THE UNIVERSITY OF ARIZONA
UA South

CYBV 471 Assembly Programming for Security Professionals
Week 1

Introduction and Setup tools and Lab Environment

Agenda

- Introductions
- Course Overview
- Project Overview
- Introduction to the UA Cyber Virtual Learning Environment (VLE)
- Access Virtual Lab and Linux VM Demo
- Create first assembly program
- High Level Languages
- What is the assembly language?
- Why should we learn Assembly Language?
- Week#1 Lab Assignment

Welcome and Introduction

- Name: Mohamed Meky
- Ph.D. (1998), M.Sc. (1990), B.Sc. (1987) all in EE
- Cyber Security Certificate (University of Maryland University College, 2011)
- More than 20 years of teaching experience (several programs)

  University of Arizona South (CYBV 496, CYBV 471, developed CYBV 454)

  Southern Methodist University (SMU), Dallas

  Rigs University, Colorado

  University of Colorado, Colorado

  University of Maryland University College (UMUC), Maryland

- Telecommunications, programming, networking, OS, etc.
- Last 12 years: Research/Teach/develop Cyber security courses
- More than 22 years industrial experience (AT&T and Verizon)

Course Overview

- Define and explain binary, hexadecimal, integers and floating-point numbers
- Define and explain memory, memory mapping and the functions and uses of registers
- Identify and describe assembly programming math and bit operations
- Recognize the fundamentals behind branching & looping as well as functions
- Identify and evaluate arrays
- Define and explain C stream I/O
- Identify and describe data structures
- Be able to develop programs that can be embedded into Linux OS kernel
- Be able to construct programs that interact with a system without the layers of abstraction that are provided by many high-level languages
- Review & Final Project

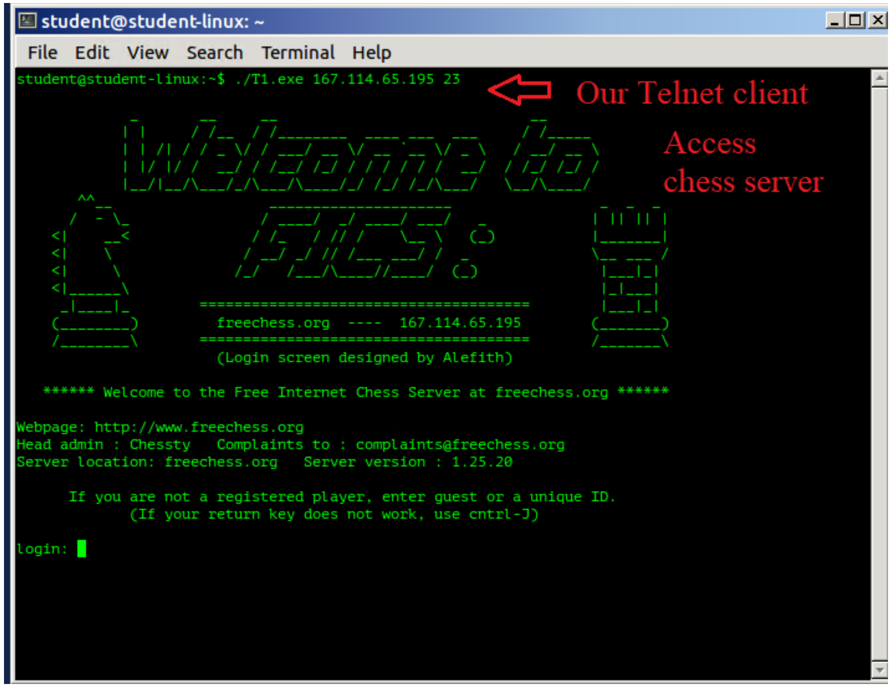
Project Overview

Telnet Client: 25%.

