



هرگاه بنده ای کارش را با نام و یاد خدا آغاز میکند ، خداوند به فرشته ها میگوید: بنده ای من کارش را با نام من آغاز کرد ، پس بر من واجب شد که کارش را آسان گردانم . امام رضا (ع)

Meisam Monsef
meisamrce@yahoo.com

meisamrce@gmail.com

آموزش قویترین دیباگر جهان Olly Debugger v1.10

نوشته : میثم منصف

دانش رو طلب کنید اگر به فرورفتن در خون ها و شکافتن دریاها باشد
دوست ندارم جوانی را مگر این که دانشمند یا دانش آموز باشد.

امام جعفر صادق (ع)

باگ (Bug):

به عمده مشکلات نرم افزاری که باعث می شود نرم افزار، کار یا عملیات خود را به درستی انجام ندهد باگ گفته میشود . این باگ ها عمدتاً ناشی از اشتباهات برنامه نویس میباشد . به بررسی این مشکلات (باگ ها) ، عملیات اشکال زدایی یا دیباگ (Debug) کردن می گویند ، که معمولاً توسط برنامه نویس انجام می شود .

دیباگر (Debugger):

برای رفع باگ های برنامه از دیباگر استفاده می شود ، دقت کنید که هر برنامه در محیط خاصی نوشته می شود ، که معمولاً به IDE معروف هست ، برنامه نویس برنامه را در حالت دیباگ اجرا کرده و حالت های متفاوت را بررسی میکند ، و با توجه به پیغام دیباگر IDE ، برنامه نویس مشکلات را حل میکند . این در حالتی میباشد که کد سورس برنامه وجود دارد و مشکلات برنامه را با تغییر در کد ، بر طرف میکند ، ولی ما می خواهیم نگاهی به دیباگر های به اندازیم که بدون داشتن فایل سورس کد و تنها با داشتن فایل اجرایی (EXE) به عملیات دیباگ کردن برنامه به پردازیم .

انواع دیباگر ها :

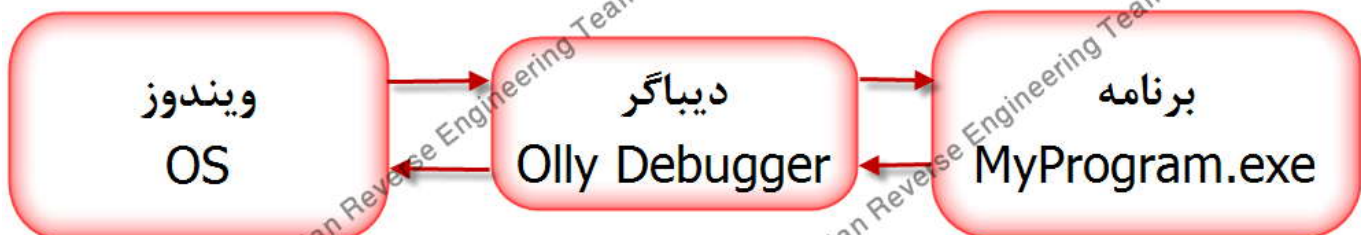
۲- دیباگرهای سیستمی :

این دیباگر ها برای آنالیز و بررسی مشکلات درایور های سیستمی مورد استفاده قرار می گیرد .

۲- دیباگرهای نرم افزاری :

این دیباگر ها برای آنالیز و بررسی مشکلات نرم افزارها مورد استفاده قرار می گیرد و معروفترین آن ها Olly Debugger می باشد .

برای درک بهتر کار دیباگرهای نرم افزاری به تصویر (1) نگاه کنید :



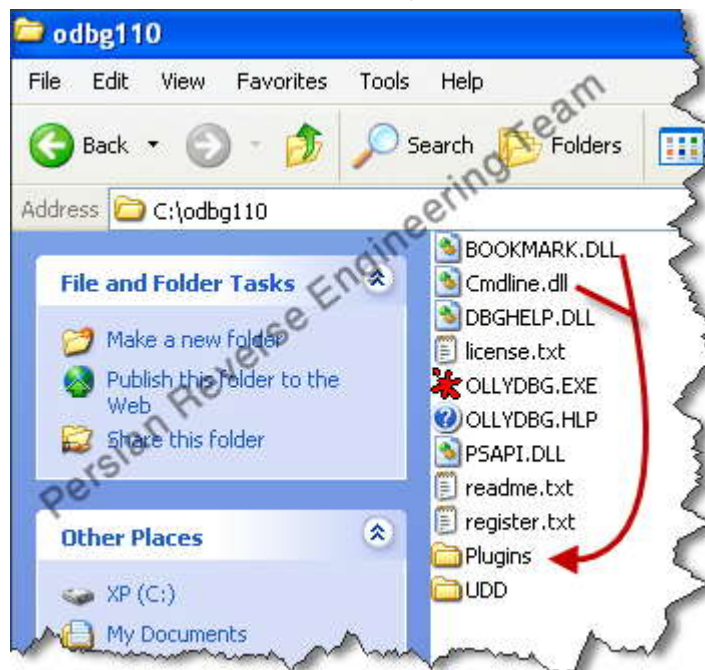
تصویر (1)

همانطور که می بینید دیباگر در بین ویندوز و برنامه قرار گرفته و هر داده های که بین این دو (ویندوز و برنامه) عبور میکند قابل رویت میباشد ، در اینجا ما می توانیم تمام داده ها و دستورات برنامه را آنالیز کنیم حتی آن ها را نیز تغییر بدهیم .

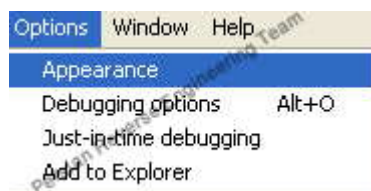


دانلود و تنظیمات Olly Debugger :

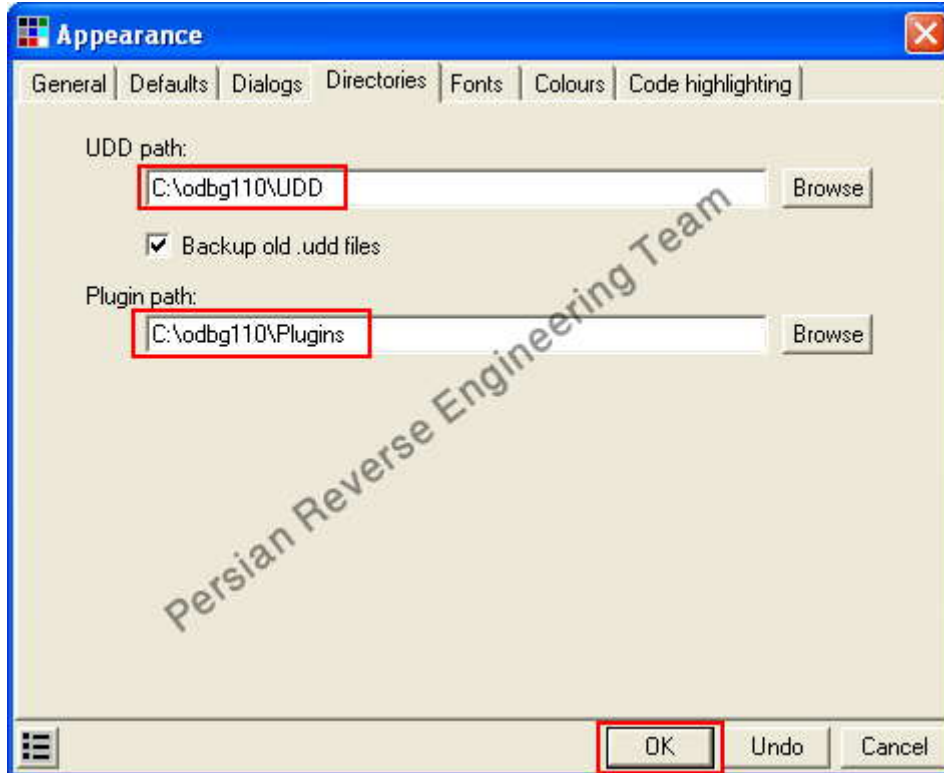
- Olly Debugger نسخه 1.10 را این لینک [دانلود](#) کنید ، و در مسیر خاصی مثلا (c:\odbg110) قرار دهید .
دقت کنید این برنامه نسخه های دیگری هم دارد ولی بهترین نسخه آن ، نسخه 1.10 می باشد .
در مسیری که Ollydbg را قرار دادید دو پوشه به نام های زیر بسازید :
۱- Plugins : این پوشه محل قرار گرفتن پلاگین های اضافی میباشد .
۲- UDD : فایل های آنالیز (udd) ، برنامه ها و فایل های پشتیبانی (bak) در این پوشه قرار می گیرد .
بعد دو فایل BOOKMARK.DLL و Cmdline.dll را در داخل پوشه Plugins قرار دهید .




