



南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异



# 汇编语言与逆向技术

## 第9章 静态逆向技术



允公允能 日新月异

# 本章知识点

- 逆向技术
- IDA Pro简介
- IDA Pro窗口
- IDA Pro的操作
- 交叉引用
- 函数分析
- 图形化显示
- 增强反汇编的相关功能



南开大学  
Nankai University



南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

# 1. 逆向技术



允公允能 日新月异

# 逆向工程

- 逆向工程（又称**逆向技术**），是一种产品设计技术再现过程
  - 对一项目标产品进行逆向分析及研究，从而演绎并得出该产品的处理流程、组织结构、功能特性及技术规格等设计要素，以制作出功能相近，但又不完全一样的产品。
- 逆向工程源于**商业及军事领域**中的硬件分析
- 其主要目的是在不能轻易获得必要的生产信息的情况下，直接从成品分析，推导出产品的设计原理



南开大学  
Nankai University



允公允能 日新月异

# 软件逆向工程

- 软件逆向工程(Software Reverse Engineering)是指根据软件程序的反汇编代码（静态）和执行过程（动态），通过逆向分析来推导出软件具体的实现方法。
- 软件逆向工程可能会被误认为是对知识产权的严重侵害，但是在实际应用上，反而可能会保护知识产权所有者。
  - 漏洞发掘
  - 取证
  - 性能分析
  - 软件保护



南开大学  
Nankai University



允公允能 日新月异

# 逆向分析技术

- 静态分析
  - IDA Pro
- 动态分析
  - OllyDbg: 用户态的动态调试
  - WinDbg: 内核态的动态调试



南开大学  
Nankai University

使用OllyDbg单步执行hello.exe程序，跟踪并分析反汇编代码的执行，这个过程是属于（）

- ☒ A 动态逆向分析
- ☐ B 静态逆向分析





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

## 2. IDA Pro简介

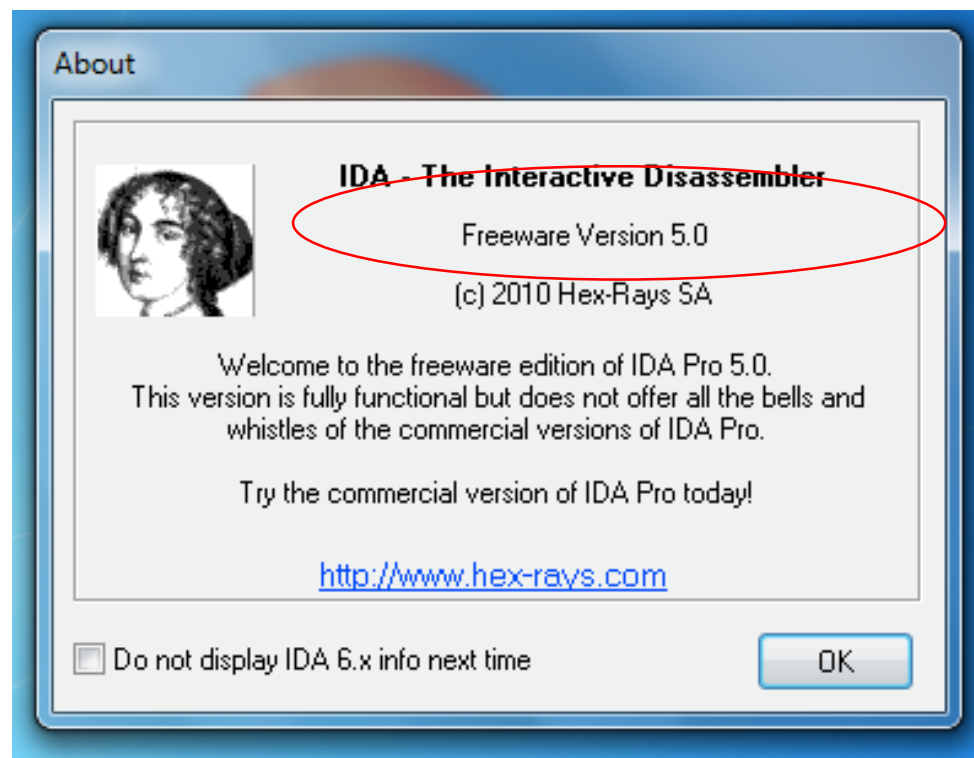




允公允能 日新月异

# IDA Pro

IDA Pro是Hex-Rays公司出品的一款交互式反汇编工具支持32位和64位程序的反汇编



南开大学  
Nankai University

IDA



**FREWARE**

Version 8.1

Hex-Rays © 2022

# IDA Freeware

IDA Freeware是Hex-Rays公司出品的一款交互式反汇编工具  
支持32位和64位程序的反汇编

- Analyze both 32-bit and 64-bit applications
- Cloud-based x64 decompiler
- Local x86/x64 debugger included
- Support x86/x64 processors
- Save your analysis results
- Perpetual license





# 下载IDA Freeware

下载地址: <https://hex-rays.com/ida-free/>

## Download your IDA Free

The freeware version of IDA v8.1 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

SHA1 checksums:

9e7ead666136ec1293c39984ae3a1cf44793dca3	arm_idafree81_mac.app.zip
55d559efb6a71ee2988892b6fcd2a24f82f536a1	idafree81_linux.run
2aa1fa642afc95317de1ee45bccdf3db2465b91a	idafree81_mac.app.zip
666df4d5e32b26e7f0fae5c18beed9c58bcd9cc7	idafree81_windows.exe



IDA Freeware for Windows (90MB)



IDA Freeware for Linux (76MB)



IDA Freeware for Mac (68MB)



IDA Freeware for Mac ARM (70MB)

# IDA PRO 专业版的价格

Target OS: Windows

IDAPRONW	IDA Pro Named License [Windows]	1975 USD
HEXARM64W	ARM64 Decompiler Fixed License [Windows]	2765 USD
HEXARMW	ARM32 Decompiler Fixed License [Windows]	2765 USD
HEXMIPS64W	MIPS64 Decompiler Fixed License [Windows]	2765 USD
HEXMIPSW	MIPS Decompiler Fixed License [Windows]	2765 USD
HEXPPC64W	PPC64 Decompiler Fixed License [Windows]	2765 USD
HEXPPCW	PPC Decompiler Fixed License [Windows]	2765 USD
HEXX64W	x64 Decompiler Fixed License [Windows]	2765 USD
HEXX86W	x86 Decompiler Fixed License [Windows]	2765 USD

Target OS: Windows

IDAPROCW	IDA Pro Computer License [Windows]	1975 USD
HEXARM64W	ARM64 Decompiler Fixed License [Windows]	2765 USD
HEXARMW	ARM32 Decompiler Fixed License [Windows]	2765 USD
HEXMIPS64W	MIPS64 Decompiler Fixed License [Windows]	2765 USD
HEXMIPSW	MIPS Decompiler Fixed License [Windows]	2765 USD
HEXPPC64W	PPC64 Decompiler Fixed License [Windows]	2765 USD
HEXPPCW	PPC Decompiler Fixed License [Windows]	2765 USD
HEXX64W	x64 Decompiler Fixed License [Windows]	2765 USD
HEXX86W	x86 Decompiler Fixed License [Windows]	2765 USD

Target OS: Windows

IDAPROFW	IDA Pro Floating License [Windows]	2960 USD
HEXARM64FW	ARM64 Decompiler Floating License [Windows]	4145 USD
HEXARMFw	ARM32 Decompiler Floating License [Windows]	4145 USD
HEXMIPS64FW	MIPS64 Decompiler Floating License [Windows]	4145 USD
HEXMIPSFW	MIPS Decompiler Floating License [Windows]	4145 USD
HEXPPC64FW	PPC64 Decompiler Floating License [Windows]	4145 USD
HEXPPCFW	PPC Decompiler Floating License [Windows]	4145 USD
HEXX64FW	x64 Decompiler Floating License [Windows]	4145 USD
HEXX86FW	x86 Decompiler Floating License [Windows]	4145 USD



允公允能 日新月异

# IDA Pro

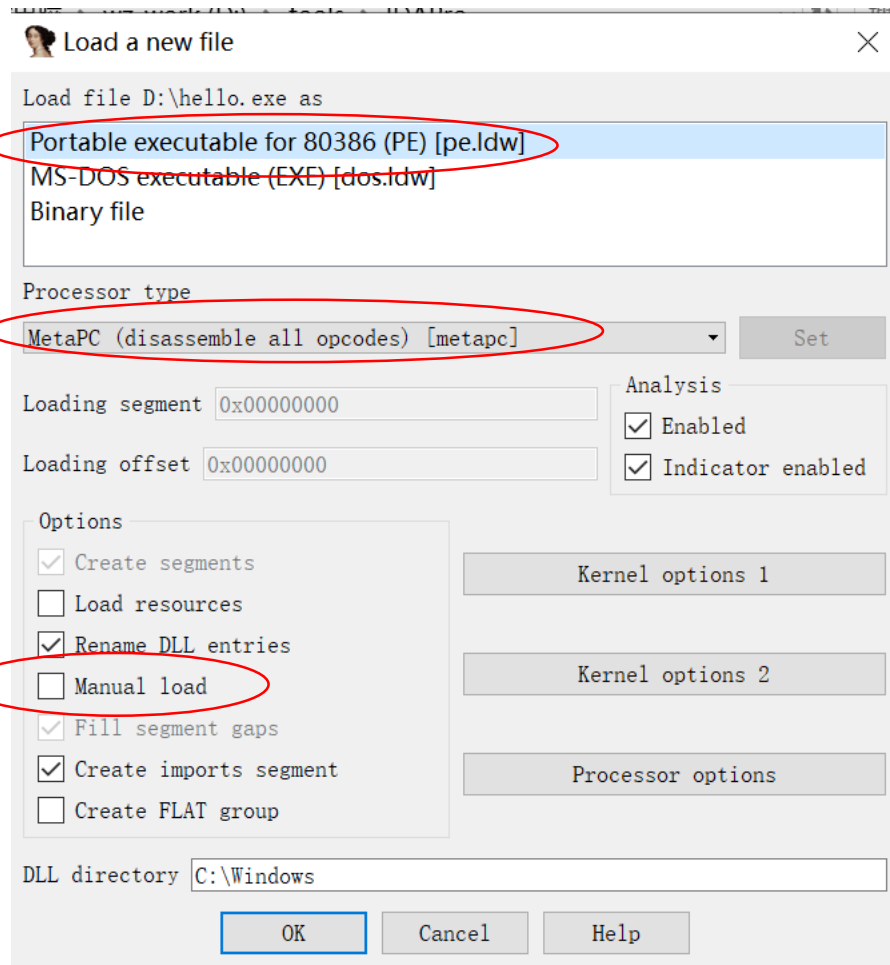
- 函数发现
- 栈分析
- 局部变量的识别
- FLIRT快速的库函数识别与标记
  - Fast Library Identification and Recognition Technology