Students must write a “Telnet Client” stand-alone assembly program without using any help from external libraries

Test your Telnet Client using any of the three commands (assume the client name T1.exe)

./T1.exe 167.114.65.195   23



```
student@student-linux:~$ ./T1.exe 167.114.65.195 23

             _       __     __                             __
            | |     / /__  / /________  ____ ___  ___     / /_____
            | | /| / / _ \/ / ___/ __ \/ __ `__ \/ _ \   / __/ __ \
            | |/ |/ /  __/ / /__/ /_/ / / / / / /  __/  / /_/ /_/ /
            |__/|__/\___/_/\___/\____/_/ /_/ /_/\___/   \__/\____/
       ^^__                  ______________________                 _  _  _
      /  - \_               / ____/  _/ ____/ ___/   _             | || || |
    <|    __<              / /_   / // /    \__ \    (_)           |_______|
    <|    \               / __/ _/ // /___ ___/ /   _              \__ ____/
    <|     \             /_/   /___/\____//____/   (_)              |___|_|
    <|______\                                                       |_|___|
     _|____|_        ======================================         |___|_|
    (_________)            freechess.org  ----  167.114.65.195        (________)
    /_________\      ======================================       /________\
                       (Login screen designed by Alefith)

   ****** Welcome to the Free Internet Chess Server at freechess.org ******

Webpage: http://www.freechess.org
Head admin : Chessty   Complaints to : complaints@freechess.org
Server location: freechess.org   Server version : 1.25.20

      If you are not a registered player, enter guest or a unique ID.
             (If your return key does not work, use cntrl-J)

login: 
```

Our Telnet client

Access chess server

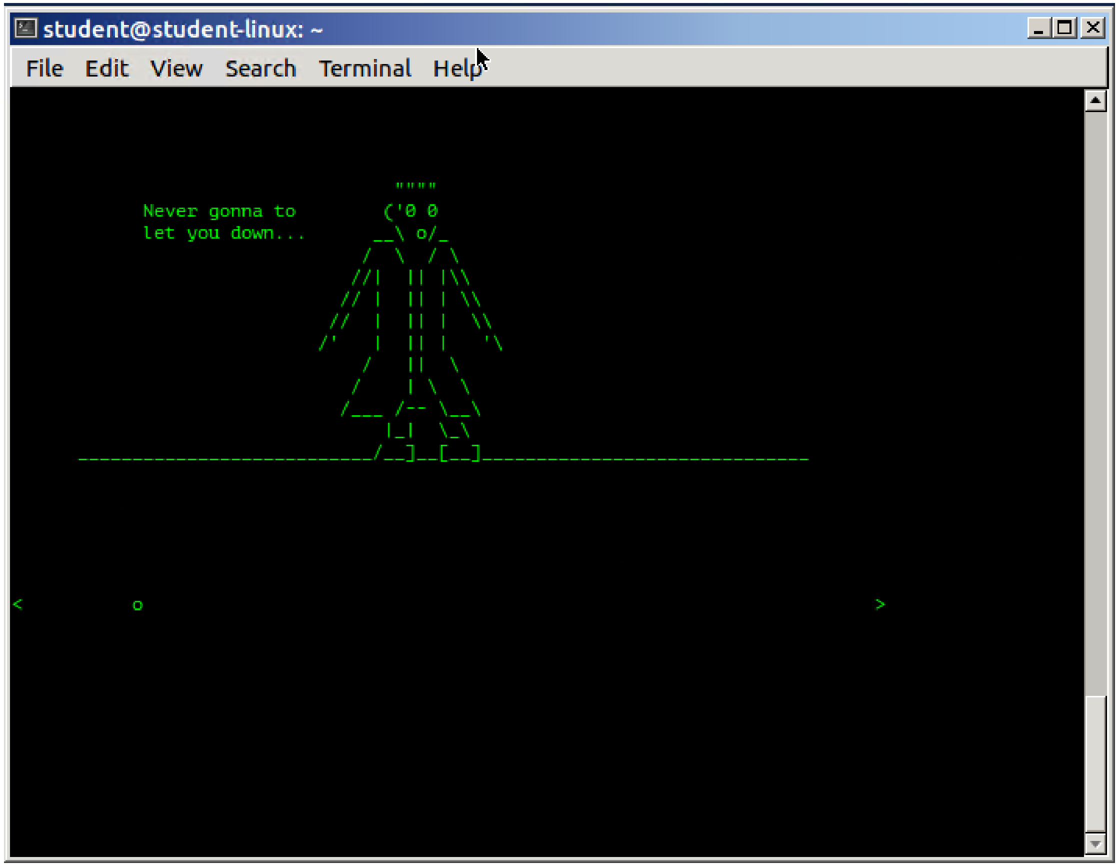
Project Overview

Test your Telnet Client using any of the three commands (assume the client name T1.exe)

./T1.exe 167.114.65.195 23

./T1.exe 10.139.201.23 24

./T1.exe 10.139.201.23 23



```
                                    """"
             Never gonna to        ('0 0
             let you down...      __\ o/_
                                 /  \  / \
                                //|  || |\\
                               // |  || | \\
                              //  |  || |  \\
                             /'   |  || |   '\
                                 /   ||  \
                                /    | \  \
                               /___ /-- \__\
                                   |_|  \_\
       ---------------------------/__]__[__]-------------------------------






<           o                                                                           >
```

Project Overview

Telnet Client: 25%.