- خب حالا فایل را اجرا کنید ، در کارد باز شده پیغام را Yes را کلیک کنید .
۱- از منوی Options گزینه Appearance یا از نوار ابزار روی آیکون کلیک کنید تا تنظیمات مربوط به مسیرها Plugins و UDD را به توانیم تنظیم کنیم.

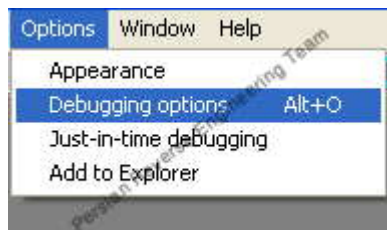


در کادر باز شده روی تب Directories کلیک کنید ، و دو مسیر UDD path و Plugin path با زدن دکمه Browse مانند شکل زیر تنظیم کنید ، و بعد روی دکمه Ok کلیک کنید .



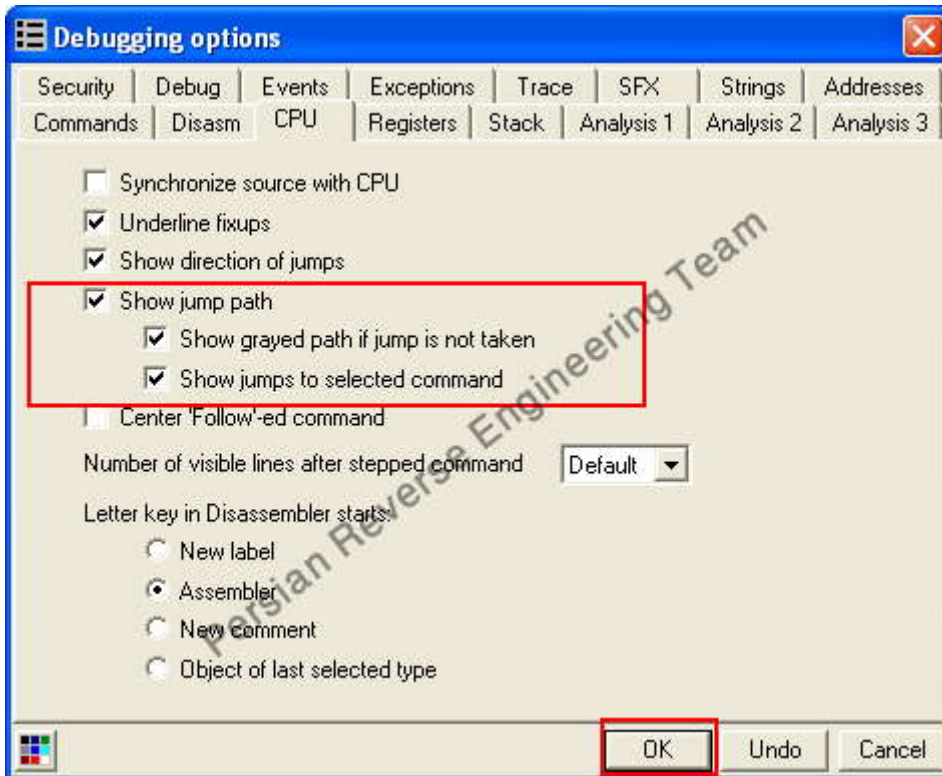
بعد پیغامی مبنی بر restart کردن برنامه میدهد که روی دکمه Ok کنید و برنامه را بسته و دوباره باز کنید .

۲- از منوی Options گزینه Debugging options یا از نوار ابزار آیکون  را کلیک کنید (میتوانید باز زدن دکمه های Alt+O نیز این کار را انجام بدهید) تا کادر تنظیمات برنامه باز شود .

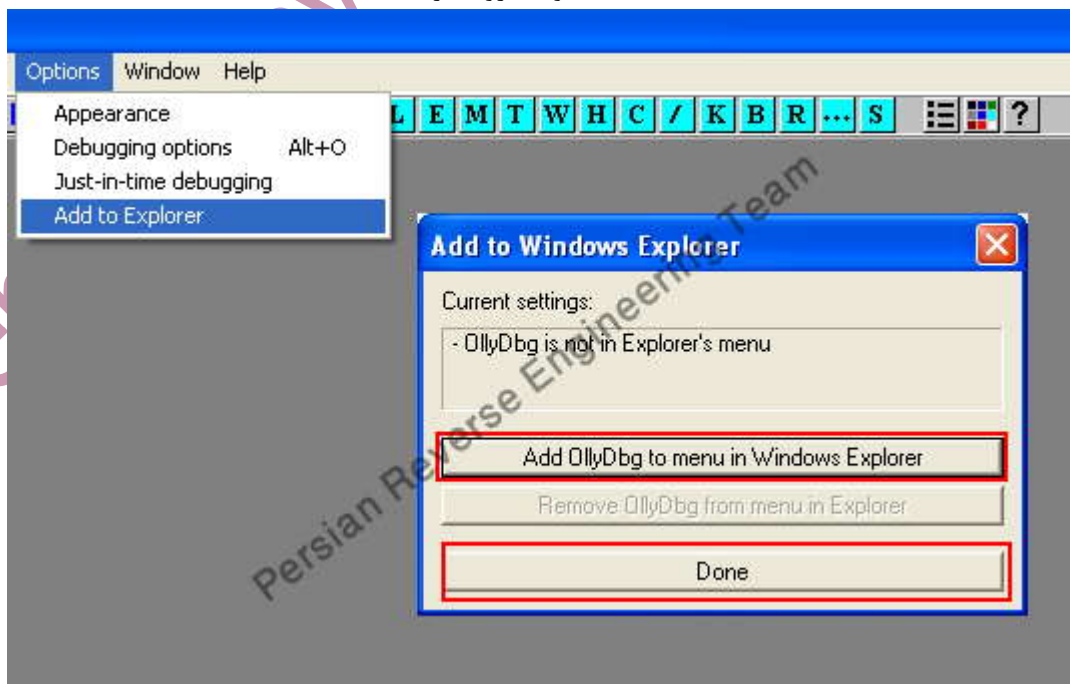




روی تب CPU کلیک کرده و با توجه به شکل زیر سه گزینه ... show را فعال کنید و بعد روی دکمه Ok کلیک کنید .

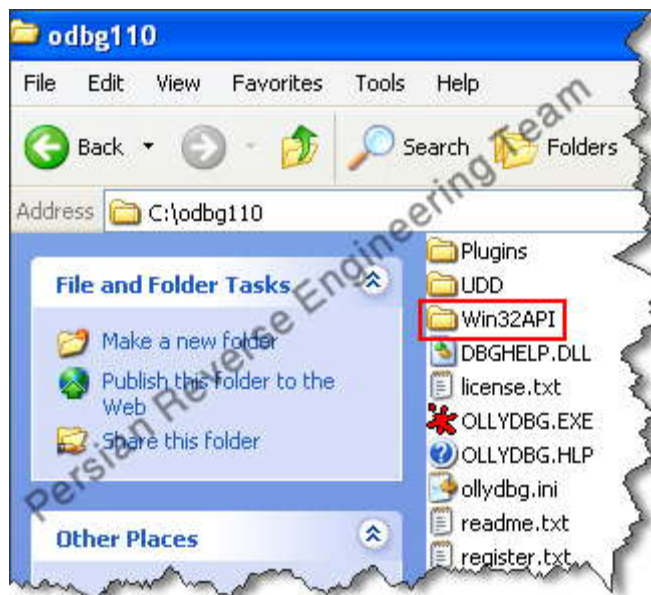


۳- از منوی Options گزینه Add to Explorer را کلیک کنید ، در کادر باز شده روی گزینه Add OllyDbg to menu Window Explorer کلیک کنید و بعد روی گزینه Done کلیک کنید.

