


<div>UNIVERSIDAD AUTONOMA TOMAS FRIAS</div> <div></div>	<div>INGENIERIA DE SISTEMAS ARQUITECTURA DE COMPUTADORAS SIS-522</div> <div></div>	Practica 9
<div>ESTUDIANTE: Univ. Cesar Armando Sanabria Cáceres</div> <div>DOCENTE: Ing. Gustvo A Puita Choque</div> <div>AUXILIAR: Univ. Aldrin Roger Pérez Miranda</div> <div>FECHA DE ENTREGA: 05 / 12 / 24</div>		Calificación
<div>GRUPO: 1</div>		

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

Es una estructura de datos que se utiliza para almacenar información temporal durante la ejecución de un programa.

El stack se utiliza principalmente para:

- Guardar la dirección de retorno cuando se llama a una subrutina, permitiendo volver al punto correcto después.
- Pasar parámetros a las subrutinas.
- Almacenar variables locales.

Para manejarlo, se usan instrucciones como:

- PUSH: Coloca un valor en el tope del stack.
- POP: Saca un valor del tope del stack.
- CALL: Guarda la dirección de retorno en el stack y salta a una subrutina.
- RET: Recupera la dirección de retorno del stack y salta a esa dirección.

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

Un escenario sería en el desarrollo de controladores de dispositivos (drivers) para sistemas operativos .

Las ventajas son las siguientes:

- Permite acceso directo y preciso al hardware, algo crucial para controlar y configurar dispositivos.
- Optimiza el rendimiento a bajo nivel, esencial en sistemas de tiempo real donde la latencia es crítica.
- Da control absoluto sobre el uso de memoria y recursos del hardware, evitando problemas.
- Facilita la depuración detallada al tener visibilidad directa sobre el estado del procesador.

En conclusion, el ensamblador es mejor cuando se requiere control preciso del hardware, como en drivers de dispositivos, sistemas operativos de bajo nivel o aplicaciones de tiempo real.

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

1. MOV AX, 5;

- Esta línea mueve el valor 5 al registro AX.
- AX es un registro de propósito general del procesador.
- Esto asigna el valor 5 al registro AX.

2. MOV BX, 10;

- Esta línea mueve el valor 10 al registro BX.
- BX es otro registro de propósito general.
- Esto asigna el valor 10 al registro BX.

3. ADD AX, BX;

- Esta línea suma el contenido de BX al contenido de AX.
- El resultado de la suma se almacena en el registro AX.
- Esto efectúa la operación $AX = AX + BX$.

4. MOV CX, AX;

- Esta línea mueve el contenido del registro AX al registro CX.
- CX es otro registro de propósito general.
- Esto copia el resultado de la suma anterior (AX) al registro CX.

4) Explique detalladamente cómo funcionan los compiladores

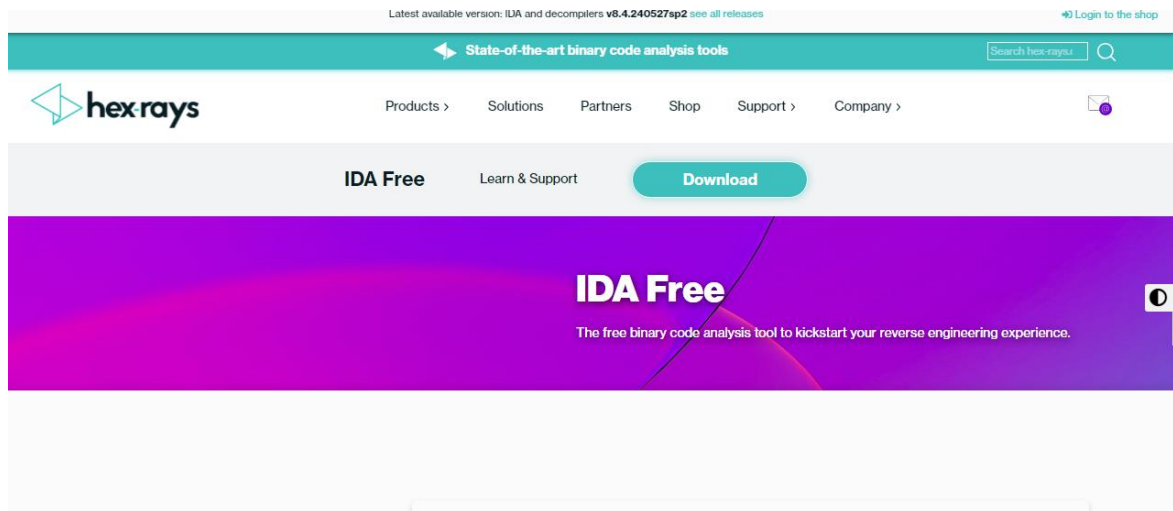
Los compiladores son herramientas que toman el código escrito en un lenguaje de alto nivel, como C, C++ o Java, y lo traducen a instrucciones de bajo nivel que la computadora pueda entender y ejecutar directamente, primero analizan sintácticamente el código fuente para verificar su estructura, luego realizan un análisis semántico para asegurar que el significado sea correcto, generan una representación intermedia que optimizan aplicando técnicas para hacerlo más eficiente, y finalmente traducen ese código optimizado a instrucciones de bajo nivel, como lenguaje ensamblador o código de máquina, que la computadora puede ejecutar sin problemas.

