WELCOME TO IDA PRO

PROF. BRENDAN SALTAFORMAGGIO

SCHOOL OF ECE

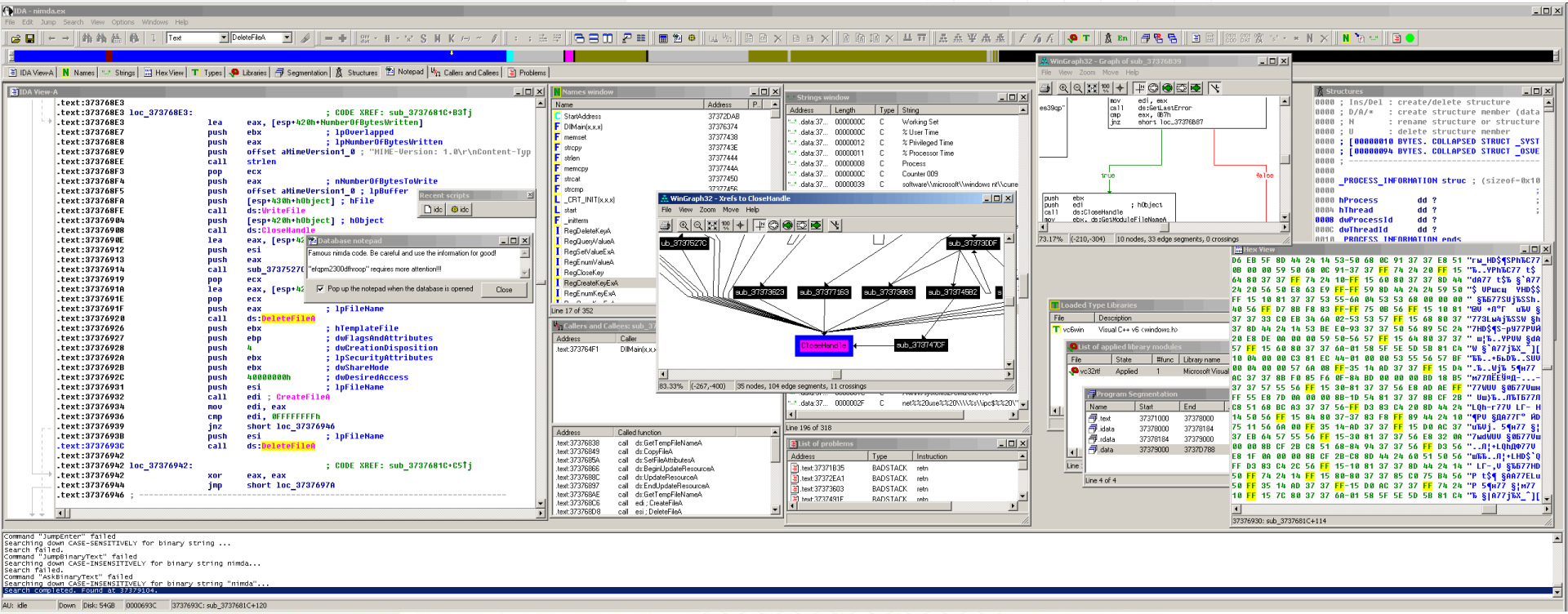
CREATING THE NEXT®

PLEASE CONSIDER THE ENVIRONMENT, AVOID PRINTING SLIDES!

WHAT IS IDA PRO?

- IDA Pro combines an interactive, programmable, multi-processor disassembler coupled to a local and remote debugger and augmented by a complete plugin environment

- IDA Pro is the industry standard for hostile code analysis, vulnerability research, software validation, interactive debugging, and much more



THANK YOU TO THE SCHOOL OF ECE!!

- IDA Pro enables multi-processor disassembly and debugging across more than 50 processor families
- IDA Pro provides a unique advantage to software security research and education
- Simply having IDA Pro experience makes you a top candidate for many careers
  - Malware analyst, low-level software developer, security researcher, …

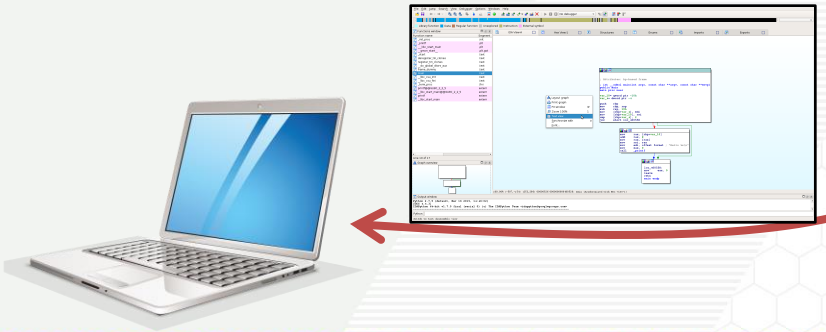
- The School of ECE saw the exciting potential of providing students access to IDA Pro

- IDA Pro is not cheap --- A single floating license costs over $2800 USD!
- The School of ECE has graciously purchased 30 floating licenses for this course!
- We are among a small group of universities which have this educational benefit

HOW TO ACCESS IDA PRO

- IDA Pro is installed in the School of ECE's cloud servers
- Four very powerful Red Hat Linux 7 machines
  - ece-linlabsrv01.ece.gatech.edu
  - 100+ GB memory and 24 cores each!

- IDA Pro runs on those machines, we can connect to the GUI in 2 ways:
  1) Standard SSH with X11 forwarding
     - Best on fast internet connections (e.g., on campus)
  2) FastX Client
     - Best on slow internet connections



ece-linlabsrv01.ece.gatech.edu

FASTX CLIENT

- FastX is a custom X server & client implementation
  - It is optimized to be more efficient over slow internet connections
  - Standard SSH with X forwarding is terrible over slow connections
- Georgia Tech OIT provides FastX Client for Windows, Linux, and Mac
  - Available at: http://software.oit.gatech.edu
- The IDA Pro Servers are running the custom FastX server
- The FastX Client handles connecting to the IDA Pro Servers and displaying the X11 GUI
- Supported by Georgia Tech OIT ☺

DO NOT FORGET TO VPN!

- We can only access these machines through the Georgia Tech VPN!
  - Even on eduroam you need the VPN
  - Only the on-campus ECE computer labs can access them without the VPN

- Georgia Tech OIT provides & supports VPN clients for Linux, Windows, Mac, …
- https://gatech.service-now.com/home?id=kb_article_view&sysparm_article=KB0026837

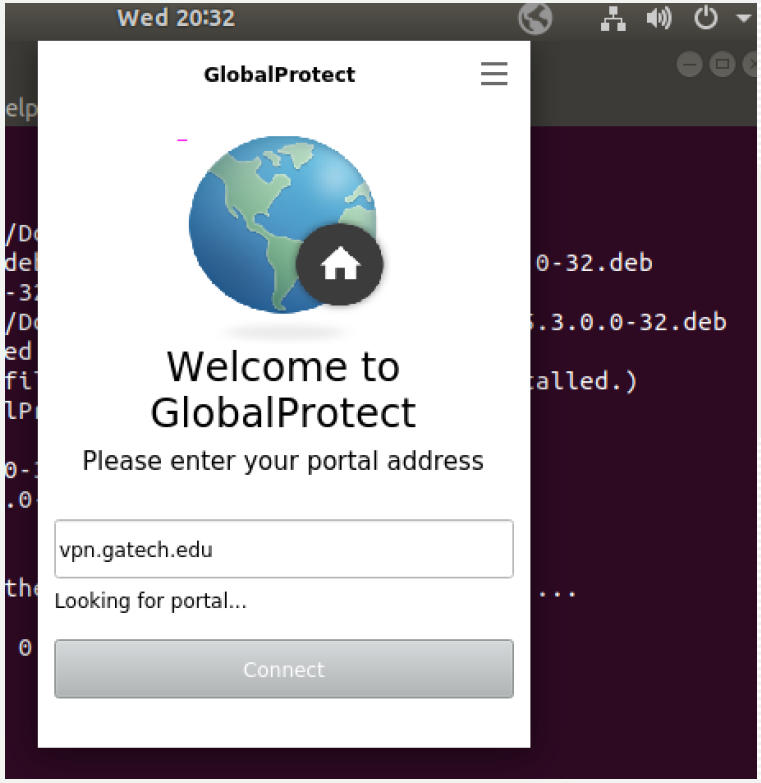
Georgia Tech

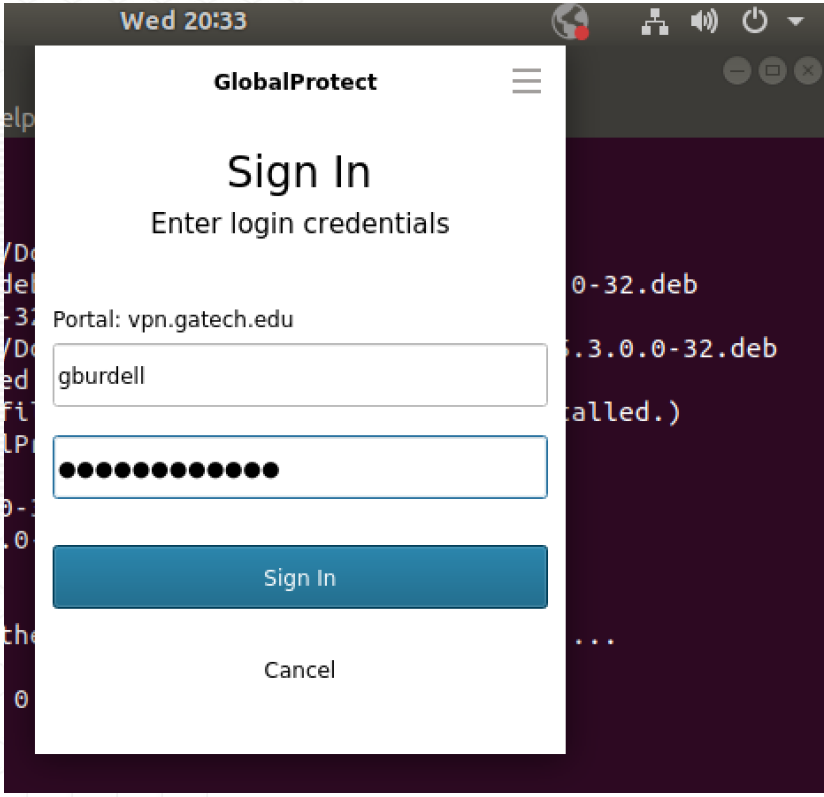
HOW TO CONNECT TO IDA PRO SERVERS FROM LINUX

CREATING THE NEXT®

LINUX VPN

https://gatech.service-now.com/home?id=kb_article_view&sysparm_article=KB0028027





AFTER VPN, STANDARD SSH WITH X11 FORWARDING

- Be sure to set X11 forwarding on, compression on, and cipher preferences for fast ciphers

Terminal - brendan@brendan-laptop: ~

File Edit View Terminal Tabs Help

```
brendan@brendan-laptop:~$ ssh -XC bds3@ece-linlabsrv01.ece.gatech.edu
```

- ece-linlabsrv01.ece.gatech.edu
- 100+ GB memory and 24 cores each!

IDA Pro runs on those machines, we can connect to the GUI in

1) Standard SSH with X11 forwarding
   - Best on fast internet connections (e.g., on campus)
2) FastX Client

ENTER PASSWORD AND YOU'RE CONNECTED!

Terminal - brendan@brendan-laptop: ~

File Edit View Terminal Tabs Help

```
brendan@brendan-laptop:~$ ssh -XC bds3@ece-linlabsrv01.ece.gatech.edu
*****************************************************************************
This computer system is the property of the Georgia Institute of Technology.
Any user of this system must comply with all Institute and Board of Regents
policies, including the Acceptable Use Policy, Cyber Security Policy and
Data Privacy Policy (http://b.gatech.edu/it-policies). Users should have no
expectation of privacy, as any and all files on this system may be
intercepted, monitored, recorded, copied, inspected, and/or disclosed to
authorized personnel in order to meet Institute obligations.

By using this system, I acknowledge and consent to these terms.
*****************************************************************************

bds3@ece-linlabsrv01.ece.gatech.edu's password:
```

Terminal - brendan@brendan-laptop: ~

File Edit View Terminal Tabs Help

```
 login session at the discretion of the CSG. Flagrant
 or repeated violations may in extreme cases be
 subject to suspension or revocation of account
 privileges. Jobs running for longer than 5 days on
 the ecelinsrv systems will be automatically
 terminated at the discretion of the CSG. If you need
 to run substantially long jobs, email help@ece
 requesting a PACE account.

If you are having problems running applications, see
 http://www.ece-help.gatech.edu/unix/cshrc.html for
 information on configuring your login environment.

bds3@ece-linlabsrv01.ece.gatech.edu>
bds3@ece-linlabsrv01.ece.gatech.edu>
bds3@ece-linlabsrv01.ece.gatech.edu>
bds3@ece-linlabsrv01.ece.gatech.edu>
bds3@ece-linlabsrv01.ece.gatech.edu>
```