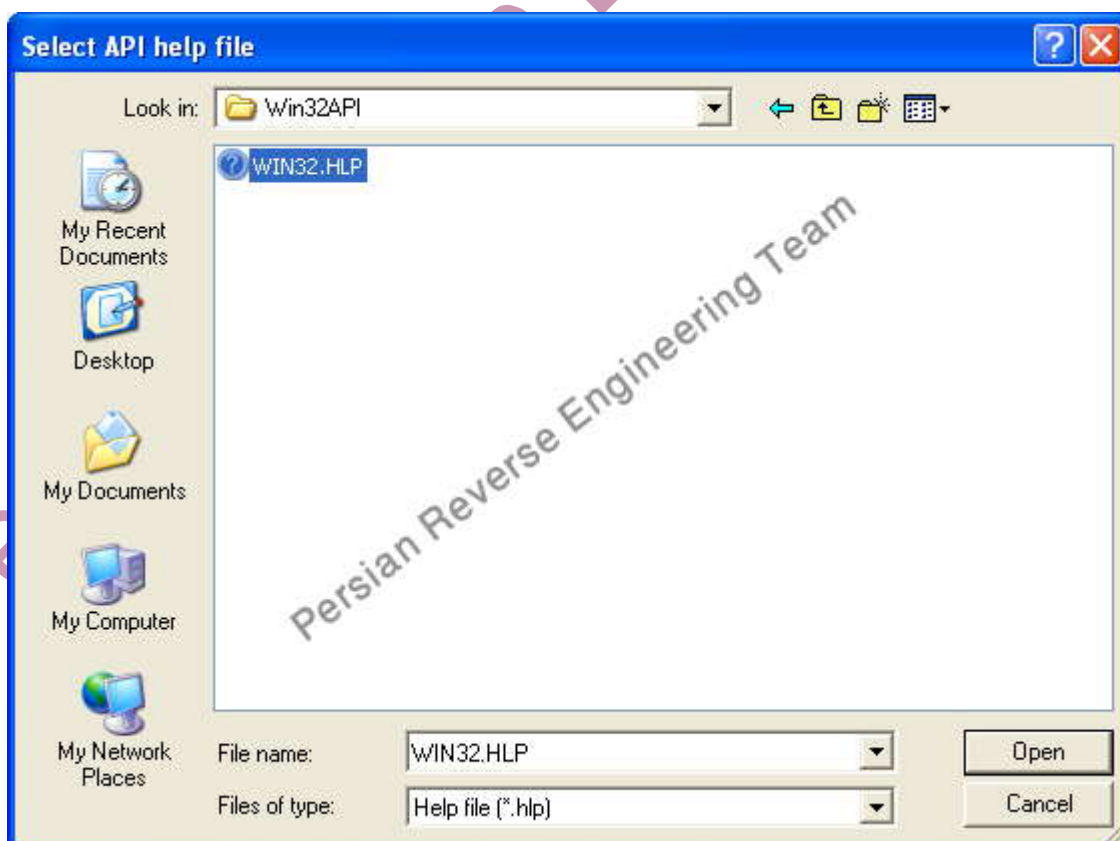




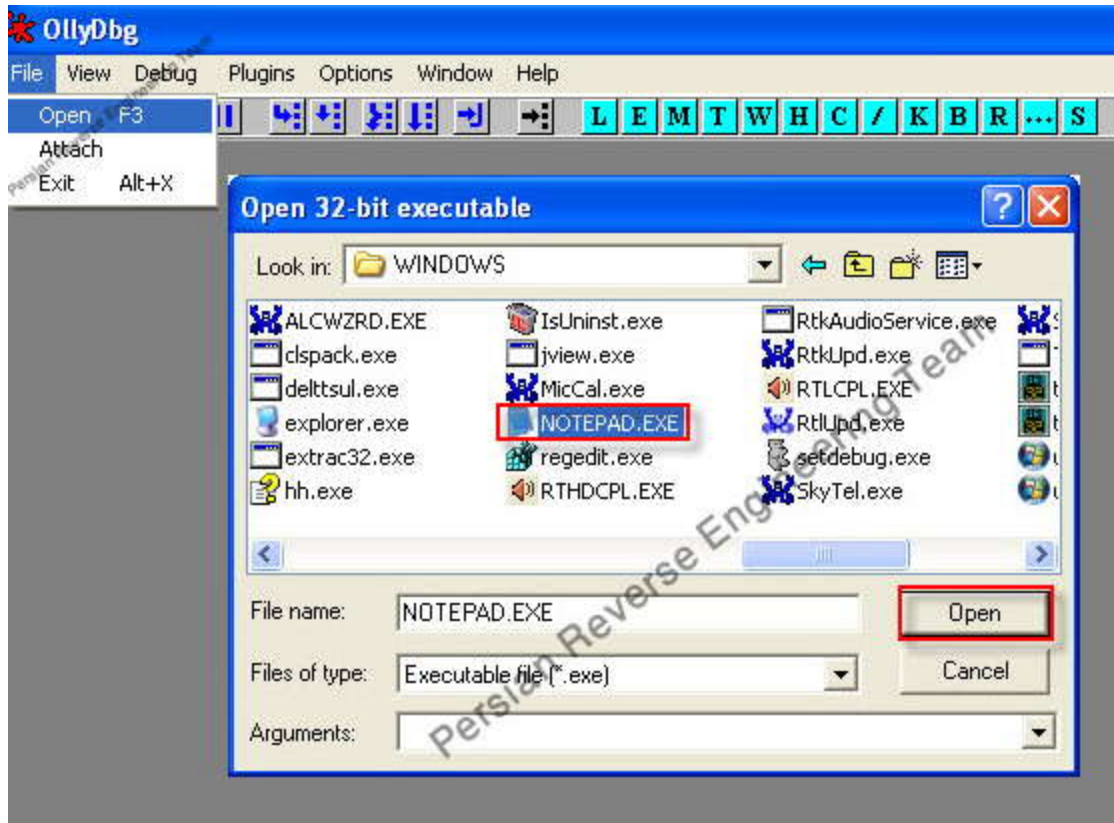
۴- فایل مرجع توابع API را از این لینک [دانلود](#) کرده و آن را در مسیر مورد نظر قرار دهید.



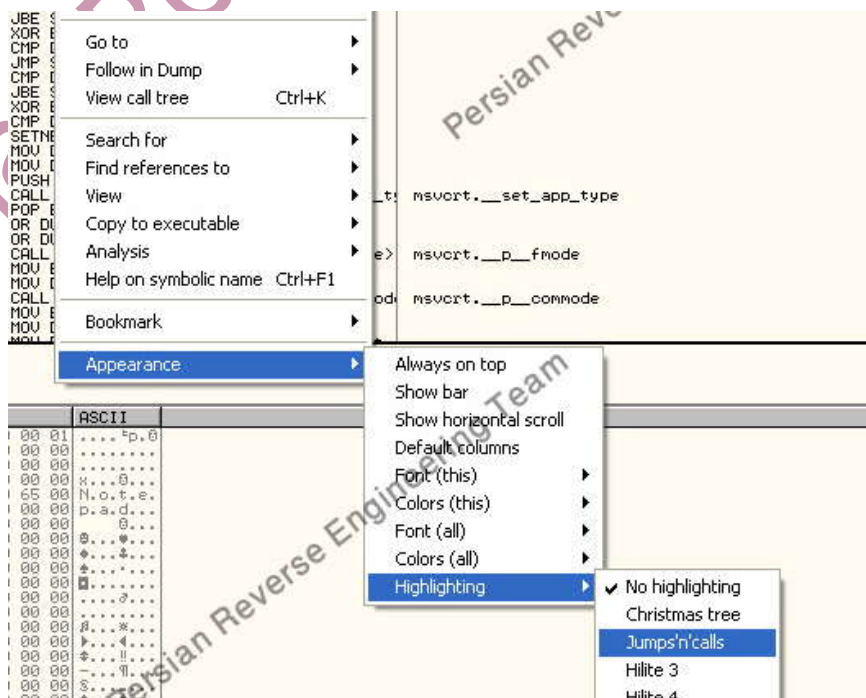
خب از منوی Help گزینه Select API help file را انتخاب کرده و در کادر باز شده فایل WIN32.HLP را از پوشه Win32API انتخاب کنید.



۵- از منوی File گزینه Open را کلیک می کنیم و یک فایل exe مثل notepad.exe را از مسیر C:\WINDOWS\ باز کنید.



در پنجره باز شده (CPU) روی صفحه اصلی کلیک راست کرده و گزینه Appearance، گزینه Highlighting و گزینه Jumps'n'calls را انتخاب کنید.