南开大学  
Nankai University

# IDA Pro

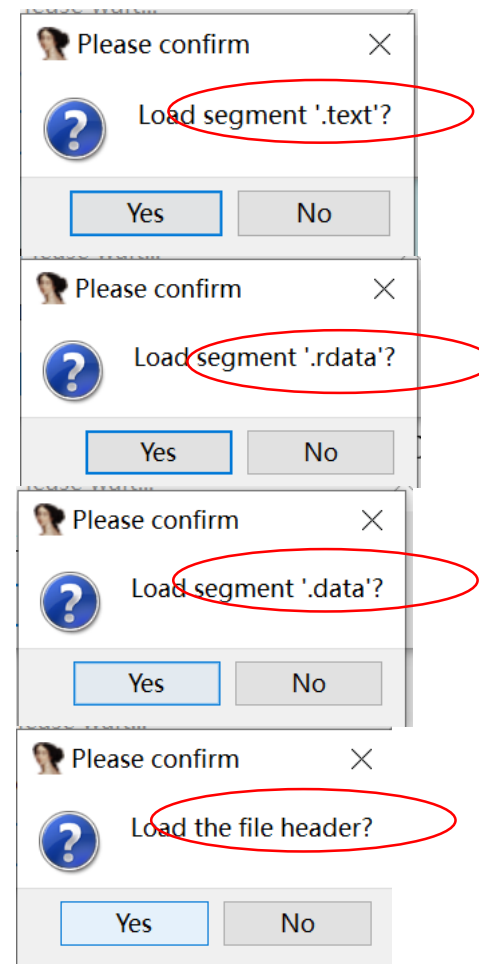
- IDA Pro除了支持**PE文件格式**，还支持DOS、UNIX、Mac、Java、.NET等平台的文件格式
- IDA Pro会自动识别处理器类型





# IDA Pro

- IDA是按**区块装载**PE文件的，例如.text(代码块)、.data(数据块)、.rsrc(资源块)等。
- 在默认情况下，IDA Pro的反汇编代码中不包含PE头或资源节。







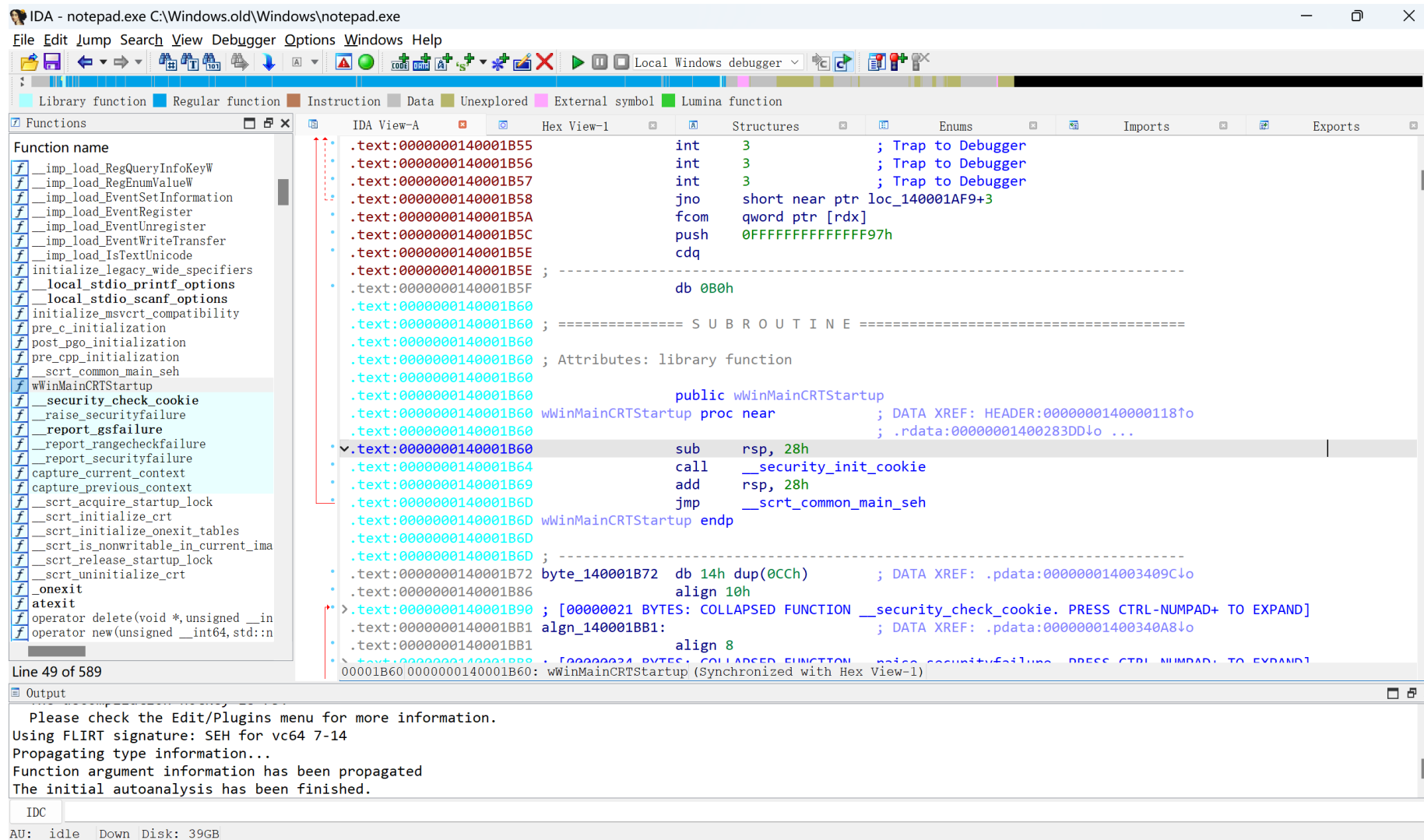
南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