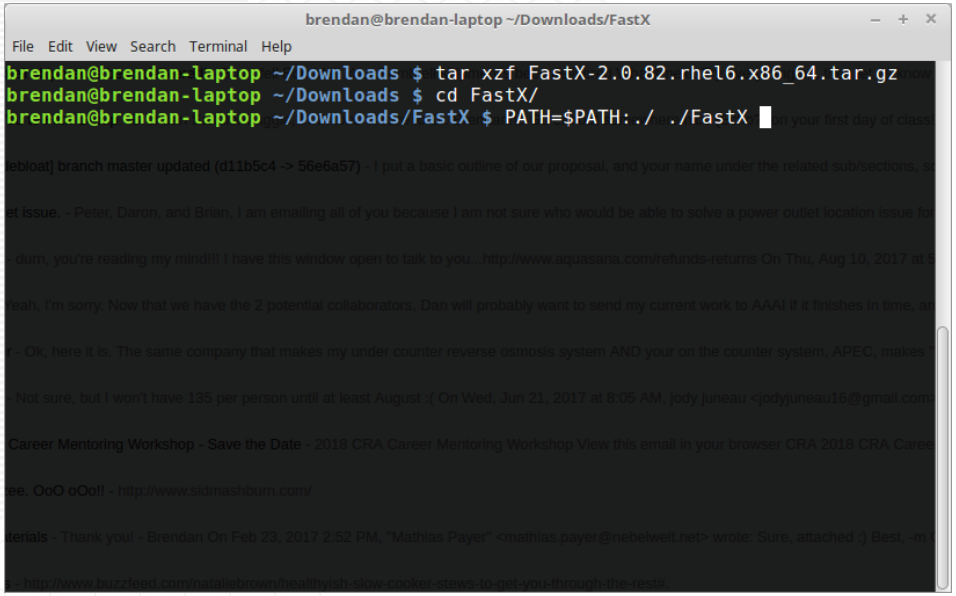
- Now you have a terminal with X11 forwarding from ece-linlabsrv01.ece.gatech.edu

ANOTHER LINUX OPTION: LINUX FASTX CLIENT

- SSH with X forwarding is **terrible** over slow connections

- FastX is optimized for slow connections
  - Also, Georgia Tech OIT will always recommend that you use FastX

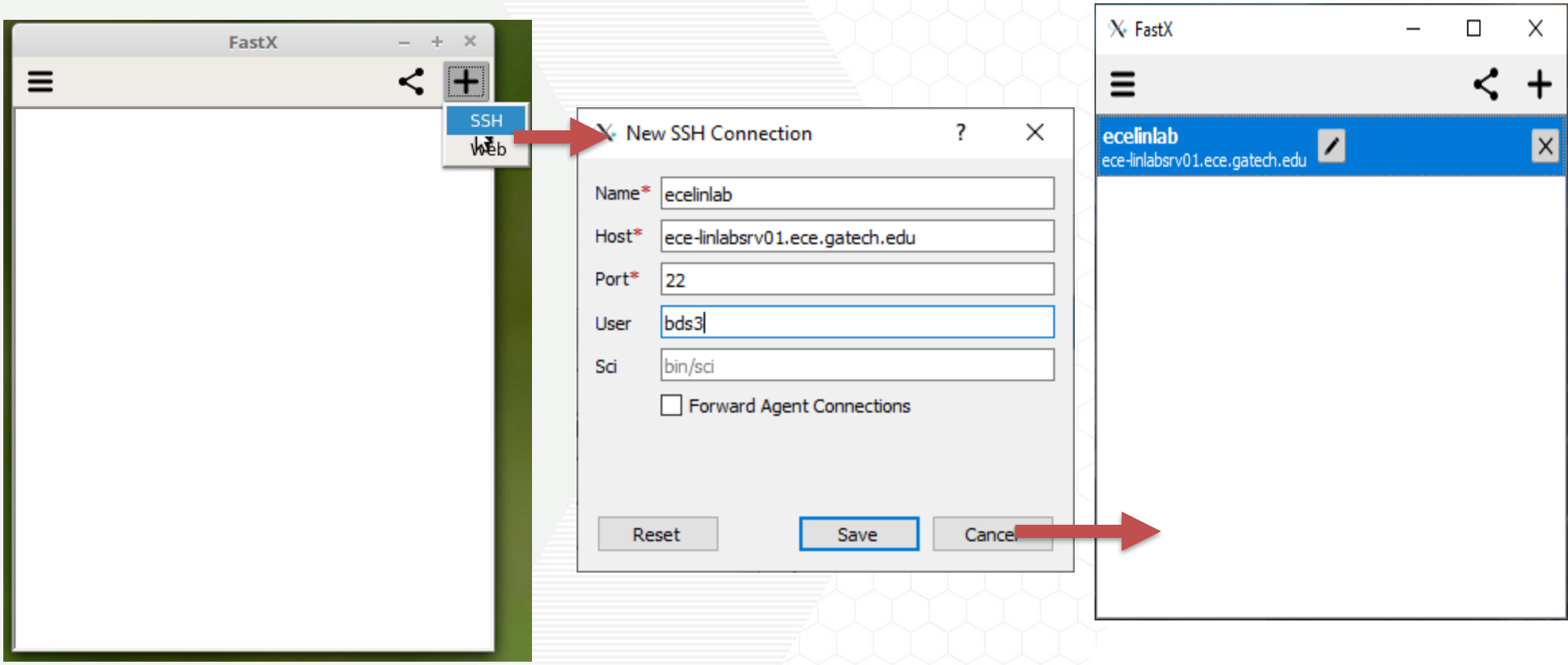
1. Download FastX from GT OIT
   - http://software.oit.gatech.edu
2. Extract the tar.gz file
3. cd to the extracted FastX directory
4. Execute: $ PATH=$PATH:./ ./FastX



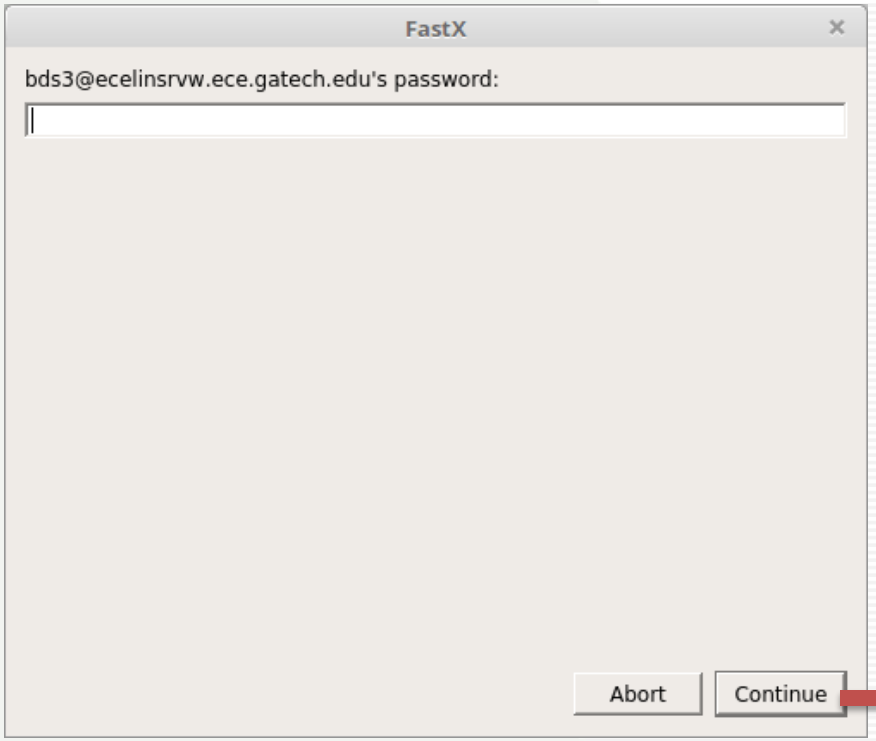
- There is a BUG in the Linux FastX client!
- It needs the FastX directory in the PATH to correctly find its dependencies!

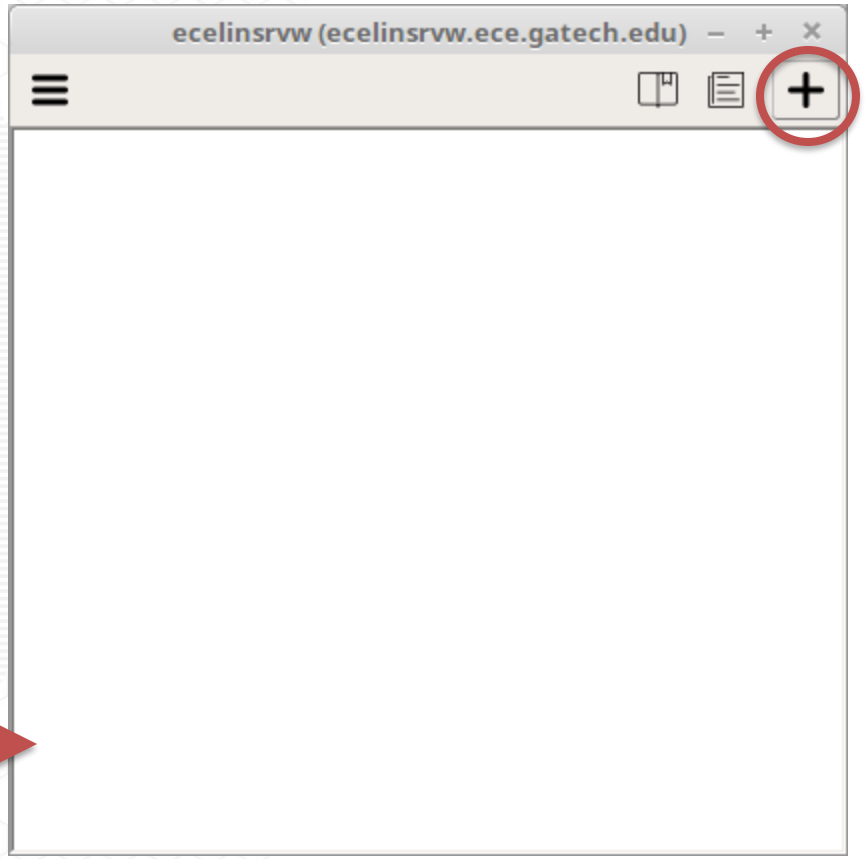
CREATING THE NEXT®

ANOTHER LINUX OPTION: LINUX FASTX CLIENT

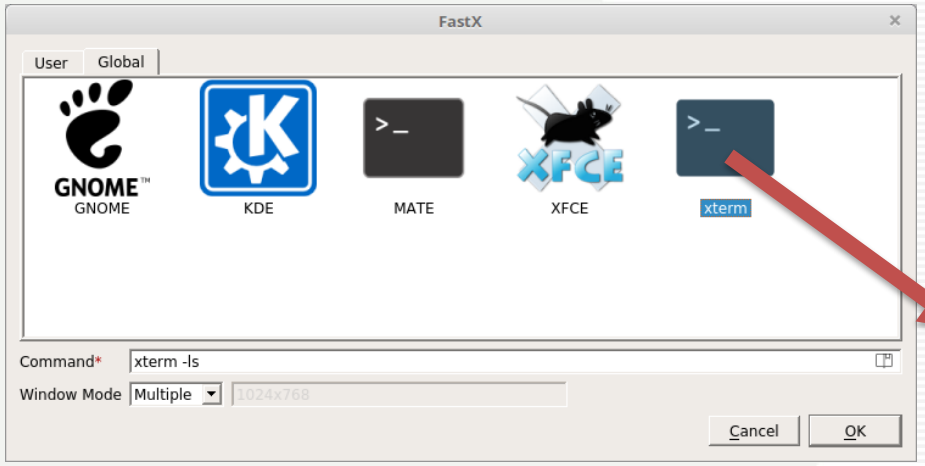


ANOTHER LINUX OPTION: LINUX FASTX CLIENT

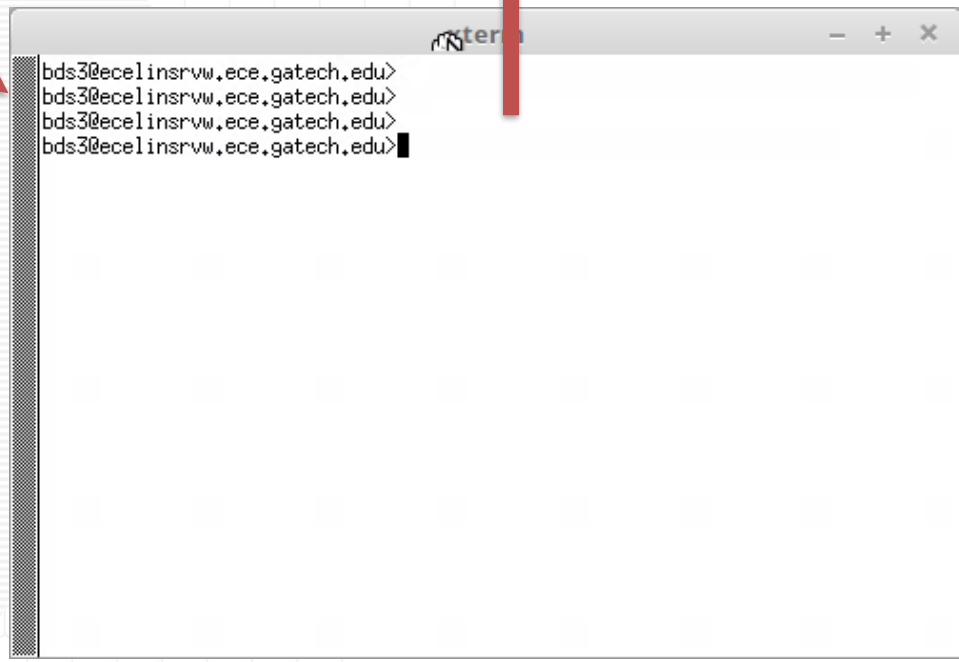




ANOTHER LINUX OPTION: LINUX FASTX CLIENT



- Now you have a terminal with X11 forwarding from ece-linlabsrv01.ece.gatech.edu