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/>



Download your IDA Free

The Free version of IDA v8.3 comes with the following limitations:

- no commercial use is allowed
- cloud-based decompiler lacks certain advanced commands
- lacks support for many processors, file formats, etc...
- comes without technical support



IDA Free for Windows (90MB)



IDA Free for Linux (76MB)



IDA Free for Mac (68MB)



IDA Free for Mac ARM (70MB)

SHA256 checksums:

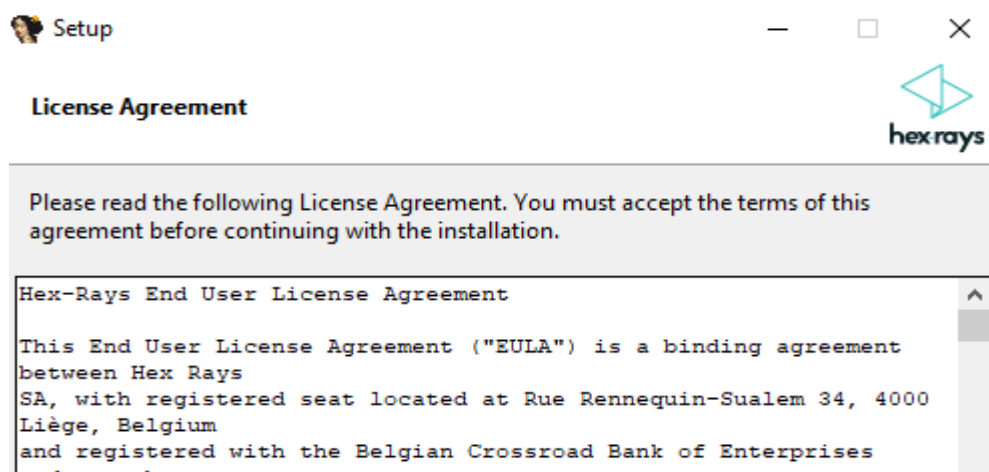
dec15875d071a398080a05320ac45e1971940de279abb0e2c57054833da18f6 arm_idafree84_mac.app.zip
941f228ce489b0f0a14268980edad16140be297e0749245d7839a2a30b2070a62 idafree84_linux.run
c4c78035d02d217351c0738c0a0d70b017aac3e7421e0b426c9fe2451e6f5 idafree84_mac.app.zip