با اجرای این تنظیمات ، دستورات پرشی به صورت زرد رنگ و دستورات توابع به رنگ سبز شود .

```
00 $ 6A 70 PUSH 70
0F . 68 98180001 PUSH NOTEPAD.01001858
14 . E8 BF010000 CALL NOTEPAD.01007568
19 . 330B XOR EBX,EBX
1B . 53 PUSH EBX
1C . 8B3D CC10000 MOV EDI,DWORD PTR DS:[&&KERNEL32.GetMod
1E . FFD7 CALL EDI
1F . 66:8138 4D50 CMP WORD PTR DS:[EAX] 504D
24 . 75 1F JNZ SHORT NOTEPAD.010073DA
2B . 8B48 3C MOV ECX,DWORD PTR DS:[EAX+3C]
2E . 03C8 ADD ECX,EAX
30 . 8139 50450000 CMP DWORD PTR DS:[ECX],4550
36 . 75 12 JNZ SHORT NOTEPAD.010073DA
38 . 0FB741 18 MOVZX EAX,WORD PTR DS:[ECX+18]
3C . 3D 0B010000 CMP EAX,10B
3E . 74 1F JE SHORT NOTEPAD.010073F2
40 . 3D 0B020000 CMP EAX,20B
42 . 74 05 JE SHORT NOTEPAD.0100730F
44 . 895D E4 MOV DWORD PTR SS:[EBP-1C],EBX
46 . EB 27 JMP SHORT NOTEPAD.01007406
48 . 83B9 84000000 CMP DWORD PTR DS:[ECX+84],0E
4C . 76 F2 JBE SHORT NOTEPAD.010073DA
4E . 33C0 XOR EAX,EAX
50 . 3999 F8000000 CMP DWORD PTR DS:[ECX+F8],EBX
52 . EB 0E JMP SHORT NOTEPAD.01007400
54 . 8379 74 0E CMP DWORD PTR DS:[ECX+74],0E
56 . 76 E2 JBE SHORT NOTEPAD.010073DA
58 . 33C0 XOR EAX,EAX
5A . 3999 E8000000 CMP DWORD PTR DS:[ECX+E8],EBX
5C . 0F95C0 SETNE AL
5E . 8945 E4 MOV DWORD PTR SS:[EBP-1C],EAX
```

خب تنظیمات برنامه ollydbg پایان رسید حال به آموزش قسمت های برنامه می پردازیم .

نوارعنوان Title Bar :



1

2

۱- نام برنامه مورد آنالیز

۲- نام ماژول فعال برنامه : هر برنامه دارای قسمت های مختلف می باشد که به آن ماژول می گویند ، البته فایل های dll در برنامه نیز به عنوان ماژول محسوب میشوند . برای دیدن لیست ماژول های برنامه روی پنجره کلیک راست کرده و از منوی باز شده ، گزینه View را انتخاب کرده ، همینطور که می بینید ، لیست تمام ماژول و dll های استفاده در برنامه لیست می شود ، با کلیک روی نام هر ماژول ، کدهای هر ماژول نمایش داده می شود .



Address	Hex dump	Disassembly	Comment
0100739D	40 2a	PUSH 2a	
0100739F		Backup	
010073A4		Copy	
010073A9		Binary	[<&KERNEL32.GetModul
010073AB		Assemble	Space
010073AC		Label	:
010073B2		Comment	
010073B4		Breakpoint	
010073B9		Hit trace	
010073BB		Run trace	
010073BE		New origin here	Ctrl+Gray *
010073C0		Go to	
010073C6		Follow in Dump	
010073C8		Search for	
010073C9		Find references to	
010073CA		View	Module 'UxTheme'
010073CB		Copy to executable	Module 'ShimEng'
010073CC		Analysis	Module 'serwvdrv'
010073CD		Bookmark	Module 'LPK'
010073CE		Dump debugged process	Module 'AcGeneral'
010073CF		Appearance	Module 'WINSPOOL'
010073D0	A1 2C130001	MOV EAX,DWORD PTR DS:	Module 'USP10'
010073D1	8B00	MOV EAX,DWORD PTR DS:	Module 'IMM32'
010073D2	A3 A4AB0001	MOV DWORD PTR DS:[14	Module 'comdlg32'
010073D3	E8 A7010000	CALL NOTEPAD.010075F	Module 'USERENV'
010073D4	391D 08960000	CMP DWORD PTR DS:[14	Module 'WINMM'
010073D5	75 0C	JNZ SHORT NOTEPAD.01	Module 'OLEAUT32'
010073D6	68 F4750001	PUSH NOTEPAD.010075F	Module 'COMCTL32'
010073D7	FF15 28130000	CALL DWORD PTR DS:[14	Module 'ole32'
010073D8	59	POP ECX	Module 'MSACM32'
010073D9	E8 77010000	CALL NOTEPAD.010075F	Module 'VERSION'
010073DA	68 10900001	PUSH NOTEPAD.0100900	Module 'msvcrt'
010073DB	68 0C900001	PUSH NOTEPAD.0100900	Module 'ADVAPI32'
010073DC	E8 5D010000	CALL <JMP.&msvcrt.	Module 'RPCRT4'
010073DD	A1 B09A0001	MOV EAX,DWORD PTR DS:	Module 'GDI32'
010073DE	8945 DC	MOV DWORD PTR SS:[E	
010073DF	8D45 DC	LEA EAX,DWORD PTR SS	
010073E0	50	PUSH EAX	
010073E1	FF35 AC9A0000	PUSH DWORD PTR DS:[
010073E2	8D45 D4	LEA EAX,DWORD PTR SS	
010073E3	50	PUSH EAX	
010073E4	8D45 D0	LEA EAX,DWORD PTR SS	
010073E5	50	PUSH EAX	
010073E6	8D45 CC	LEA EAX,DWORD PTR SS	

لیست ماژول های برنامه

نوار منو Menu Bar :



1

2

- ۱- نام پنجره فعال
- ۲- منو های اصلی برنامه

نوار ابزار Tool Bar :



این قسمت میانبر های از منو ها می باشد .

معرفی منوهای پر کاربرد :

کاربرد	آیکون نوار ابزار	کلید میانبر	نام منو
باز کردن فایل Exe-dll برای آنالیز		F3	File → Open
بارگذاری دوباره برنامه برای آنالیز		Ctrl+F2	Debug → Restart
بستن برنامه باز شده برای آنالیز		Alt+F2	Debug → Close
شروع و اجرای برنامه برای آنالیز		F9	Debug → Run
توقف آنالیز برنامه		F12	Debug → Pause
آنالیز خط به خط برنامه و رفتن داخل تابع ها		F7	Debug → Step into
آنالیز خط به خط برنامه و رد کردن تابع ها		F8	Debug → Step over
نمایش پنجره اصلی برنامه CPU		Alt+C	View → CPU
مدیریت نقاط توقف Break Points		Alt+B	View → Breakpoints

دقت داشته باشید که منوها و گزینه های دیگر در مقالات بعدی آموزش خواهد داده شد .



معرفی پنجره اصلی (CPU) برنامه ollydbg :

این پنجره از ۷ قسمت تشکیل شده و مهمترین پنجره از ، پنجره های دیگر میباشد ، به توضیح هر قسمت از این پنجره (CPU) می پردازیم .

The screenshot displays the OllyDbg interface for the process NOTEPAD.EXE. The main window is divided into several panes:

- Registers (FPU):** Shows the state of CPU registers. Register EAX is highlighted with a red box labeled '2'. The value of EAX is 00000000.
- Disassembly:** Shows the current instruction being executed. Instruction 1 is highlighted with a red box labeled '1'. It is a CALL instruction: CALL NOTEPAD.01007568.
- Information:** A section at the bottom left with a red box labeled '3'.
- Memory Dump:** Shows the memory dump at the current instruction pointer. Instruction 5 is highlighted with a red box labeled '5'. The dump shows hex values and their ASCII representation.
- Registers (FPU) - Value:** A table showing the values of registers. Address 0007FFC4 is highlighted with a red box labeled '4'. The value is 7C910738.
- Program entry point:** A section at the bottom with a red box labeled '7'.
- Paused:** A red box labeled '6' is located at the bottom right of the interface.



قسمت 1 پنجره CPU :

این قسمت شامل چهار بخش می باشد :

