Universidad del Valle de Guatemala

Programación de Microprocesadores

Sección No. 10

Docente: Juan Celada

Proyecto #2 de desarrollo:

Algoritmo de clasificación Par-Impar

Alejandro José Gomez Hernández 20347

Ana Paola de León Molina 20361

Cristian Eduardo Aguirre Duarte 20231

Gabriela Paola Contreras Guerra 20213

1. **Índice**

[Introducción](#_wup4mzclwtuj) **3**

[Antecedentes del proyecto](#_xgw93y99jvjq) **4**

[Algoritmo Descriptivo](#_6yn7efan9458) **5**

[Control de versiones](#_tivwp5kpirtt) **6**

[Demostración del temario](#_dv8lasz8cqw) **6**

[Cuerpo](#_crxbxyhzpk0g) **6**

[Conclusiones](#_7614i44aeliq) **7**

[Anexos](#_nd10gkmdvjgs) **8**

[Bibliografía](#_tz9w5s12qj0n) **11**

# 

# **Introducción**

Como parte del curso de programación de microprocesadores, se plantea el presente trabajo, el cual tiene como finalidad identificar y explotar las características de la paralelización potencial en algoritmos usando pthreads y el lenguaje C++. El objetivo principal fue aplicar los conceptos teóricos vistos a lo largo del curso. Además, de implementar operaciones del tipo mutex, variables de condición y barreras.

El temario asignado corresponde a la clasificación par-impar, para la realización de operaciones paralelas sin dependencia hacia los resultados de otros hilos. Este algoritmo se encarga de establecer las fases del proceso, en la cual se cambian los valores locales, que son clasificados posteriormente por el vecino correspondiente. Finalmente, se clasifican de forma local secuencialmente las llaves que se intercambiaron, para finalizar con un número limitado de fases, en la cual las llaves se encontrarán clasificadas.

Se logró concluir que el paralelismo es una forma eficaz de realizar un programa puesto a que este logra distribuir las tareas en varios bloques, permitiendo así, la realización de tareas en menos tiempo. Se recomienda correr el programa en una Raspberry Pi para evitar cualquier tipo de inconveniente al compilar.

# 

# **Antecedentes del proyecto**

Los números siempre han jugado un papel fundamental en la vida del ser humano, siendo utilizados desde el lejano año de 7.000 a.C., durante la época egipcia. Sin embargo, no fue hasta el año 1926 que se estableció un principio matemático. Este, fue llamado principio de paridad y se dio a conocer en el libro de Modern Puzzles de Henrry Ernest Dudeney. Henrry nunca realizó estudios universitarios de matemáticas, aunque provenía de una familia de maestros matemáticos. Este principio Henrry lo definió como “Si *un objeto o elemento cualquiera que pueda tener asociados dos estados diferentes. Si cambia su estado un número par de veces, volverá a su estado original. Por el contrario, si cambia un número impar de veces, entonces se modificará su estado”.* En palabras más simples, este principio permite establecer si un número cuenta con atributos de ser par o impar.

Un número par se puede definir como aquellos números que pueden dividirse dentro de dos, dentro de estos se encuentran los números 2,4,6,8,10,12, etc. Una forma de identificar estos números es que lacifra de las unidades de estos resultados siempre es cero o un número par. Por otro lado, los números impares son aquellos que no pueden agruparse de dos en dos como lo son los números: 1, 3, 5, 7, 9, 11… etc.

Como se mencionó anteriormente, los números son parte de casi todas las actividades que se realizan. Entre todas las que se encuentran, existe una, muy importante para la carrera de Ciencias de la Computación, que es toda la parte de la programación. Estos son fundamentales para realizar la comunicación interna de los dispositivos. Adicionalmente, se puede mencionar que permiten la realización de cálculos. Esto es de suma importancia debido a que, al ser una computadora, las operaciones matemáticas a realizar con números no tienen mayor límite más que la capacidad y potencia de la computadora a utilizar.

El paralelismo es una técnica de computación que se basa en dividir las tareas en varias tareas pequeñas y resolverlas al mismo tiempo. Esto permite la ejecución de instrucciones en un tiempo menor. Este se distingue de la computación secuencial en que varias operaciones pueden ocurrir simultáneamente. Así mismo, es importante destacar que esta técnica ha sido empleada durante muchos años, sobre todo para la computación de alto rendimiento, teniendo en cuenta las generaciones de [procesadores](https://www.ecured.cu/Microprocesadores). Este permite resolver problemas en un tiempo razonable y ofrece un mejor balance entre el rendimiento y el costo computacional. Sin embargo, la implementación de esta es compleja al momento de programar y dificulta la sincronización y comunicación entre tareas. El paralelismo se ha utilizado para muchas temáticas diferentes, desde [bioinformática](https://www.ecured.cu/Bioinform%C3%A1tica) para hacer plegamiento de [proteínas](https://www.ecured.cu/Prote%C3%ADnas), hasta economía para hacer simulaciones en [matemática](https://www.ecured.cu/Matem%C3%A1tica) [financiera](https://www.ecured.cu/index.php?title=Financiera&action=edit&redlink=1).