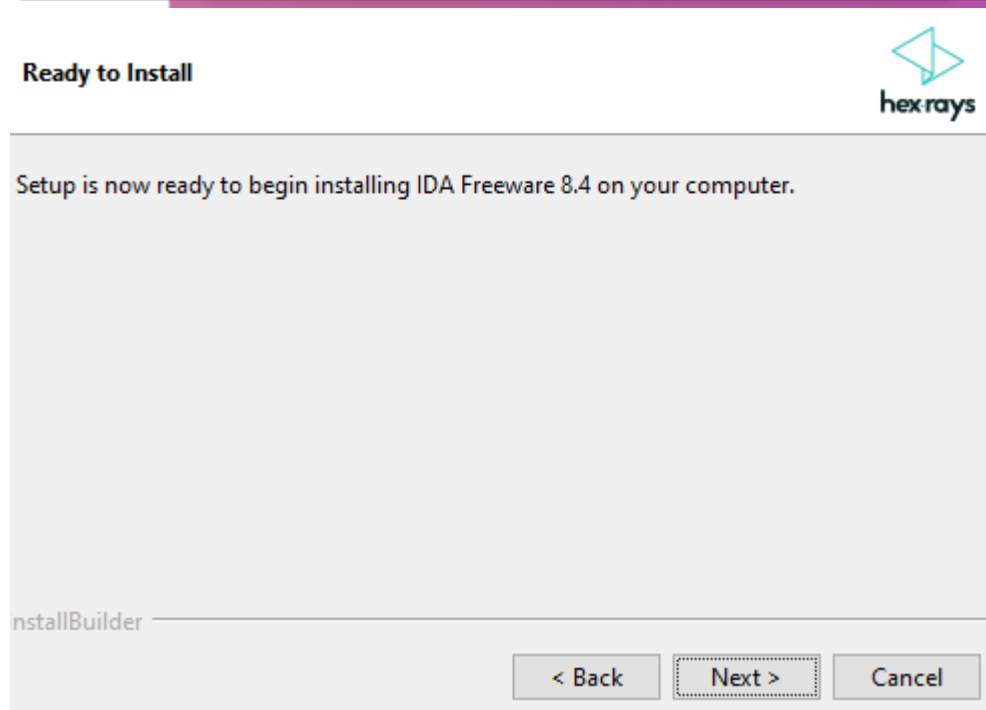
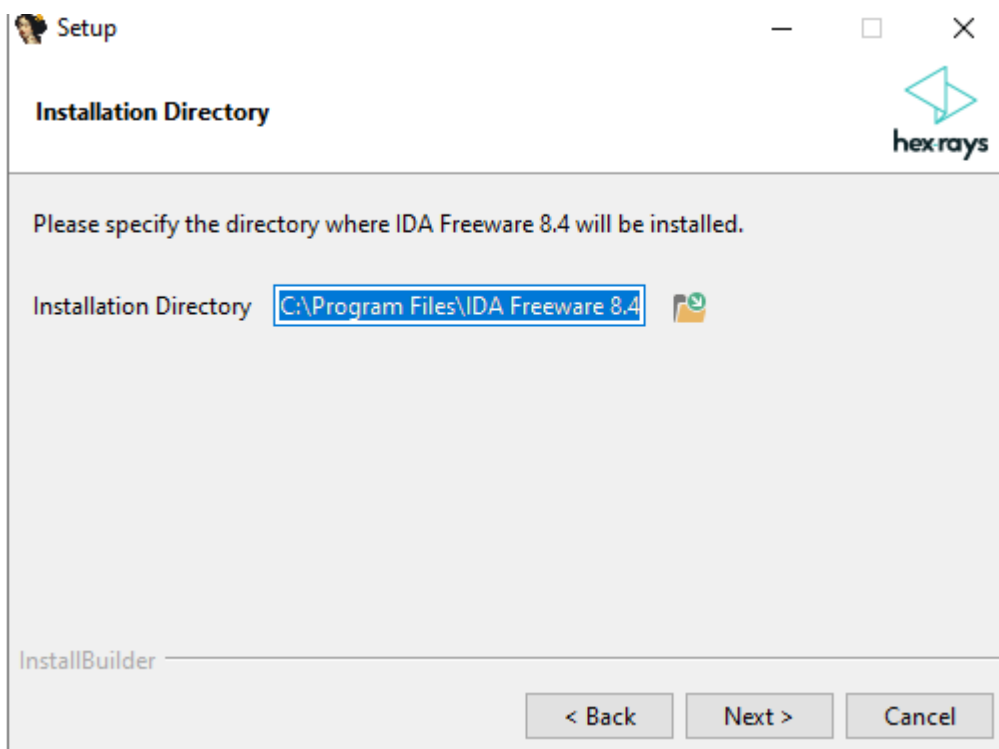
We use cookies to improve your experience on our website. [More info](#) de4112d66016ac2091 idafree84_windows.exe