Address	Hex dump	Disassembly	Comment
01007390	6A 70	PUSH 70	
0100739F	68 98180001	PUSH NOTEPAD.01001898	
010073A4	E8 BF010000	CALL NOTEPAD.01007568	
010073A9	33DB	XOR EBX,EBX	
010073AB	53	PUSH EBX	
010073AC	8B3D CC10000	MOV EDI,DWORD PTR DS:[<&KERNEL32.GetMod	pModule => NULL
010073B2	FFD7	CALL EDI	kernel32.GetModuleHandleA
010073B4	66:8138 4D5A	CMP WORD PTR DS:[EAX],5A4D	GetModuleHandleA
010073B9	75 1F	JNZ SHORT NOTEPAD.010073DA	
010073BB	8B48 3C	MOV ECX,DWORD PTR DS:[EAX+3C]	
010073BE	03C8	ADD ECX,EAX	
010073C0	3D 00010000	CMP DWORD PTR DS:[ECX],4550	
010073C6	75 12	JNZ SHORT NOTEPAD.010073DA	
010073C8	0EB7 18	MOVZX EAX,WORD PTR DS:[ECX+18]	
010073D1	74 1F	CMP EAX,10B	
010073D3	3D 0B020000	CMP EAX,20B	
010073D8	74 05	JE SHORT NOTEPAD.010073DF	
010073DA	895D E4	MOV DWORD PTR SS:[EBP-1C],EBX	
010073DD	EB 27	JMP SHORT NOTEPAD.01007406	
010073DF	83B9 8400000	CMP DWORD PTR DS:[ECX+84],0E	
010073E6	76 F2	JBE SHORT NOTEPAD.010073DA	
010073E8	33C0	XOR EAX,EAX	
010073EA	3999 F800000	CMP DWORD PTR DS:[ECX+F8],EBX	
010073F0	EB 0E	JMP SHORT NOTEPAD.01007400	
010073F2	8379 74 0E	CMP DWORD PTR DS:[ECX+74],0E	
010073F6	76 E2	JBE SHORT NOTEPAD.010073DA	
010073F8	33C0	XOR EAX,EAX	
010073FA	3999 F800000	CMP DWORD PTR DS:[ECX+E8],EBX	
01007400	0F95C0	SETNE AL	
01007403	8945 E4	MOV DWORD PTR SS:[EBP-1C],EAX	
01007406	895D FC	MOV DWORD PTR SS:[EBP-4],EBX	
01007409	6A 02	PUSH 2	
0100740B	FF15 3813000	CALL DWORD PTR DS:[<&msvrt.__set_app_t	msvrt.__set_app_type
01007411	59	POP ECX	
01007412	830D 9CAB000	OR DWORD PTR DS:[100AB9C],FFFFFFF	
01007419	830D A0AB000	OR DWORD PTR DS:[100ABA0],FFFFFFF	
01007420	FF15 3413000	CALL DWORD PTR DS:[<&msvrt.__p_fm	msvrt.__p_fm
01007426	8B0D B89A000	MOV ECX,DWORD PTR DS:[1009AB8]	
0100742C	8908	MOV DWORD PTR DS:[EAX],ECX	
0100742E	FF15 3013000	CALL DWORD PTR DS:[<&msvrt.__p_com	msvrt.__p_com
01007434	8B0D B49A000	MOV ECX,DWORD PTR DS:[1009AB4]	
0100743A	8908	MOV DWORD PTR DS:[EAX],ECX	
0100743C	A1 2C130001	MOV EAX,DWORD PTR DS:[<&msvrt.__adjust_	
01007441	8BAA	MOV EDI,DWORD PTR DS:[EAX]	

۱-۱ (Address) : این قسمت آدرس دستور اسمبلی را نمایش می دهد .

۱-۲ (Hex dump) : OpCode یا کد ماشین هر دستور اسمبلی را نمایش می دهد ، دقت کنید که اعدادی که در ollydbg نمایش داده می شود بر مبنای هگزاد دسیمال (Hexadecimal) می باشد و هر دو عدد تشکیل یک بایت را می دهد ، هر دستور اسمبلی تشکیل شده از یک یا چند بایت opcode ، برای اینکه این موضوع را بهتر درک کنید ، تصویر زیر را ببینید :

Address	Hex dump	Disassembly
01007390	6A 70	PUSH 70
0100739F	68 98180001	PUSH NOTEPAD.01001898
010073A4	E8 BF010000	CALL NOTEPAD.01007568
010073A9	33DB	XOR EBX,EBX
010073AB	53	PUSH EBX
010073AC	8B3D CC10000	MOV EDI,DWORD PTR DS:[<&KERNEL32.
010073B2	FFD7	CALL EDI
010073B4	66:8138 4D5A	CMP WORD PTR DS:[EAX],5A4D
010073B9	75 1F	JNZ SHORT NOTEPAD.010073DA
010073BB	8B48 3C	MOV ECX,DWORD PTR DS:[EAX+3C]
010073BE	03C8	ADD ECX,EAX
010073C0	3D 00010000	CMP DWORD PTR DS:[ECX],4550
010073C6	75 12	JNZ SHORT NOTEPAD.010073DA
010073C8	0EB7 18	MOVZX EAX,WORD PTR DS:[ECX+18]
010073D1	74 1F	CMP EAX,10B
010073D3	3D 0B020000	CMP EAX,20B
010073D8	74 05	JE SHORT NOTEPAD.010073DF
010073DA	895D E4	MOV DWORD PTR SS:[EBP-1C],EBX
010073DD	EB 27	JMP SHORT NOTEPAD.01007406
010073DF	83B9 8400000	CMP DWORD PTR DS:[ECX+84],0E
010073E6	76 F2	JBE SHORT NOTEPAD.010073DA
010073E8	33C0	XOR EAX,EAX
010073EA	3999 F800000	CMP DWORD PTR DS:[ECX+F8],EBX
010073F0	EB 0E	JMP SHORT NOTEPAD.01007400
010073F2	8379 74 0E	CMP DWORD PTR DS:[ECX+74],0E

اگر به تصویر زیر خب نگاه کنید ، یکسری نماد های کوچک توسط برنامه نمایش داده شده :

Address	Hex dump	Disassembly	Comment
01001BE3	CC	INT3	
01001BE4	CC	INT3	
01001BE5	CC	INT3	
01001BE6	CC	INT3	
01001BE7	CC	INT3	
01001BE8	08FF	MOV EDI,EDI	
01001BE9	55	PUSH EBP	
01001BEA	8BEC	MOV EBP,ESP	
01001BED	8B45 08	MOV EAX,WORD PTR SS:[EBP+8]	
01001EF0	56	PUSH ESI	
01001EF1	33F6	XOR ESI,ESI	
01001EF3	33C9	XOR ECX,ECX	
01001EF5	EB 26	JMP SHORT NOTEPAD.01001C10	
01001EF7	66:83F9 20	CMP CX,20	
01001EFB	74 06	JE SHORT NOTEPAD.01001C08	
01001BFD	66:83F9 09	CMP CX,9	
01001C01	75 04	JNZ SHORT NOTEPAD.01001C07	
01001C03	85F6	TEST ESI,ESI	
01001C05	74 1E	JE SHORT NOTEPAD.01001C25	
01001C07	66:83F9 22	CMP CX,22	
01001C08	75 09	JNZ SHORT NOTEPAD.01001C16	
01001C0B	33C9	XOR ECX,ECX	
01001C0F	85F6	TEST ESI,ESI	
01001C11	0F94C	SETC CL	
01001C14	8BF1	MOV ESI,ECX	
01001C16	50	PUSH EAX	
01001C17	FF15 4412000	CALL DWORD PTR DS:[&USER32.CharNextW]	CharNextW
01001C1D	66:8B08	MOV CX,WORD PTR DS:[EAX]	CharNextW
01001C20	66:85C9	TEST CX,CX	
01001C23	75 02	JNZ SHORT NOTEPAD.01001BF7	
01001C25	5E	POP ESI	
01001C26	66:8B08	MOV CX,WORD PTR DS:[EAX]	
01001C29	66:83F9 20	CMP CX,20	
01001C2D	74 06	JE SHORT NOTEPAD.01001C35	
01001C2F	66:83F9 09	CMP CX,9	
01001C33	75 04	JNZ SHORT NOTEPAD.01001C39	
01001C35	40	INC EAX	
01001C36	40	INC EAX	
01001C37	EB ED	JMP SHORT NOTEPAD.01001C26	
01001C39	5D	POP EBP	
01001C3B	C2 0400	RETN 4	
01001C3D	CC	INT3	
01001C3E	CC	INT3	
01001C3F	CC	INT3	
01001C40	CC	INT3	

نمادهای مهم

یک سری از این نماد ها برای ما خیلی مهم است که به تشریح هریک می پردازیم :

1- این نماد نشان می دهد که مجموعه ای از کد ها ، تشکیل یک بلوک تابع را میدهد ، و نیز شروع و پایان تابع را تعیین می کند .

