UNIVERSIDAD AUTÓNOMA "TOMAS FRÍAS" CARRERA DE INGENIERÍA DE SISTEMAS				
Materia:	Arquitectura de computadoras (SIS-522)			
Docente: **Auxiliar:**	Ing. Gustavo A. Puita Choque Univ. Aldrin Roger Perez Miranda			N° Práctica
16/06/2024	**Fecha publicación**			9
01/07/2024	**Fecha de entrega**			
Grupo:	**1**	**Sede**	**Potosí**	

Nombre: Jorge Eduardo Chavarria Condori

1) ¿Qué es el 'stack' en el contexto del lenguaje ensamblador y cómo se utiliza?

La pila del programa (stack) es una de las regiones que forma parte de los programas en ejecución. Su tamaño es dinámico, crece o disminuye según se requiera. Crece hacia abajo desde la dirección más alta.

2) Describe un escenario práctico donde el uso de ensamblador sería más ventajoso que el uso de un lenguaje de alto nivel.

Los usos típicos son controladores/manejadores (drivers) de dispositivo, sistemas embebidos de bajo nivel, y sistemas de tiempo real. Históricamente, un gran número de programas han sido escritos enteramente en lenguaje ensamblador.

Escenario: Sistemas Embebidos en Automóviles

Contexto:

Microcontroladores en automóviles manejan tareas críticas como sistemas de frenado ABS, control de tracción, y gestión del motor. Estos sistemas necesitan alta eficiencia y rapidez.

Ventajas del Ensamblador:

Eficiencia y Rendimiento:

Código altamente optimizado, esencial para reacciones inmediatas (ej. sistema ABS).

Control Directo del Hardware:

Acceso rápido y preciso a registros y periféricos del hardware, crucial para control de motores y lectura de sensores.

Minimización del Uso de Memoria:

Uso eficiente y reducido de memoria RAM y ROM, vital en microcontroladores con recursos limitados.

Determinismo en el Tiempo de Ejecución:

Control detallado del flujo de ejecución para cumplir con tiempos predeterminados, esencial en sistemas de tiempo real.

Ejemplo Específico: Control de Inyección de Combustible

Lectura de Sensores: Rutinas en ensamblador para leer valores de sensores rápidamente.

Cálculo del Tiempo de Inyección: Operaciones aritméticas optimizadas en ensamblador.

Generación de Señales de Control: Precisión temporal en la generación de señales para inyectores.

3) Explique cada línea del siguiente código del lenguaje ensamblador y diga que es lo que se está haciendo

MOV AX, 5 ; Línea 1

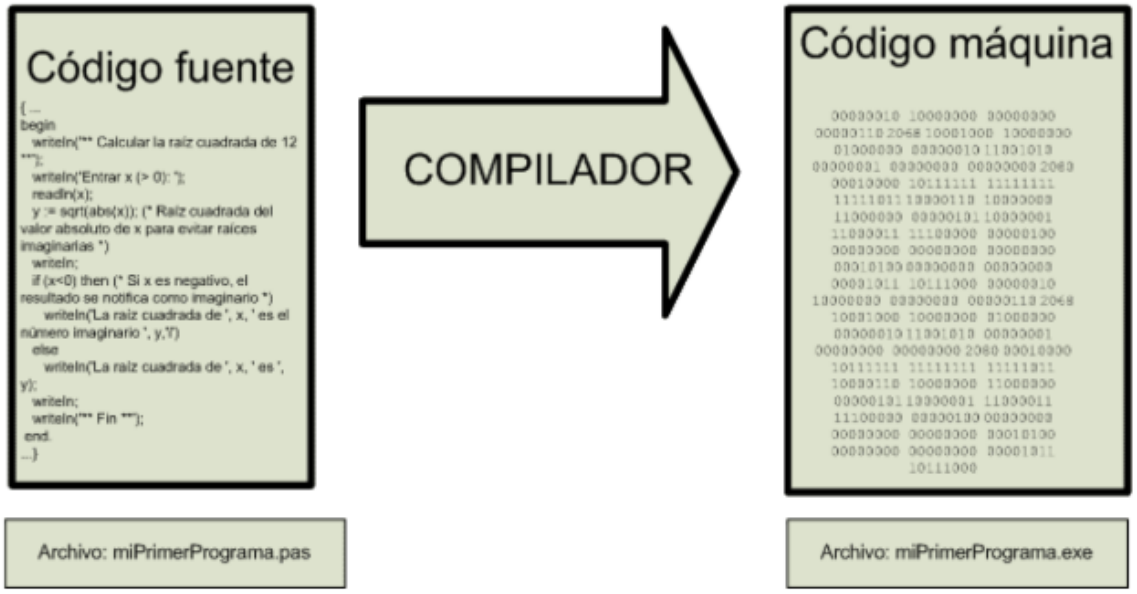
MOV BX, 10 ; Línea 2

ADD AX, BX ; Línea 3

MOV CX, AX ; Línea 4

Línea 1: MOV AX, 5 Instrucción: MOV Operando Destino: AX Operando Fuente: 5 Acción: Se mueve (copia) el valor inmediato 5 al registro AX. Comentario: ; Línea 1 Resultado: AX ahora contiene el valor 5.	Línea 2: MOV BX, 10 Instrucción: MOV Operando Destino: BX Operando Fuente: 10 Acción: Se mueve (copia) el valor inmediato 10 al registro BX. Comentario: ; Línea 2 Resultado: BX ahora contiene el valor 10.
Línea 3: ADD AX, BX Instrucción: ADD Operando Destino: AX Operando Fuente: BX Acción: Se suma el valor del registro BX al valor del registro AX y el resultado se almacena en AX. Comentario: ; Línea 3 Resultado: AX ahora contiene el valor 15 (porque 5 + 10 = 15).	Línea 4: MOV CX, AX Instrucción: MOV Operando Destino: CX Operando Fuente: AX Acción: Se mueve (copia) el valor del registro AX al registro CX. Comentario: ; Línea 4 Resultado: CX ahora contiene el valor 15.
El código en ensamblador realiza las siguientes acciones paso a paso: Inicializa el registro AX con el valor 5. Inicializa el registro BX con el valor 10. Suma el valor de BX al valor de AX, almacenando el resultado en AX. Copia el resultado de AX al registro CX. Al finalizar, AX y CX contienen el valor 15, y BX contiene el valor 10.	