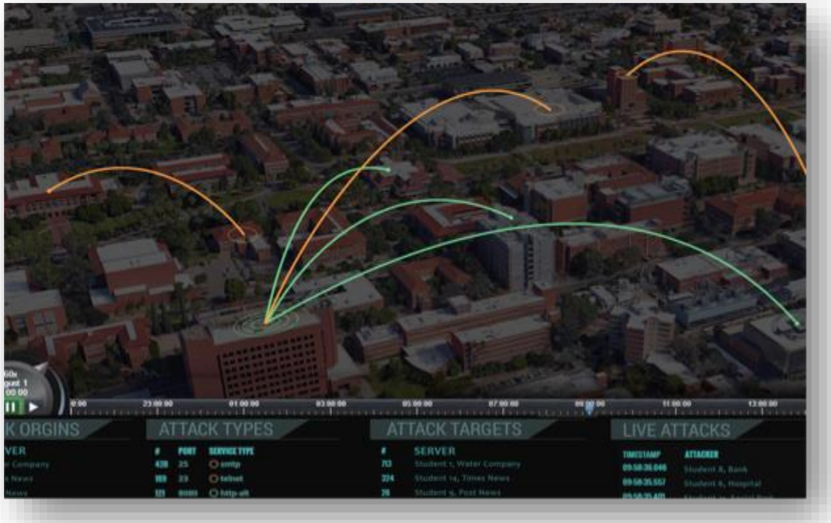
Students must write a “Telnet Client” stand-alone assembly program without using any help from external libraries

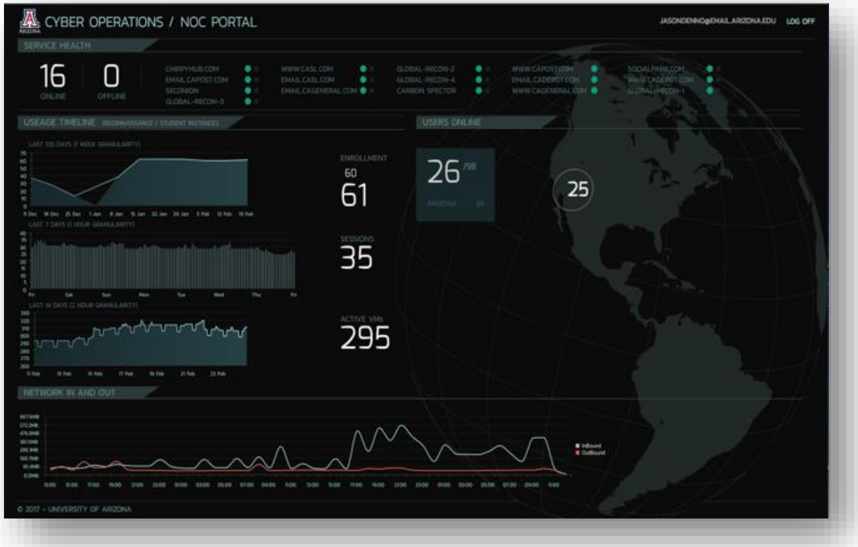
Test your Telnet Client

Introduction to the VLE

- **UA Virtual Learning Environment (VLE)**
  - **Student Portal**
  - **CyberApolis**
    - Websites
    - Network Architectures
    - Global vs. Local
    - Virtual Personas
    - Content and Metadata
    - Networking
  - **Capture the Flag Arena**
  - **Malware Sandbox**
  - **Forensics Lab**
  - **IoT Labs**
  - **Honeynet Lab**