001BE5	CC	INT3	
001BE6	CC	INT3	
001BE7	CC	INT3	
001BE8	08FF	MOV EDI,EDI	
001BE9	55	PUSH EBP	
001BEA	8BEC	MOV EBP,ESP	
001BED	8B45 08	MOV EAX,WORD PTR SS:[EBP+8]	
001EF0	56	PUSH ESI	
001EF1	33F6	XOR ESI,ESI	
001EF3	33C9	XOR ECX,ECX	
001EF5	EB 26	JMP SHORT NOTEPAD.01001C10	
001EF7	66:83F9 20	CMP CX,20	
001EFB	74 06	JE SHORT NOTEPAD.01001C08	
001BFD	66:83F9 09	CMP CX,9	
001C01	75 04	JNZ SHORT NOTEPAD.01001C07	
001C03	85F6	TEST ESI,ESI	
001C05	74 1E	JE SHORT NOTEPAD.01001C25	
001C07	66:83F9 22	CMP CX,22	
001C08	75 09	JNZ SHORT NOTEPAD.01001C16	
001C0B	33C9	XOR ECX,ECX	
001C0F	85F6	TEST ESI,ESI	
001C11	0F94C1	SETC CL	
001C14	8BF1	MOV ESI,ECX	
001C16	50	PUSH EAX	
001C17	FF15 4412000	CALL DWORD PTR DS:[&USER32.CharNextW]	CharNextW
001C1D	66:8B08	MOV CX,WORD PTR DS:[EAX]	CharNextW
001C20	66:85C9	TEST CX,CX	
001C23	75 02	JNZ SHORT NOTEPAD.01001BF7	
001C25	5E	POP ESI	
001C26	66:8B08	MOV CX,WORD PTR DS:[EAX]	
001C29	66:83F9 20	CMP CX,20	
001C2D	74 06	JE SHORT NOTEPAD.01001C35	
001C2F	66:83F9 09	CMP CX,9	
001C33	75 04	JNZ SHORT NOTEPAD.01001C39	
001C35	40	INC EAX	
001C36	40	INC EAX	
001C37	EB ED	JMP SHORT NOTEPAD.01001C26	
001C39	5D	POP EBP	
001C3B	C2 0400	RETN 4	
001C3D	CC	INT3	
001C3E	CC	INT3	
001C3F	CC	INT3	
001C40	CC	INT3	

شروع تابع

پایان تابع



۲- این نماد نشان می دهد که مجموعه ای از کد ها ، تشکیل یک حلقه یا Loop را می دهد .

```
08 MOV EDX,DWORD PTR SS:[EE
PUSH ESI
10 MOV ESI,DWORD PTR SS:[EE
0E MOV CX,WORD PTR DS:[ESI]
C9 TEST CX,CX
0A JE SHORT NOTEPAD.010024
MOV WORD PTR DS:[EDX],C
INC EDX
INC EDX
INC ESI
INC ESI
0C DEC DWORD PTR SS:[EBP+C
0C 00 JNZ SHORT NOTEPAD.01002
CMP DWORD PTR SS:[EBP+C]
POP ESI
0C JNZ SHORT NOTEPAD.010024
DEC EDX
```

۳- این نماد نشان میدهد که از این آدرس (01002413) به یک آدرس دیگر (0100241C) پرش شده ، به این پرش ها ، پرش های به سمت پایین گفته می شود .

```
0100240C . 8BEC MOV EBP,ESP
0100240E . 33C0 XOR ESI,ESI
01002410 . 3945 0C CMP ESI,EAX
01002413 > 75 07 JNZ SHORT NOTEPAD.0100241C
01002415 > B8 57000780 MOV EBX,780057
0100241A > EB 2D JMP SHORT NOTEPAD.0100241A
0100241C > 8B55 08 MOV ESI,EBX
0100241F . 56 PUSH ESI
01002420 . 8B75 10 MOV ESI,DWORD PTR SS:[EBP+10]
01002423 > 66:8B0E MOV CX,WORD PTR DS:[ESI]
01002426 . 66:85C9 TEST CX,CX
01002429 > 74 0C JZ SHORT NOTEPAD.0100242B
0100242B . 66:890A MOV WORD PTR DS:[EDX],CX
0100242E . 42 INC EDX
0100242F . 42 INC EDX
01002430 . 46 INC ESI
01002431 . 46 INC ESI
01002432 . FF4D 0C DEC DWORD PTR DS:[EDI]
01002435 > 75 EC JNZ SHORT NOTEPAD.01002423
```

از اینجا به

این جا پرش شده

۴- این نماد نشان میدهد که از این آدرس (01002435) به یک آدرس دیگر (01002423) پرش شده ، به این پرش ها ، پرش های به سمت بالا گفته می شود .

```
01002408 . CC INT3
01002409 . 8BFF MOV EDI,EBP
0100240B . 55 PUSH EBX
0100240C . 8BEC MOV EBP,ESP
0100240E . 33C0 XOR ESI,ESI
01002410 . 3945 0C CMP ESI,EAX
01002413 > 75 07 JNZ SHORT NOTEPAD.0100241C
01002415 > B8 57000780 MOV EBX,780057
0100241A > EB 2D JMP SHORT NOTEPAD.0100241A
0100241C > 8B55 08 MOV ESI,EBX
0100241F . 56 PUSH ESI
01002420 . 8B75 10 MOV ESI,DWORD PTR SS:[EBP+10]
01002423 > 66:8B0E MOV CX,WORD PTR DS:[ESI]
01002426 . 66:85C9 TEST CX,CX
01002429 > 74 0C JZ SHORT NOTEPAD.0100242B
0100242B . 66:890A MOV WORD PTR DS:[EDX],CX
0100242E . 42 INC EDX
0100242F . 42 INC EDX
01002430 . 46 INC ESI
01002431 . 46 INC ESI
01002432 . FF4D 0C DEC DWORD PTR DS:[EDI]
01002435 > 75 EC JNZ SHORT NOTEPAD.01002423
01002437 > 837D 0C 00 CMP DWORD PTR DS:[EDI],0
0100243D . 5E POP ESI
0100243E . 75 07 JNZ SHORT NOTEPAD.01002437
0100243E . 4A DEC EDI
0100243F . 4A DEC EDI
01002440 . 80 70000700 MOV EBX,70000700
```

این جا پرش شده

از اینجا به

۵- این نماد نشان میدهد که از جایی به این اینجا پرش شده ، برای این که بفهمیم از کدام آدرس یا آدرس ها به این جا پرش شده ، روی آدرس راست کلیک میکنم و از منوی ظاهر شده ، منوی Go to را کلیک می کنیم ، و در زیر منوی ظاهر شده آدرس یا آدرس های که به این اینجا پرش شده را نمایش می دهد و با کلیک بر هر یک از آدرس ها به آن محل پرش می کنیم .

Address	Hex dump	Disassembly	Comment
010026E5	50	PUSH EAX	
010026E6	53	PUSH EBX	
010026E7	FF15 FC1000	CALL DWORD PTR DS:[&KERNEL32.lstrcpyW]	
010026E8	FFD5 C000FF	CALL DWORD PTR DS:[&FDD-2147]	
010026ED	FF15 F81000	CALL DWORD PTR DS:[&F81000]	
010026F3	33C0	XCX	Backup
010026F9	8B4D FC	MOVBX,EBX	Copy
010026FE	5F	PI	Binary
010026FF	5E	PI	Binary
01002700	5B	PI	Binary
01002701	E8 414A0000	JMP EBX	Assemble Space
01002706	C9	LI	Label
01002707	C2 1000	RI	Label
0100270A	CC	II	Comment
0100270B	CC	II	Comment
0100270C	CC	II	Breakpoint
0100270D	CC	II	Breakpoint
0100270E	CC	II	Hit trace
0100270F	8BFF	MOVBX,EBX	Run trace
01002711	55	PI	Run trace
01002712	8BEC	MOVBX,EBX	
01002714	81EC 10020000	SHR EBX,10	
0100271A	A1 04960001	MOVBX,[04960001]	New origin here Ctrl+Gray *
0100271F	53	PI	
01002720	56	PI	
01002721	57	PI	
01002722	8945 FC	MOVBX,EBX	Go to
01002725	330B	XCX	Follow in Dump
01002727	980B	XCX	View call tree Ctrl+K
01002729	86:899D F4FD	MOVBX,EBX	
01002730	B9 81000000	MOVBX,EBX	
01002735			
0100273B			
0100273D			
0100273F			
01002740			
01002746			
01002748			
0100274E	89B5 F0F0FFF	MOVBX,EBX	Analysis
01002754	74 14	JLE EBX,14	JLE from 01002685
01002756	53	PI	JE from 0100268D
01002757	53	PI	
01002758	6A 0E	MOVBX,EBX	Bookmark
0100275A	FF35 3898000	MOVBX,EBX	Dump debugged process
0100275B	FF07	MOVBX,EBX	

۳-۱ Disassembly: این بخش دستورات اسمبلی را نوشته ، برای این که بتوانیم این دستورات را خوب متوجه و آنالیز کنیم باید با زبان اسمبلی آشنا باشیم . برای تغییر دستور مورد نظر کافی روی دستور ، دوبار کلیک کرده و دستور مورد نظر را جایگزین کنیم . و بعد دکمه Assemble را کلیک کنید ، تا تغییرات اعمال شود ، دقت کنید که فقط یکبار دکمه Assemble را کلیک کنید ، اگر یک بار دیگر دکمه Assemble را بزنید ، دستورهای بعدی در آدرس های بعدی هم با دستور جدید جایگزین می شود و برنامه Crash می شود .