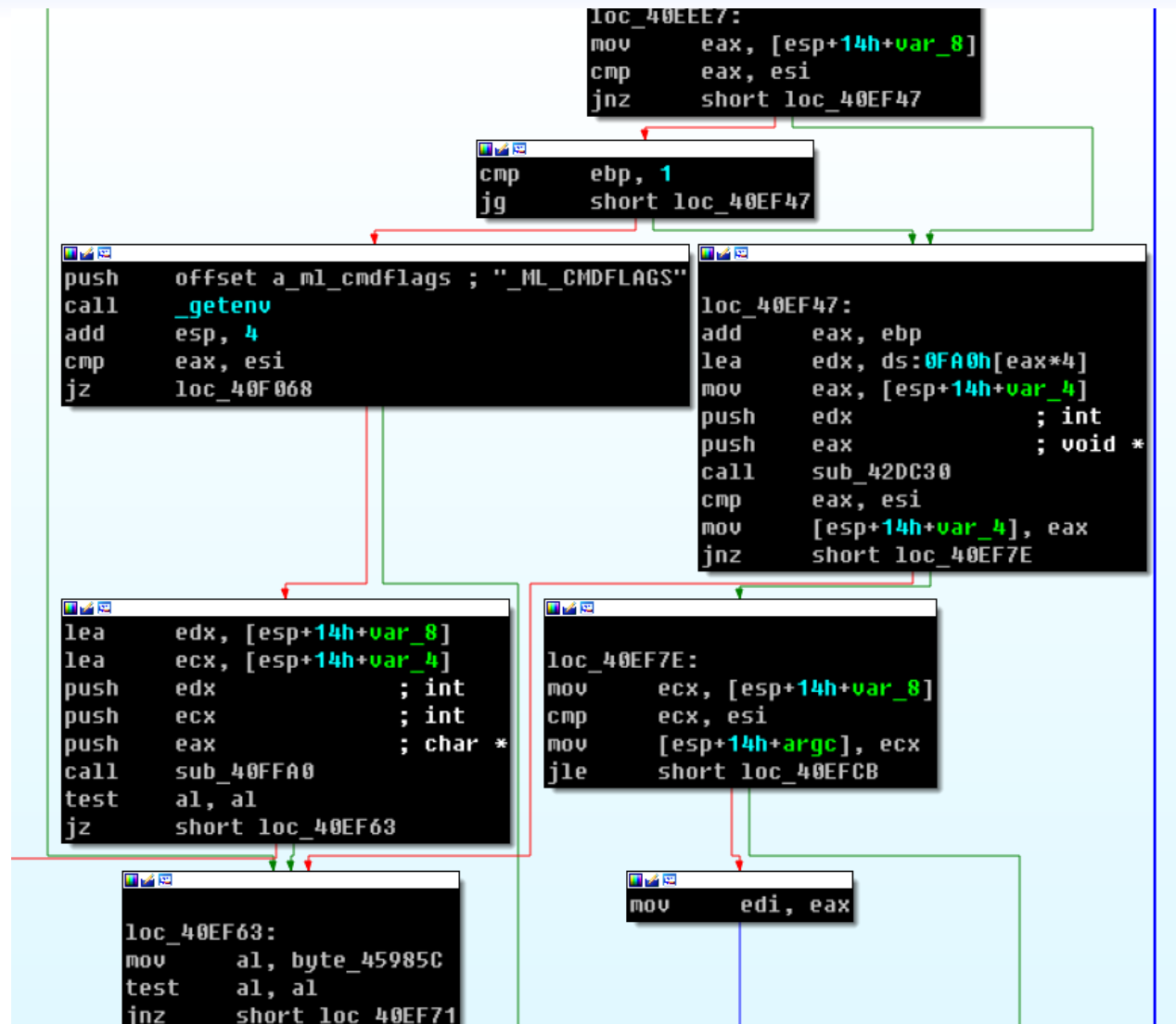
### 3. IDA 窗口

# IDA Freeware





# 图形模式



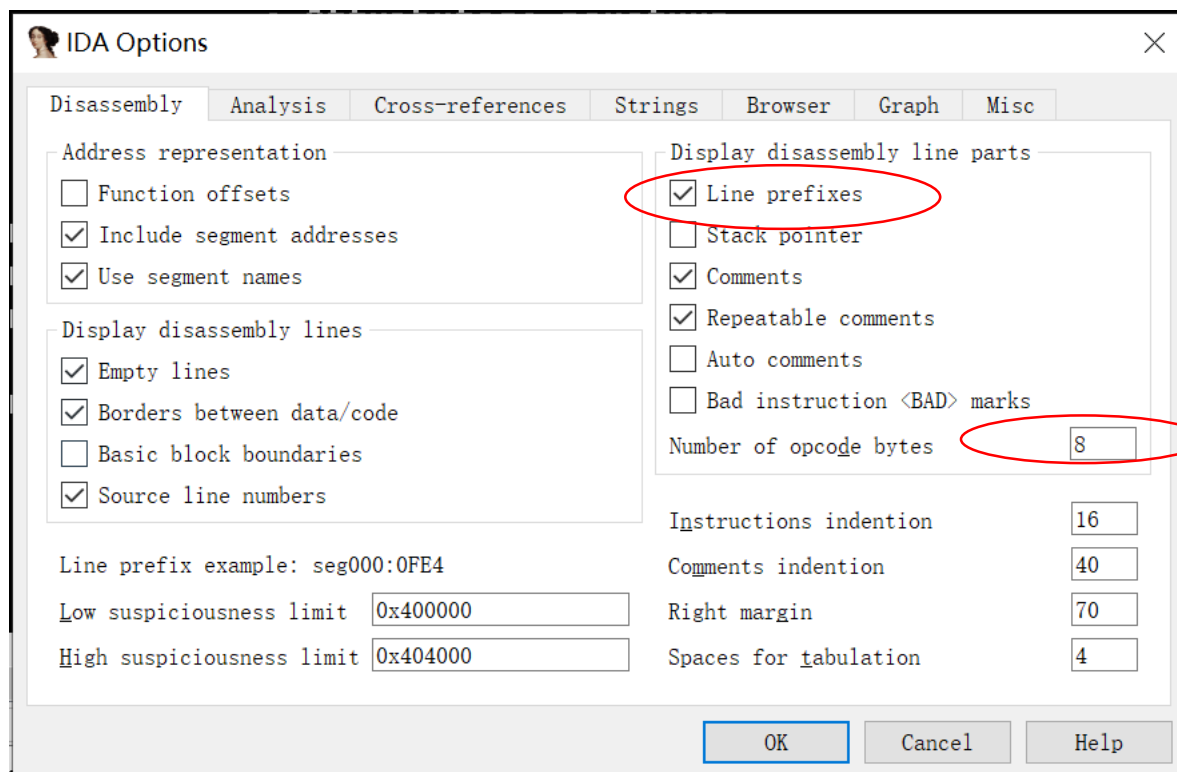


# 文本模式

图形模式	文本模式	空格键
<pre>.text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 .text:0040EE60 • .text:0040EE60 83 EC 08 • .text:0040EE63 55 • .text:0040EE64 56 • .text:0040EE65 57 • .text:0040EE66 33 F6 • .text:0040EE68 68 00 80 00 00 • .text:0040EE6D 6A 01 • .text:0040EE6F 89 74 24 18</pre>	<pre>; int _cdecl main(int argc, const char **argv, const char **e _main proc near var_8 = dword ptr -8 var_4 = dword ptr -4 argc = dword ptr 4 argv = dword ptr 8 envp = dword ptr 0Ch sub esp, 8 push ebp push esi push edi xor esi, esi push 8000h ; int push 1 ; int mov [esp+1Ch+var_4], esi</pre>	<pre>; CODE XREF: start+AF↓</pre>



# 反汇编窗口

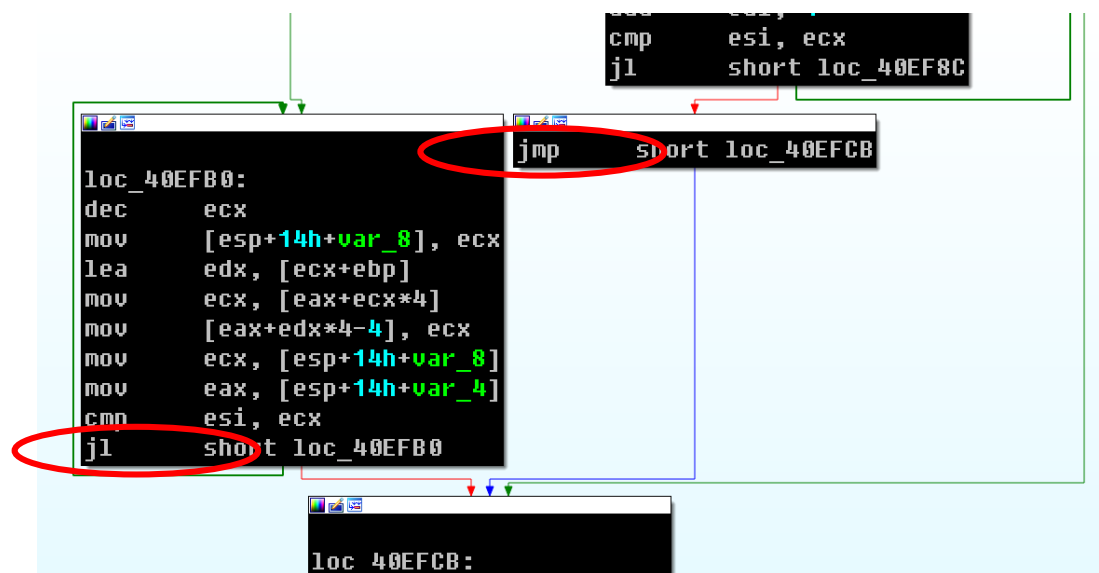


在图形模式中，IDA Pro默认不显示行号、操作码



# 箭头

- 红色: False分支
- 绿色: True分支
- 蓝色: 无条件跳转
- 循环: 向上的箭头





# 文本模式

箭头

实线 = 无条件跳转

虚线 = 条件跳转

向上箭头 = 循环

节信息

内存地址

注释

.text:00401015	jz	short loc_40102B	
.text:00401017	push	offset aSuccessInterne	; "Success: Internet Connection\n"
.text:0040101C	call	sub_40105F	
.text:00401021	add	esp, 4	
.text:00401024	mov	eax, 1	
.text:00401029	jmp	short loc_40103A	
.text:0040102B			
.text:0040102B			
.text:0040102B	loc_40102B:		; CODE XREF: sub_401000+15↑j
.text:0040102B	push	offset aError1_1NoInte	; "Error 1.1: No Internet\n"
.text:00401030	call	sub_40105F	
.text:00401035	add	esp, 4	
.text:00401038	xor	eax, eax	
.text:0040103A			
.text:0040103A	loc_40103A:		; CODE XREF: sub_401000+29↑j
.text:0040103A	mov	esp, ebp	
.text:0040103C	pop	ebp	



















# 函数窗口














- 列举所有函数
  - 可以发现规模庞大的函数、规模很小的函数。

Function name	Segment	Start	Length	Locals	Arguments	R	F	L	S	B	T	=
 sub 40B620	.text	0040B620	00000109	00000008	00000008	R	.	.	.	.	.	.
 sub 40B730	.text	0040B730	000000AF	00000000	00000004	R	.	.	.	.	.	.
 sub 40B7E0	.text	0040B7E0	00000095	00000000	00000004	R	.	.	.	.	.	.
 sub 40B880	.text	0040B880	000000AD	0000000C	00000004	R	.	.	.	.	.	.
 sub 40BC30	.text	0040BC30	0000015E	0000012C	00000008	R	.	.	.	.	.	.
 sub 40BDB0	.text	0040BDB0	0000005D	0000000C	00000004	R	.	.	.	.	.	.
 sub 40C000	.text	0040C000	000006C7	0000000C	0000000C	R	.	.	.	.	.	.
 sub 40D040	.text	0040D040	000001E3	0000001C	0000000C	R	.	.	.	.	.	.
 sub 40D230	.text	0040D230	000001ED	00000010	0000000D	R	.	.	.	.	.	.
 sub 40D420	.text	0040D420	00000889	00000010	00000010	R	.	.	.	.	.	.
 sub 40DD10	.text	0040DD10	00000277	00000010	00000010	R	.	.	.	.	.	.
 sub 40DF90	.text	0040DF90	000006FF	00000024	00000010	R	.	.	.	.	.	.