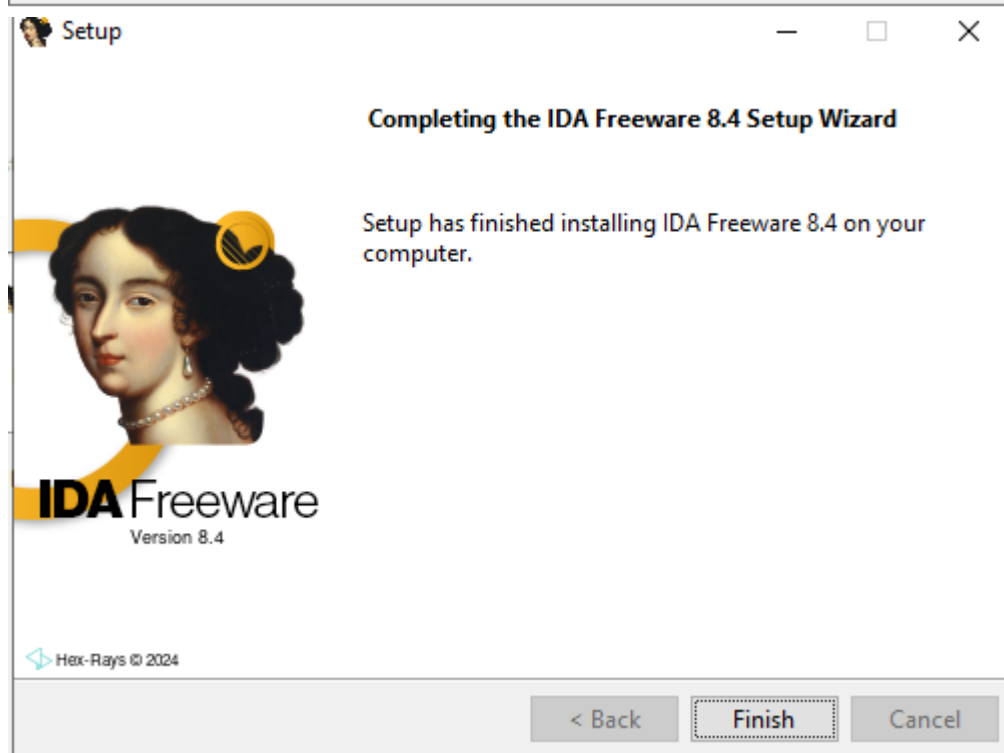
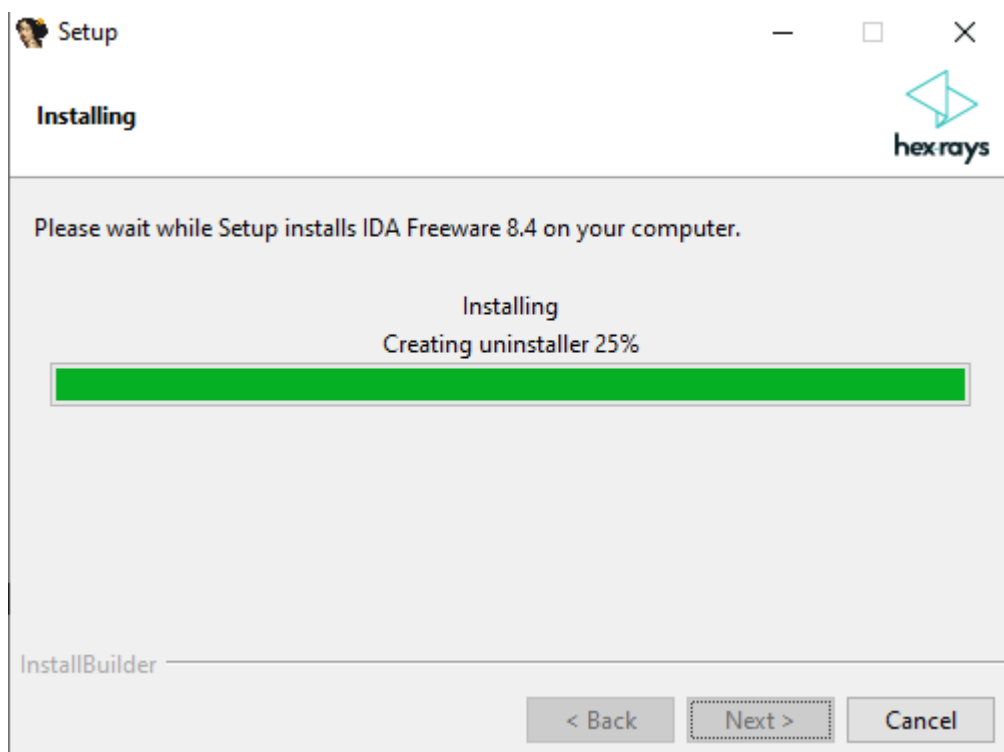
I'm okay with that

PASO 2

INSTALACION, Y SELECCIONE NEXT, Y SEGUI CON EL PROCESO DE INSTALACION, ACPETANDO LOS TERMINOS Y CONDICIONES, LA UBICACIÓN DE LA INSTALACION.