4) Explique detalladamente cómo funcionan los compiladores



Explicación Detallada de Cómo Funcionan los Compiladores

Un compilador es un programa que traduce el código fuente escrito en un lenguaje de programación de alto nivel a un lenguaje de bajo nivel, típicamente código máquina o código ensamblador, que puede ser ejecutado directamente por la computadora. A continuación, se describen las fases principales del proceso de compilación:

1. Análisis Léxico (Lexical Analysis):

- **Función:** Escanea el código fuente y convierte la secuencia de caracteres en una secuencia de tokens. Los tokens son las unidades léxicas básicas del lenguaje, como palabras clave, identificadores, operadores y símbolos.
- **Salida:** Una lista de tokens.

2. Análisis Sintáctico (Syntax Analysis):

- **Función:** Toma la secuencia de tokens y verifica que estén estructurados de acuerdo con las reglas gramaticales del lenguaje de programación. Construye un árbol sintáctico (parse tree) que representa la estructura jerárquica del programa.
- **Salida:** Un árbol sintáctico (AST - Abstract Syntax Tree).

3. Análisis Semántico (Semantic Analysis):

- **Función:** Examina el árbol sintáctico para asegurarse de que el programa tiene sentido desde el punto de vista lógico y semántico. Realiza verificaciones de tipos, control de variables, y otras verificaciones contextuales.
- **Salida:** Un árbol sintáctico anotado con información semántica.

4. Optimización (Optimization):

- **Función:** Mejora el código intermedio generado en términos de eficiencia y rendimiento sin cambiar su significado. Esto puede incluir la eliminación de código muerto, la optimización de bucles, y otras transformaciones.
- **Salida:** Código intermedio optimizado.

5. Generación de Código Intermedio (Intermediate Code Generation):

- **Función:** Traduce el árbol sintáctico anotado en una representación intermedia que es más fácil de manipular y optimizar que el código fuente original. Este código intermedio es independiente de la máquina.
- **Salida:** Código intermedio.

6. Generación de Código (Code Generation):

- **Función:** Convierte el código intermedio optimizado en código máquina específico para la arquitectura del procesador de destino.
- **Salida:** Código máquina o ensamblador.

7. Ensamblado y Enlazado (Assembly and Linking):

- **Función:** Si la salida es código ensamblador, se ensambla en código máquina. Luego, se enlaza con otras unidades de código y bibliotecas necesarias para crear un ejecutable completo.
- **Salida:** Un archivo ejecutable (por ejemplo, miPrimerPrograma.exe).

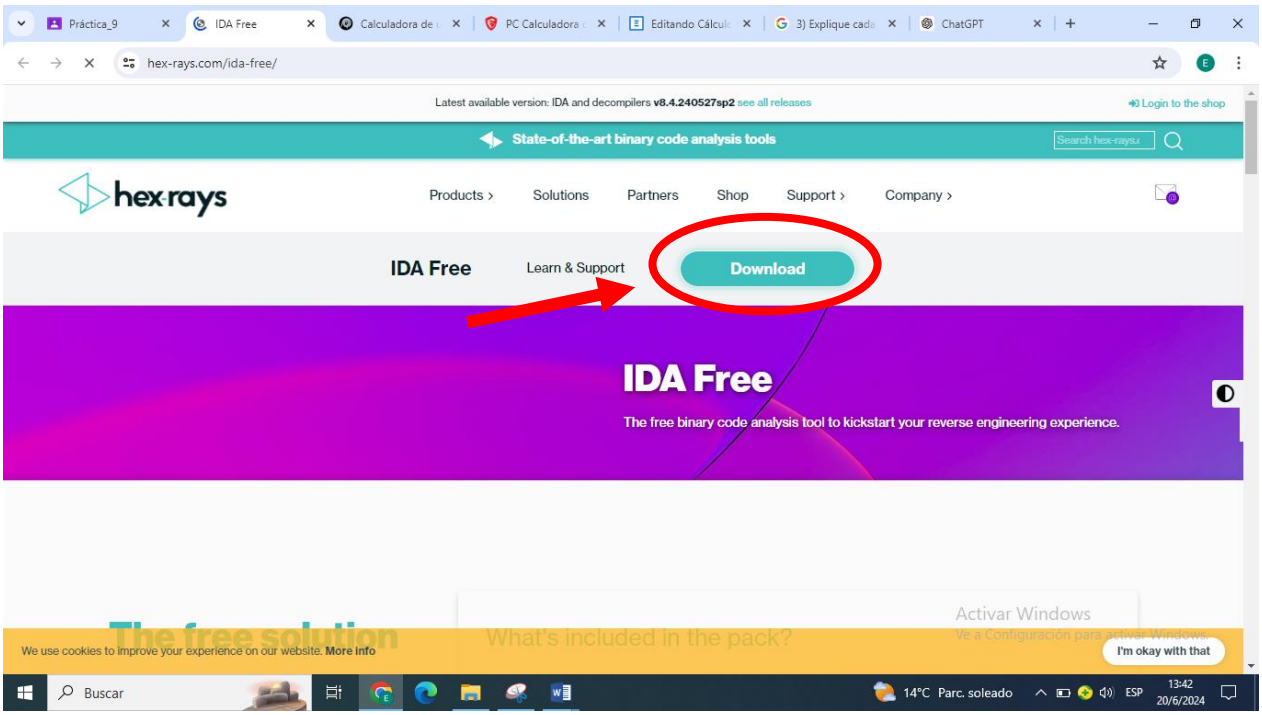
5) Realizar capturas de pantalla del siguiente procedimiento:

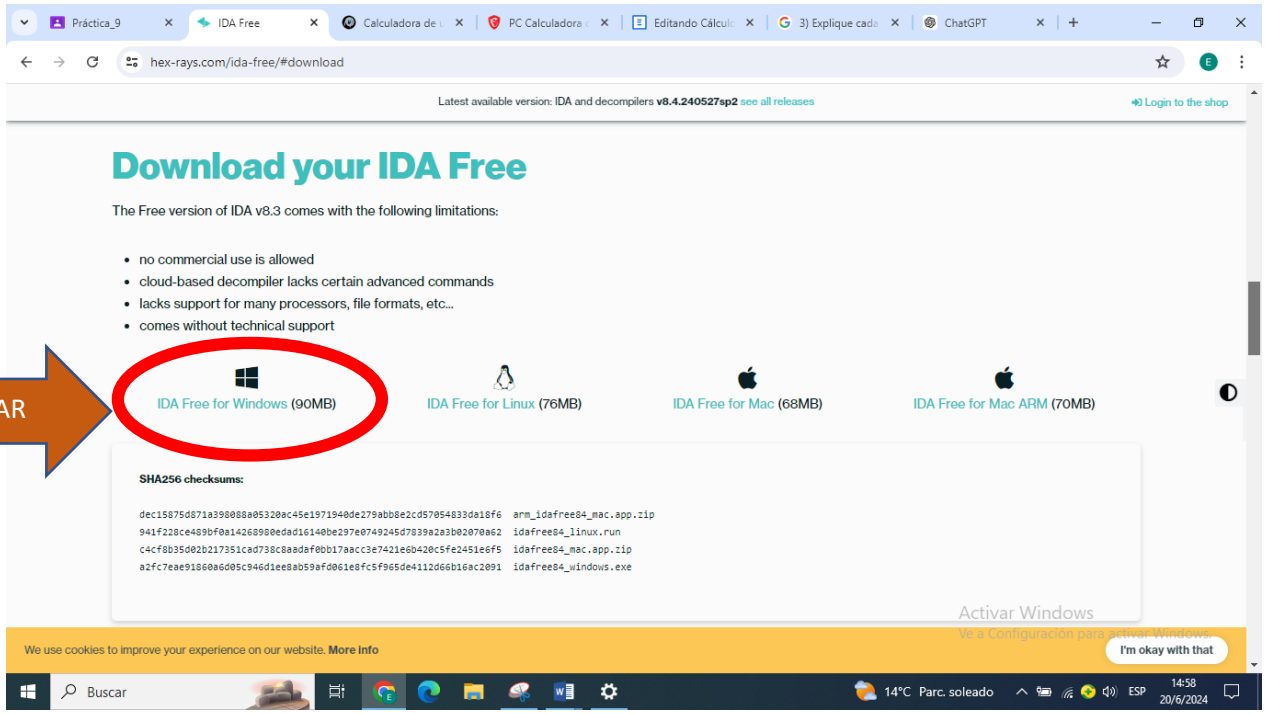
EL PROCEDIMIENTO LO DEBE HACER COMO UN LABORATORIO PASO A PASO Y EXPLICAR QUE ES LO QUE SE ESTA HACIENDO CON SU RESPECTIVA CAPTURA USTED DEBE SELECCIONAR CUALQUIER SERVICIO DE SU PREFERENCIA

IDA: Es una de las herramientas más conocidas y potentes para el análisis de código binario y desensamblado. En este laboratorio se instalará IDA FREE pero también se tiene la versión de paga IDA PRO

Paso 1:

Descargar el software IDA FREE el cual lo podrá a hacer del siguiente enlace: https://hex-rays.com/ida-free/