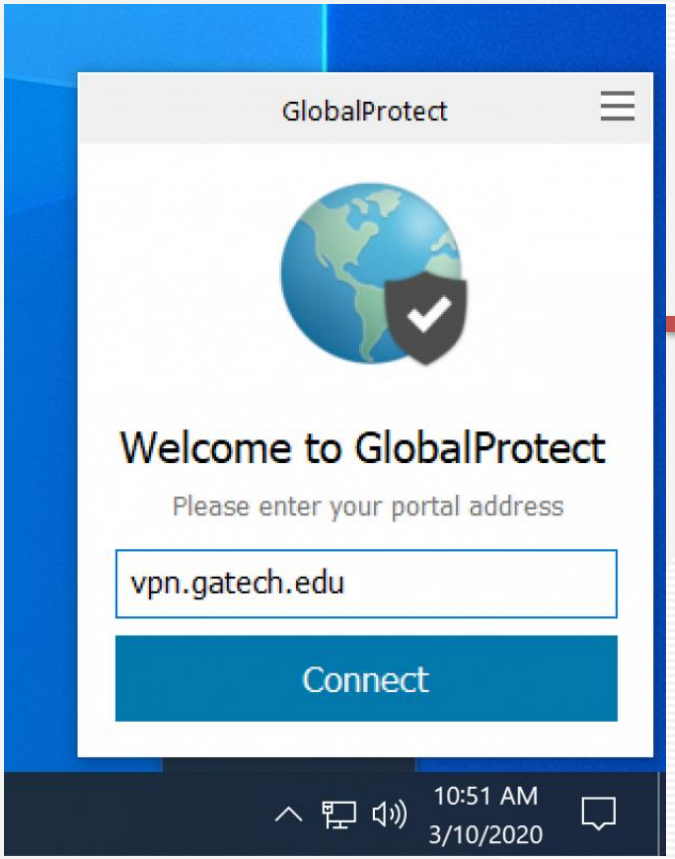
Georgia Tech

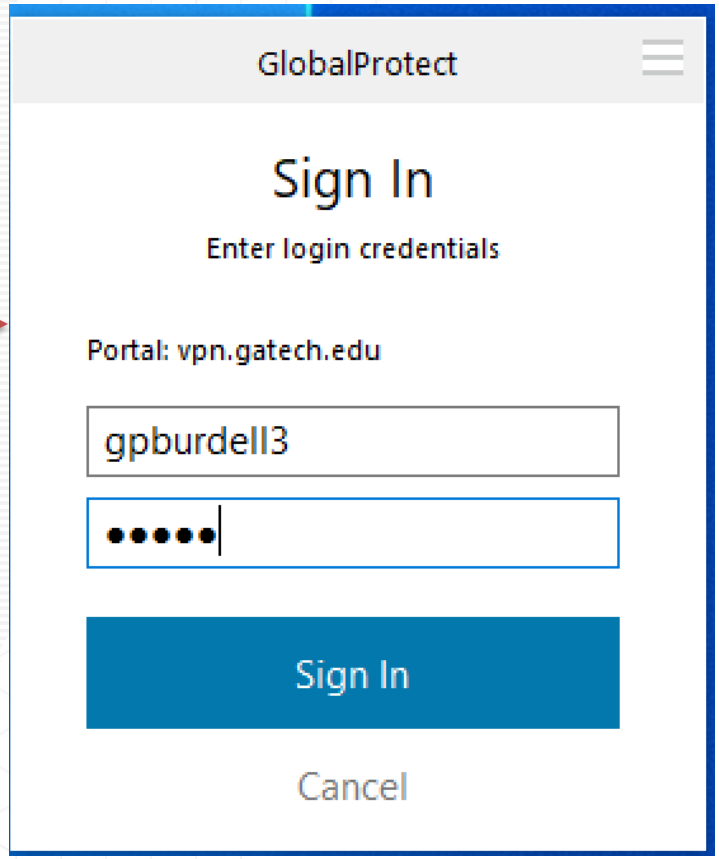
HOW TO CONNECT TO IDA PRO SERVERS FROM WINDOWS

CREATING THE NEXT®

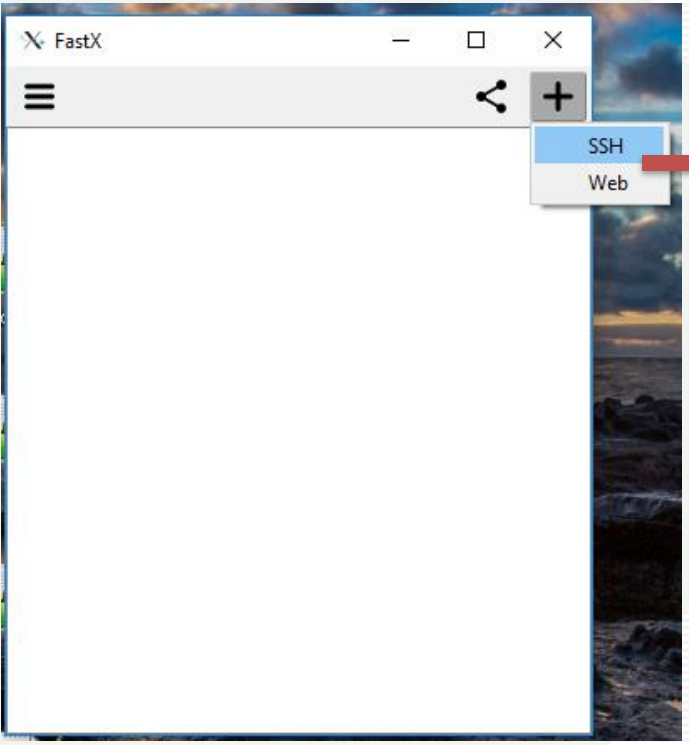
WINDOWS VPN

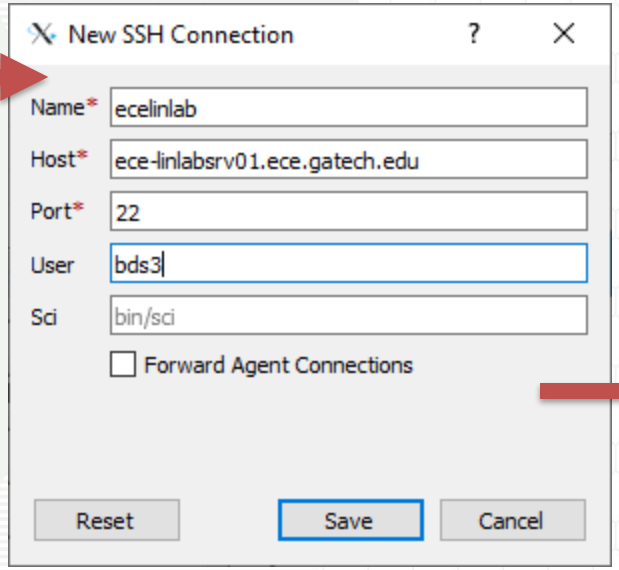
https://gatech.service-now.com/home?id=kb_article_view&sysparm_article=KB0026742