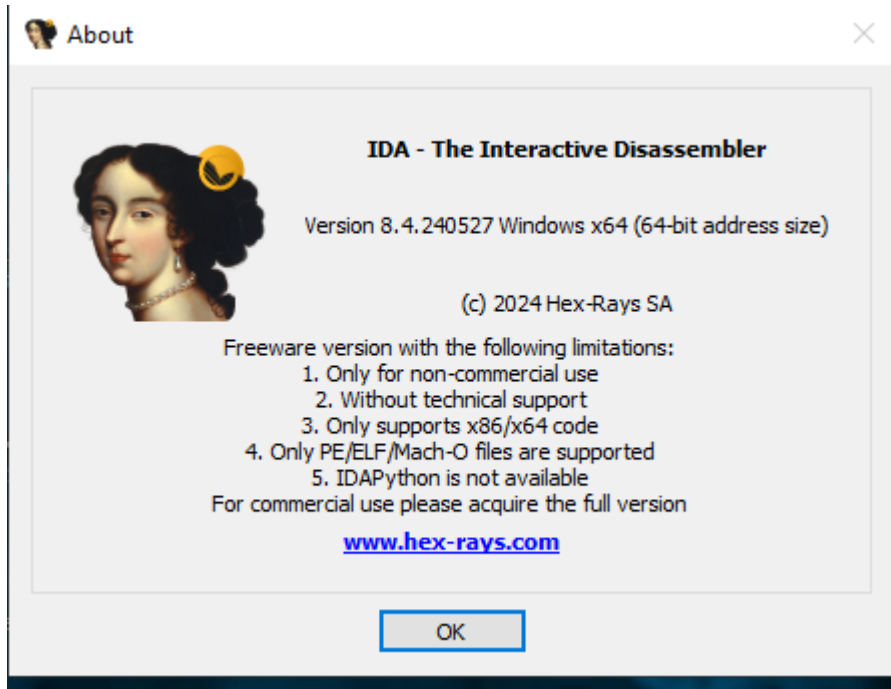




AQUÍ YA SE CREO EL ACCESO DIRECTO

PASO 3

SE PROCEDE A ABRIR EL PROGRAMA Y SELECCIONAR UN SERVICIO DE WINDOWS QUE PARA ESTE CASO ES UN SERVICIO DE OPERA GX, SE LO SELECCIONO DESDE EL ADMINISTRADOR DE TAREAS Y SE SE COPIO LA DIRECCION DEL SERVICIO DESDE LA UBICACIÓN DEL ARCHIVO.