En cuanto al principio de paridad, se puede mencionar que este ha sido implementado en muchísimas ocasiones en ejercicios y ejemplos de estructuras de datos. Inclusive, llegando al punto de crear un algoritmo llamado clasificación impar-par, como el que fue presentado para el desarrollo de este proyecto. Este algoritmo se divide en dos fases, siendo estas la fase impar y la par. El algoritmo se ejecuta hasta que los elementos de la matriz se ordenan y en cada iteración se producen las dos fases. La fase impar, es la encargada de realizar una clasificación del tipo bubblesort de los elementos indexados impares y la fase par realiza una clasificación del tipo bubblesort en elementos indexados pares.

Finalmente, es importante mencionar que el temario a desarrollar para esta entrega cuenta con mucha ventana de oportunidad para realizar la paralelización de este. Cabe mencionar que, este algoritmo utiliza fases de intercambio para comparar elementos que ocurren de forma simultánea en otras fases. Por estos motivos, el temario presentado es una correcta aplicación de los conceptos relacionados al paralelismo.

# **Algoritmo Descriptivo**

1. Se define el encabezado
2. Se hace llamado a librerías
3. Se declaran variables globales
4. Se declara variables tipo pthread, mutex, barrera
5. Se define subrutina ObtencionNum
   * Se solicita al usuario el ingreso del número
   * Se muestra mensaje de clasificación
   * Se crea un ciclo for con los elementos definidos
     1. Se recorre y se obtienen números aleatorios
   * Se retorna la lista con los elementos que obtuvo
6. Se define la subrutina de Bubblesort
   * Ciclo for para obtener el valor mínimo
     1. Ciclo for para ordenar la lista
        1. Ciclo if para clasificar los números
     2. XXXXXXXXXXXXXXXXXXX
7. Se define la subrutina ClasificacionNum
   * HH
   * Se realiza la división del dominio
   * Se crea el ciclo for para acceder al dominio
     1. Se llama la subrutina Bubblesort
   * XXXXXXXXX
8. Se define la subrutina Main
   * Llamado al ID de cada hilo
   * Llamado a la función de ObtencionNum
   * Declaración de variables internas
   * Declaración del ciclo while, el cual dura la cantidad de hilos utilizados
     1. Creación de hilos
     2. Creación del ciclo for
        1. Verificación de la creación del hilo
     3. Suma al contador de hilos
   * Ciclo for para imprimir listas
   * Ciclo while para realizar el join
   * Destrucción de hilos y mutex

# **Control de versiones**

Como parte de los requerimientos del presente proyecto, se optó por implementar la herramienta de GitHub para llevar control del manejo de versiones. Esto, facilitó bastante la realización de los procedimientos a desarrollar por cada uno de los miembros del equipo, evitando así cualquier tipo de confusión.

<https://github.com/aleg001/Proyecto2-Micro>

# **Demostración del temario**

Adicionalmente, se grabó un vídeo en el cual se demuestra el correcto funcionamiento del temario asignado. Esto, con la finalidad de presentar la versión final del programa.

LINK DEL youtube xxdxd

# **Cuerpo**

El presente proyecto tiene un enfoque en las operaciones paralelas para la clasificación par e impar. Entre las funciones se encuentra que, al realizar estas operaciones paralelas, no depende directamente de los resultados obtenidos de otros hilos durante su ejecución. Básicamente, el algoritmo desarrollado en código consiste en varias fases de proceso, en la cual se realiza la separación de datos, los cuales posteriormente se ordenan de forma secuencial dentro del hilo. Posteriormente, se intercambian los valores locales que se clasificaron con su vecino correspondiente, para mandarlos a una pareja. Finalmente, este algoritmo regresa localmente las llaves intercambiadas, para su clasificación.

En cuanto a los retos encontrados para la implementación del paralelismo dentro de un algoritmo de clasificación, se encontró con bastantes conflictos en cuanto a la compilación. Por la naturaleza de este proyecto, no se pudo compilar en ningún compilador en línea como método de prueba. Sin embargo, esto fácilmente se solucionó implementando el uso de una Raspberry Pi para el corrimiento del programa. Cabe destacar que, adicional a esto, también se realizaron pruebas en una Mac, utilizando Clang. A diferencia del compilador en línea, se obtuvieron resultados prácticamente iguales que al correrlos en una Raspberry Pi, haciendo de esta una alternativa para correr el código.