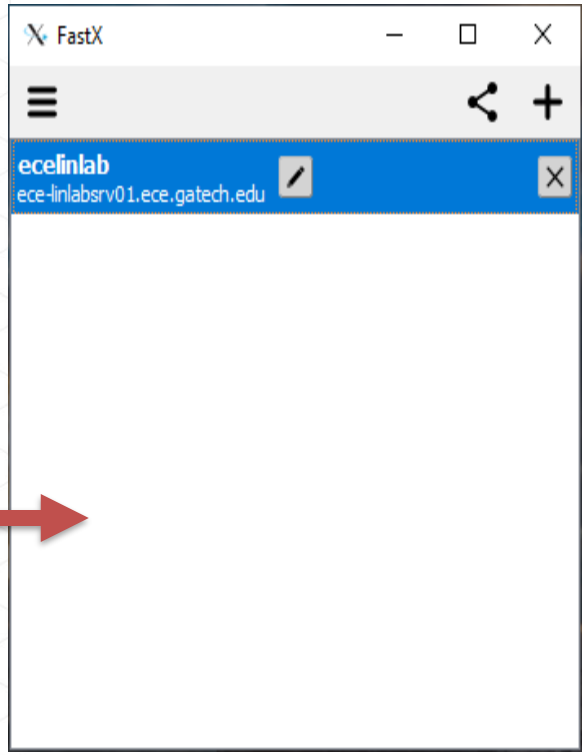




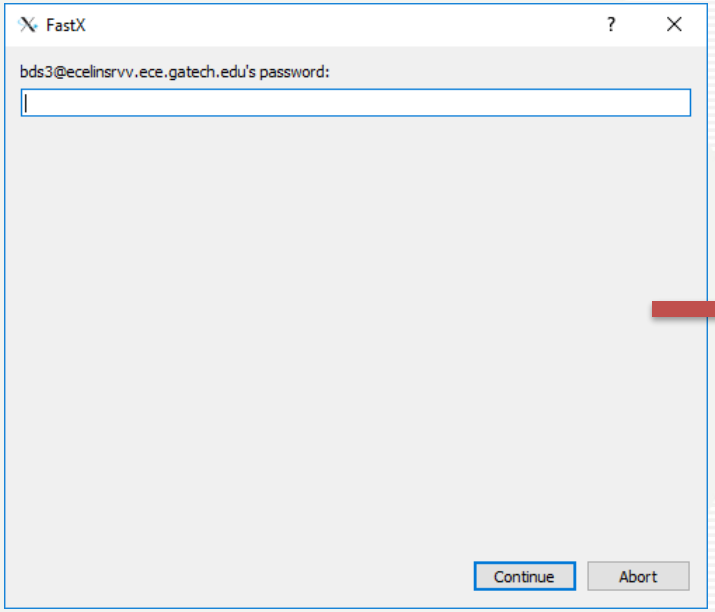
AFTER VPN, WINDOWS FASTX CLIENT

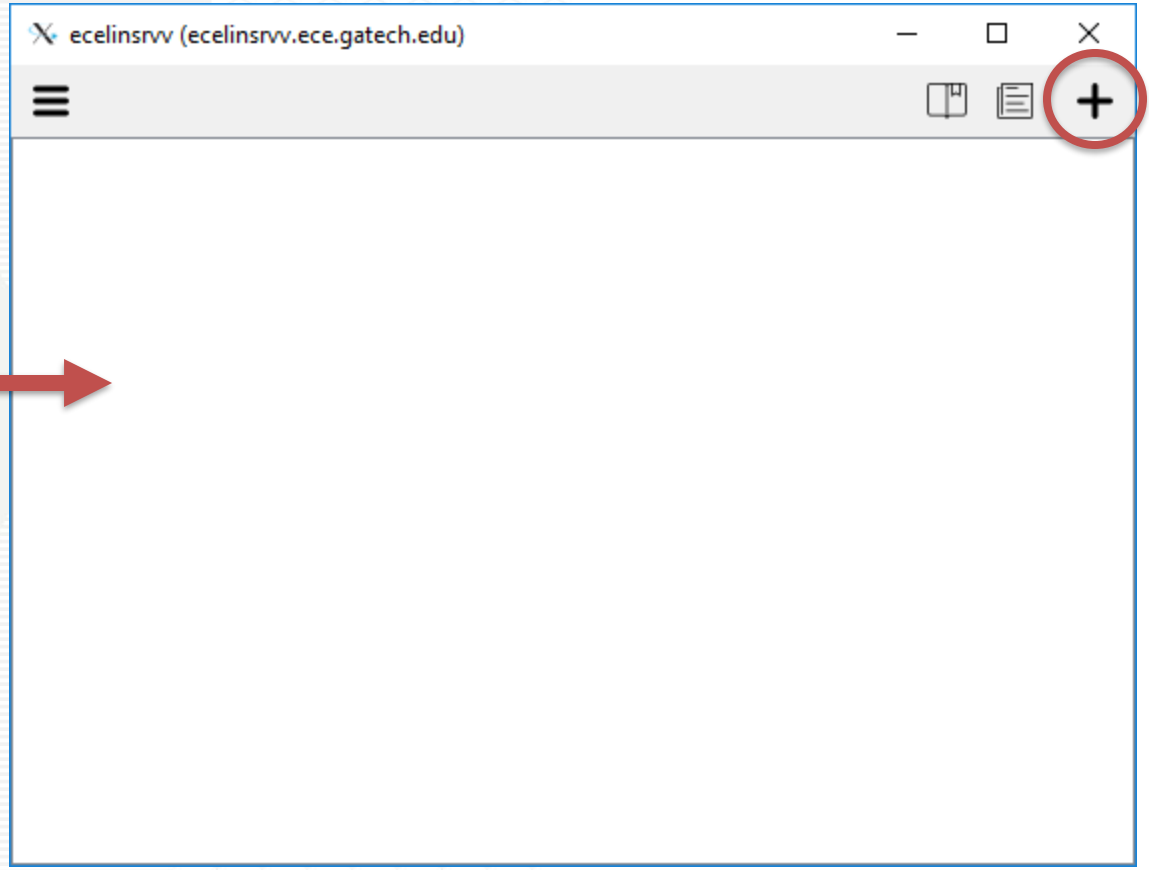




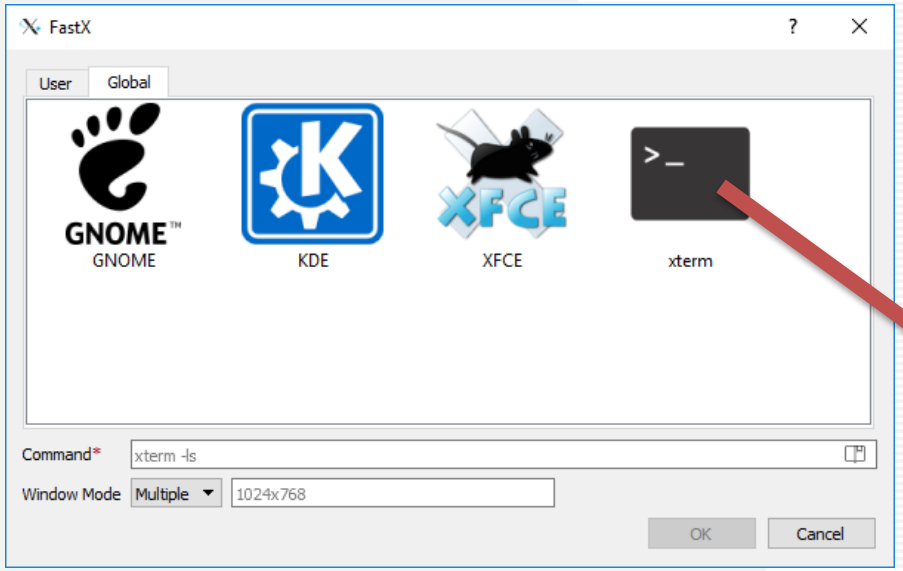


LAUNCH APPS IN FASTX

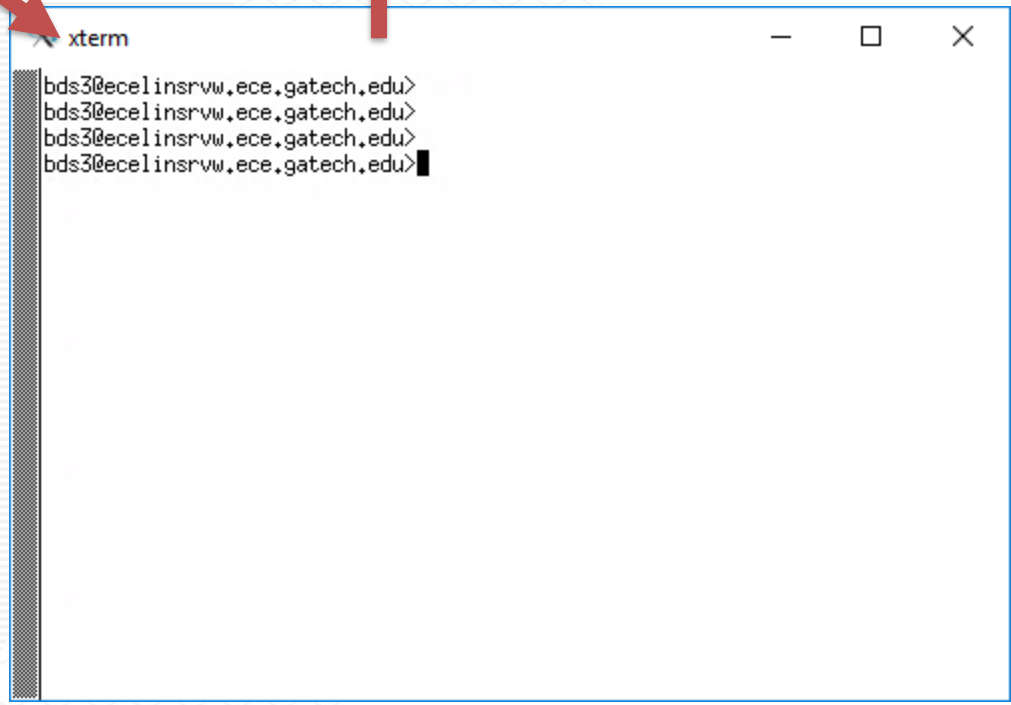




LAUNCH APPS IN FASTX



- Now you have a terminal with X11 forwarding from ece-linlabsrv01.ece.gatech.edu



ONCE YOU ARE CONNECTED TO THE IDA PRO SERVERS…

YOU ARE IN ECE TERRITORY!

ALL PROBLEMS MUST BE SENT TO:

HELP@ECE.GATECH.EDU

CREATING THE NEXT®

IDA PRO ENVIRONMENT

- Before you can do anything with IDA Pro, you **must** first set up the running environment
- Use the following command: $ source /tools/software/hex-rays/idapro.csh

- Note: You may not see the /tools/software/hex-rays/ directory until you execute that command
  - It is our "secret" key!

- Only needs to be done once (per terminal) to set up the environment

- Everything related to IDA Pro is available in /tools/software/hex-rays/
  - We have read & execute permissions in that folder (after executing "$ source …")
- Feel free to look through the directories, read the docs, and check out the SDK!

A TALE OF TWO IDAS

- IDA Pro has two different executables
  - Which executable you use depends on what binary you are analyzing

- **ida** for working with 32-bit binaries
- **ida64** for working with 64-bit binaries

- You will get an error if you use the wrong one

- I recommend executing them with “&” after to put the process in the background

NOTICE THE TCSH!

- The default shell on the ECE machines should be /bin/tcsh
- THIS IS NOT BASH! But it is similar
  - https://help.ece.gatech.edu/linux/shells
- The "$ source /tools/software/hex-rays/idapro.csh" command will ONLY work in tcsh!
- You can check your current shell with "echo $0"
- After you run "$ source …" you can switch to /bin/bash if you want

```
bds3@ecelinsrvw.ece.gatech.edu>
bds3@ecelinsrvw.ece.gatech.edu>
bds3@ecelinsrvw.ece.gatech.edu>echo $0
/bin/tcsh
bds3@ecelinsrvw.ece.gatech.edu>source /tools/idapro/ida-
bds3@ecelinsrvw.ece.gatech.edu>/bin/bash
[bds3@ecelinsrvw ~]$ ida &
[1] 44579
[bds3@ecelinsrvw ~]$
[1]+  Exit 1                  ida
[bds3@ecelinsrvw ~]$ ida64 &
[1] 44602
[bds3@ecelinsrvw ~]$
[1]+  Exit 1                  ida64
[bds3@ecelinsrvw ~]$
```

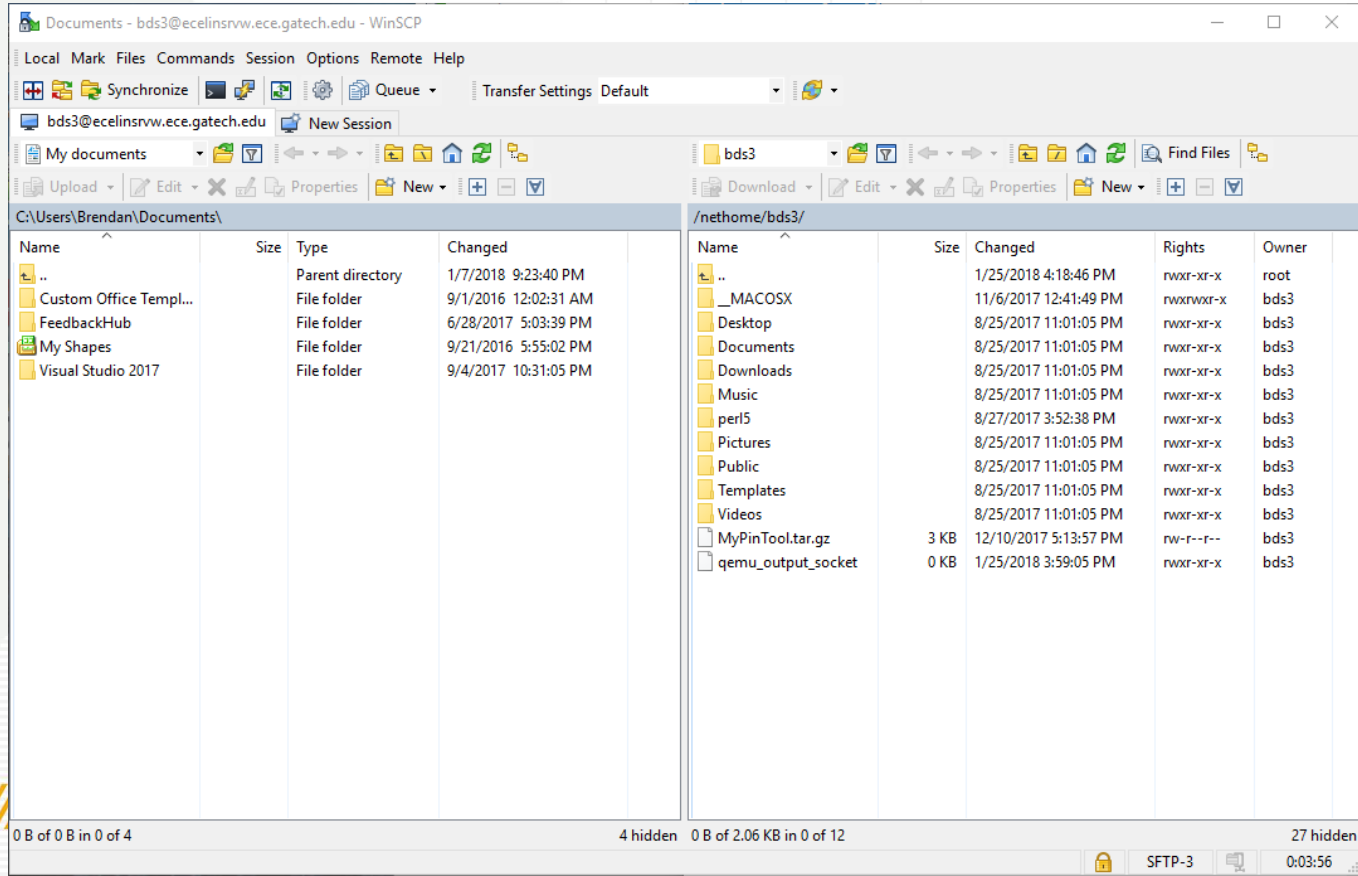
CLOUD IDA

- Notice that IDA Pro will be executing on the remote server
- So you have to move any files/test cases/etc. to your home directory before you begin
- You also need to pull your answers/submission filed back down when you're finished
- Many remote file copy utilities exist for every platform
- On Linux, use the scp command:
  - Please see this helpful cheat-sheet for a range of scp uses: http://www.hypexr.org/linux_scp_help.php

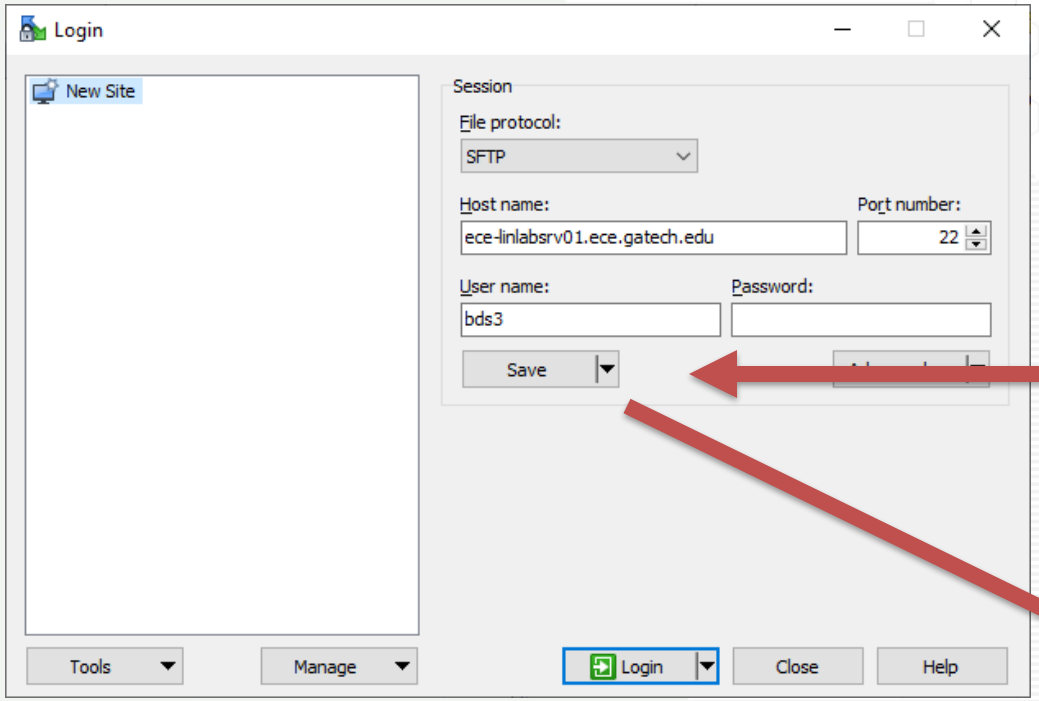
```
brendan@brendan-laptop ~/homeworks/hello $ scp ./hello bds3@ecelinsrvw.ece.gatech.edu:~/homeworks/
*****************************************************************************
This computer system is the property of the Georgia Institute of Technology.
Any user of this system must comply with all Institute and Board of Regents
policies, including the Acceptable Use Policy, Cyber Security Policy and
Data Privacy Policy (http://b.gatech.edu/it-policies). Users should have no
expectation of privacy, as any and all files on this system may be
intercepted, monitored, recorded, copied, inspected, and/or disclosed to
authorized personnel in order to meet Institute obligations.

By using this system, I acknowledge and consent to these terms.
*****************************************************************************

bds3@ecelinsrvw.ece.gatech.edu's password:
hello                                                 100% 8608     8.4KB/s   00:00
brendan@brendan-laptop ~/homeworks/hello $
```

WINSCP FOR THOSE WHO LIKE GUIS

- Another great option on Windows is WinSCP
- Download from: https://winscp.net/eng/download.php
- The exact same source code as SCP, just wrapped in a Windows GUI
- Simply drag and drop files between the two machines ☺