Administrador de tareas

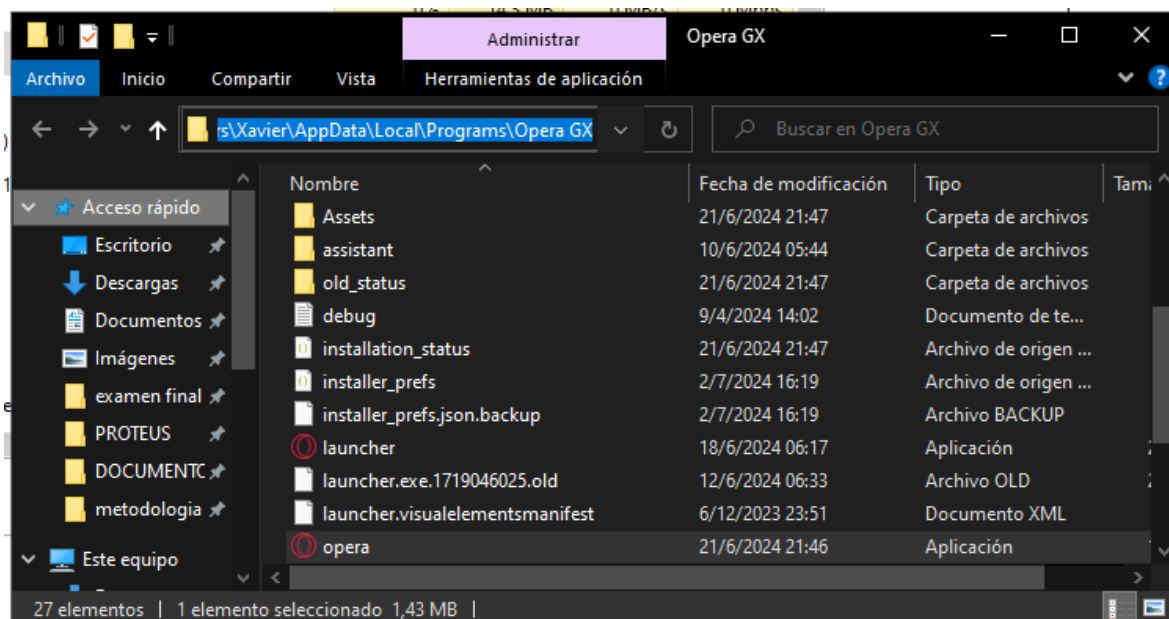
Archivo Opciones Vista

Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios

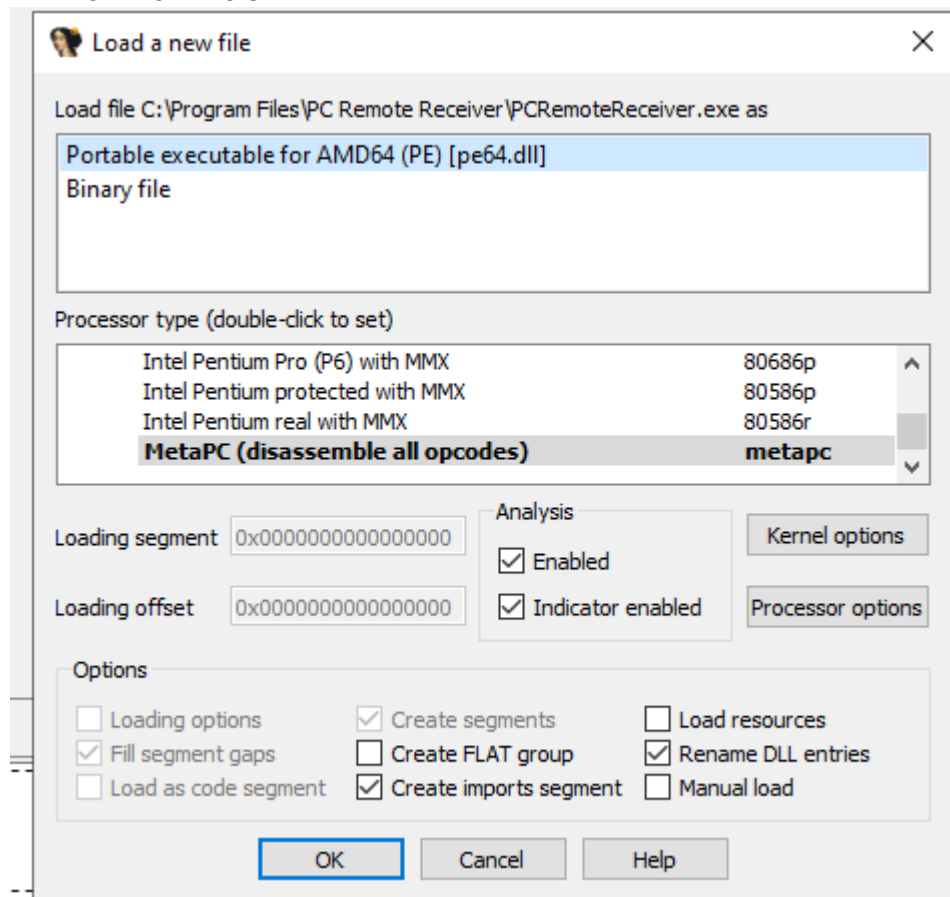
Nombre	Estado	12% CPU	67% Memoria	0% Disco	0% Red	C
Radeon Settings: Source Extension		0%	19,6 MB	0 MB/s	0 Mbps	
Opera GX Internet Browser		0%	18,1 MB	0,1 MB/s	0 Mbps	
> Host de servicio: Servicio local (sin red) (3)		0%	16,5 MB	0 MB/s	0 Mbps	
Radeon Settings: Host Service		0%	14,3 MB	0 MB/s	0 Mbps	
Opera GX Internet Browser		0%	13,5 MB	0 MB/s	0 Mbps	
> Host de servicios: Unistack Service Group (6)		0%	13,3 MB	0 MB/s	0 Mbps	
> Host de servicio: Servicio local (red restringida) (5)		0%	12,4 MB	0 MB/s	0 Mbps	
> Host de servicio: Sistema local (red restringida) (11)		0%	12,0 MB	0 MB/s	0 Mbps	
Opera GX Internet Browser		0%	11,8 MB	0 MB/s	0 Mbps	
> LocalServiceNoNetworkFirewall (2)		0%	11,4 MB	0 MB/s	0 Mbps	
> Host de servicio: Servicio local (11)		0%	10,8 MB	0 MB/s	0 Mbps	
> Recortes de pantalla		0%	10,4 MB	0 MB/s	0 Mbps	
> Microsoft Text Input Application		0%	9,4 MB	0 MB/s	0 Mbps	