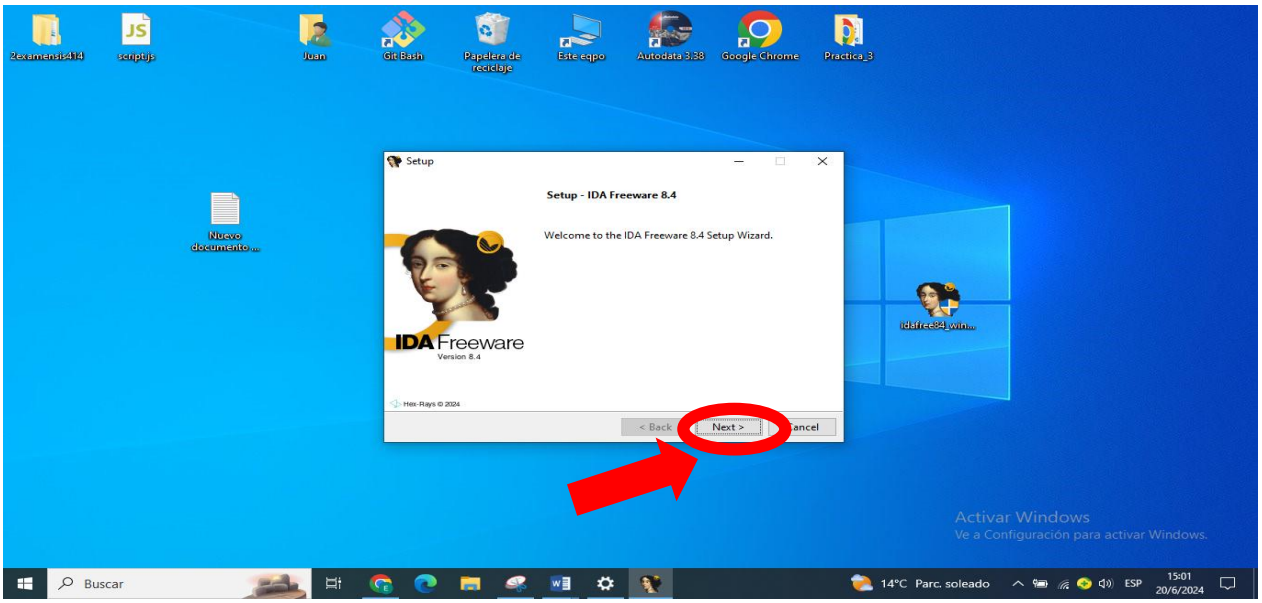


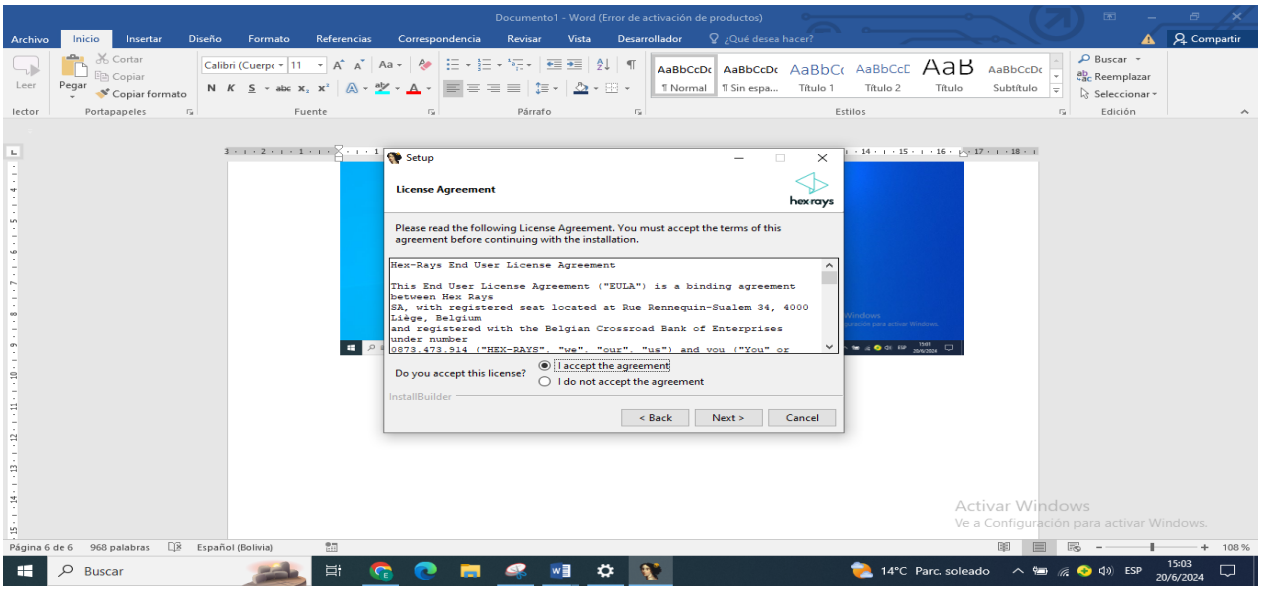
PRESIONAR

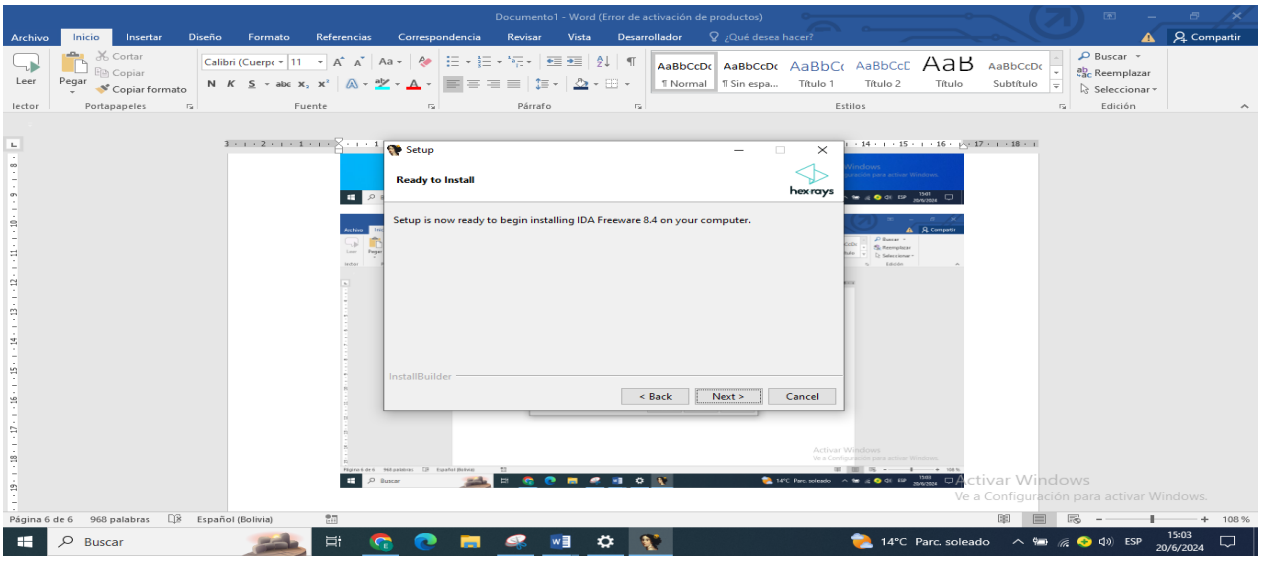
Paso 2:

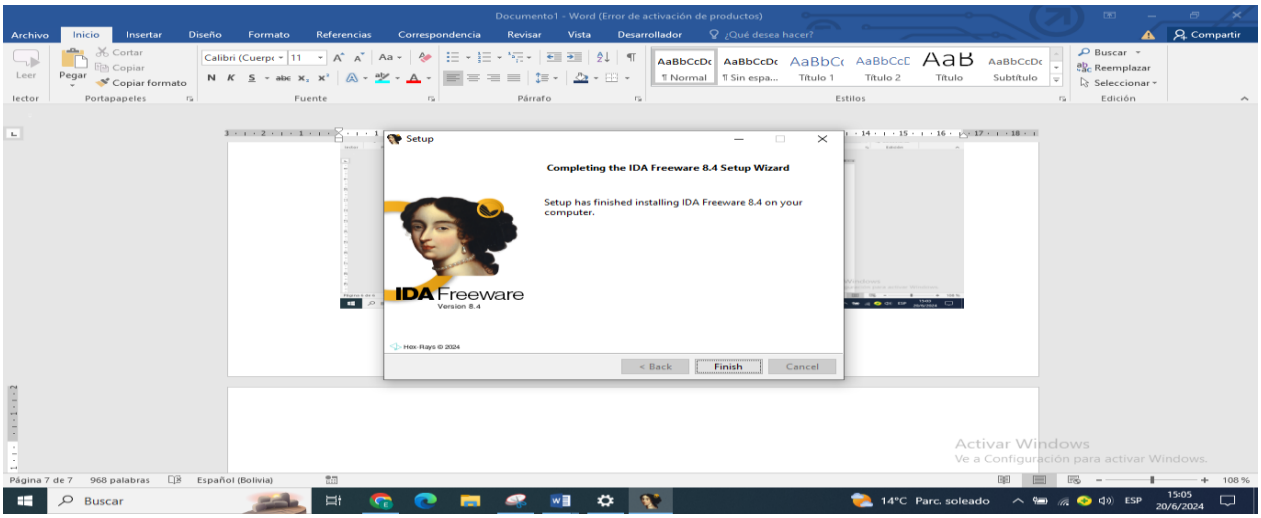
Instalación

Seleccionar Next

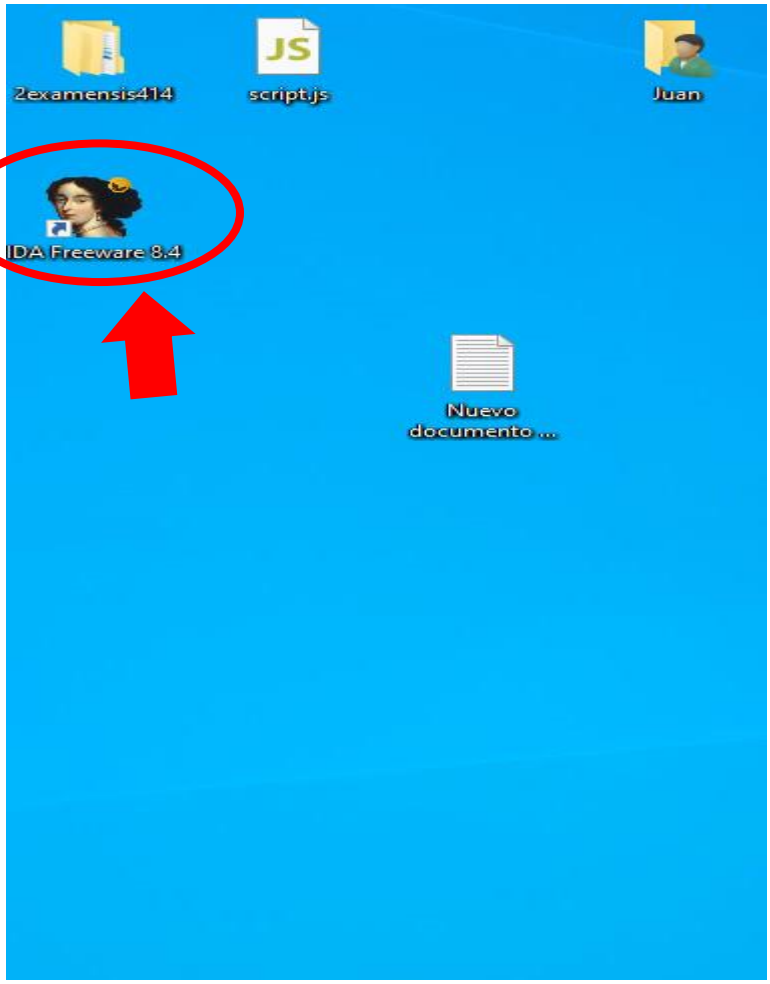

2examensis414
scriptjs
Juan
Git Bash
Papelera de reciclaje
Este equipo
Autodata 3.38
Google Chrome
Practica_3
Nuevo documento...
idafree84_win...
Setup
Setup - IDA Freeware 8.4
Welcome to the IDA Freeware 8.4 Setup Wizard.
IDA Freeware
Version 8.4
Hex-Rays © 2024
< Back
Next >
Cancel
Activar Windows
Ve a Configuración para activar Windows.
Buscar
14°C Parc. soleado
ESP
15:01
20/6/2024


Documento1 - Word (Error de activación de productos)
Archivo
Inicio
Insertar
Diseño
Formato
Referencias
Correspondencia
Revisar
Vista
Desarrollador
¿Qué desea hacer?
Compartir
Setup
License Agreement
hex rays
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
Hex-Rays End User License Agreement
This End User License Agreement ("EULA") is a binding agreement between Hex Rays
SA, with registered seat located at Rue Rennequin-Sualem 34, 4000 Liège, Belgium
and registered with the Belgian Crossroad Bank of Enterprises under number
0873.473.914 ("HEX-RAYS", "we", "our", "us") and you ("You" or
Do you accept this license?
I accept the agreement
I do not accept the agreement
InstallBuilder
< Back
Next >
Cancel
Activar Windows
Ve a Configuración para activar Windows.
Página 6 de 6
968 palabras
Español (Bolivia)
108 %
Buscar
14°C Parc. soleado
ESP
15:03
20/6/2024


Documento1 - Word (Error de activación de productos)
Archivo
Inicio
Insertar
Diseño
Formato
Referencias
Correspondencia
Revisar
Vista
Desarrollador
¿Qué desea hacer?
Compartir
Setup
Ready to Install
hex rays
Setup is now ready to begin installing IDA Freeware 8.4 on your computer.
InstallBuilder
< Back
Next >
Cancel
Activar Windows
Ve a Configuración para activar Windows.
Página 6 de 6
968 palabras
Español (Bolivia)
108 %
Buscar
14°C Parc. soleado
ESP
15:03
20/6/2024