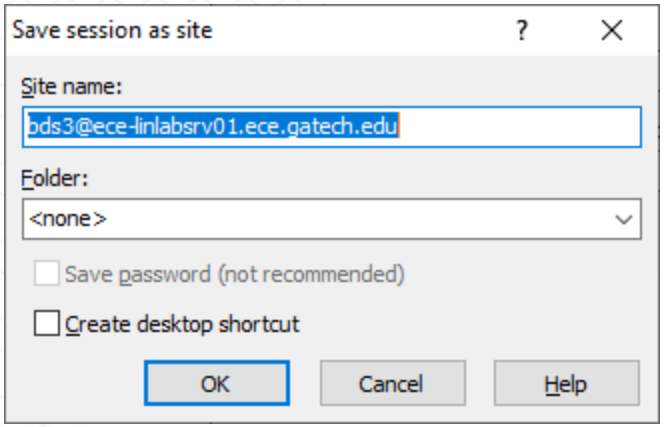


SIGNING INTO WINSCP



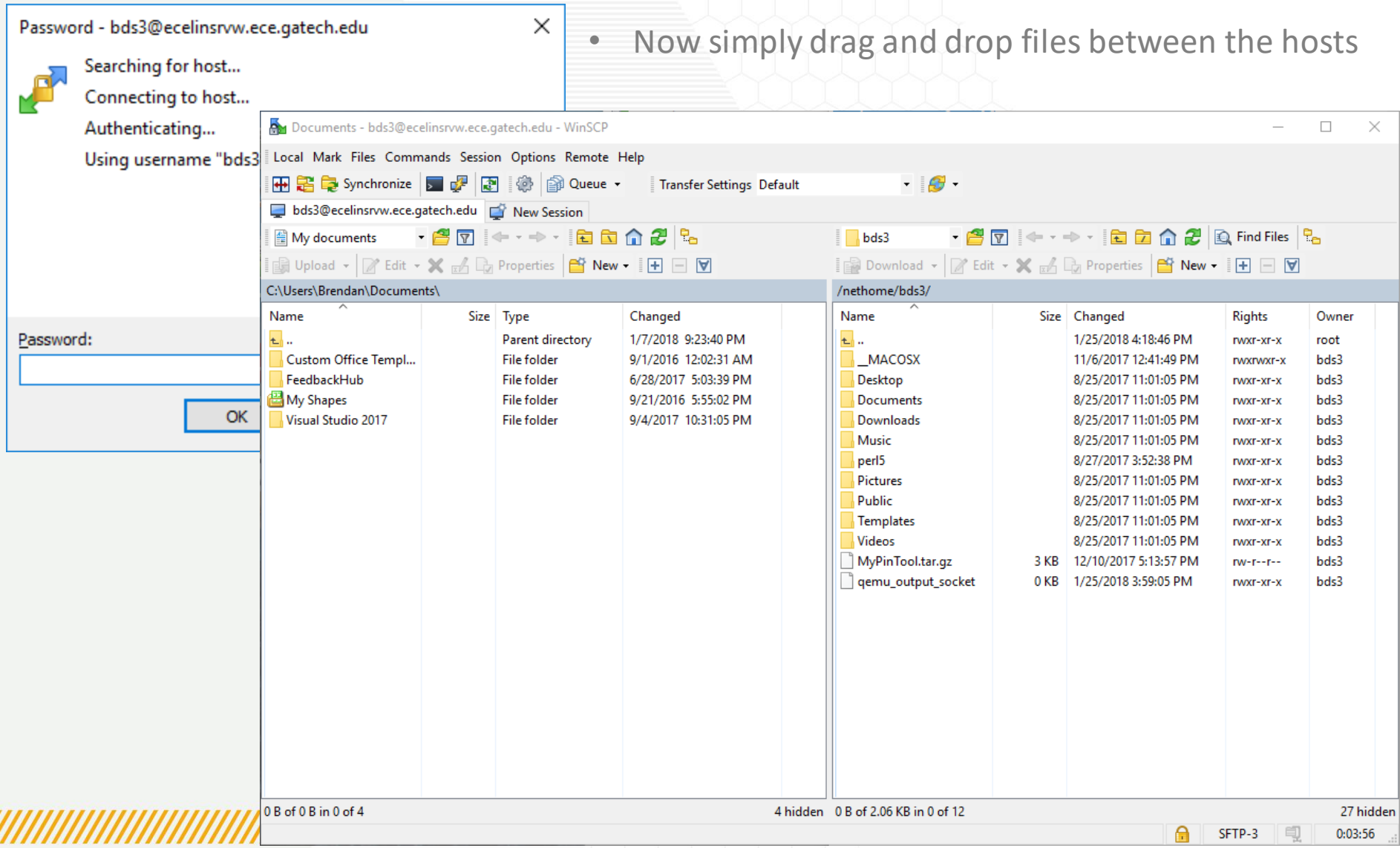
- Protocol = SFTP
- Fill in the host name and username fields
- Port = 22
- Remember to save!

- Give the session a nice name



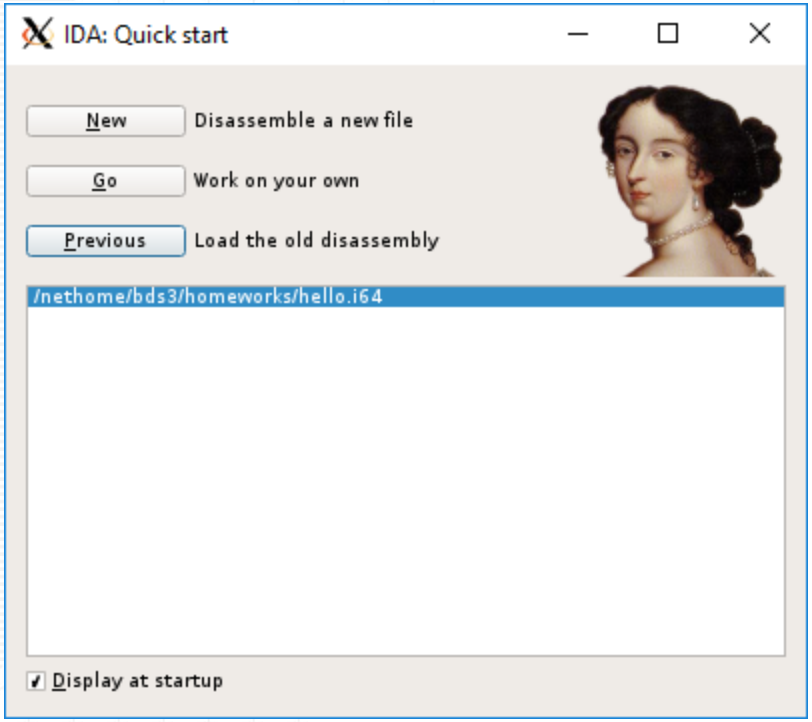
ENTER PASSWORD & YOU'RE ALL SET!

- Now simply drag and drop files between the hosts

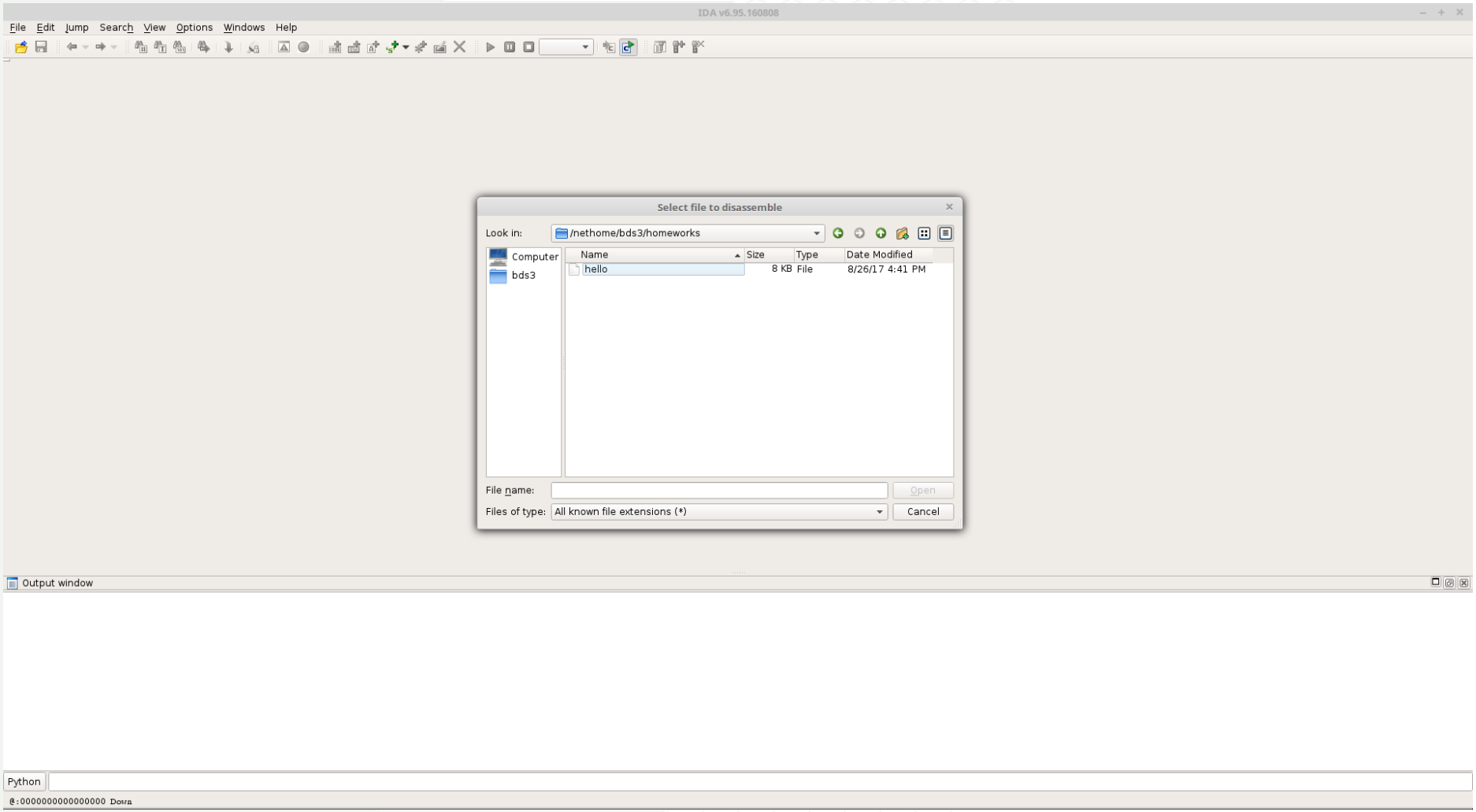


SELECT YOUR ANALYSIS TARGET!

- When IDA Pro starts it will ask you to start a new disassembly or open a previous one



LOADING A NEW FILE



IDA WILL THEN ASK FOR LOADING INSTRUCTIONS

- The defaults are almost always correct … unless you are dealing with nasty malware!

IDA v6.95.160808

File Edit Jump Search View Options Windows Help

Load a new file

Load file /nethome/bds3/homeworks/hello as

ELF64 for x86-64 (Executable) [elf64.llx64]
Binary file

Processor type

MetaPC (disassemble all opcodes) [metapc] Set

Loading segment 000000000000
Loading offset 000000000000

Analysis
- [x] Enabled
- [x] Indicator enabled

Kernel options 1 | Kernel options 2
Processor options

Options
- [ ] Loading options
- [ ] Fill segment gaps
- [x] Create segments
- [ ] Create FLAT group
- [ ] Load as code segment
- [ ] Load resources
- [x] Rename DLL entries
- [ ] Manual load
- [ ] Create imports segment