- **Getting Started**
  - Register
  - Download Software
  - Accessing the VLE





Access your VM at VLE
Demo/Steps

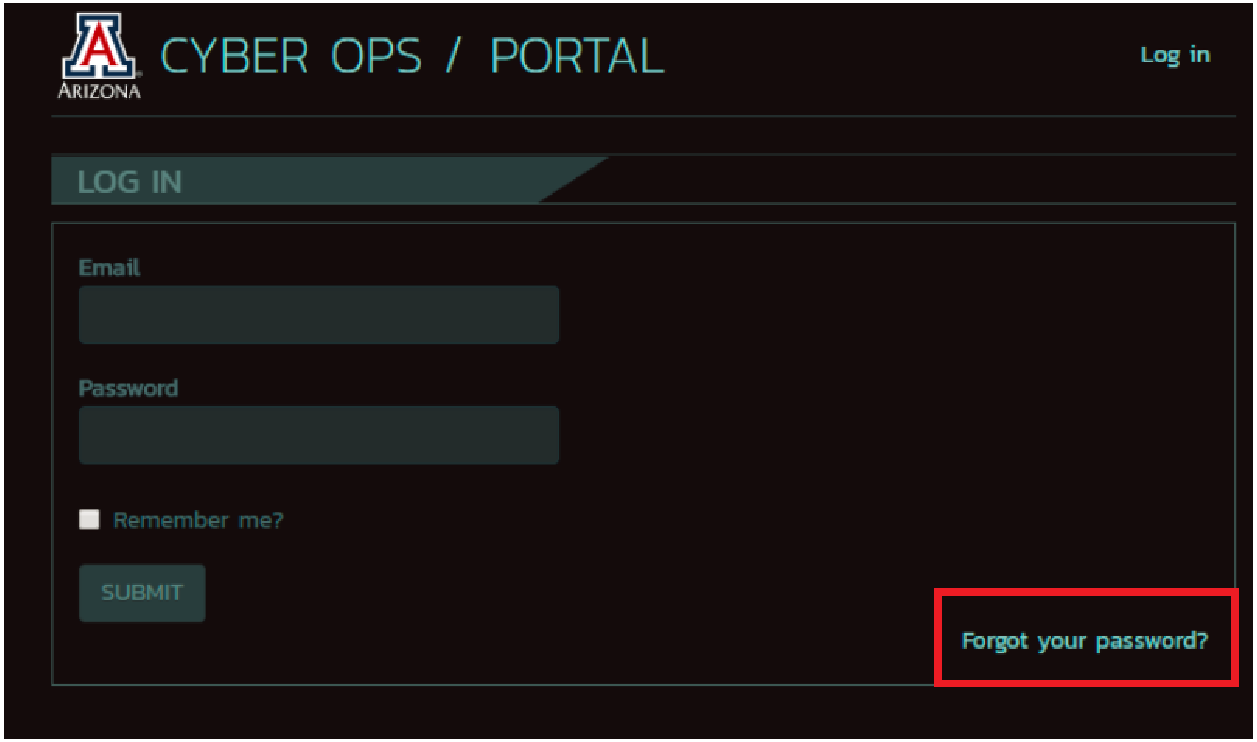
Access Virtual Lab/Machines

To access the virtual lab, please follow the following steps

1- Access the following link https://portal.cyberapolis.com/login

2- Click "Forgot your password.

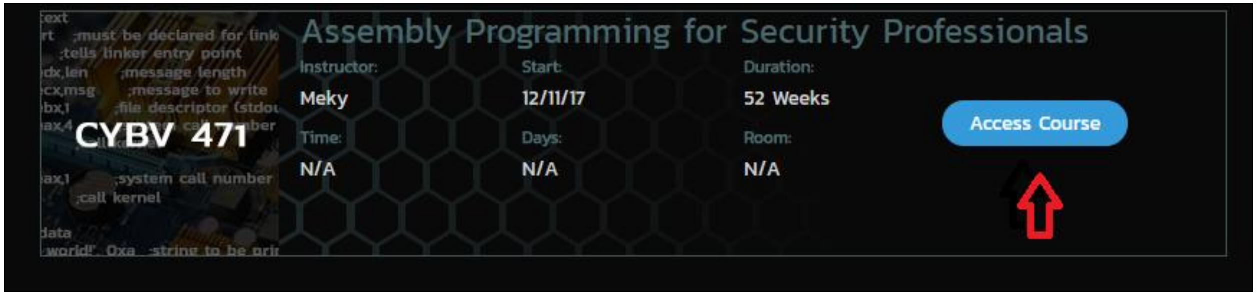
3- Provide your AZ account information and wait for an email with a password



