Diseño e implementación:

Antes que nada, cabe informar que se utilizó una estructura llamada “complejo” para poder representarlos. La misma consiste de dos atributos de tipo float los cuales representan a la parte real e imaginaria del numero.

El programa consiste primeramente en encontrar cada argumento indicado (pasado como parámetro al programa por línea de comandos), analizar si éstos son correctos, guardarlos en sus respectivas variables y, en caso de no ser correctos, informar el error y cortar la ejecución del programa. Esto se hizo con la ayuda de funciones auxiliares para el manejo de strings de librerías como string o unistd, a través de una estructura de switch. Todos los parámetros fueron guardados en variables estáticas, y en caso de no estar presentes se reemplazaron por aquellos dados en el enunciado por defecto. En caso de no poder crear un archivo de salida, también se sale del programa.

La parte central del algoritmo, donde se itera en el plano complejo de forma discreta, se utilizan dos ciclos definidos para el eje real y el eje imaginario, manteniendo un acumulador para definir el brillo del pixel, el cual luego es guardado en el archivo de salida.

Se escribió un programa secuencial completamente, sin el agregado de muchas estructuras adicionales ni funciones auxiliares, para que luego el seguimiento en lenguaje ensamblador sea más simple y entendible.

EXPLICAR EL ALGORITMO

Como obtener el ejecutable y correr pruebas:

Para mayor comodidad se dispone de un archivo llamado “script.sh” que al ejecutarlo (esto es, posicionándose sobre el directorio donde se encuentran los archivos y ejecutando el comando ./script.sh) automáticamente compilara el programa y ejecutará una seria de comandos de prueba y sus respectivas salidas.

El código fuente:

PEGAR CODIGO FUENTE