# 名字窗口













- 列举内存地址的名字，包括函数名、代码的名字、数据的名字和字符串

Name	Address	Public
 comexecmd 0	0044088C	
 freebuf	004408E0	
 strpbrk	00440910	
 listnext	00440924	
 listdone	00440931	
 dstnext	00440934	
 dstdone	00440944	
 strrchr	00440950	
 returndi	00440971	
 toend 1	00440973	
 access	00440977	
 cenvarg	004409BB	
 tolower	00440BBF	



# 字符串窗口

- 显示内存中识别出来的所有字符串

Address	Length	Type	String
 .rdata:00442D0C 00000016		C	SunMonTueWedThuFriSat
 .rdata:00442D24 00000025		C	JanFebMarAprMayJunJulAugSepOctNovDec
 .rdata:00442D6C 00000005		C	PATH
 .rdata:00442D74 00000013		C	GetLastActivePopup
 .rdata:00442D88 00000010		C	GetActiveWindow
 .rdata:00442D98 0000000C		C	MessageBoxA
 .rdata:00442DA4 0000000B		C	user32.dll
 .rdata:00442DB0 00000005		C	.com
 .rdata:00442DB8 00000005		C	.exe
 .rdata:00442DC0 00000005		C	.bat
 .rdata:00442DC8 00000005		C	.cmd
 .rdata:004433A4 0000000D		C	KERNEL32.dll





# 导入表窗口

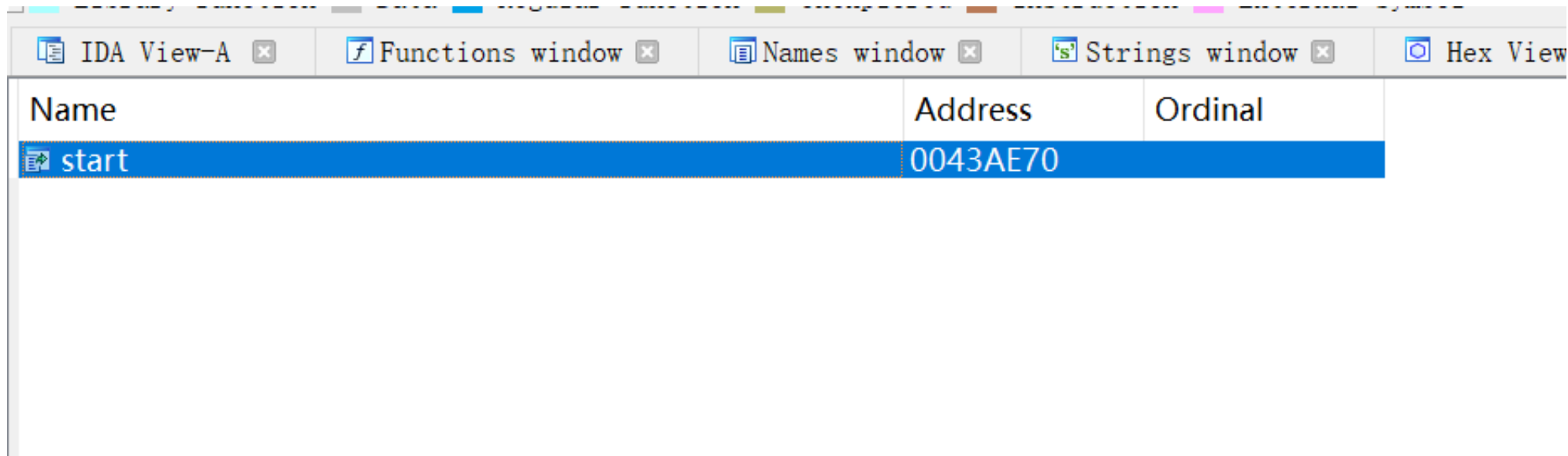
- 列出程序导入的所有函数

Address	Ordinal	Name	Library
00442000		VirtualFree	KERNEL32
00442004		HeapFree	KERNEL32
00442008		ExitProcess	KERNEL32
0044200C		TerminateProcess	KERNEL32
00442010		GetCurrentProcess	KERNEL32
00442014		GetTimeZoneInformation	KERNEL32
00442018		GetSystemTime	KERNEL32
0044201C		GetLocalTime	KERNEL32
00442020		GetLastError	KERNEL32
00442024		SetFilePointer	KERNEL32
00442028		WriteFile	KERNEL32
0044202C		ReadFile	KERNEL32
00442030		CloseHandle	KERNEL32
00442034		GetFileType	KERNEL32



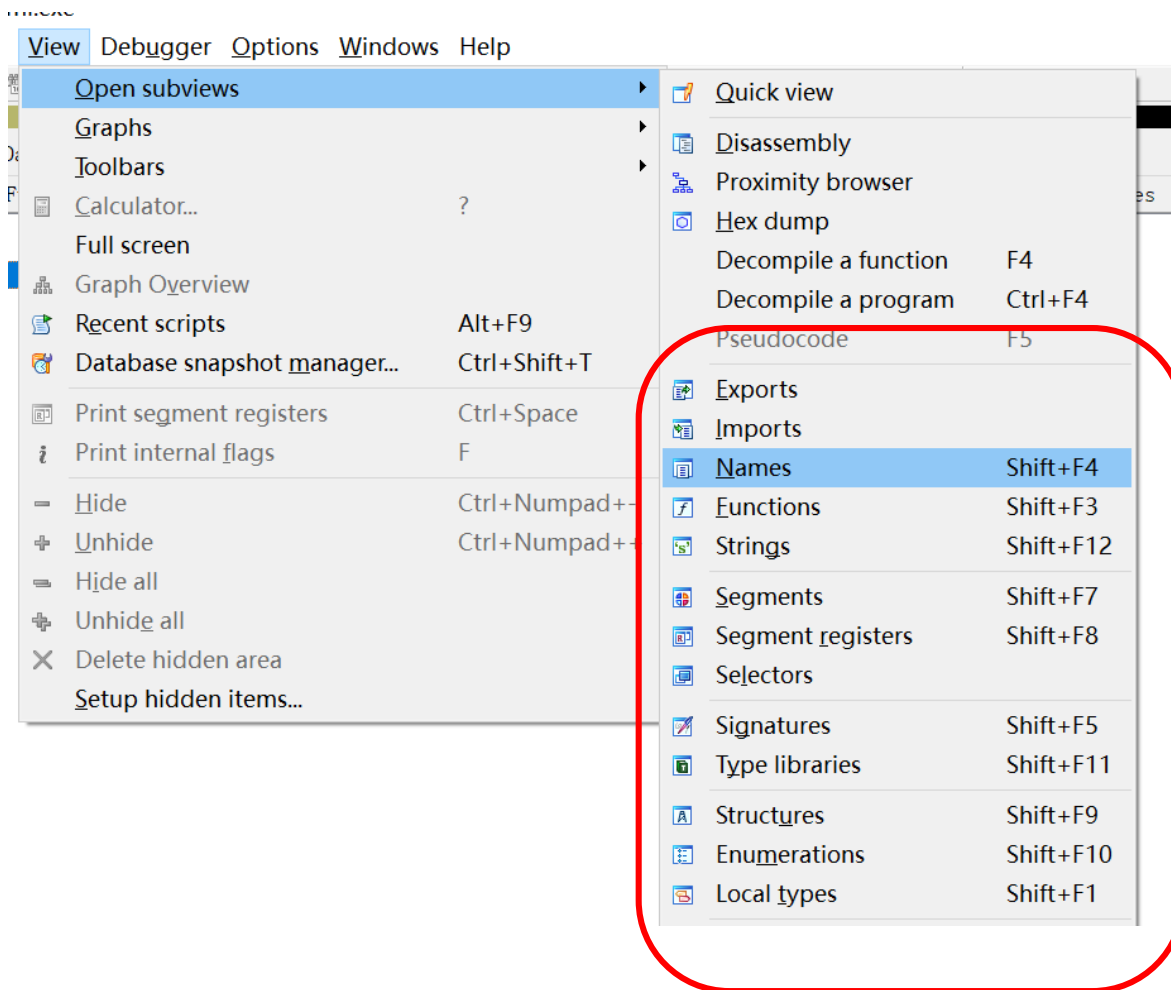
## 导出表窗口

- 列出一个函数所有导出的函数



Name	Address	Ordinal
start	0043AE70	

## 其它窗口



以下IDA Pro的哪个窗口显示EXE程序调用的WindowsAPI函数信息

- ☐ A 字符串窗口(Strings)
- ☒ B 导入表窗口(Import)
- ☐ C 导出表窗口(Export)
- ☐ D 函数窗口(Function)

提交





下面哪个IDA Pro窗口的显示内容包含函数名？

- ☒ A 字符串窗口(Strings)
- ☒ B 名字窗口 (Name)
- ☒ C 函数窗口 (Function)
- ☒ D 导入表窗口 (Import Table)

提交





南開大學

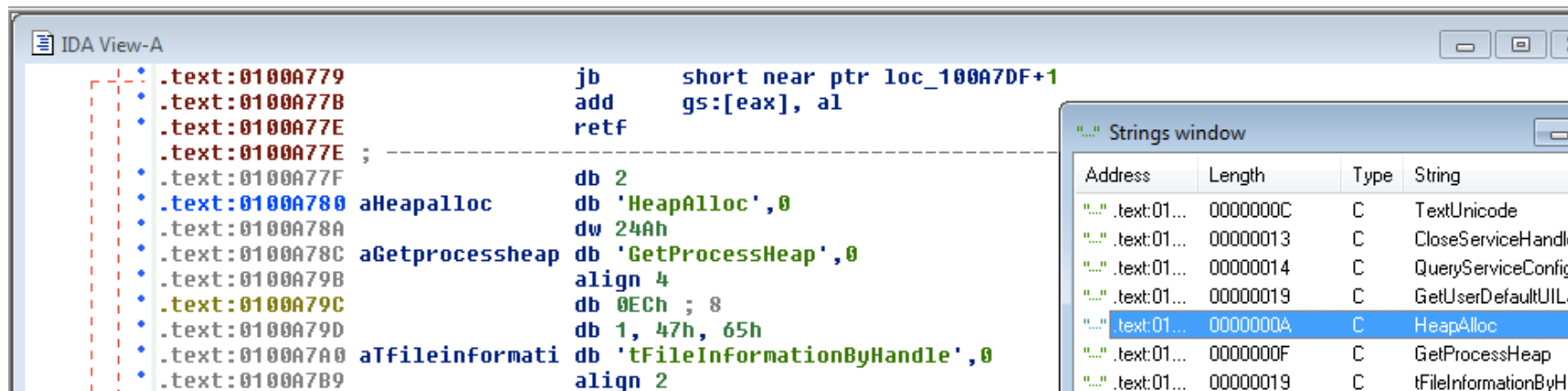
NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