Access Virtual Lab/Machines

4- After getting a password by email, access the following link https://portal.cyberapolis.com/login

5- Access your course



CYBV 471

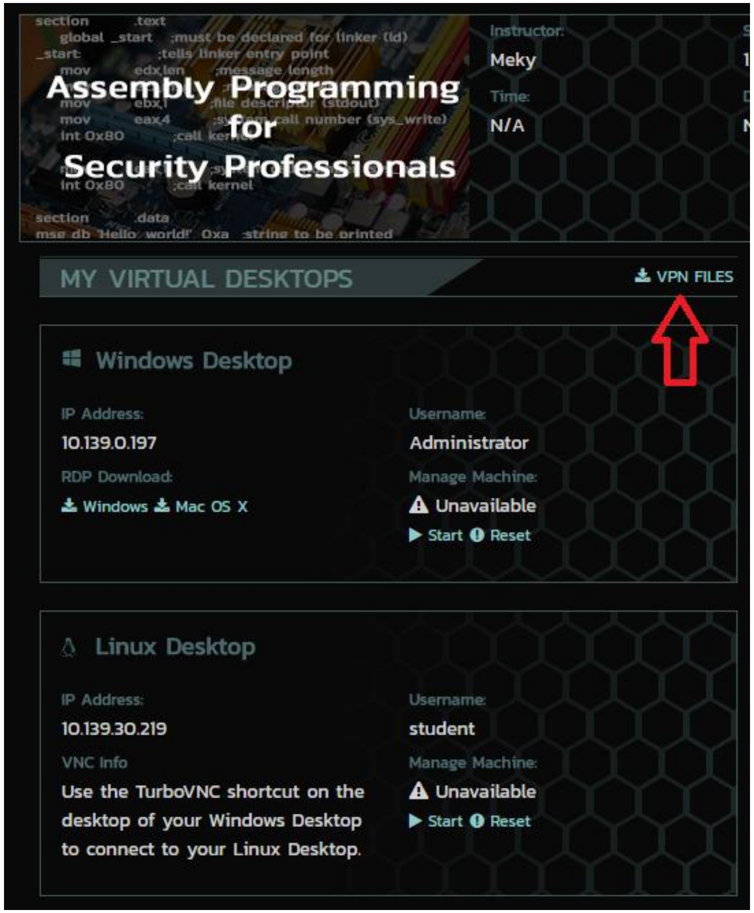
Assembly Programming for Security Professionals

Instructor:	Start:	Duration:
Meky	12/11/17	52 Weeks
Time:	Days:	Room:
N/A	N/A	N/A