OK | Cancel | Help

Output window

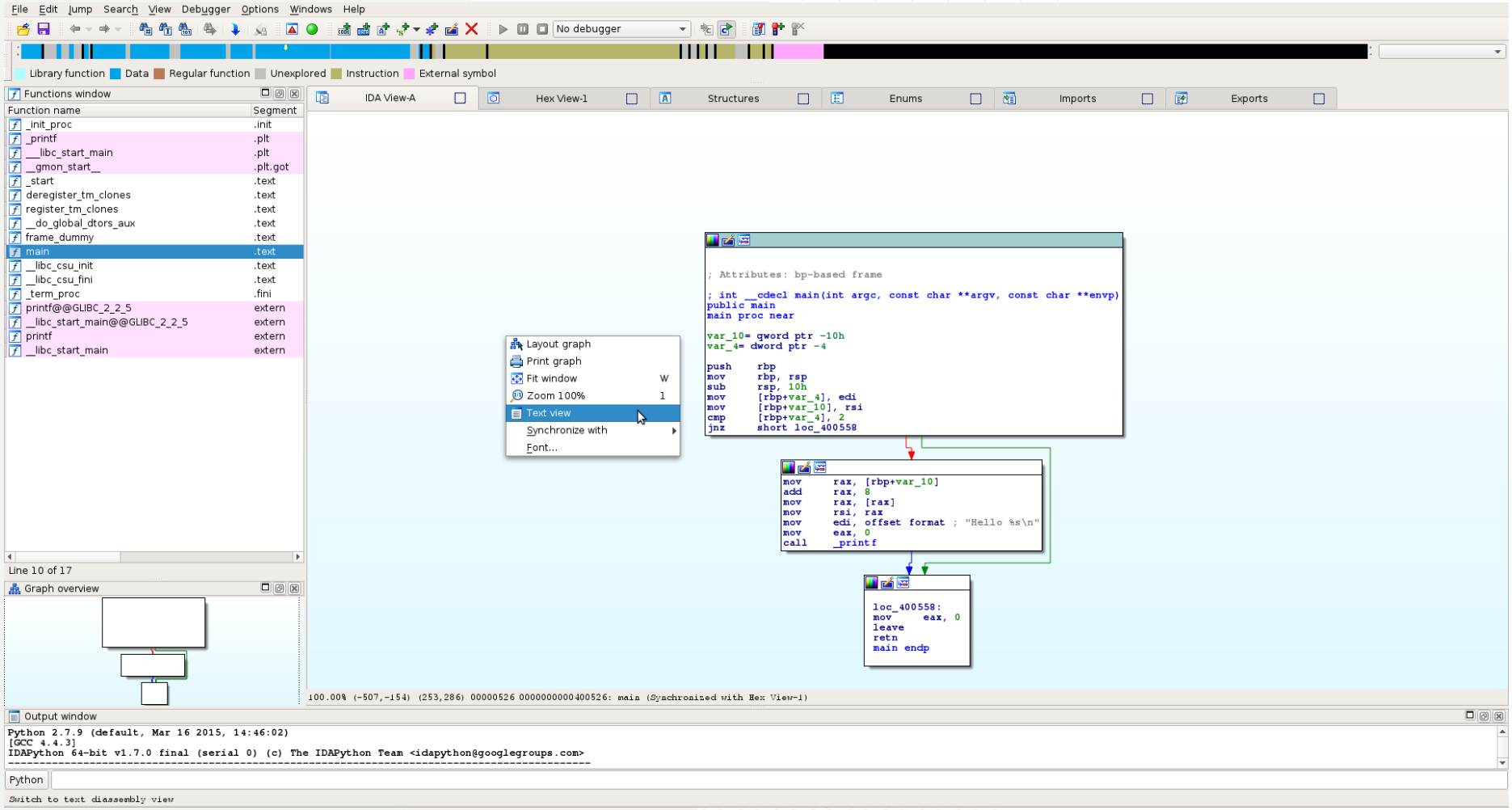
```
  bytes   pages size description
--------- ----- ---- --------------------------------------------
   262144    32 8192 allocating memory for b-tree...
    65536     8 8192 allocating memory for virtual array...
   262144    32 8192 allocating memory for name pointers...
----------------------------------------------------------------
   589824            total memory allocated

Loading processor module /tools/idapro/ida-695/procs/pc64.ilx64 for metapc...OK
Autoanalysis subsystem has been initialized.
Possible file format: ELF64 for x86-64 (Executable) (/tools/idapro/ida-695/loaders/elf64.llx64)
```

Python

@:0000000000000000 Down

IDA WILL OPEN IN CONTROL FLOW GRAPH VIEW

- Right-click and select Text View to view the flat disassembled code



COMMENTS: RIGHT-CLICK -> ENTER COMMENTS (OR PRESS ":")

Georgia Tech

File Edit Jump Search View Debugger Options Windows Help

No debugger

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Functions window

Function name	Segment
_init_proc	.init
_printf	.plt
__libc_start_main	.plt
__gmon_start__	.plt.got
_start	.text
deregister_tm_clones	.text
register_tm_clones	.text
__do_global_dtors_aux	.text
frame_dummy	.text
main	.text
__libc_csu_init	.text
__libc_csu_fini	.text
_term_proc	.fini
printf@@GLIBC_2_2_5	extern
__libc_start_main@@GLIBC_2_2_5	extern
printf	extern
__libc_start_main	extern

Line 10 of 17

IDA View-A | Hex View-1 | Structures | Enums | Imports | Exports

```
.text:0000000000400510                 mov     eax, 0
.text:0000000000400515                 test    rax, rax
.text:0000000000400518                 jz      short loc_40050B
.text:000000000040051A                 push    rbp
.text:000000000040051B                 mov     rbp, rsp
.text:000000000040051E                 call    rax
.text:0000000000400520                 pop     rbp
.text:0000000000400521                 jmp     register_tm_clones
.text:0000000000400521 frame_dummy     endp
.text:0000000000400521
.text:0000000000400526
.text:0000000000400526 ; =============== S U B R O U T I N E =======================================
.text:0000000000400526
.text:0000000000400526 ; Attributes: bp-based frame
.text:0000000000400526
.text:0000000000400526 ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:0000000000400526                 public main
.text:0000000000400526 main            proc near               ; DATA XREF: _start+1D↑o
.text:0000000000400526
.text:0000000000400526 var_10          = qword ptr -10h
.text:0000000000400526 var_4           = dword ptr -4
.text:0000000000400526
.text:0000000000400526                 push    rbp             ; save the previous stack frame base pointer
.text:0000000000400527                 mov     rbp, rsp
.text:000000000040052A                 sub     rsp, 10h
.text:000000000040052E                 mov     [rbp+var_4], edi
.text:0000000000400531                 mov     [rbp+var_10], rsi
.text:0000000000400535                 cmp     [rbp+var_4], 2
.text:0000000000400539                 jnz     short loc_400558
.text:000000000040053B                 mov     rax, [rbp+var_10]
.text:000000000040053F                 add     rax, 8
.text:0000000000400543                 mov     rax, [rax]
.text:0000000000400546                 mov     rsi, rax
.text:0000000000400549                 mov     edi, offset format
.text:000000000040054E                 mov     eax, 0
.text:0000000000400553                 call    _printf
.text:0000000000400558
.text:0000000000400558 loc_400558:
.text:0000000000400558                 mov     eax, 0
.text:000000000040055D                 leave
.text:000000000040055E                 retn
.text:000000000040055E main            endp
.text:000000000040055E
.text:000000000040055E ; ---------------------------------------------------------------------------
.text:000000000040055F                 align 20h
.text:0000000000400560
.text:0000000000400560 ; =============== S U B R O U T I N E =======================================
.text:0000000000400560
.text:0000000000400560
.text:0000000000400560 ; void _libc_csu_init(void)
.text:0000000000400560                 public __libc_csu_init
.text:0000000000400560 __libc_csu_init proc near               ; DATA XREF: _start+16↑o
.text:0000000000400560                 push    r15
.text:0000000000400562                 push    r14
.text:0000000000400564                 mov     r15d, edi
.text:0000000000400567                 push    r13
```

List cross references to...	Ctrl+X
Enter comment...	:
Enter repeatable comment...	;
Edit function...	Alt+P
Hide	Ctrl+Numpad+-
Graph view	
Proximity browser	Numpad+-
Undefine	U
Synchronize with	
Add breakpoint	F2
Xrefs graph to...	
Xrefs graph from...	
Font...	

Enter a regular comment

00000527 0000000000400527: main+1 (Synchronized with Hex View-1)

Output window

```
Python 2.7.9 (default, Mar 16 2015, 14:46:02)
[GCC 4.4.3]
IDAPython 64-bit v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
-------------------------------------------------------------------------------
```

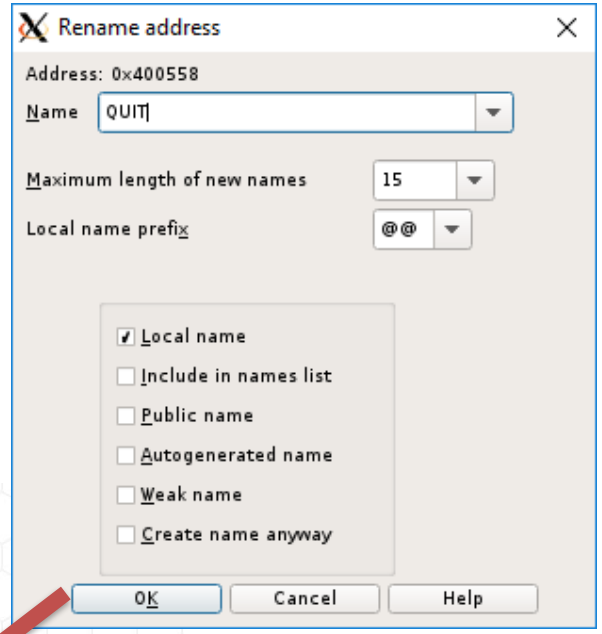
Python

Enter a regular comment

CREATING THE NEXT®

PRO TIP: RENAME LABELS AS YOU GO!

```
text:0000000000400526                 push    rbp
text:0000000000400527                 mov     rbp, rsp
text:000000000040052A                 sub     rsp, 10h
text:000000000040052E                 mov     [rbp+var_4], edi
text:0000000000400531                 mov     [rbp+var_10], rsi
text:0000000000400535                 cmp     [rbp+var_4], 2
text:0000000000400539                 jnz     short loc_400558
text:000000000040053B                 mov     rax, [rbp+var_10]
text:000000000040053F                 add     rax, 8
text:0000000000400543                 mov     rax, [rax]
text:0000000000400546                 mov     rsi, rax
text:0000000000400549                 mov     edi, offset format ; "Hello %s\n"
text:000000000040054E                 mov     eax, 0
text:0000000000400553                 call    _printf
text:0000000000400558
text:0000000000400558 loc_400558:                             ; CODE XREF: main+13↑j
text:0000000000400558                 mov     eax, 0
text:000000000040055D                 leave
text:000000000040055E                 retn
text:000000000040055E main            endp
text:000000000040055E
```



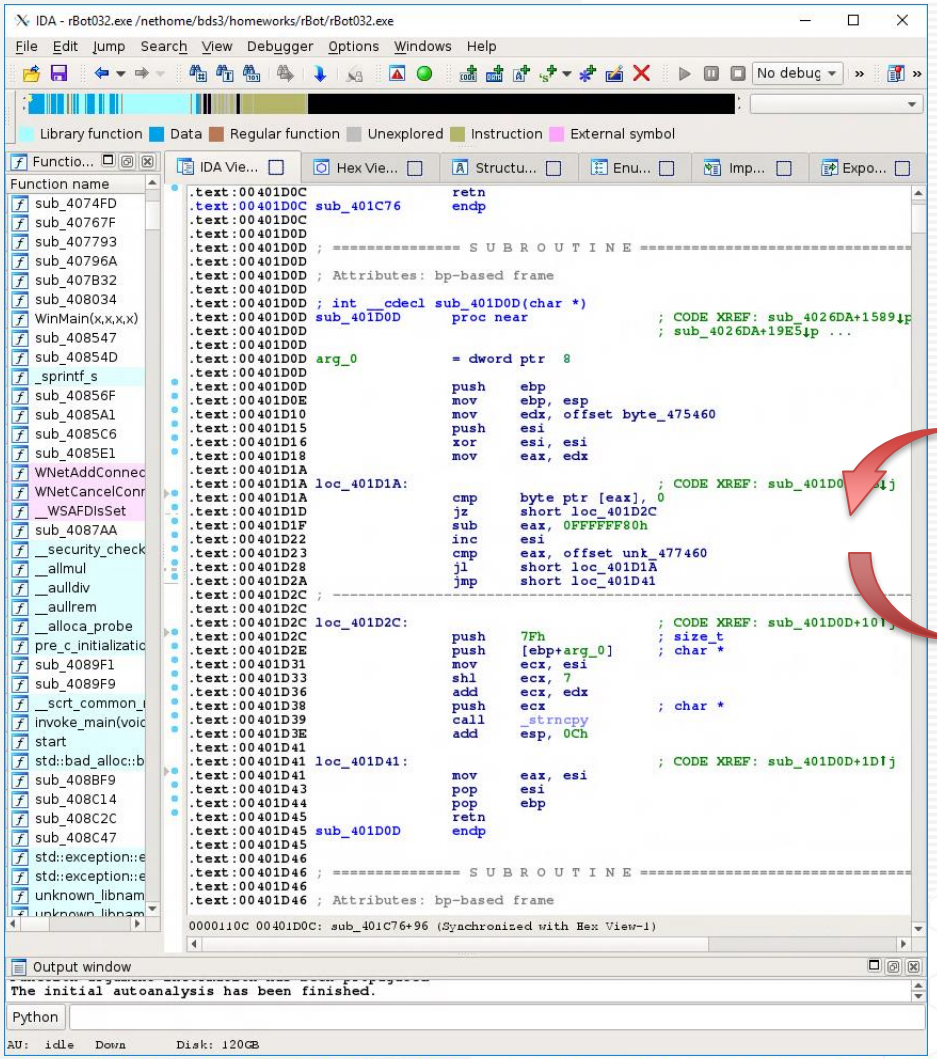
1. Click on the element to rename
2. Press the “n” key
3. Enter name and settings (if any)
4. Enjoy easier to read assembly!

