

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Arquitectura de Computadores y Ensambladores
Sección B

Manual Técnico

Proyecto 2: Hojas de Cálculo en Ensamblador

Henri Eduardo Martínez Duarte

Carné: 201704312

Segundo semestre, año 2024

Índice

Proyecto 2: Hojas de Cálculo en Ensamblador.....	1
Índice.....	2
Índice.....	3
Funcionalidad General.....	3
Índice.....	3
Requisitos Mínimos para la Ejecución.....	3
Compilación y Ejecución.....	3
Descripción de las Etiquetas Clave.....	3

Funcionalidad General

Este manual técnico detalla la estructura y funcionamiento del programa de hojas de cálculo desarrollado en ensamblador ARM64, diseñado para operar en una cuadrícula limitada con operaciones matemáticas y lógicas. Aquí se describen las etiquetas principales en el código, el flujo del programa, los requisitos mínimos y el entorno necesario para su ejecución.

Requisitos Mínimos para la Ejecución

1. **Entorno ARM64:** El programa fue escrito en ensamblador para ARM64. Para ejecutarlo, es necesario contar con un entorno que soporte esta arquitectura, como una Raspberry Pi con un procesador ARMv8 o superior, o emuladores de arquitectura ARM en sistemas x86_64.
2. **QEMU:** Se utiliza QEMU (Quick Emulator) para ejecutar el programa en sistemas que no cuentan con un procesador ARM64 nativo. QEMU permite emular una máquina ARM y facilita la ejecución del código ensamblador en distintas plataformas.
3. **GNU Assembler (as):** Necesario para ensamblar el código.
4. **Memoria:** Aproximadamente 512 MB de RAM libre para evitar limitaciones durante la ejecución de operaciones y cálculos en la cuadrícula.
5. **Sistema Operativo:** Linux es el sistema operativo recomendado para ejecutar y compilar el programa debido a su compatibilidad con herramientas de ensamblador y emulación.

Compilación y Ejecución

Para compilar y ejecutar el programa en un sistema con soporte ARM64 o utilizando QEMU:

bash:

```
as -o proyecto.o proyecto.s      # Ensamblar el archivo
ld -o proyecto proyecto.o        # Vincular el archivo objeto
qemu-aarch64 ./proyecto          # Ejecutar el programa en QEMU
```

Descripción de las Etiquetas Clave

1. `print_matrix`

- **Función:** Esta etiqueta se encarga de imprimir la matriz completa, fila por fila y columna por columna, en la consola.
- **Detalle de Operación:**
 - Inicializa la matriz y define los contadores de filas y columnas.

- Primero imprime los encabezados de columna, luego recorre cada fila de la matriz.
- En cada celda, convierte el valor numérico a ASCII para su impresión.
- Formatea la salida con espacios entre columnas y saltos de línea entre filas para simular una cuadrícula.

2. **proc_export**

- **Función:** Gestiona la exportación de datos de la matriz a un archivo HTML.
- **Detalle de Operación:**
 - Carga el comando de exportación, y verifica que el formato y parámetros sean correctos.
 - Extrae la columna inicial, el número de columnas a exportar, y el nombre del archivo de destino.
 - En caso de error en la entrada, muestra un mensaje de error específico.
 - Llama a la subrutina `export_data` para realizar la exportación.

3. **export_data**

- **Función:** Realiza la escritura de los datos de la matriz en el archivo HTML especificado.
- **Detalle de Operación:**
 - Abre el archivo de destino y comienza a escribir el encabezado de la tabla HTML.
 - Carga y escribe los encabezados de cada columna, solicitando el nombre de cada uno al usuario.
 - Itera sobre las filas y columnas de la matriz, escribiendo cada valor como una celda en el archivo HTML.
 - Completa la tabla y cierra el archivo una vez que se han escrito todos los datos.

4. **proc_import**

- **Función:** Gestiona la importación de datos desde un archivo CSV o separado por tabulaciones.
- **Detalle de Operación:**
 - Carga el comando de importación y verifica que el formato sea correcto.
 - Extrae el nombre del archivo y el separador utilizado (coma o tabulación).
 - Llama a la subrutina `import_data` para procesar la importación.

5. **import_data**

- **Función:** Realiza la lectura y carga de datos desde el archivo CSV o separado por tabulaciones en la matriz.
- **Detalle de Operación:**
 - Abre el archivo de origen y lee los encabezados.
 - A cada columna leída le asigna un índice dentro de la matriz.
 - Convierte cada valor leído del archivo a formato numérico y lo guarda en la posición correspondiente de la matriz.
 - Una vez completada la importación, imprime un mensaje de éxito.

6. **proc_op**

- **Función:** Realiza operaciones aritméticas entre dos operandos.
- **Detalle de Operación:**
 - Recibe dos valores y el tipo de operación (suma, resta, multiplicación, etc.).
 - Realiza la operación y guarda el resultado en una variable de retorno.
 - Incluye manejo de errores para evitar desbordamiento y otros problemas aritméticos.

7. **proc_prom**

- **Función:** Calcula el promedio de un rango de celdas.
- **Detalle de Operación:**
 - Suma todos los valores en el rango especificado.
 - Divide la suma entre el número de celdas para obtener el promedio.
 - Guarda el promedio en una variable de retorno para su posterior uso.

8. **itoa**

- **Función:** Convierte un número entero a su representación en ASCII (caracteres).
- **Detalle de Operación:**
 - Recibe un número en un registro y lo convierte a una cadena de caracteres ASCII.
 - La cadena se guarda en un buffer para su impresión o almacenamiento en el archivo.

9. **atoi**

- **Función:** Convierte una cadena de caracteres ASCII en un número entero.
- **Detalle de Operación:**
 - Lee una cadena de caracteres y la convierte en su valor numérico correspondiente.
 - Almacena el resultado en un registro para su uso en cálculos y operaciones.

10. **readCSV**

- **Función:** Lee datos de un archivo CSV y los carga en la matriz.
- **Detalle de Operación:**
 - Itera sobre cada línea del archivo, separando los valores por comas.
 - Convierte cada valor en entero y lo almacena en la celda correspondiente de la matriz.
 - Verifica los saltos de línea para mover a la siguiente fila en la matriz.