< >

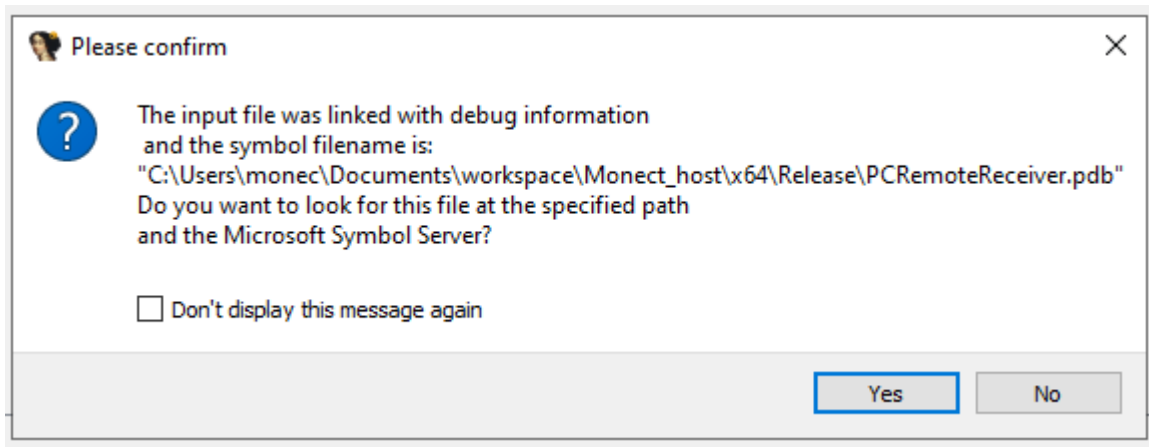
Menos detalles Finalizar tarea



UNA VES SE COPIO LA DIRECCION DEL ARCHIVO AL NUEVO PROYECTO DE IDA, SELECCIONAMOS OK PARA QUE ANALICE EL SERVICIO.