Para futuras réplicas de este temario de proyecto, se recomienda tener bastante presente los conceptos de paralelismo para poder realizarlo de forma correcta. Adicional a esto, se recomienda realizar el algoritmo descriptivo o diagrama de flujo, previo al desarrollo del código. Al realizarlo de esta manera, resultará mucho más fácil programar este algoritmo de clasificación par e impar. Así mismo, la implementación de herramientas como GitHub para llevar un control de versiones, resulta bastante útil. Se usó esta herramienta como método de completar las distintas tareas o conflictos existentes con el código desarrollado.

# **Conclusiones**

* + Por medio de la investigación y la programación en paralelo, se cumplió con los objetivos propuestos en este proyecto, logrando así, la resolución del temario asignado.
  + Se recomienda para futuras réplicas de este temario de proyecto, tomar en consideración que, adicionalmente al análisis previo, es necesaria una ardua investigación para contar con amplios conocimientos sobre el paralelismo, aplicandolo en la resolución de este temario.
  + Se recomienda compilar este tipo de programas en una Raspberry Pi, esto, debido a que al correrlo en un compilador en línea, muestra mensajes de error.

# 

# **Anexos**

Tabla: No.2: Catálogo de funciones

|  | **Variables** | **Tipos** | **Descripción de uso** |
| --- | --- | --- | --- |
| **Entradas** | Variable |  |  |
| **Salidas** |  |  |  |
| **Descripción**  **(Propósito de función)** |  |  |  |

Tabla: No.3: Bitácora de pruebas:

| **Número de prueba** | **Resultado** |
| --- | --- |
| Prueba No.1: |  |
| Prueba No.2: |  |
| Prueba No.3: |  |
| Prueba No.4: |  |
| Prueba No.5: |  |
| Prueba No.6: |  |
| Prueba No.7: |  |
| Prueba No.8: |  |
| Prueba No.9: |  |
| Prueba No.10: |  |
| Prueba No.11: |  |
| Prueba No.12: |  |
| Prueba No.13: |  |
| Prueba No.14: |  |
| Prueba No.15: |  |