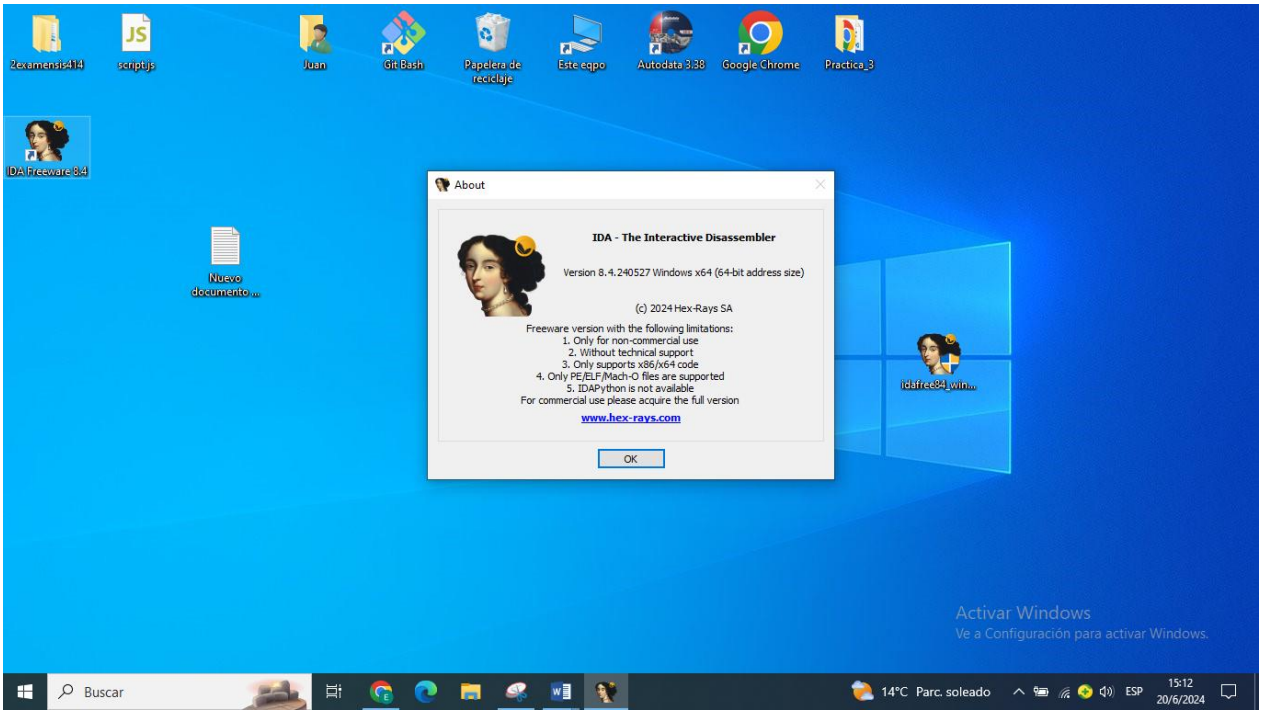


Una vez descargado e instalado deberán abrir el ejecutable .exe

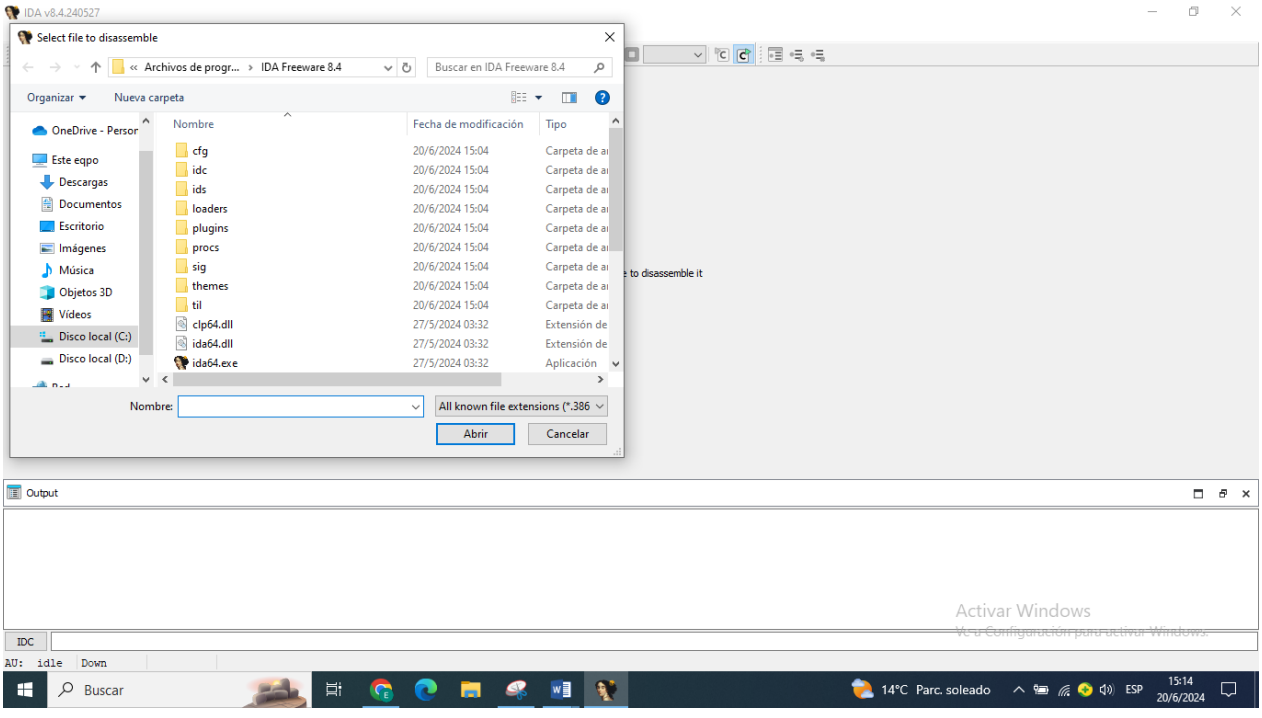


Paso 3:

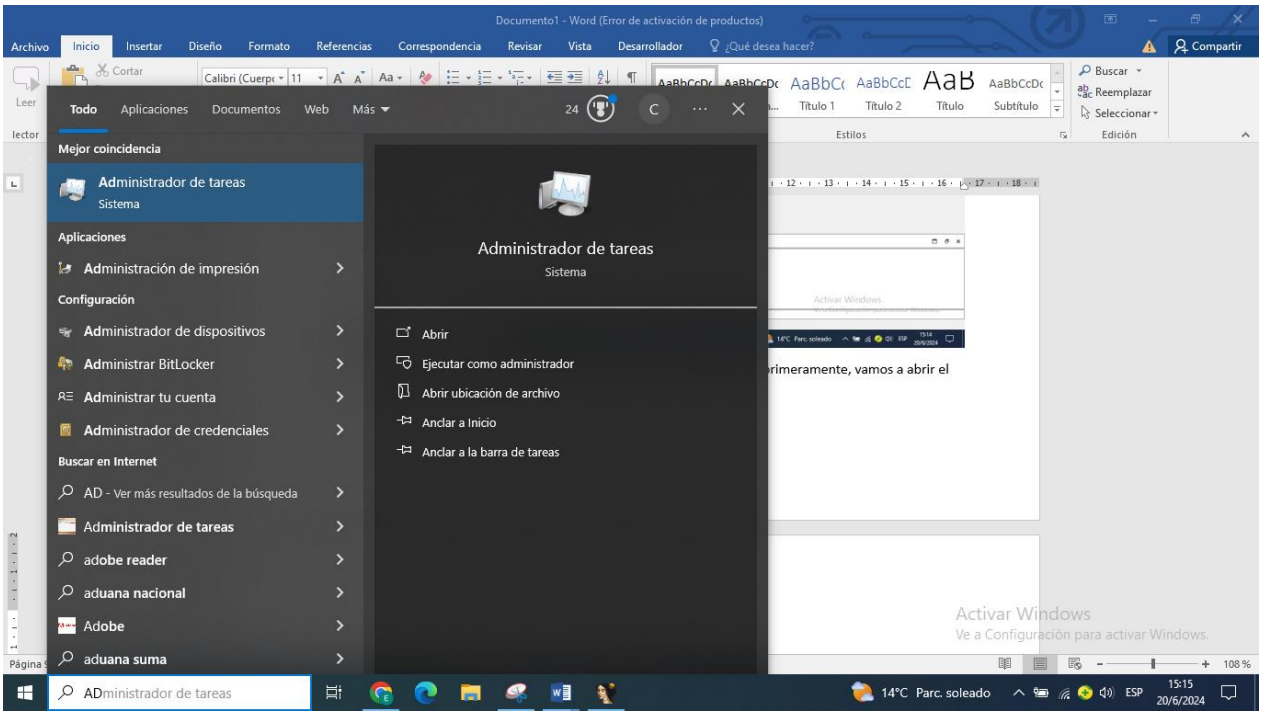
Procederemos a abrir un servicio en Windows