## 4. IDA 操作

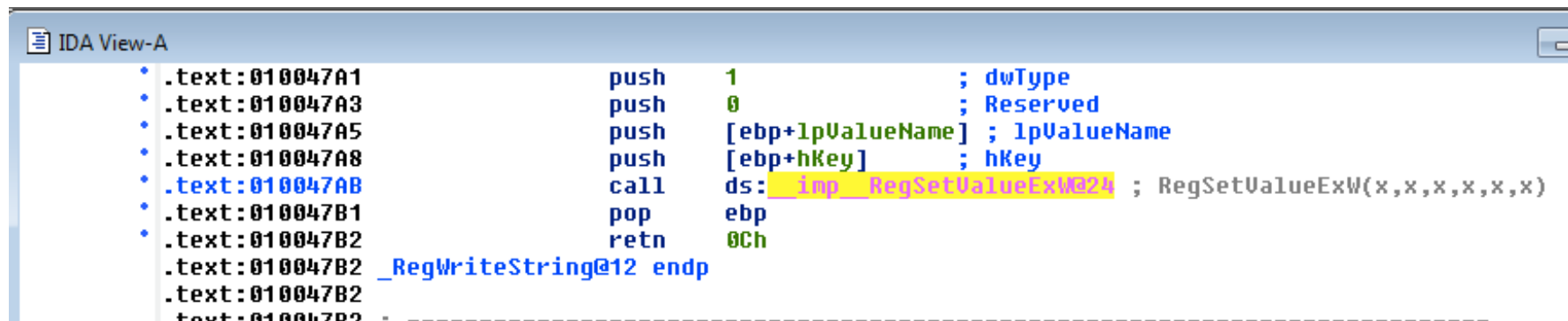
# Import和Strings窗口的导航

- 双击字符串会跳转到反汇编窗口中相应字符串的使用位置



# 链接Link

- 在反汇编窗口双击地址，IDA Pro会跳转到地址所在的反汇编窗口



```

IDA View-A
• .text:010047A1      push     1           ; dwType
• .text:010047A3      push     0           ; Reserved
• .text:010047A5      push     [ebp+lpValueName] ; lpValueName
• .text:010047A8      push     [ebp+hKey]   ; hKey
• .text:010047AB      call     ds: Imp__RegSetValueExW@24 ; RegSetValueExW(x,x,x,x,x,x)
• .text:010047B1      pop      ebp
• .text:010047B2      retn     0Ch
.text:010047B2      _RegWriteString@12 endp
.text:010047B2

```



允公允能 日新月异

# 链接类型

- sub前缀的链接
  - 函数的地址
- loc前缀的链接
  - 跳转地址



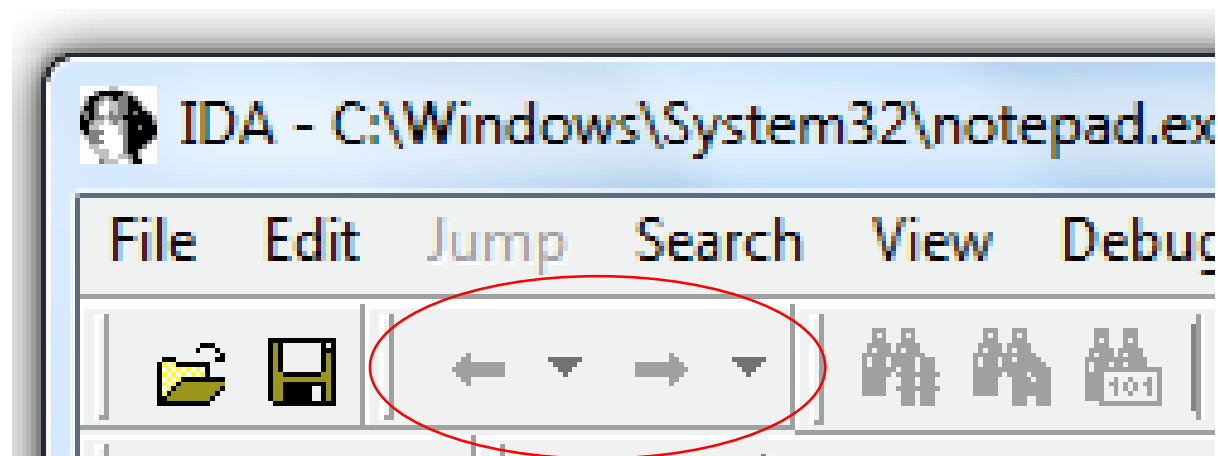
南开大学  
Nankai University



允公允能 日新月异

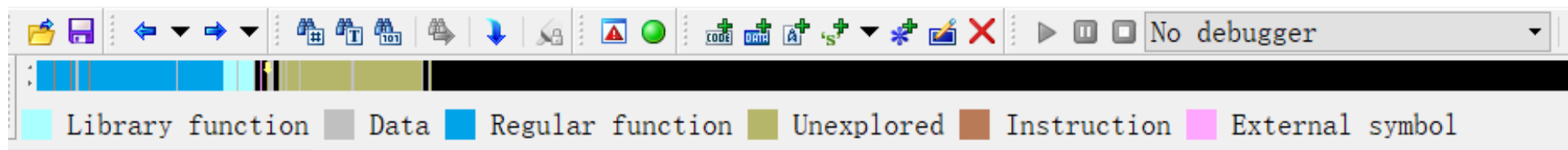
# IDA 的操作记录（History）

前进、后退按钮，跳转到之前或者周后的操作状态



南开大学  
Nankai University

# 导航栏Navigation Band



- 浅蓝色: 链接库的代码Library code
- 红色: 编译器代码Compiler-generated code
- 深蓝色: 用户写的代码 – **Analyze this**