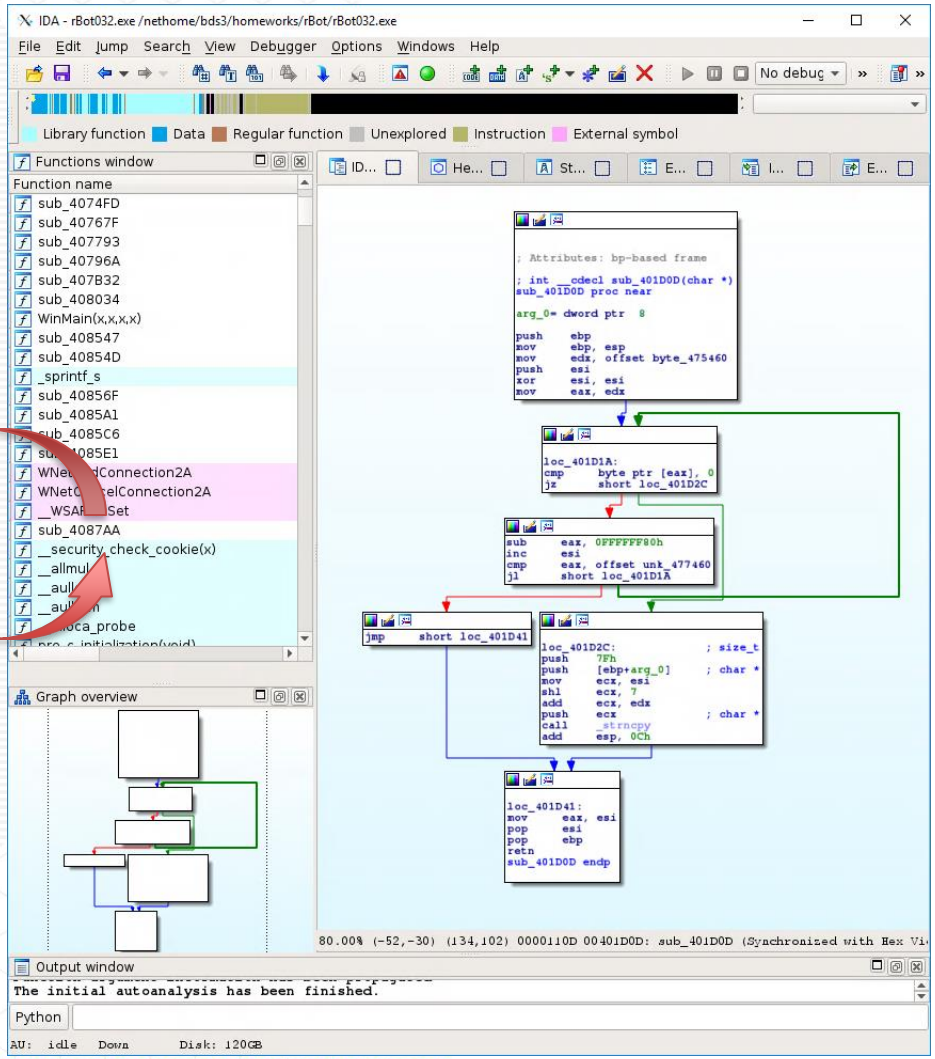
```
text:000000000040052A                 sub     rsp, 10h
text:000000000040052E                 mov     [rbp+var_4], edi
text:0000000000400531                 mov     [rbp+var_10], rsi
text:0000000000400535                 cmp     [rbp+var_4], 2
text:0000000000400539                 jnz     short QUIT
text:000000000040053B                 mov     rax, [rbp+var_10]
text:000000000040053F                 add     rax, 8
text:0000000000400543                 mov     rax, [rax]
text:0000000000400546                 mov     rsi, rax
text:0000000000400549                 mov     edi, offset format ; "Hello %s\n"
text:000000000040054E                 mov     eax, 0
text:0000000000400553                 call    _printf
text:0000000000400558
text:0000000000400558 QUIT:                                   ; CODE XREF: main+13↑j
text:0000000000400558                 mov     eax, 0
text:000000000040055D                 leave
text:000000000040055E                 retn
text:000000000040055E main            endp
text:000000000040055E
```

CREATING THE NEXT®

PRO TIP #2: SWITCH BETWEEN VIEWS!

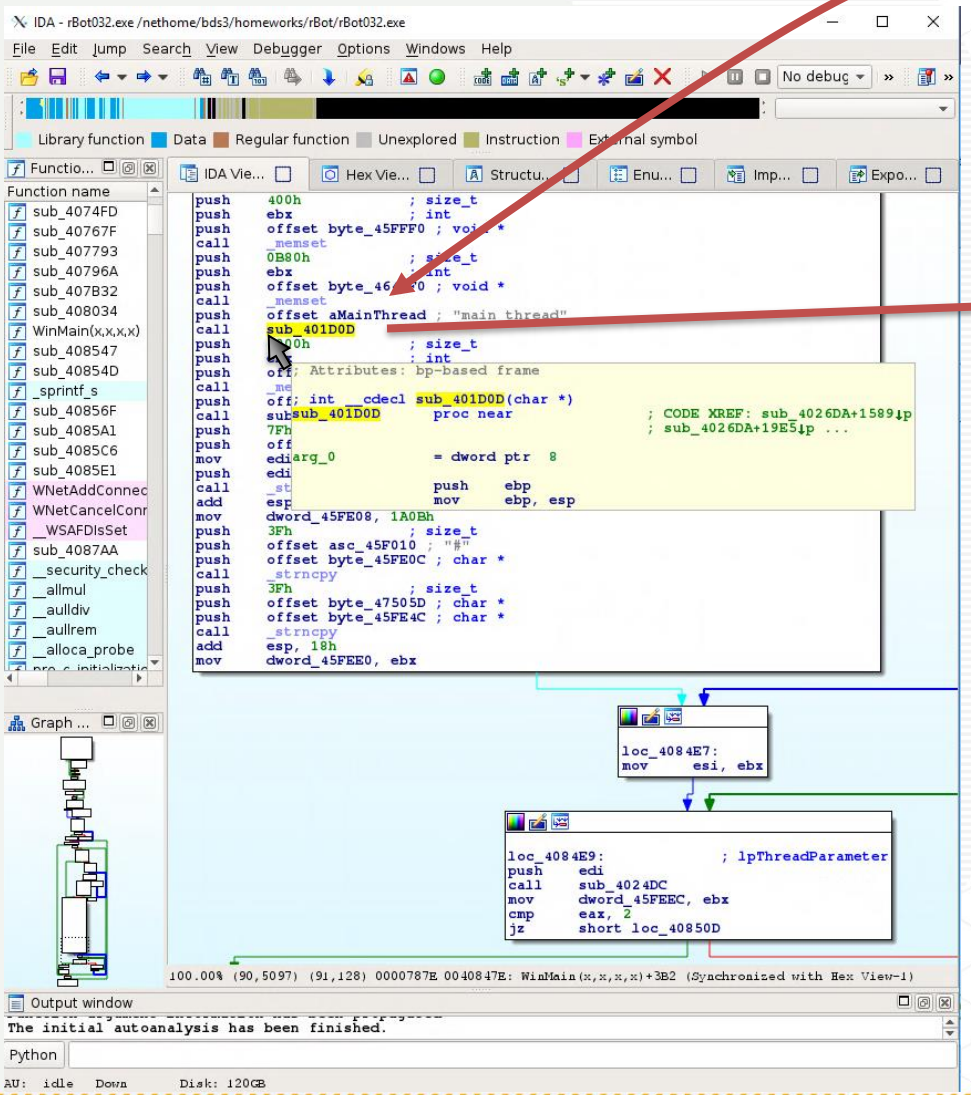
- Text View may be easier to read, but Graph View gives better context
- Comments still show in both views ☺

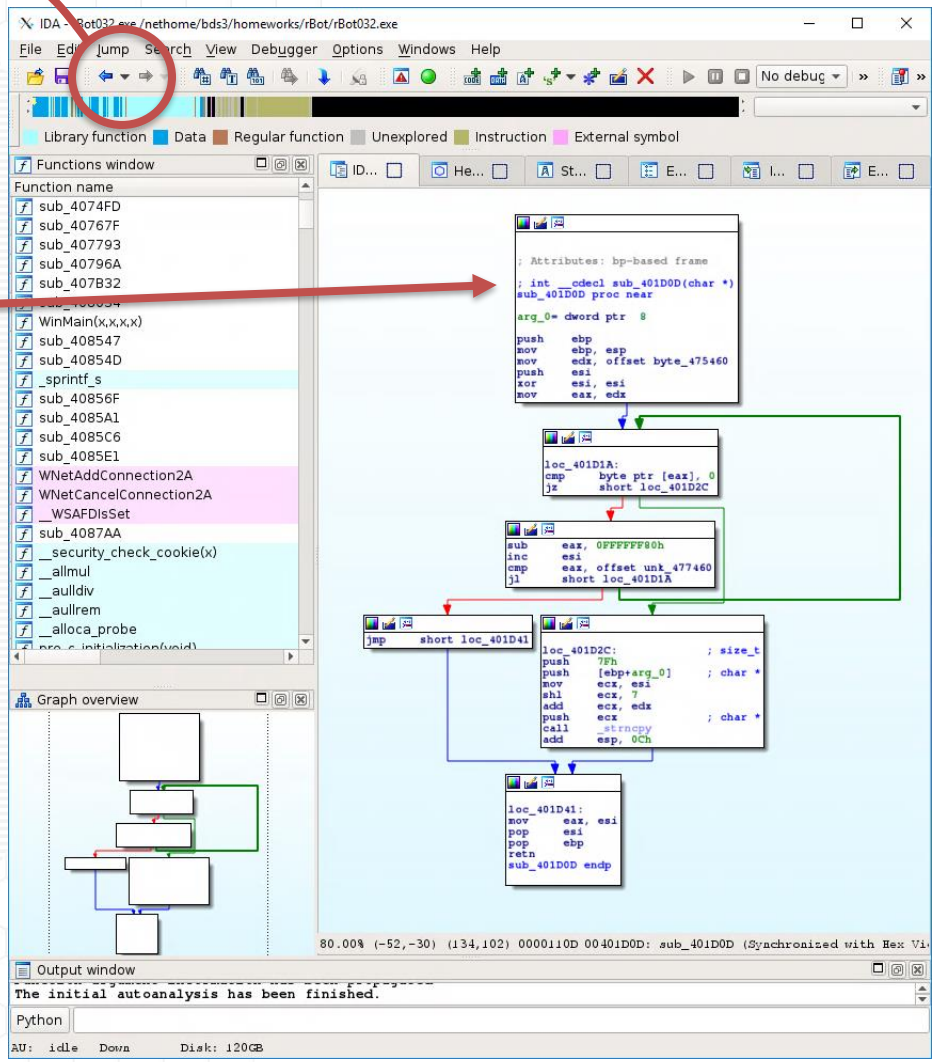




PRO TIP #3: NAVIGATION BUTTONS!

- Double-click on a label to jump to it. Want to go back? IDA remembers!





PRO TIP #4: RENAME SYMBOLIC CONSTANTS

- IDA's FLIRT signatures know the arguments for common APIs
- But IDA also knows the symbolic names for most defined constants!
- You just have to tell IDA what value you are looking for

```
LONG WINAPI RegCreateKeyEx(
  _In_        HKEY                    hKey,
  _In_        LPCTSTR                 lpSubKey,
  _Reserved_  DWORD                   Reserved,
  _In_opt_    LPTSTR                  lpClass,
  _In_        DWORD                   dwOptions,
  _In_        REGSAM                  samDesired,
  _In_opt_    LPSECURITY_ATTRIBUTES   lpSecurityAttributes,
  _Out_       PHKEY                   phkResult,
  _Out_opt_   LPDWORD                 lpdwDisposition
)
```

```
lea     eax, [ebp+phkResult]
push    ecx                 ; lpdwDisposition
push    eax                 ; phkResult
push    ecx                 ; lpSecurityAttributes
push    0F003Fh             ; samDesired
push    ecx                 ; dwOptions
push    ecx                 ; lpClass
push    ecx                 ; Reserved
push    offset aSoftwareMicr_1 ; "Software\\Microsoft\\Windows\\CurrentVe"...
push    80000002h           ; hKey
call    ebx ; RegCreateKeyExA
```

PRO TIP #4: RENAME SYMBOLIC CONSTANTS (2)

- Right Click -> Use standard symbolic constant
- Then simply find the constant name you are looking for

```
push    80000002h       ; hKey
call    ebx ; Reg
push    0Ch
lea     eax, [ebp
push    eax
push    1
push    0
push    offset aC
push    [ebp+phkR
call    edi ; Reg
push    [ebp+phkR
call    esi ; Reg
xor     ecx, ecx
lea     eax, [ebp
push    ecx
push    eax
push    ecx
push    0F003Fh
push    ecx
push    ecx
push    ecx
push    offset aS
push    80000001h
call    ebx ; Reg
push    0Ch
lea     eax, [ebp
xor     ebx, ebx
push    eax
```

- Jump to operand
- Jump in a new window
- Jump in a new hex window
- Use standard symbolic constant
- 2147483650
- 20000000002o
- 1000000000000000000000000000010b
- Manual...
- Edit function...
- Hide
- Graph view
- Proximity browser
- Undefine
- Synchronize with
- Add breakpoint
- Xrefs graph to...

