campo una descripción o comentario explicando qué no se logró y por qué.

***Logrado Parcialmente**, significa que el usuario:

- Puede jugar, pero se detecta algún error durante la ejecución.
- Hay un incumplimiento o inconsistencia clara en el programa.
- Se realizó una implementación parcial de lo solicitado, pero hay ejecución que permite al usuario jugar ciertos escenarios.

*****Logrado:**

-El programa funciona 100% según lo descrito por el profesor, eso implica que no hay errores de ejecución ni problemas al jugar.

Extras:

1. Se calificará un 5% extra del valor del proyecto si el mismo se entrega 1 semana antes de la fecha de entrega. Para optar por esta porcentaje, el proyecto deberá estar completamente terminado y no tener ningún pendiente de acuerdo con lo solicitado.
2. Se calificará un 10% extra del valor del proyecto si el mismo se entrega 2 semanas antes de la fecha de entrega. Para optar por esta porcentaje, el proyecto deberá estar completamente terminado y no tener ningún pendiente de acuerdo con lo solicitado.

➤ Aspectos de entrega

1. La fecha de entrega máxima de este proyecto es el Miércoles 16 de Noviembre del 2022 a las 11:00am (GMT-6). No se aceptarán entregas posterior a dicha fecha.
2. El medio de entrega será:
 - A) El informe impreso, se entregará presencialmente el día de la fecha de entrega, en el aula que oportunamente el profesor comunique.
 - B) El código desarrollado (únicamente código fuente), se entregará digitalmente a la dirección entregasITCR@gmail.com

Sobre este punto:

El asunto del correo debe indicar: **IIS2022_PP2_ArquitecturaDeComputadoras_GRUPO_X**
Donde la X: Será corresponde al número de grupo de estudiantes.

El anexo código fuente: **sudoku.asm**

- C) La defensa se realizará el 16/11/2022, dado que ese día ya no hay clases, el profesor asignará espacios de 20 minutos para que cada grupo defienda el proyecto. No es necesario llevar computadora ya que todo se revisará en el computador del profesor con los códigos enviados. Sin excepción, todos los integrantes deben de estar presentes durante la revisión.

Otros Aspectos

1. Para dudas en cuanto al entendimiento del proyecto (no en cuanto a diseño, implementación o corrección de errores), el profesor las atenderá vía exclusiva por medio del correo entregasITCR@gmail.com o durante las horas destinadas a consulta. El correo a enviar será con el Asunto o Subject: **IIS2022-PP2-Arquitectura-Consulta- Grupo X.**

Anexo 1. Portada estandarizada a utilizar en el trabajo:

Instituto Tecnológico de Costa Rica



Segundo Proyecto Programado:

Programando el juego de Sudoku en ensamblador utilizando NASM

Grupo X:

Estudiante#1 – Carné: XXXXXXXX

Estudiante#2 – Carné: XXXXXXXX

Estudiante#3 – Carné: XXXXXXXX

Profesor:

Ing. Emmanuel Ramírez Segura

Fecha de entrega:

16/11/2022

II Semestre, 2022