Imagen No.1: Referencia del código del libro Pachecho 3.8 & Grama et al 9.3

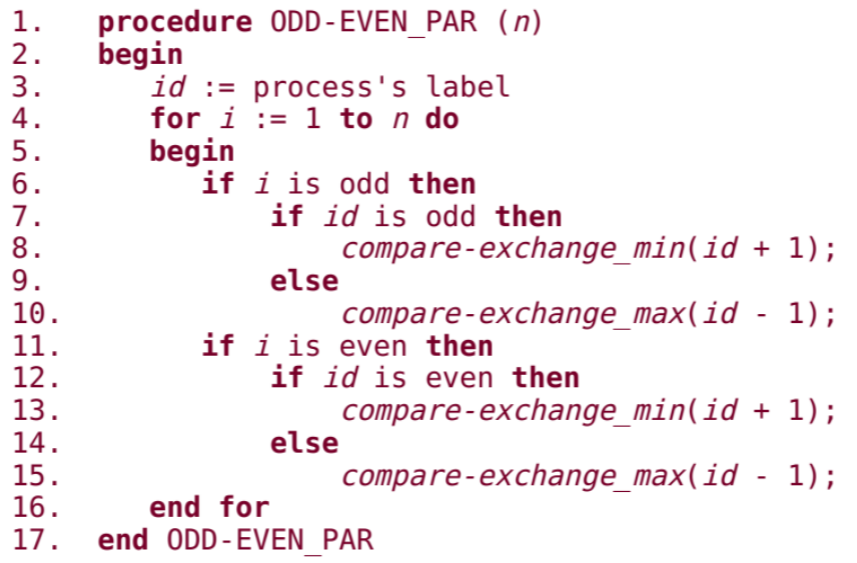


Imagen No.2: Referencia de código para la realización del sort

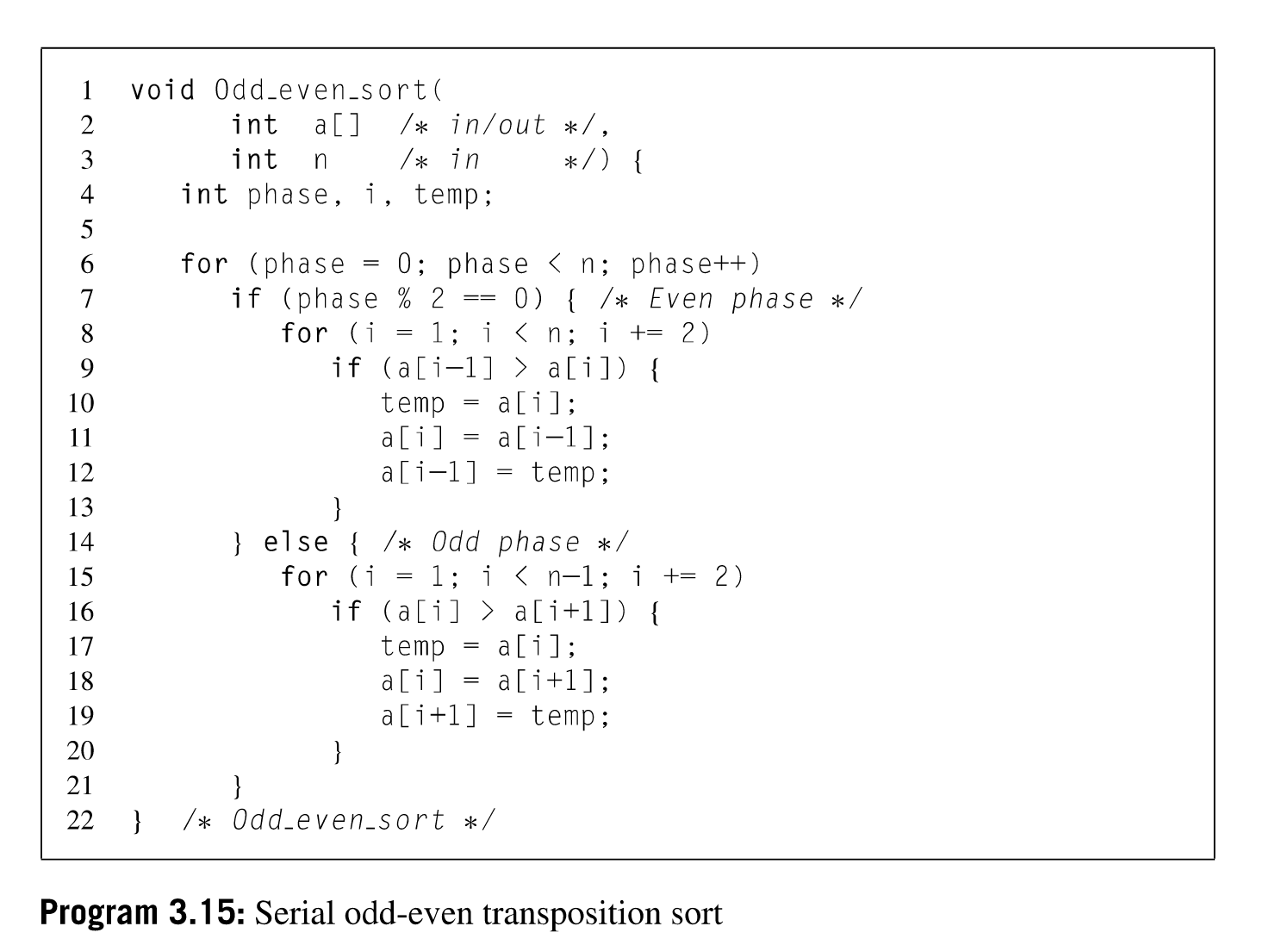


Imagen No.3: Planteamiento original del ordenamiento por fases



# **Bibliografía**

* + A. (2017, 24 octubre). *Principio de paridad*. Ilusiones Matemáticas. <https://www.ilusionesmatematicas.com/principio-de-paridad/>
  + Grama, A. (2003). *Introduction to Parallel Computing (2nd Edition)*. TBS.
  + Universidad de Carlos III Madrid. (s. f.). Programación con Hilos. Programación con Hilos. Recuperado 4 de octubre de 2021, de <http://www.it.uc3m.es/pbasanta/asng/CES/M2/concurrent_1_es.pdf>
  + Pacheco, P. (2011). *An Introduction to Parallel Programming (English Edition)* (1.a ed.). Morgan Kaufmann.
  + Primaria, M. (2021, 14 junio). *▷ Números pares e impares【¿Qué son? Curiosidades】*. Mundo Primaria. <https://www.mundoprimaria.com/recursos-matematicas/numeros-pares-e-impares>
  + GeeksforGeeks. (2021, 28 junio). *Odd-Even Sort / Brick Sort*. <https://www.geeksforgeeks.org/odd-even-sort-brick-sort/>
  + EcuRed. (s. f.). Paralelismo ( informática) - EcuRed. Recuperado 5 de octubre de 2021, de <https://www.ecured.cu/Paralelismo_(_inform%C3%A1tica)>
  + C++ Multithreading - Tutorialspoint. (2010). tutorialspoint. Recuperado el 5 de octubre. <https://www.tutorialspoint.com/cplusplus/cpp_multithreading.htm>
  + Rauber, T., & Runger, G. (n.d.). Parallel programming: For multicore and cluster systems. Springer Books.