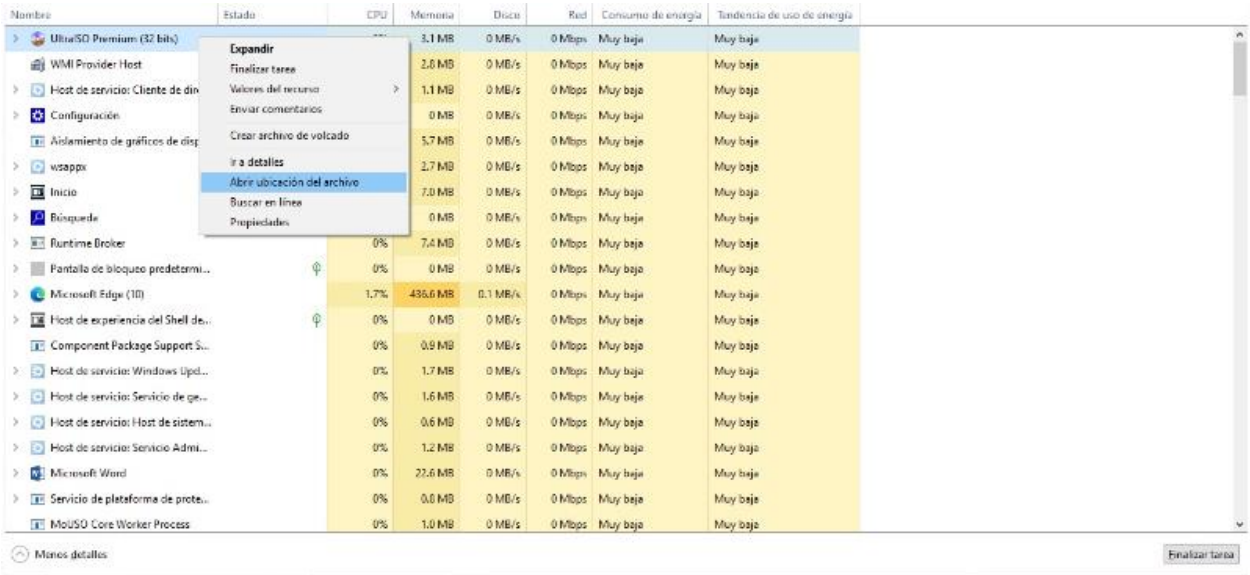
Lo que deberá hacer ahora es seleccionar “New”



Ahora deberá seleccionar algún servicio de su administrador de tareas, primeramente, vamos a abrir el administrador de tareas

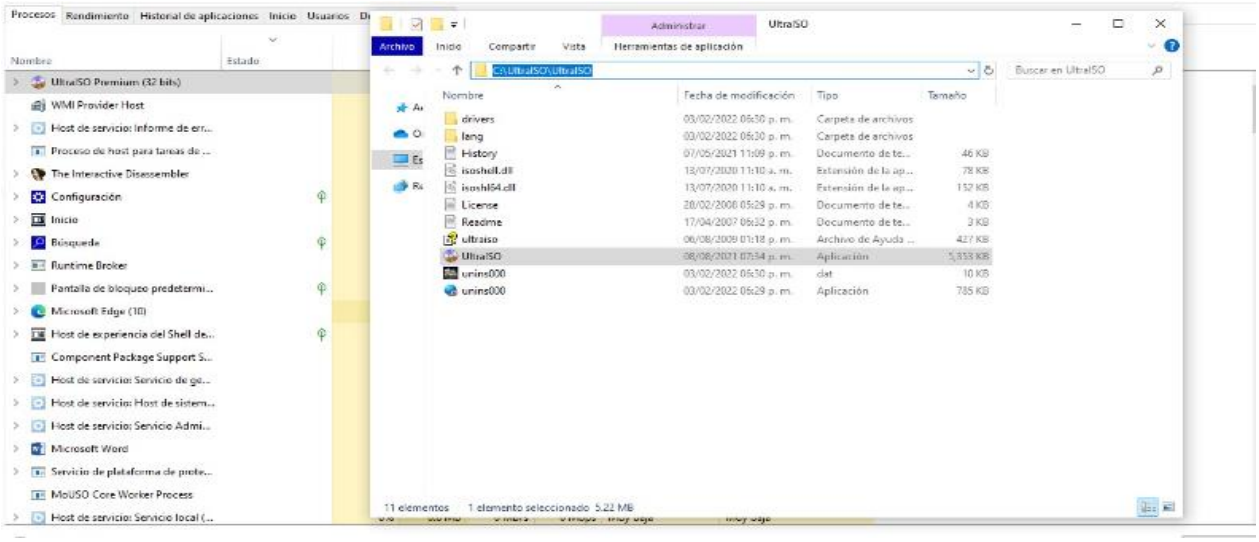


Ahora en la pestaña procesos deberá buscar cualquier servicio que se este ejecutando en tiempo real, y hacer un clic izquierdo sobre el servicio que le interesará ver el código ensamblador de este y después con un clic derecho seleccionar "Abrir ubicación del archivo".