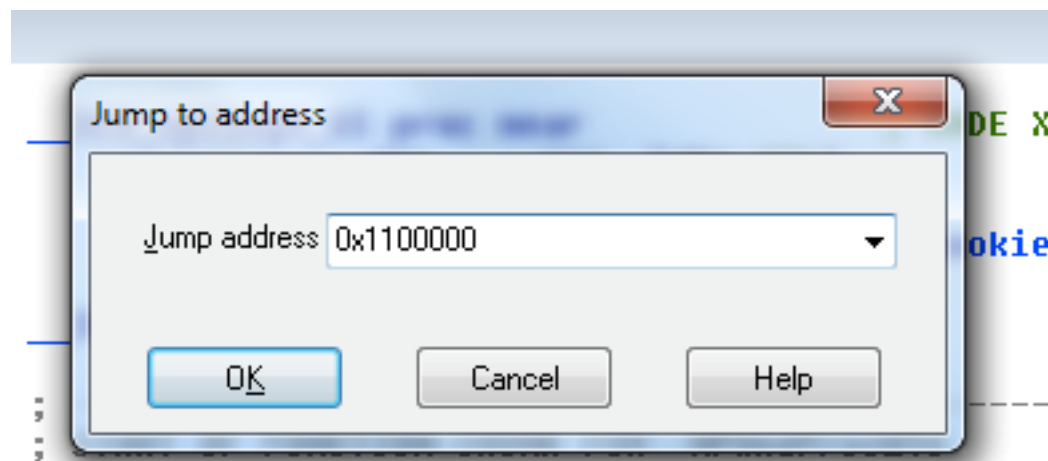




允公允能 日新月异

## 跳转到指定地址

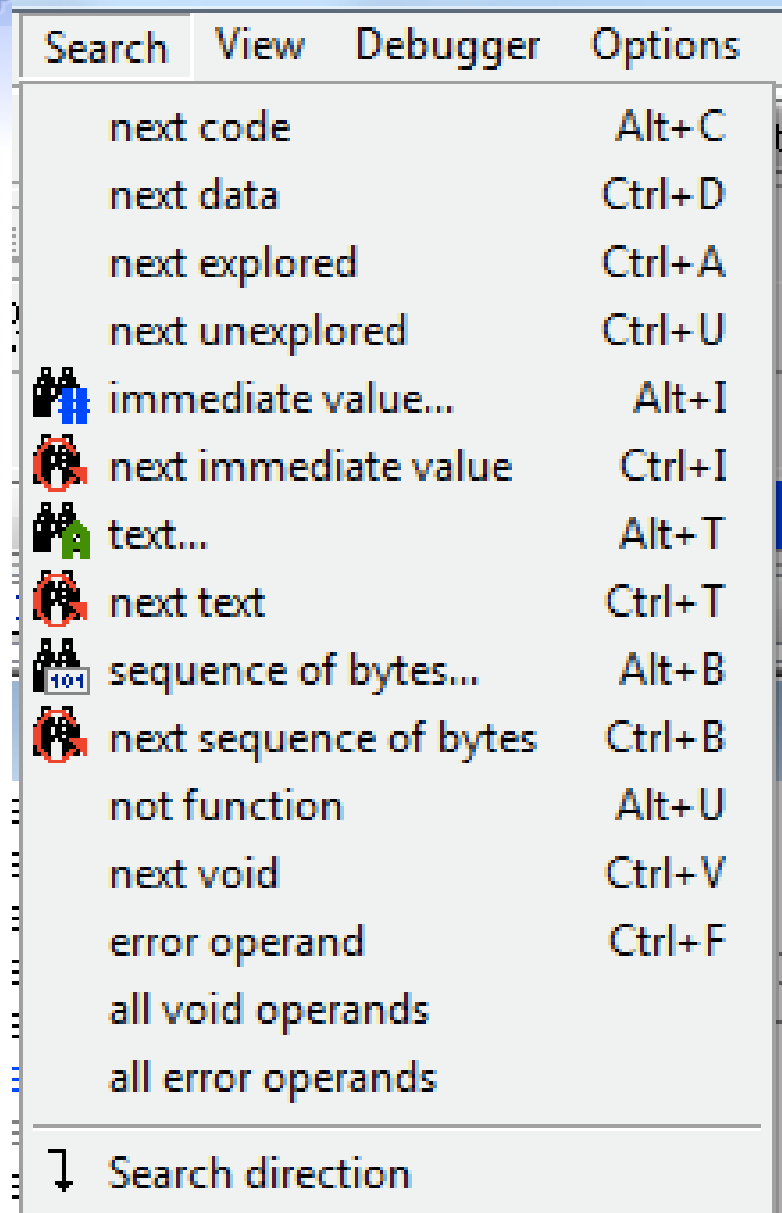
- 快捷键 **g**
- 跳转到内存地址或者地址名



# 搜索

- 在反汇编窗口搜索
  - 立即数
  - 字符串
  - 字节
  - 字节序列

IDA Freeware版本只支持字符串



以下哪个选项可以在IDA 中进行搜索？

- ☒ A 立即数
- ☒ B 字符串
- ☒ C 字节
- ☒ D 字节序列

提交





南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

## 5. 交叉引用 Cross-References

# 代码的交叉引用 CODE XREF

```
.text:00401440
.text:00401440 ; !!!!!!!!!!!!!!! S U B R O U T I N E !!!!!!!!!!!!!!!
.text:00401440
.text:00401440
.text:00401440 ; int __cdecl main(int argc,const char **argv,const char *envp)
.text:00401440 _main          proc near          ; CODE XREF: start+DE↓
.text:00401440
.text:00401440 var_44          = dword ptr -44h
.text:00401440 var_40          = dword ptr -40h
.text:00401440 var_3C          = dword ptr -3Ch
.text:00401440 var_38          = dword ptr -38h
.text:00401440 var_34          = dword ptr -34h
.text:00401440 var_30          = dword ptr -30h
.text:00401440 var_2C          = dword ptr -2Ch
.text:00401440 var_28          = dword ptr -28h
.text:00401440 var_24          = dword ptr -24h
.text:00401440 var_20          = dword ptr -20h
.text:00401440 var_1C          = dword ptr -1Ch
.text:00401440 var_18          = dword ptr -18h

                                push     offset unk_403000
                                call     _initterm
                                call     ds: _p__initenv
                                mov      ecx, [ebp+envp]
                                mov      [eax], ecx
                                push     [ebp+envp]          ; envp
                                push     [ebp+argv]           ; argv
                                push     [ebp+argc]           ; argc
                                call     main
                                add      esp, 30h
```

- CODE XREF 显示该函数在什么地方被调用了
- 默认设置只显示两处被调用的内存地址



允公允能 日新月异

## 代码的交叉引用CODE XREF

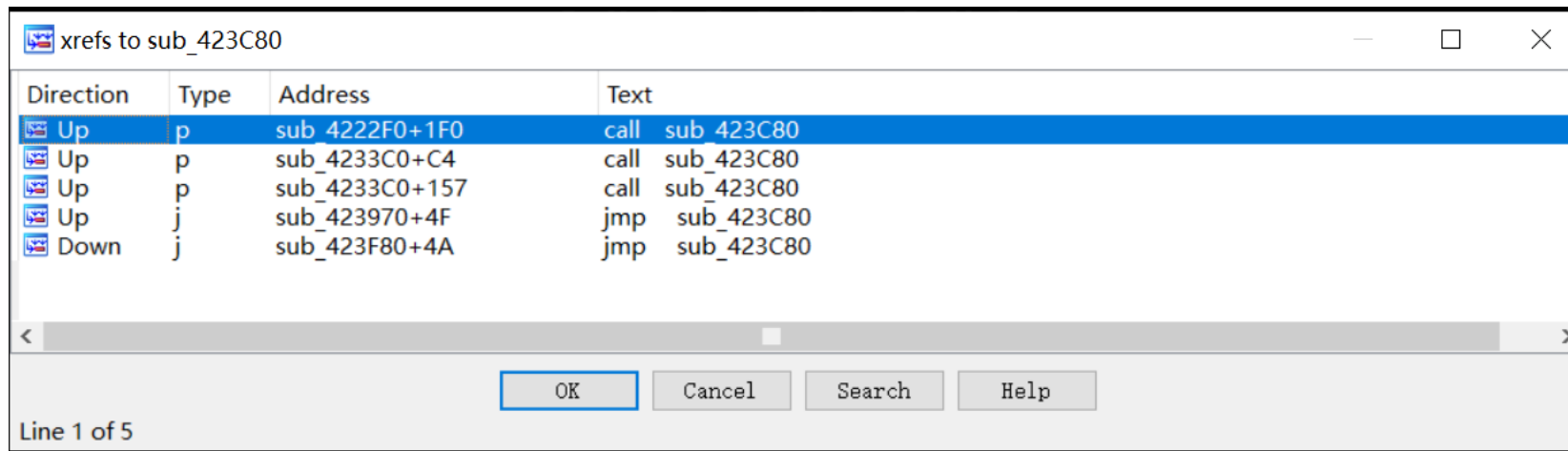
- 鼠标放置在CODE XREF的地址上，会弹出引用该数据地址上的反汇编信息
- 双击CODE XREF的地址，会跳转到该地址的反汇编窗口



南开大学  
Nankai University

# 查看所有的XREF地址

- View->Open subviews-> Cross references







# 数据的交叉引用DATA XREF

```
align 2
word_44329E dw 90h ; DATA XREF: .rdata:00442F10↑o
64 43 6C 6F 73+ db 'FindClose',0
word_4432AA dw 0B2h ; DATA XREF: .rdata:00442F14↑o
65 45 6E 76 69+ db 'FreeEnvironmentStringsA',0
word_4432C4 dw 0B3h ; DATA XREF: .rdata:00442F18↑o
65 45 6E 76 69+ db 'FreeEnvironmentStringsW',0
word_4432DE dw 106h ; DATA XREF: .rdata:00442F1C↑o
45 6E 76 69 72+ db 'GetEnvironmentStrings',0
word_4432F6 dw 108h ; DATA XREF: .rdata:00442F20↑o
45 6E 76 69 72+ db 'GetEnvironmentStringsW',0
align 10h
word_443310 dw 153h ; DATA XREF: .rdata:00442F24↑o
53 74 72 69 6E+ db 'GetStringTypeA',0
align 2
word_443322 dw 156h ; DATA XREF: .rdata:00442F28↑o
53 74 72 69 6E+ db 'GetStringTypeW',0
align 4
```



允公允能 日新月异

## 数据的交叉引用DATA XREF

- 鼠标放置在DATA XREF的地址上，会弹出引用该数据地址上的反汇编信息
- 双击DATA XREF的地址，会跳转到该地址的反汇编窗口



南开大学  
Nankai University



南開大學

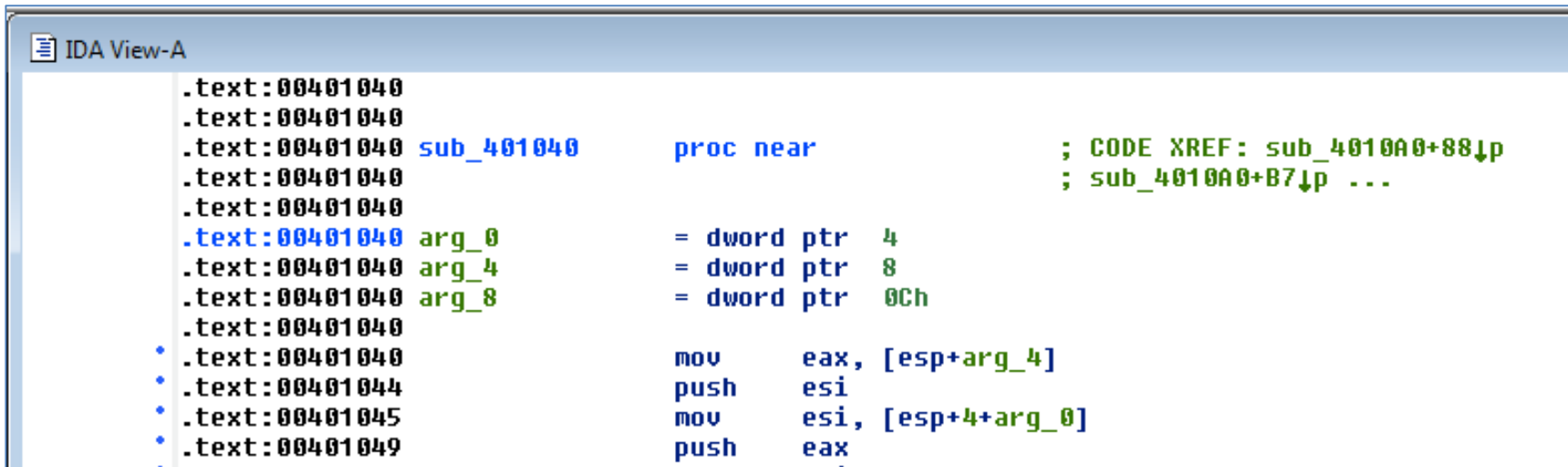
NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

## 6. 函数分析

## 函数和参数的识别

- IDA Pro 会自动识别函数，并给函数、函数的参数、函数的局部变量进行命名



```

IDA View-A
.text:00401040
.text:00401040
.text:00401040 sub_401040      proc near          ; CODE XREF: sub_4010A0+88↓p
.text:00401040                                     ; sub_4010A0+B7↓p ...
.text:00401040
.text:00401040 arg_0          = dword ptr 4
.text:00401040 arg_4          = dword ptr 8
.text:00401040 arg_8          = dword ptr 0Ch
.text:00401040
* .text:00401040      mov     eax, [esp+arg_4]
* .text:00401044      push    esi
* .text:00401045      mov     esi, [esp+4+arg_0]
* .text:00401049      push    eax
    
```



允公允能 日新月异

# 默认命名规则

- 局部变量（local variable）
  - 前缀: var\_
  - 后缀: 相对EBP的偏移值
  - 偏移值为负值





允公允能 日新月异

# 默认命名规则

- 参数（argument）
  - 前缀： arg\_
  - 后缀： 相对于EBP的偏移值
  - 偏移为正值





# 参数和局部变量

```
var_14      = dword ptr -14h
var_10      = dword ptr -10h
var_C       = dword ptr -0Ch
var_8       = dword ptr -8
var_4       = dword ptr -4
arg_0       = dword ptr 8
arg_4       = dword ptr 0Ch

push        ebp
mov         ebp, esp
sub         esp, 30h
push        esi
push        edi
mov         [ebp+var_C], 0
mov         word ptr [ebp+var_8], 0
mov         word ptr [ebp+var_4], 0
mov         byte ptr [ebp+var_10], 0
mov         eax, [ebp+arg_4]
```



