

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Computo de alto desempeño

Profesor: M.I. Rene Oscar Valdez Casillas

Equipo: Daniel Ponce de León Súchil

Gómez de la Cruz Luis Octavio

Semestre: 2020-1

Índice

[Objetivos 2](#_Toc25237705)

[Problema 2](#_Toc25237706)

[Análisis del problema 2](#_Toc25237707)

[Introducción 2](#_Toc25237708)

[Desarrollo 3](#_Toc25237709)

[Análisis y Diseño: 3](#_Toc25237710)

[Funciones de MPI 3](#_Toc25237711)

[Análisis de Speed-up 4](#_Toc25237712)

[Manual de Uso del programa 4](#_Toc25237713)

[Conclusiones Generales 4](#_Toc25237714)

[Bibliografía 5](#_Toc25237715)

## Objetivos

Analizaremos, desarrollaremos e implementaremos un programa de forma paralela.

Identificaremos en qué casos o circunstancias óptimas para implementar el computo en paralelo con las herramientas adecuadas en este caso se usar OpenMP.

Realizaremos el cifrado de Hill en forma serial para posteriormente desarrollar la versión paralela del mismo.

## Problema

Se requiere realizar un programa en ANSI C en su versión serial y paralela que realice el cifrado de Hill en base al alfabeto indicado en este documento. En los parámetros de entrada se recibirán las dimensiones de la matriz, n, los valores de la matriz A y el mensaje en texto claro a cifrar.

## Análisis del problema

Es esencial la versión del programa de forma serial para poder dar inicio a la resolución de este programa se debe conocer el cifrado Hill para poder entender completamente el proceso que se realiza y asi mismo se podrá identificar en la estructura que sección esta susceptible para implementar computo paralelo.

## Introducción

En 1929, Lester S. Hill desarrolla un sistema criptográfico de sustitución polialfabético, es decir, un mismo signo, en este caso una misma letra, puede ser representado en un mismo mensaje con más de un carácter.

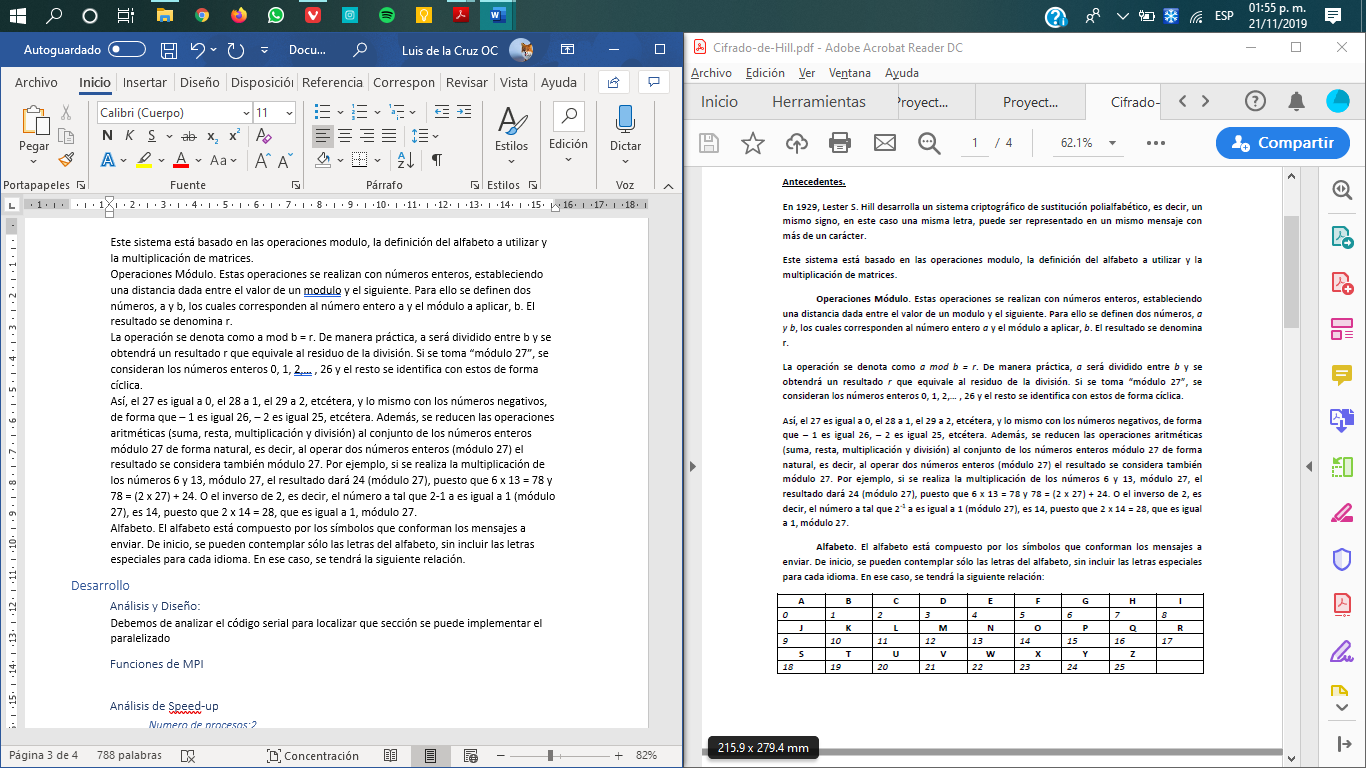
Este sistema está basado en las operaciones modulo, la definición del alfabeto a utilizar y la multiplicación de matrices.