Please choose a symbol

Type name	Declaration	Type library
CHANGER_PREDISMOUNT_ALIGN_TO_DRIVE	80000002	MS SDK (Windows XP)
DSOP_DOWNLEVEL_FILTER_LOCAL_GROUPS	80000002	MS SDK (Windows XP)
DeviceDsmAction_Notification	FFFFFFFF80000002	MS SDK (Windows XP)
EXCEPTION_DATATYPE_MISALIGNMENT	80000002	MS SDK (Windows XP)
HKEY_LOCAL_MACHINE	80000002	MS SDK (Windows XP)
MQCONN_CREATE_SOCKET_FAILURE	80000002	MS SDK (Windows XP)
PERF_QUERY_COSTLY	FFFFFFFF80000002	MS SDK (Windows XP)
PID_SECURITY	00000013	MS SDK (Windows XP)
STATUS_DATATYPE_MISALIGNMENT	80000002	MS SDK (Windows XP)
TAPE_DRIVE_TENSION	FFFFFFFF80000002	MS SDK (Windows XP)
tomItalic	80000002	MS SDK (Windows XP)
EXCEPTION_DATATYPE_MISALIGNMENT	80000002	Visual C++ v6
HKEY_LOCAL_MACHINE	80000002	Visual C++ v6
HKEY_PERF_ROOT	FFFFFFFF80000002	Visual C++ v6
LINEERR_BADDEVICEID	80000002	Visual C++ v6
PERF_QUERY_COSTLY	FFFFFFFF80000002	Visual C++ v6
PID_SECURITY	FFFFFFFF80000002	Visual C++ v6
STATUS_DATATYPE_MISALIGNMENT	80000002	Visual C++ v6
TAPE_DRIVE_TENSION	FFFFFFFF80000002	Visual C++ v6

OK | Cancel | Search | Help

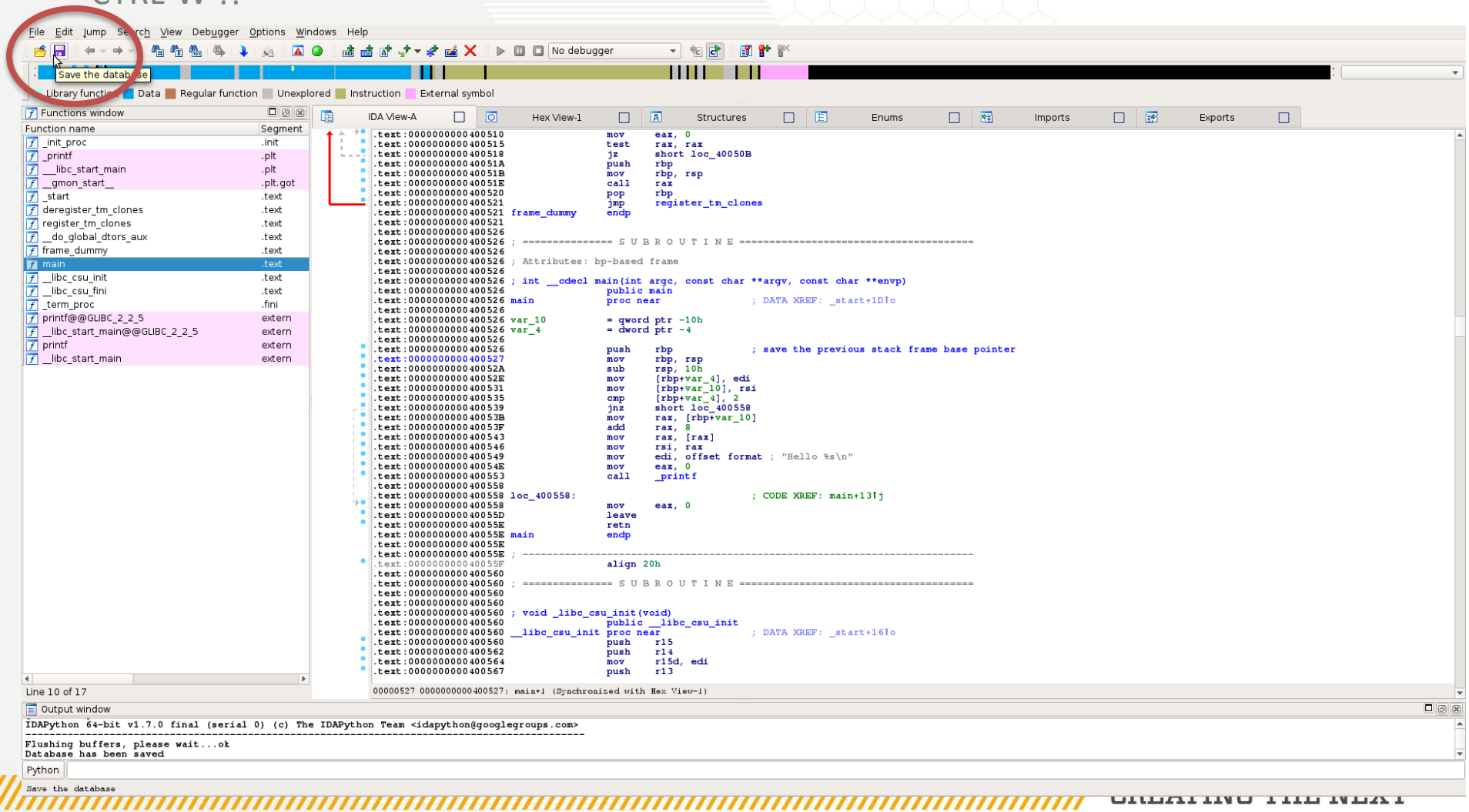
Line 5 of 19

```
push    offset aSoftwareMicr_1 ; "S
push    HKEY_LOCAL_MACHINE ; hKey
call    ebx ; RegCreateKeyExA
push    0Ch                ; cbData
```

- If you can't find the symbol, you may need to add it
- First, look up the symbol's header file definition
- Second, read about adding new Enums and symbolic constants here: https://www.hex-rays.com/products/ida/support/idadoc/499.shtml

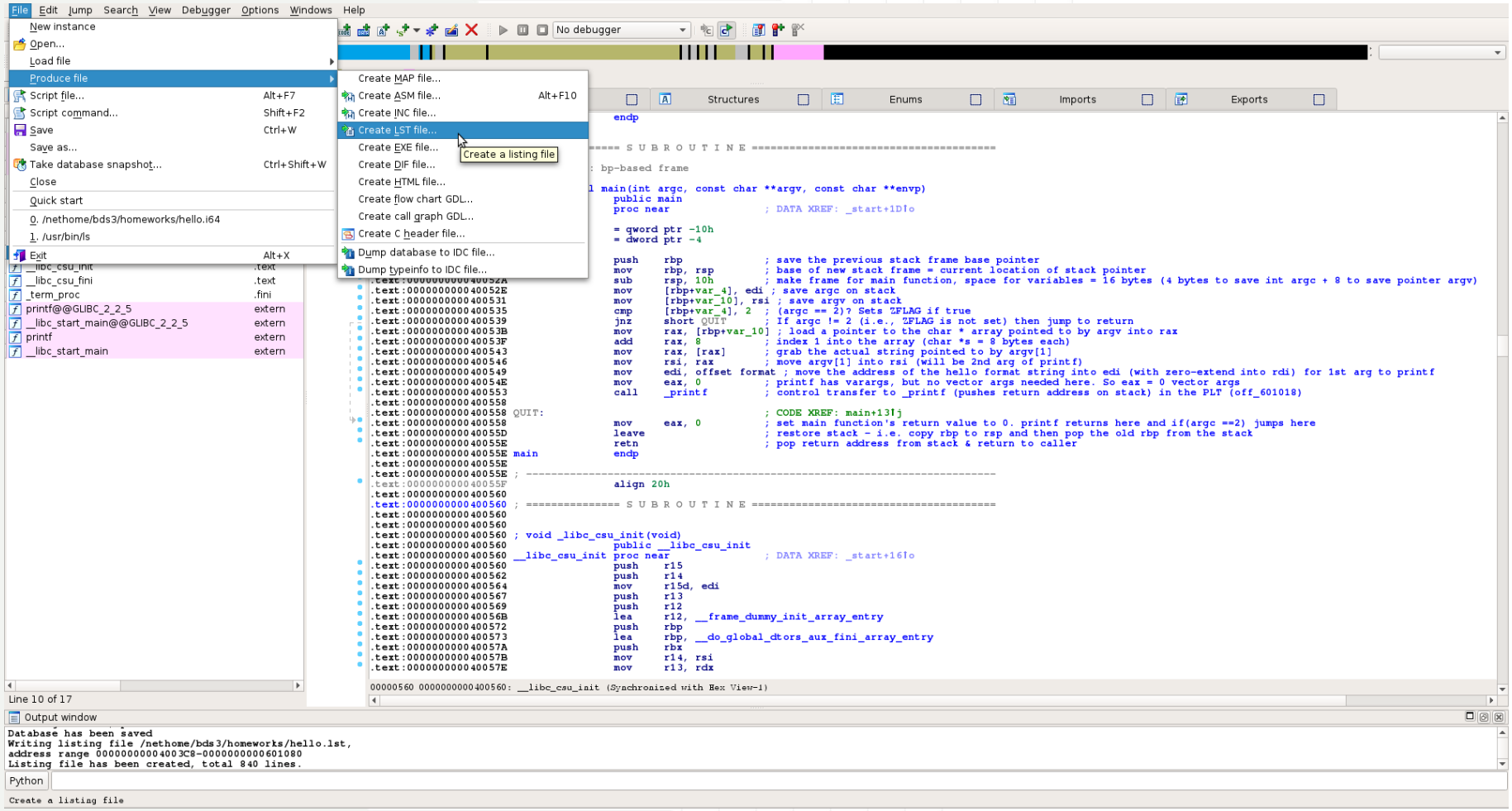
SAVE OFTEN!

- Losing hours of reverse engineering can be hazardous to your health!
- CTRL-W !!



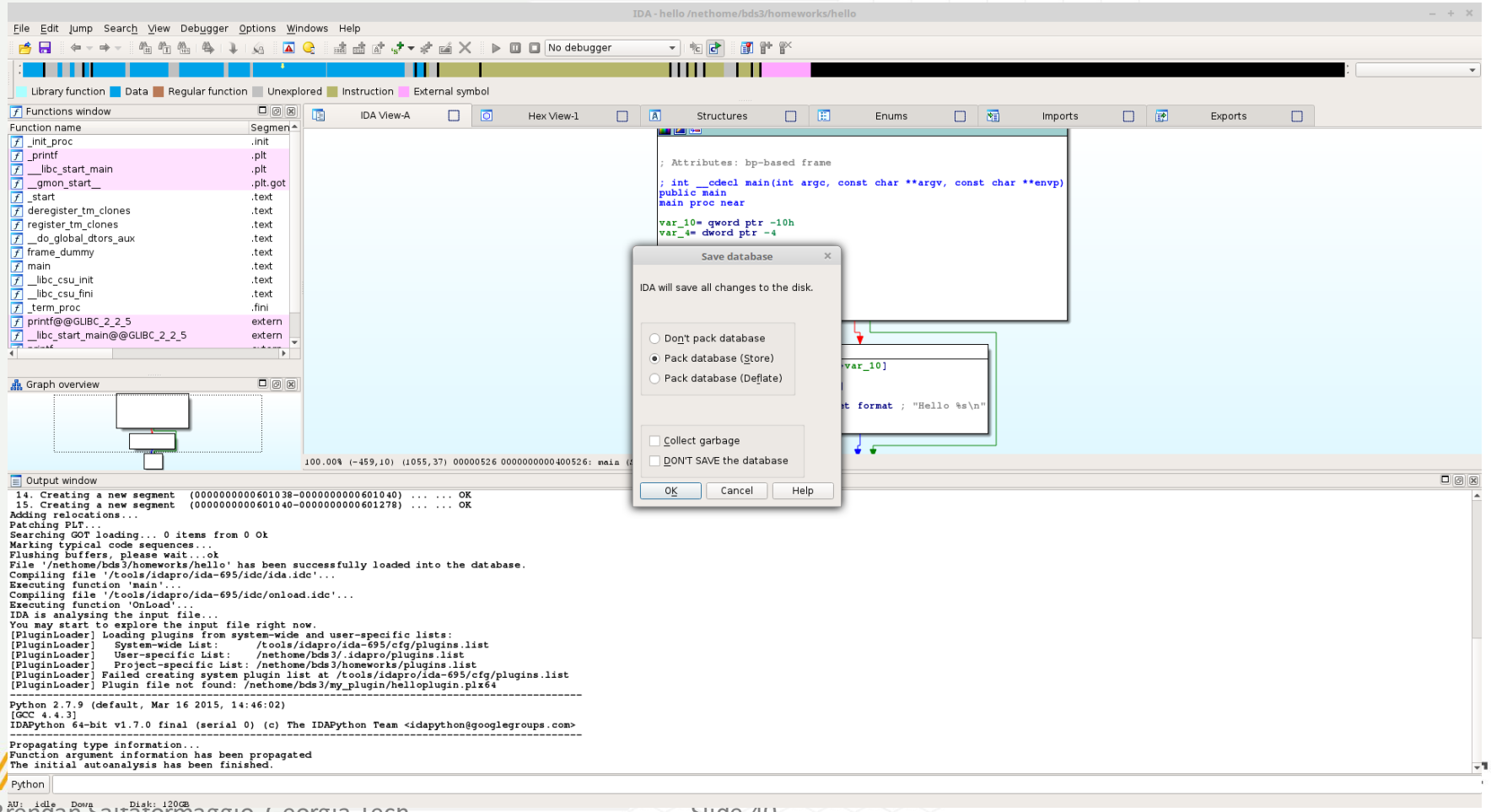
EXPORT YOUR ASSEMBLY LISTING

- IDA can export all the Text View content to a “Listing” file
  - You will often turn these in for reverse engineering assignments



READY TO CLOSE IDA? SAVE YOUR DATABASE (OR NOT)

- IDA will compress all of its data into a single database to save your progress
  - The save file is called an idb file for 32-bit and i64 for 64-bit
- You can also tell IDA to not save a database (i.e., you lose everything you've done)



AS YOU WORK, SO DOES IDA

- IDA performs very substantial analysis for you and saves the results in a number of files while you work
- If you kill IDA (or lose connection to the IDA servers), these files will be corrupted!
- So save often!! Saving will create/update the database file

```
File  Edit  View  Search  Terminal  Help
bds3@ecelinsrvw.ece.gatech.edu>cd homeworks/
bds3@ecelinsrvw.ece.gatech.edu>ls
hello  hello.id0  hello.id1  hello.id2  hello.nam  hello.til
bds3@ecelinsrvw.ece.gatech.edu>
[1]    Done                  idaq64
bds3@ecelinsrvw.ece.gatech.edu>ls
hello  hello.i64
bds3@ecelinsrvw.ece.gatech.edu>
```

ADDITIONAL READINGS (OPTIONAL)

- Chris Eagle. The IDA Pro Book. No Starch Press (2nd Edition), 2011. ISBN: 978-1593272890
  - You can probably find the PDF version online!

- Guided Hacking's "How To Reverse Engineer" Videos
  - https://www.youtube.com/playlist?list=PLt9cUwGw6CYFXtAElzDLob2aOaSfZqHJc

QUESTIONS?

CREATING THE NEXT®