在反汇编代码中，函数参数的地址，相对于EBP的偏移是一个正值还是负值？

☒ A 正值

☐ B 负值

提交





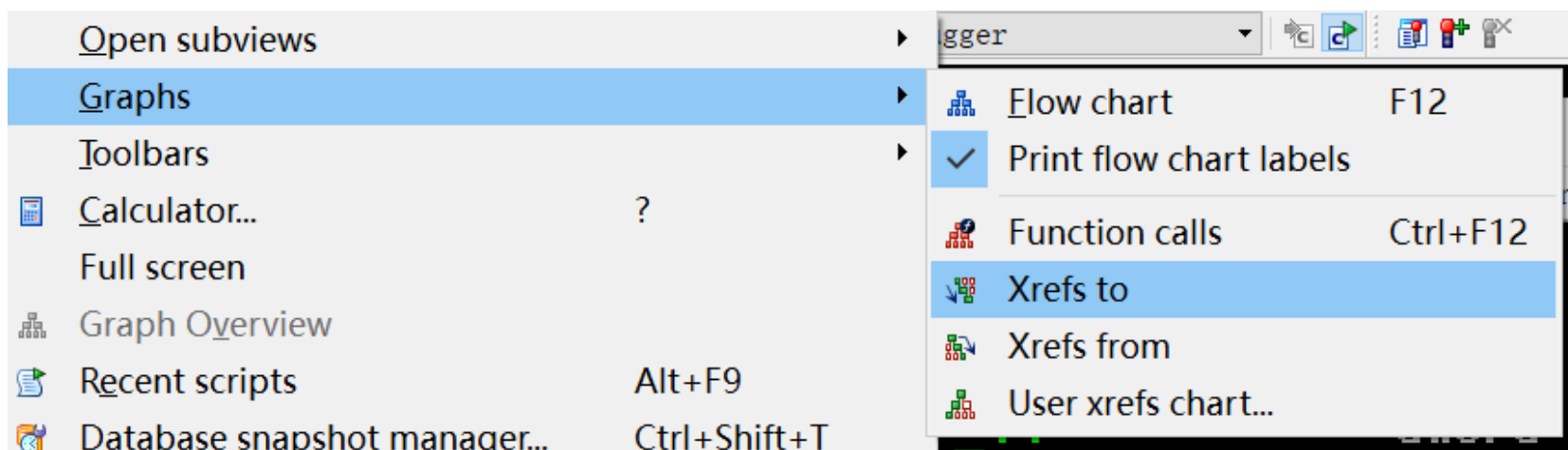
南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