Address	Hex dump	Disassembly
0100746B	68 0C900001	PUSH NOTEPAD.0100900C
01007470	E8 5D010000	CALL <JMP.&msvcrt._initterm>
01007475	A1 009A0001	MOV EAX,DWORD PTR DS:[1009A001]
0100747A	8945 DC	MOVBX,EBX
0100747D	8D45 DC	MOVBX,EBX
01007480	50	PI
01007481	FF35 AC9A000	MOVBX,EBX
01007487	8D45 D4	MOVBX,EBX
0100748A	50	PI
0100748B	8D45 D0	MOVBX,EBX
0100748E	50	PI
0100748F	8D45 CC	MOVBX,EBX
01007492	50	PI
01007493	FF15 2013000	CALL DWORD PTR DS:[2013000]
01007499	8945 C8	MOVBX,EBX
0100749C	68 08900001	PUSH NOTEPAD.01009000
010074A1	68 00900001	PUSH NOTEPAD.01009000
010074A6	E8 27010000	CALL <JMP.&msvcrt._initterm>
0100740B	8304 24	MOVBX,EBX

دقت داشته باشید که گزینه Fill with NOP's را فعال کنید ، این گزینه باعث میشود اگر دستور جدید opcode کمتری داشت ، بقیه Opcode ها باقی مانده با دستور nop (opcode 90) جایگزین شود . برای درک این موضوع به دو تصویر زیر دقت کنید :



دستور اصلی :

```

0100745A . FF15 2813000 CALL DWORD PTR DS:[00401000],setuserma
01007460 . 59 POP ECX
01007461 > E8 77010000 CALL NOTEPAD.01007500
01007466 . 68 10900001 PUSH NOTEPAD.01009010
01007469 . 68 0C900001 PUSH NOTEPAD.0100900C
01007470 . E8 5D010000 CALL <JMP.&msvcrt._initterm>
01007475 . A1 B09A0001 MOV EAX,DWORD PTR DS:[1009AB0]
0100747A . 8D45 DC MOV DWORD PTR SS:[EBP-24],EAX
0100747D . 8D45 DC LEA EAX,DWORD PTR SS:[EBP-24]
    
```

۵ بایت

دستور جایگزین :

```

01007475 . B0 01 MOV AL,1
01007477 . 90 NOP
01007478 . 90 NOP
01007479 . 90 NOP
0100747A . 8945 DC MOV DWORD PTR SS:[EBP-24],EAX
0100747D . 8D45 DC LEA EAX,DWORD PTR SS:[EBP-24]
01007480 . 50 PUSH EAX
01007481 . FF35 AC9A0000 PUSH DWORD PTR DS:[00401000]
01007487 . 8D45 D4 LEA EAX,DWORD PTR SS:[EBP-18]
0100748A . 50 PUSH EAX
    
```

۵ بایت

همان طور که در تصاویر بالا می بینید ، سه عدد دستور nop به کد اضافه شد تا opcode های باقی مانده ، برنامه را خراب نکند (باعث می شد که برنامه (Crash) شود) اگر خواستید ، به حالت اول (دستور اصلی) برگردید ، تعداد خط های را که دستکاری کردید (opcode آنها قرمز رنگ شده) همه را باهم انتخاب کرده و کلید های Alt+Backspace را بزنید .

```

01007461 > E8 77010000 CALL NOTEPAD.01007500
01007466 . 68 10900001 PUSH NOTEPAD.01009010
01007469 . 68 0C900001 PUSH NOTEPAD.0100900C
01007470 . E8 5D010000 CALL <JMP.&msvcrt._initterm>
01007475 B0 01 MOV AL,1
01007477 90 NOP
01007478 90 NOP
01007479 90 NOP
0100747A . 8945 DC MOV DWORD PTR SS:[EBP-24],EAX
0100747D . 8D45 DC LEA EAX,DWORD PTR SS:[EBP-24]
01007480 . 50 PUSH EAX
01007481 . FF35 AC9A0000 PUSH DWORD PTR DS:[00401000]
01007487 . 8D45 D4 LEA EAX,DWORD PTR SS:[EBP-18]
0100748A . 50 PUSH EAX
0100748B . 8D45 D0 LEA EAX,DWORD PTR SS:[EBP-1C]
    
```

۴-۱ Comment : این قسمت مربوط به توضیحات می باشد ، یکسری از توضیحات را خود برنامه آنالیز کرده به ما نشان می دهد. ما هم میتوانیم به هر خط توضیحات اضافه کنیم ، کافی است روی خط مورد نظر رفته تا آن خط به رنگ خاکستری در آید و با زدن دکمه ؛ از کی برد ، کادری ظاهر شده و میتوان توضیحات مورد نظر را تایپ کرده و بعد روی دکمه OK کلیک کنید تا توضیحات مورد نظر ثبت شود .

Address	Hex dump	Disassembly	Comment
01007390	CC	INT3	
01007391	CC	INT3	
01007392	FF25 BC12000	JMP DWORD PTR DS:[00401000],WINSPOOL.OpenPrinterW	WINSPOOL.OpenPrinterW
01007398	CC	INT3	
01007399	CC	INT3	
0100739A	CC	INT3	
0100739B	CC	INT3	
0100739C	CC	INT3	
0100739D	6A 70	PUSH 70	
0100739F	68 98180001	PUSH NOTEPAD.01001898	
010073A4	E8 BF010000	CALL NOTEPAD.01007568	
010073A9	330B	XOR EBX,EBX	
010073AB	53	PUSH EBX	
010073AC	8B3D CC10000	MOV EDI,DWORD PTR DS:[00401000]	
010073AD	FFD7	CALL EDI	
010073B4	66:8130 405A	CMP WORD PTR DS:[00401000],00000000	
010073B9	75 1F	JNZ SHORT NO	
010073BB	8B48 3C	MOV ECX,DWORD PTR DS:[00401000]	
010073BE	0808	ADD ECX,EAX	
010073C0	0139 5045000	CMP DWORD PTR DS:[00401000],00000000	
010073C6	75 12	JNZ SHORT NO	
010073C8	0FB741 18	MOVZX EAX,WORD PTR DS:[00401000]	
010073CC	3D 0B010000	CMP EAX,10B	
010073D1	74 1F	JE SHORT NOTI	
010073D3	3D 0B020000	CMP EAX,20B	
010073D8	74 05	JE SHORT NOTI	
010073DA	895D E4	MOV DWORD PTR DS:[00401000],EAX	
010073DD	EB 27	JMP SHORT NO	
010073DF	83B9 8400000	CMP DWORD PTR DS:[00401000],00000000	
010073E6	75 F2	JBE SHORT NO	
010073E8	33C0	XOR EAX,EAX	
010073EA	3999 F800000	CMP DWORD PTR DS:[00401000],00000000	
010073F0	EB 0E	JMP SHORT NO	
010073F2	8379 74 0E	CMP DWORD PTR DS:[00401000],00000000	
010073F5	75 F2	JBE SHORT NO	

Add comment at 010073A9

EBX=0

OK Cancel

```

010073A4 . E8 BF010000 CALL NOTEPAD.01007568
010073A9 . 330B XOR EBX,EBX
010073AB . 53 PUSH EBX
010073AC . 8B3D CC10000 MOV EDI,DWORD PTR DS:[00401000]
    
```




قسمت 2 پنجره CPU :

این قسمت ، برای نمایش مقادیر ثابت های سی پی یو (CPU Registers) می باشد . دقت کنید که مقادیر در مبنای Hex می باشد .

The screenshot shows the CPU registers window with several callouts in red rounded rectangles:

- ثبات های اصلی و مقادیر آنها**: Points to the top section of registers (EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, EIP).
- ثبات های سگمنت**: Points to the segment registers (CS, SS, DS, FS, GS).
- ثبات Eflags و مقادیرشان**: Points to the EFLAGS register.

اگر بخواهیم مقادیر رجیستر ها را دستکاری کنیم ، کافی است روی مقادیر رجیستر ها (EAX -EDX) دوبار کلیک کنید (مقدار EIP را نمیتوانیم تغییر بدهیم) در کادر باز شده می توان مقدار رجیستر را تغییر بدهیم .

The screenshot shows the 'Modify EAX' dialog box with several callouts in red rounded rectangles:

- مقدار عددی در مبنای Hex**: Points to the Hexadecimal input field.
- مقدار عددی با علامت در مبنای Dec**: Points to the Signed/Unsigned radio buttons.
- مقدار عددی بدون علامت در مبنای Dec**: Points to the Char input field.
- مقدار ASCII**: Points to the Char input field.

اگر بخواهیم مقادیر فلگهای C-P-A-Z-S-T-D-O را دستکاری کنیم (0 یا 1) کافی است روی مقدار هر یک کلیک کنید تا تغییر کند .