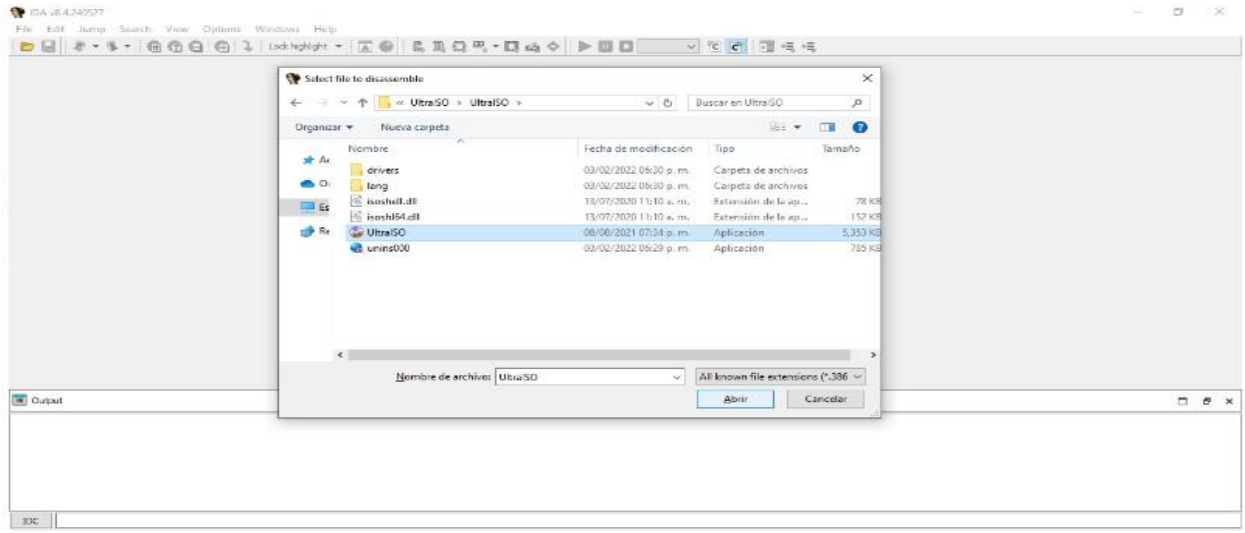


Una vez hecho esto se abrirá la ubicación del servicio

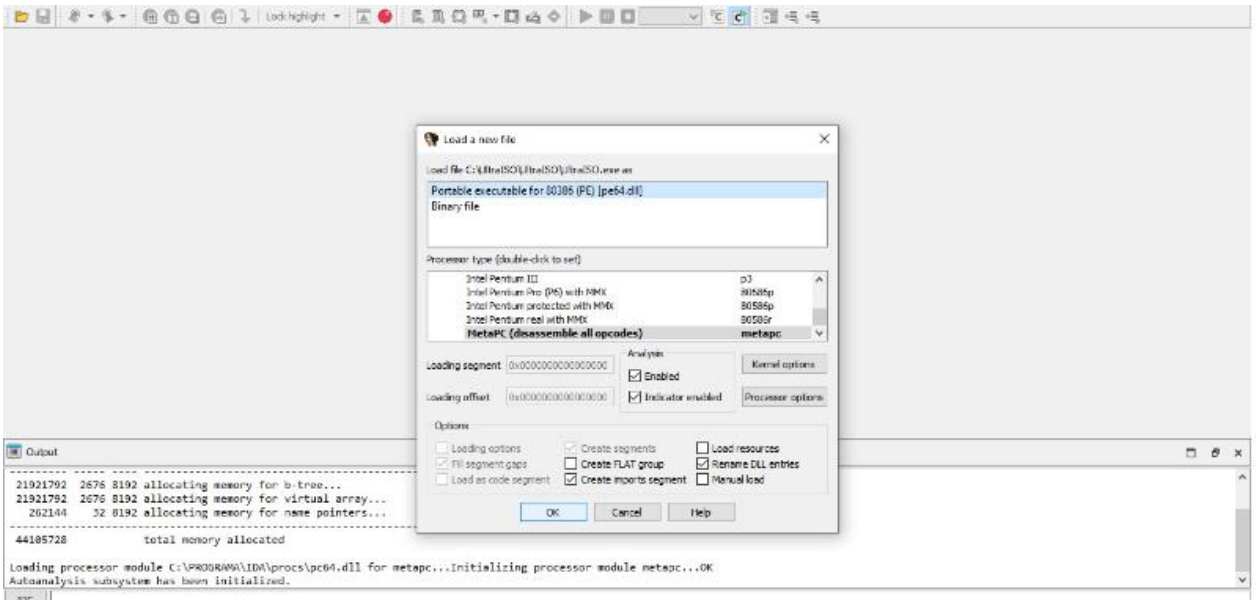
copiando la ruta del servicio



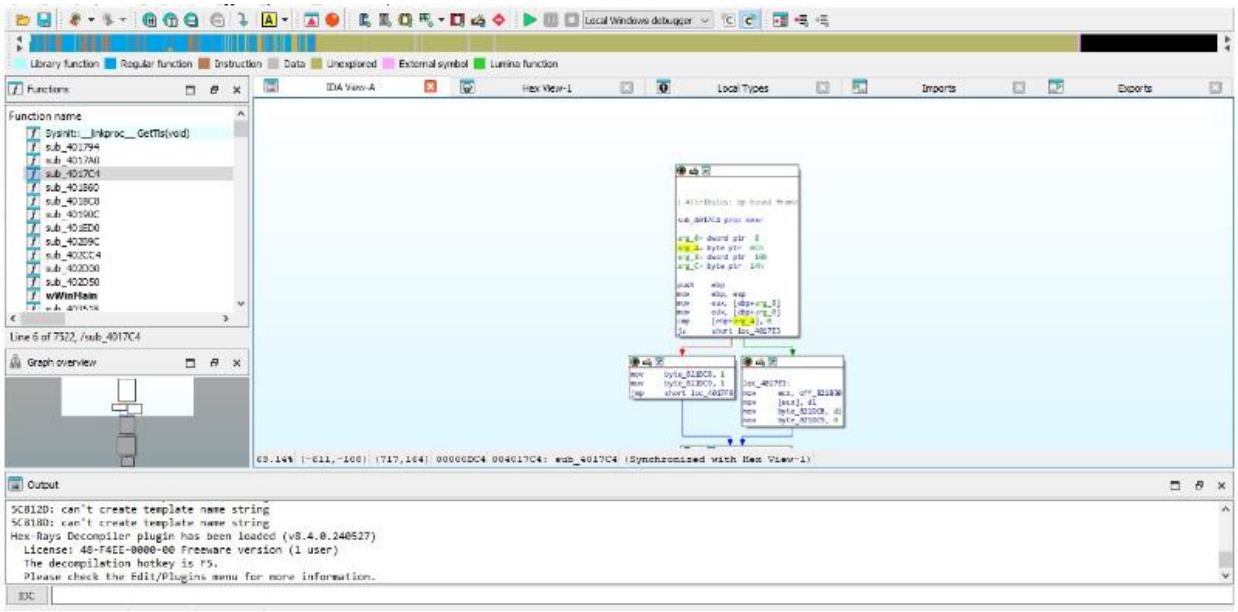
colocar la ruta en IDA, seleccionar el servicio



"ok"



Estructura



codigo assembler