Operaciones Módulo. Estas operaciones se realizan con números enteros, estableciendo una distancia dada entre el valor de un módulo y el siguiente. Para ello se definen dos números, a y b, los cuales corresponden al número entero a y el módulo a aplicar, b. El resultado se denomina r.

La operación se denota como a mod b = r. De manera práctica, a será dividido entre b y se obtendrá un resultado r que equivale al residuo de la división. Si se toma “módulo 27”, se consideran los números enteros 0, 1, 2,… , 26 y el resto se identifica con estos de forma cíclica.

Así, el 27 es igual a 0, el 28 a 1, el 29 a 2, etcétera, y lo mismo con los números negativos, de forma que – 1 es igual 26, – 2 es igual 25, etcétera. Además, se reducen las operaciones aritméticas (suma, resta, multiplicación y división) al conjunto de los números enteros módulo 27 de forma natural, es decir, al operar dos números enteros (módulo 27) el resultado se considera también módulo 27. Por ejemplo, si se realiza la multiplicación de los números 6 y 13, módulo 27, el resultado dará 24 (módulo 27), puesto que 6 x 13 = 78 y 78 = (2 x 27) + 24. O el inverso de 2, es decir, el número a tal que 2-1 a es igual a 1 (módulo 27), es 14, puesto que 2 x 14 = 28, que es igual a 1, módulo 27.

Alfabeto. El alfabeto está compuesto por los símbolos que conforman los mensajes a enviar. De inicio, se pueden contemplar sólo las letras del alfabeto, sin incluir las letras especiales para cada idioma. En ese caso, se tendrá la siguiente relación.



## Desarrollo

### Análisis y Diseño:

Debemos de analizar el código serial para localizar que sección se puede implementar el paralelizado

### Funciones de MPI

### Análisis de Speed-up

#### Tiempo de ejecución del programa paralelo con diferentes tamaños del mensaje

#### Tiempo de ejecución del programa paralelo con diferentes tamaños del mensaje y diferentes numero de procesos

##### Numero de procesos:2

##### Numero de procesos:4

##### Numero de procesos:6

##### Numero de procesos:8

### Manual de Uso del programa

1. Iniciamos el sistema operativo donde se tenga implementado la herramienta MPI, en este caso se usó Linux, distribución Ubuntu.
2. Ejecutamos la consola(terminal), nos posicionamos en la dirección donde se encuentre el código fuente respectivo para la ejecución del mismo

### Conclusiones Generales

Como podemos comprobar en las gráficas aumentando el número de procesos es más rápido la ejecución de las sentencias que se paralelizaron, pero no en todos los problemas computacionales no es recomendable hacer uso de la paralelización, y en otros casos no abrumar con el número de procesos esto puede entorpecer la operación.

#### Conclusiones Personales

##### Luis Octavio Gómez de la Cruz

Fue un poco complicado controlar o hacer que la comunicación de los hilos o procesos ya que en cierto punto solo se realizaba una cuarta parte de la transformación de la imagen correctamente y el resto no llego o no se sobrescribió, este fue el mayor problema que presentamos

##### . Daniel Ponce de León Súchil

El programa

# Bibliografía