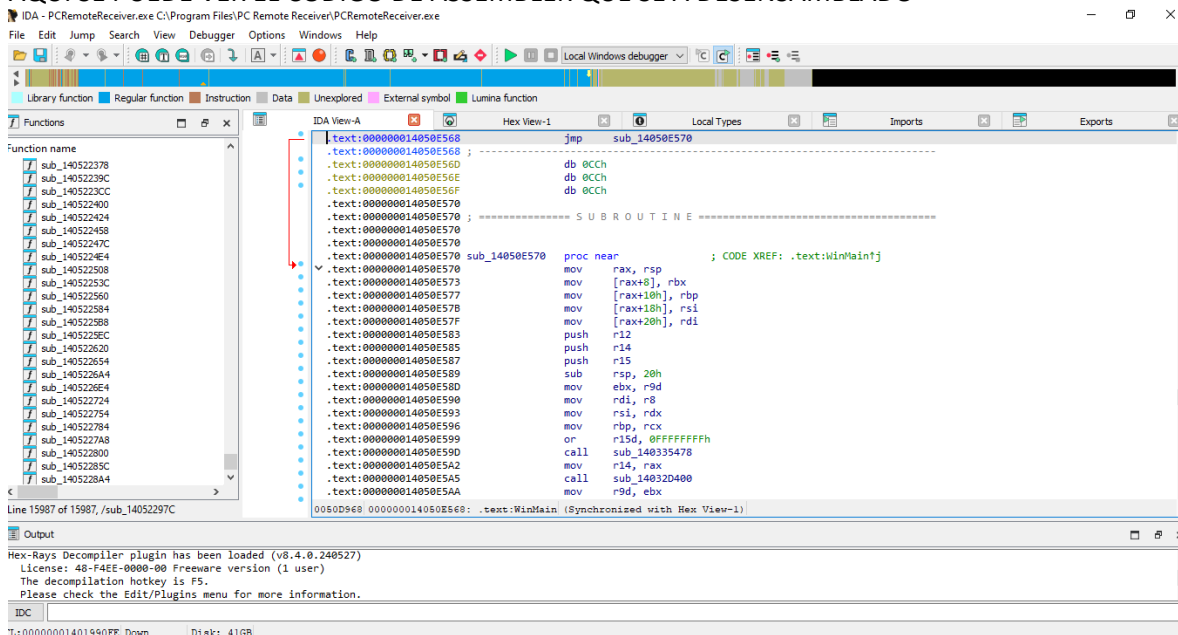


LUEGO COLOCAMOS NO

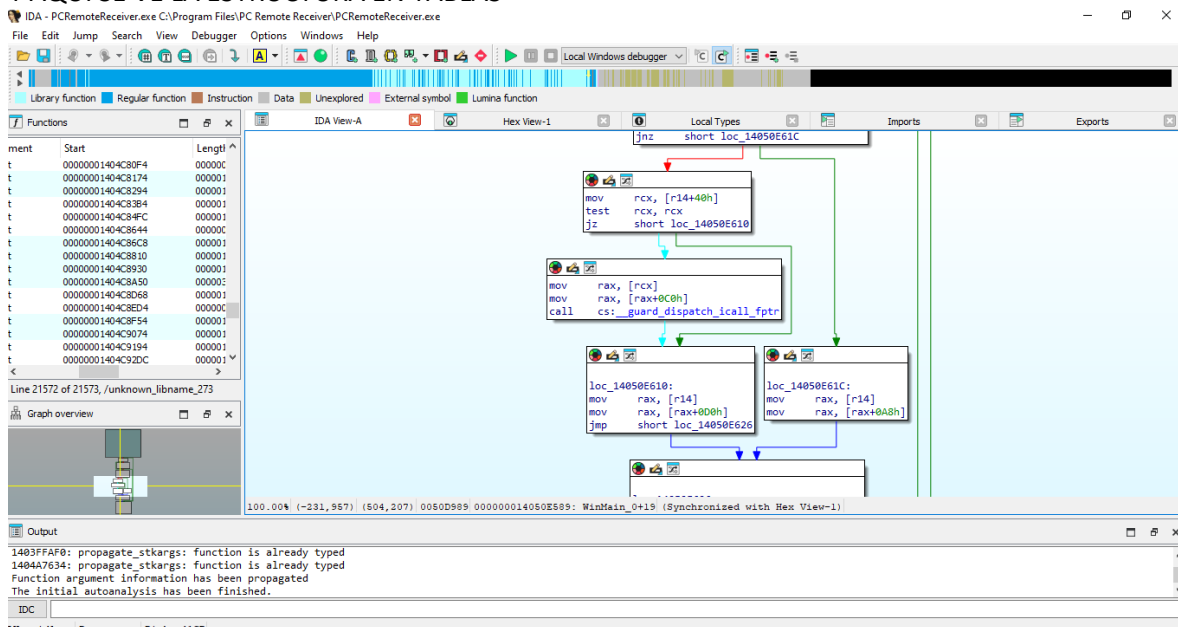


PASO 4

AQUÍ SE PUEDE VER EL CODIGO DE ASSEMBLER QUE SE A DESENSAMBLADO



Y AQUÍ SE VE LA ESTRUCTURA EN TABLAS



Y ESTO ES MAS DEL CODIGO ASSEMBLER

IDA - PCRemoteReceiver.exe C:\Program Files\PC Remote Receiver\PCRemoteReceiver.exe

File Edit Jump Search View Debugger Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions

ment	Start	Length
t	0000000140C80F4	00000C
t	0000000140C8174	000001
t	0000000140C8294	000001
t	0000000140C8384	000001
t	0000000140C84FC	000001
t	0000000140C8644	00000C
t	0000000140C86C8	000001
t	0000000140C8810	000001
t	0000000140C8930	000001
t	0000000140C8A50	00000C
t	0000000140C8D68	000001
t	0000000140C8ED4	00000C
t	0000000140C9074	000001
t	0000000140C9194	000001
t	0000000140C92DC	000001
t	0000000140C9424	00000C
t	0000000140C94A8	000001
t	0000000140C95F0	000001
t	0000000140C9710	000001
t	0000000140C9830	00000C
t	0000000140C9B48	00000C
t	0000000140CA0C4	00000C
t	0000000140CA640	00000F
t	0000000140CB0B8	00000F

Line 21572 of 21573, /unknown_libname_273

Current location

Hex View-1

```
.text:000000014050E590 mov rdi, r8
.text:000000014050E593 mov rsi, rdx
.text:000000014050E596 mov rbp, rcx
.text:000000014050E599 or r15d, 0FFFFFFFh
.text:000000014050E59D call ?AfxGetThread@YAPEAVCWinThread@@@KZ ; AfxGetThread(void)
.text:000000014050E5A2 mov r14, rax
.text:000000014050E5A5 call ?AfxGetModuleState@YAPEAVAFX_MODULE_STATE@@@KZ ; AfxGetModuleState(void)
.text:000000014050E5AA mov r9d, ebx
.text:000000014050E5AD mov r8, rdi
.text:000000014050E5B0 mov rdx, rsi
.text:000000014050E5B3 mov rcx, rbp
.text:000000014050E5B6 mov r12, [rax+8]
.text:000000014050E5BA call unknown_libname_470 ; MFC 7-14 64bit
.text:000000014050E5BF test eax, eax
.text:000000014050E5C1 jz short loc_14050E632
.text:000000014050E5C3 test r12, r12
.text:000000014050E5C6 jz short loc_14050E5E0
.text:000000014050E5C8 mov rax, [r12]
.text:000000014050E5CC mov rcx, r12
.text:000000014050E5CF mov rax, [rax+158h]
.text:000000014050E5D6 call cs:_guard_dispatch_icall_fptr
.text:000000014050E5DC test eax, eax
.text:000000014050E5DE jz short loc_14050E632
.text:000000014050E5E0 loc_14050E5E0: ; CODE XREF: WinMain_0+56fj
.text:000000014050E5E8 mov rax, [r14]
.text:000000014050E5E3 mov rcx, r14
0050D9CF 000000014050E5CF: WinMain_0+5F (Synchronized with Hex View-1)
```

Output

1403FFAF0: propagate_stkargs: function is already typed
1404A7634: propagate_stkargs: function is already typed
Function argument information has been propagated
The initial autoanalysis has been finished.

IDA

AU: idle Down Disk: 41GB