قسمت 3 پنجره CPU :

این قسمت را زمانی میتوان استفاده کرد که برنامه را بصورت خط به خط (F8-F7) اجرا کرده باشیم ، نشان میدهد که دستور جاری (آدرس مشکی) چه مقادیری در ثبات یا حافظه وجود دارد .

Address	Hex dump	Disassembly
01007390	6A 70	PUSH 70
0100739F	68 98180001	PUSH NOTEPAD.01001898
010073A4	E8 BF010000	CALL NOTEPAD.01007568
010073A9	33DB	XOR EBX,EBX
010073AB	53	PUSH EBX
010073AC	8B3D CC10000	MOV EDI,DWORD PTR DS:[&KERNEL32.C
010073B2	FFD7	CALL EDI
Information		
EBX=00000000		

همان طور که می بیند در خط فعال (010073AB) دستور PUSH EBX مقدار رجیستر EBX را درون پشته قرار می دهد ،
همنطور که می بینید در این قسمت به ما می گوید که مقدار EBX=0 می باشد .

قسمت 4 پنجره CPU :

این قسمت مربوط به حافظه پشته یا STACK می باشد. این قسمت هم از سه بخش تشکیل شده است .

Address	Value	Comment
0007FFC4	7C816FF7	RETURN to kernel32.7C816FF7
0007FFC8	7C910738	ntdll.7C910738
0007FFCC	FFFFFFFF	
0007FFD0	7FFD7000	
0007FFD4	00000640	
0007FFD8	0007FFC8	
0007FFDC	86FB44B0	
0007FFE0	FFFFFFFF	End of SEH chain
0007FFE4	7C839A30	SE handler
0007FFE8	7C817000	kernel32.7C817000
0007FFEC	00000000	
0007FFF0	00000000	
0007FFF4	00000000	
0007FFF8	0100739D	NOTEPAD.<ModuleEntryPoint>
0007FFFC	00000000	

4-1

4-2

4-3

۴-۱ : آدرس پشته

۴-۲ : مقدار درون آدرس پشته

۴-۳ : توضیحات برنامه ، برای آدرس و مقدار

قسمت 5 پنجره CPU :

در این قسمت اطلاعات مربوط به حافظه برنامه و مقادیر آنها می دهد . این قسمت هم از سه بخش تشکیل شده است .

Address	Hex dump	ASCII
01009000	00 00 00 00 04 70 00 01	...p.0
01009008	00 00 00 00 00 00 00 00	...
01009010	00 00 00 00 00 00 00 00	...
01009018	78 00 00 00 01 00 00 00	...s...
01009020	4E 00 6F 00 74 00 65 00	...N o t e
01009028	70 00 61 00 00 00 00 00	...
01009030	FF FF FF FF	...
01009038	02 00 00 00	...
01009040	04 00 00 00 05 00 00 00	...*
01009048	06 00 00 00 07 00 00 00	...
01009050	08 00 00 00 09 00 00 00	...
01009058	0A 00 00 00 0B 00 00 00	...
01009060	0C 00 00 00 0D 00 00 00	...
01009068	0E 00 00 00 0F 00 00 00	...
01009070	10 00 00 00 11 00 00 00	...<...>
01009078	12 00 00 00 13 00 00 00	...<...>
01009080	14 00 00 00 15 00 00 00	...<...>
01009088	16 00 00 00 17 00 00 00	...<...>
01009090	18 00 00 00 19 00 00 00	...<...>
01009098	1A 00 00 00 1B 00 00 00	...<...>
010090A0	1C 00 00 00 1D 00 00 00	...<...>
010090A8	1E 00 00 00 1F 00 00 00	...<...>

1 Byte

2 Byte = 1 Word

4 Byte = 2 Word = 1 Double Word

5-1

5-2

5-3

Persian Reverse Engineering Team

۵-۱ : آدرس حافظه

۵-۲ : مقدار حافظه بر مبنای Hex

۵-۳ : مقدار حافظه ، معادل کد Ascii



قسمت 6 پنجره CPU :

این قسمت وضعیت برنامه را تعیین می کند .

برنامه متوقف شده است . **Paused**

برنامه در حال اجرا می باشد . **Running**

برنامه بسته شده . **Terminated**

قسمت 7 پنجره CPU :

این قسمت یک سری اطلاعات در مورد آنالیز کردن برنامه میدهد ، مثلا در تصویر زیر به ما میگوید که در آدرس جاری که هستیم ، نقطه ورود برنامه یا Entry Point است .

Program entry point

نقاط توقف Break Points :

اگر بخواهیم در یکسری از آدرسها برنامه متوقف شود تا بتوانیم دستورات آن محدوده یا مقادیر حافظه ها ، ثبات ها را ببینیم و آنالیز کنیم ، از نقاط توقف (BP) استفاده می کنیم ، برای این کار روی آدرس یا آدرس های مورد نظر دکمه F2 را می زنیم ، تا آن آدرس یا آدرس ها به رنگ قرمز در آیند .

0100739D	6A 70	PUSH 70	
0100739F	68 98180001	PUSH NOTEPAD.01001898	
010073A4	E8 BF010000	CALL NOTEPAD.01007568	
010073A9	33DB	XOR EBX,EBX	
010073AB	53	PUSH EBX	
010073AC	9B3D CC10000	MOV EDI,DWORD PTR DS:[<&KERNEL32.GetModu	pModule => NULL
010073B2	FFD7	CALL EDI	kernel32.GetModuleHandleA
010073B4	66:8138 4D5A	CMP WORD PTR DS:[EAX],5A4D	GetModuleHandleA
010073B5	75 1F	JNZ SHORT NOTEPAD.010073DA	
010073BB	0D48 3C	MOV ECX,DWORD PTR DS:[EAX+3C]	
010073BE	03C8	ADD ECX,EAX	
010073C0	8139 50450000	CMP DWORD PTR DS:[ECX],4550	
010073C6	75 12	JNZ SHORT NOTEPAD.010073DA	
010073C8	0FB741 18	MOVZBL WORD PTR DS:[ECX+1	
010073CC	3D 0A010000	CMP EAX,10B	
010073CD	74 1F	JE SHORT NOTEPAD.010073F2	
010073D3	3D 0A020000	CMP EAX,20B	
010073D8	74 05	JE SHORT NOTEPAD.010073DF	
010073DB	895D E4	MOV DWORD PTR SS:[EBP-1C],E	
010073DD	EB 27	JMP SHORT NOTEPAD.01007406	
010073DF	83B9 84000000	CMP DWORD PTR DS:[ECX+84],0	
010073E6	76 F2	JBE SHORT NOTEPAD.010073DA	
010073E8	33C0	XOR EAX,EAX	
010073EA	3959 F8000000	CMP DWORD PTR DS:[ECX+F8],EB	
010073F0	EB 0E	JMP SHORT NOTEPAD.01007406	
010073F2	8379 74 0E	CMP DWORD PTR DS:[ECX+74],0E	
010073F6	76 E2	JBE SHORT NOTEPAD.010073DA	
010073F8	33C0	XOR EAX,EAX	
010073FA	3999 E8000000	CMP DWORD PTR DS:[ECX+E8],EBX	
01007400	0F95C0	SETNE AL	
01007403	8945 E4	MOV DWORD PTR SS:[EBP-1C],EAX	
01007406	895D FC	MOV DWORD PTR SS:[EBP-4],EBX	
01007409	6A 02	PUSH 2	
0100740B	FF15 3813000	CALL DWORD PTR DS:[<&msvcrt.__set_app_t	msvcrt.__set_app_type
01007411	59	POP ECX	
01007412	830D 9CAB000	OR DWORD PTR DS:[100AB9C],FFFFFFF	
01007419	830D A0AB000	OR DWORD PTR DS:[100ABA0],FFFFFFF	
01007420	FF15 3413000	CALL DWORD PTR DS:[<&msvcrt.__p_fm	msvcrt.__p_fm
01007426	8B0D B89A000	MOV ECX,DWORD PTR DS:[1009AB0]	
0100742C	8908	MOV DWORD PTR DS:[EAX],ECX	
0100742E	FF15 3013000	CALL DWORD PTR DS:[<&msvcrt.__p_commod	msvcrt.__p_commode
01007434	8B0D B49A000	MOV ECX,DWORD PTR DS:[1009AB4]	
0100743A	8908	MOV DWORD PTR DS:[EAX],ECX	

وقتی برنامه را اجرا کنیم (F9) و اگر برنامه به آدرسی که BP گذاشته باشیم برسد در آن نقطه متوقف میشود و ما می توانیم با دکمه های F8 یا F7 برنامه را خط به خط اجرا کنیم . برای برداشتن BP ها در آدرس مورد نظر دوباره کلید F2 را بزنید .