## 7. 图像化显示

# IDA Pro的Graphing Options



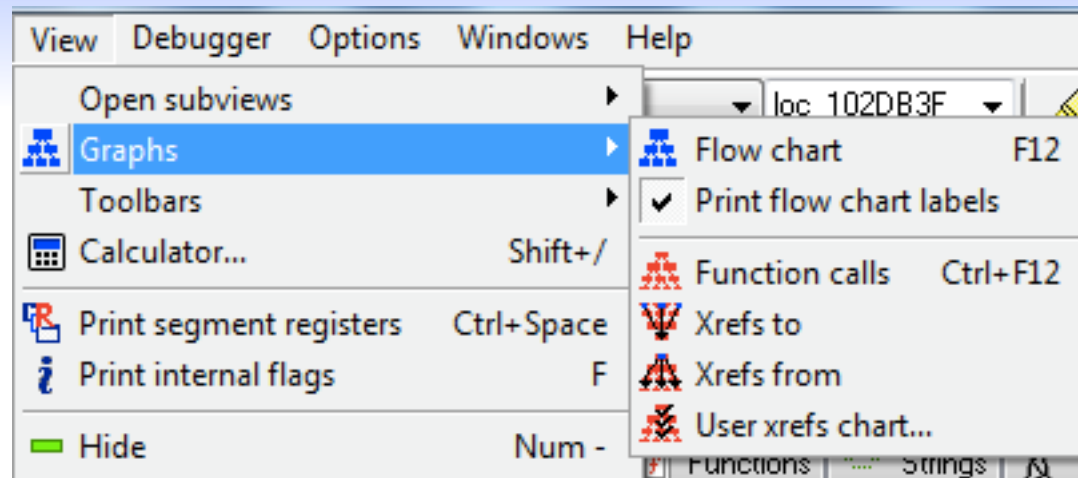
一些Freeware版本不支持



# Graphing

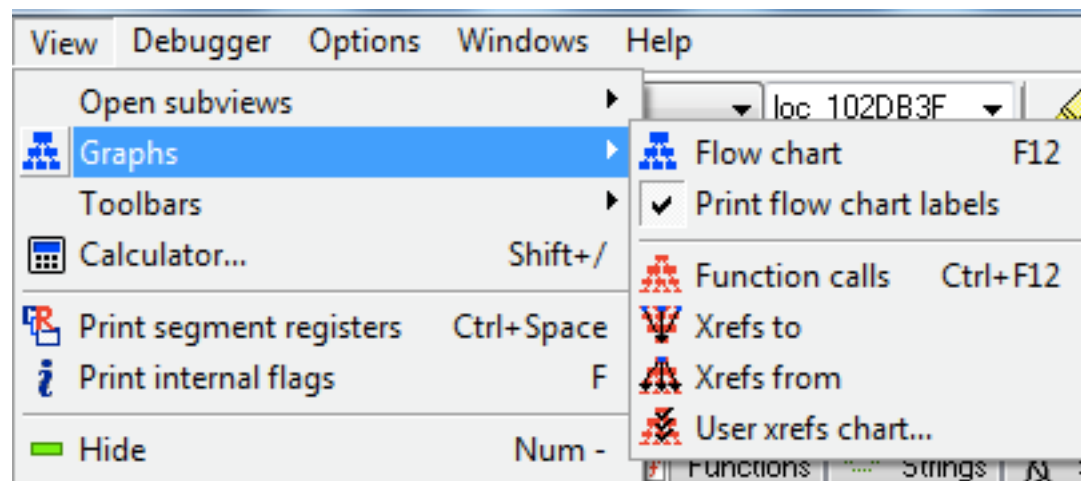
## Options

- **Flow chart**
  - 显示当前函数的控制流图
- **Function calls**
  - 显示整个程序的函数调用图





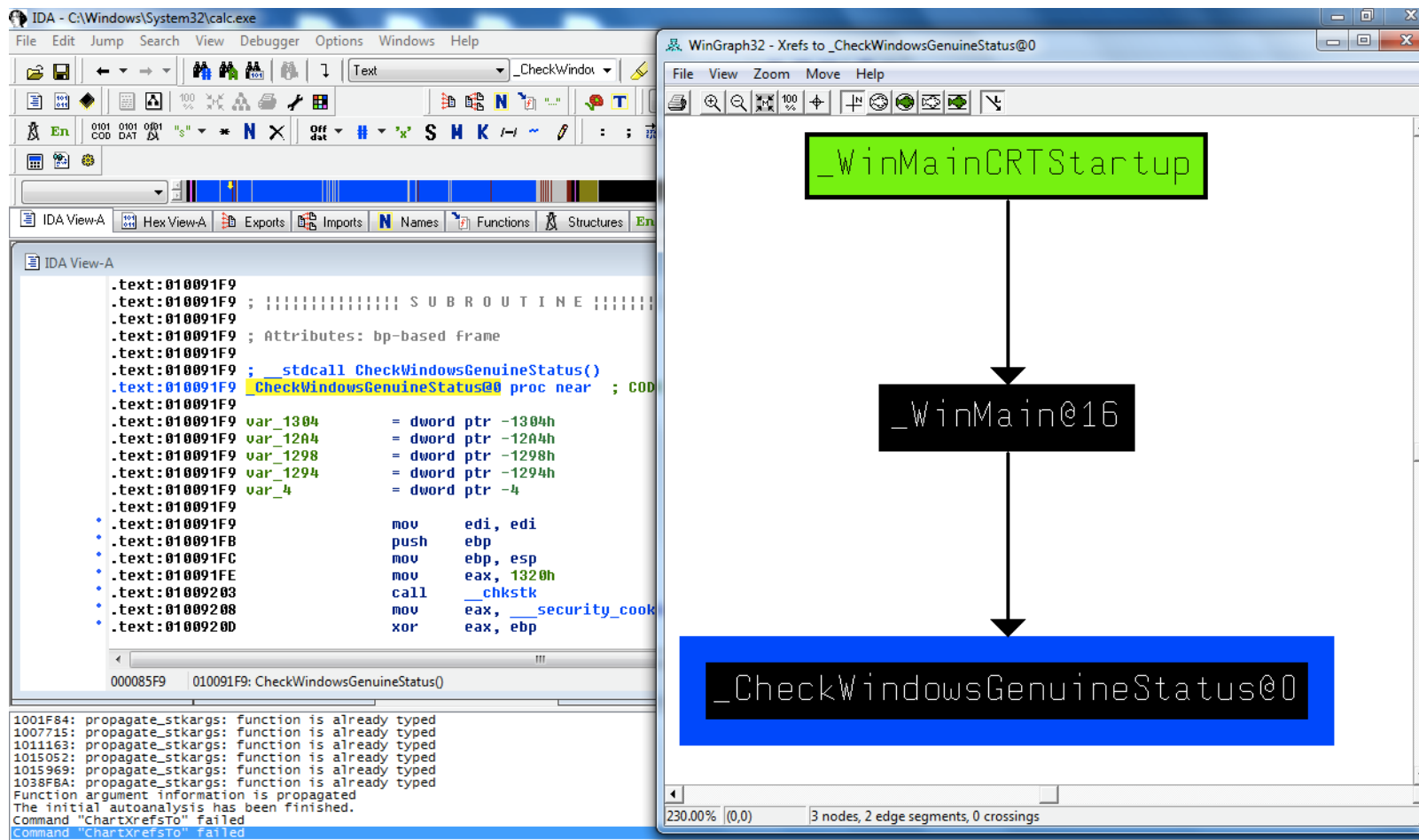
# Graphing Options



- **Xrefs to**
  - 图形化显示所有到达该函数的路径

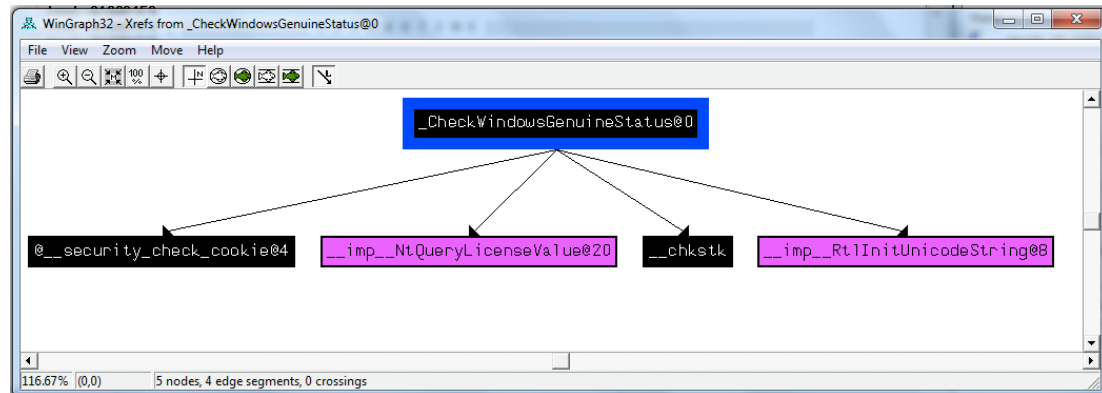


# Windows Genuine Status in Calc.exe





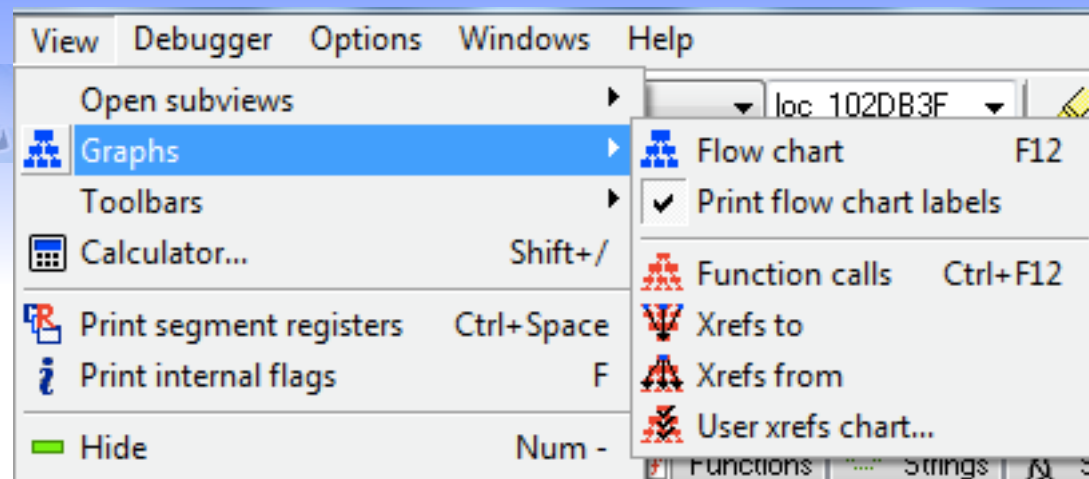
# Graphing Options



- **Xrefs from**
  - 图形化显示从该函数引出的所有路径



# Graphing Options



- **User xrefs chart...**
  - Customize graph's recursive depth, symbols used, to or from symbol, etc.
  - The only way to modify legacy graphs



下面哪一个交叉引用(XREF)的图形化功能显示的是一个函数所引出的所有路径？

- ☐ A Xrefs to
- ☒ B Xrefs from
- ☐ C Flow chart
- ☐ D Function calls



南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

## 8.增强反汇编的相关功能



允公允能 日新月异

## 内存地址的重命名

- 在IDA Pro中把一个函数名重命名为一个有意义的字符串，  
例如**sub\_401000** 重命名为**ReverseBackdoorThread**
- 重命名后，所有交叉引用的信息会自动更新



南开大学  
Nankai University



Table 6-2. Function Operand Manipulation

Without renamed arguments

```
004013C8 mov    eax, [ebp+arg_4]
004013CB push   eax
004013CC call  _atoi
004013D1 add    esp, 4
004013D4 mov    [ebp+var_598], ax
004013DB movzx  ecx, [ebp+var_598]
004013E2 test   ecx, ecx
004013E4 jnz     short loc_4013F8
004013E6 push   offset aError
004013EB call  printf
004013F0 add    esp, 4
004013F3 jmp    loc_4016FB
004013F8 ; -----
004013F8
004013F8 loc_4013F8:
004013F8 movzx  edx, [ebp+var_598]
004013FF push   edx
00401400 call  ds:htons
```

With renamed arguments

```
004013C8 mov    eax, [ebp+port_str]
004013CB push   eax
004013CC call  _atoi
004013D1 add    esp, 4
004013D4 mov    [ebp+port], ax
004013DB movzx  ecx, [ebp+port]
004013E2 test   ecx, ecx
004013E4 jnz     short loc_4013F8
004013E6 push   offset aError
004013EB call  printf
004013F0 add    esp, 4
004013F3 jmp    loc_4016FB
004013F8 ; -----
004013F8
004013F8 loc_4013F8:
004013F8 movzx  edx, [ebp+port]
004013FF push   edx
00401400 call  ds:htons
```





允公允能 日新月异

## 注释

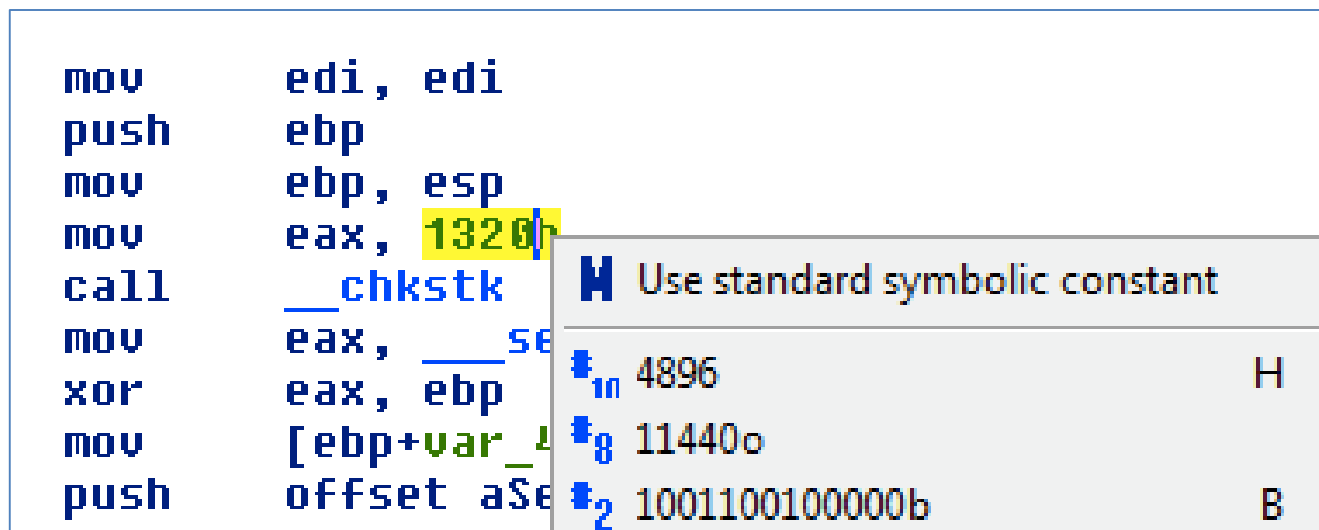
- 冒号(:), 添加注释, 不更新交叉引用XREF
- 分号(;), 添加注释, 并更新交叉引用XREF的信息





# 数字格式的转化

- 默认显示十六进制的数字
- 右键菜单中可以选择其它的数字格式







# 使用符号常量

- 使Windows API函数的参数更加清晰

Before symbolic constants	After symbolic constants
<pre>mov     esi, [esp+1Ch+argv] mov     edx, [esi+4] mov     edi, ds:CreateFileA push    0      ; hTemplateFile push    80h    ; dwFlagsAndAttributes push    3      ; dwCreationDisposition push    0      ; lpSecurityAttributes push    1      ; dwShareMode</pre>	<pre>mov     esi, [esp+1Ch+argv] mov     edx, [esi+4] mov     edi, ds:CreateFileA push    NULL   ; hTemplateFile push    FILE_ATTRIBUTE_NORMAL ; dwFlagsAndAttributes push    OPEN_EXISTING ; dwCreationDisposition push    NULL   ; lpSecurityAttributes push    FILE_SHARE_READ ; dwShareMode</pre>





允公允能 日新月异

## 重新定义代码和数字

- **U**: 撤销IDA对函数、数字的定义
- **C**: 把原始数据定义为代码
- **D**: 把原始数据定义为BYTE, WORD, DWORD
- **A**: 把原始数据定义为ASCII字符串



IDA Pro支持哪些数据格式的显示

☒ A 十六进制

☒ B 十进制

☒ C 二进制

☒ D 八进制

提交



# Plug-ins 脚本

- IDC (IDA's scripting language) 和 Python scripts available

www.openrce.org/downloads/browse/IDA_Scripts			
Decrypt Data	Unknown	IDA script to decipher data from HCU Millenium strainer stage 1 (AESCUL.EXE)	
Delphi RTTI script	RedPlait	This script deals with Delphi RTTI structures	
Export To Lib	Unknown	This script exports all functions to a lib file	
Find Format String Vulnerabilities	Unknown	A small IDC script hacked from sprintf.idc to detect format bugs currently ...	