Access Virtual Lab/Machines

7- Start your VMs

8- Download and install VPN files



Assembly Programming for Security Professionals

Instructor:
Meky

Time:
N/A

MY VIRTUAL DESKTOPS

VPN FILES

Windows Desktop

IP Address:
10.139.0.197

Username:
Administrator

RDP Download:
Windows Mac OS X

Manage Machine:
Unavailable
▶ Start Reset

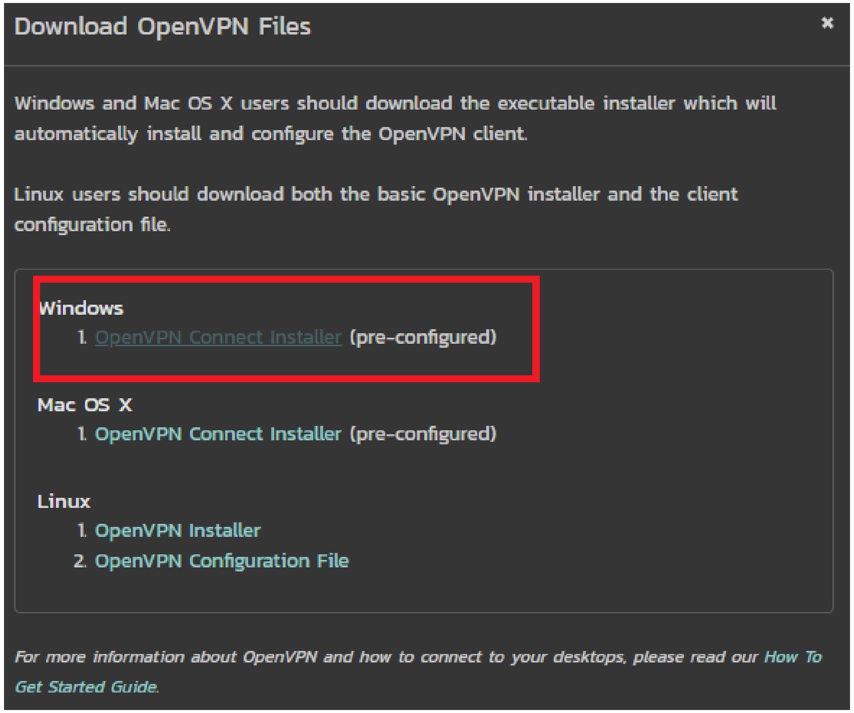
Linux Desktop

IP Address:
10.139.30.219

Username:
student

VNC Info
Use the TurboVNC shortcut on the desktop of your Windows Desktop to connect to your Linux Desktop.

Manage Machine:
Unavailable
▶ Start Reset



Download OpenVPN Files ×

Windows and Mac OS X users should download the executable installer which will automatically install and configure the OpenVPN client.

Linux users should download both the basic OpenVPN installer and the client configuration file.

Windows
1. OpenVPN Connect Installer (pre-configured)

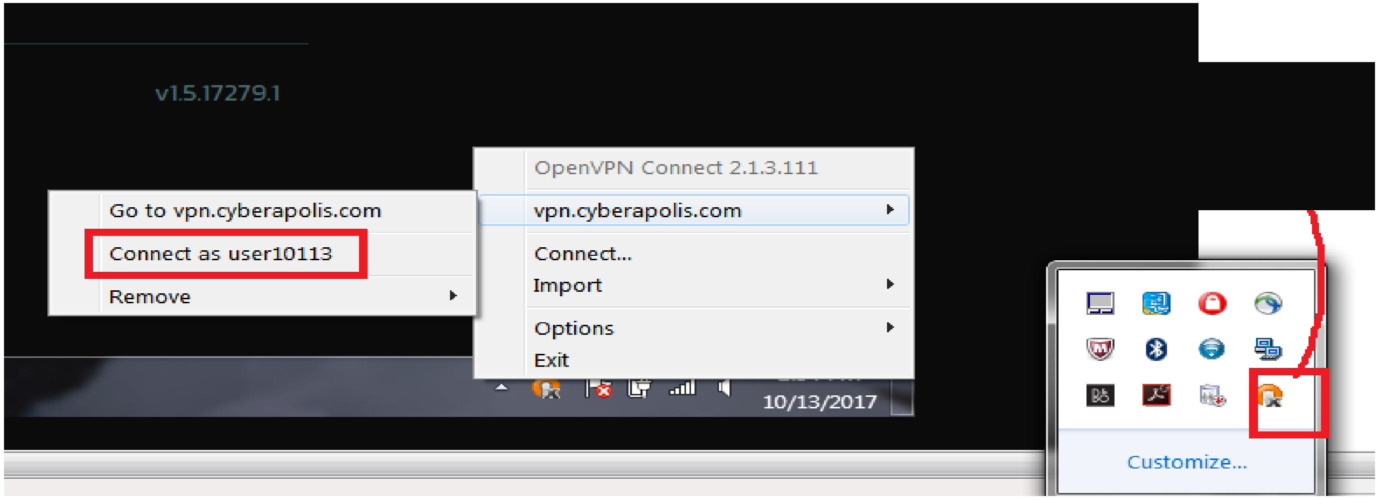
Mac OS X
1. OpenVPN Connect Installer (pre-configured)

Linux
1. OpenVPN Installer
2. OpenVPN Configuration File

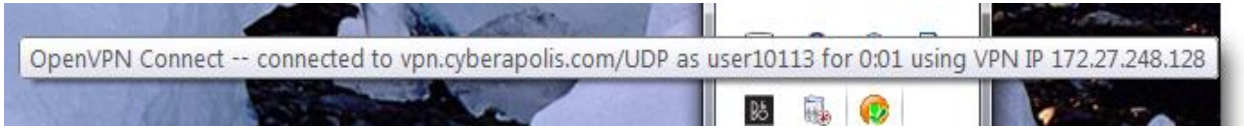
For more information about OpenVPN and how to connect to your desktops, please read our How To Get Started Guide.

Access Virtual Lab/Machines

8- After installing the VPN client, you should see the “OpenVPN” icon at the right bottom of your computer (at the start bar).

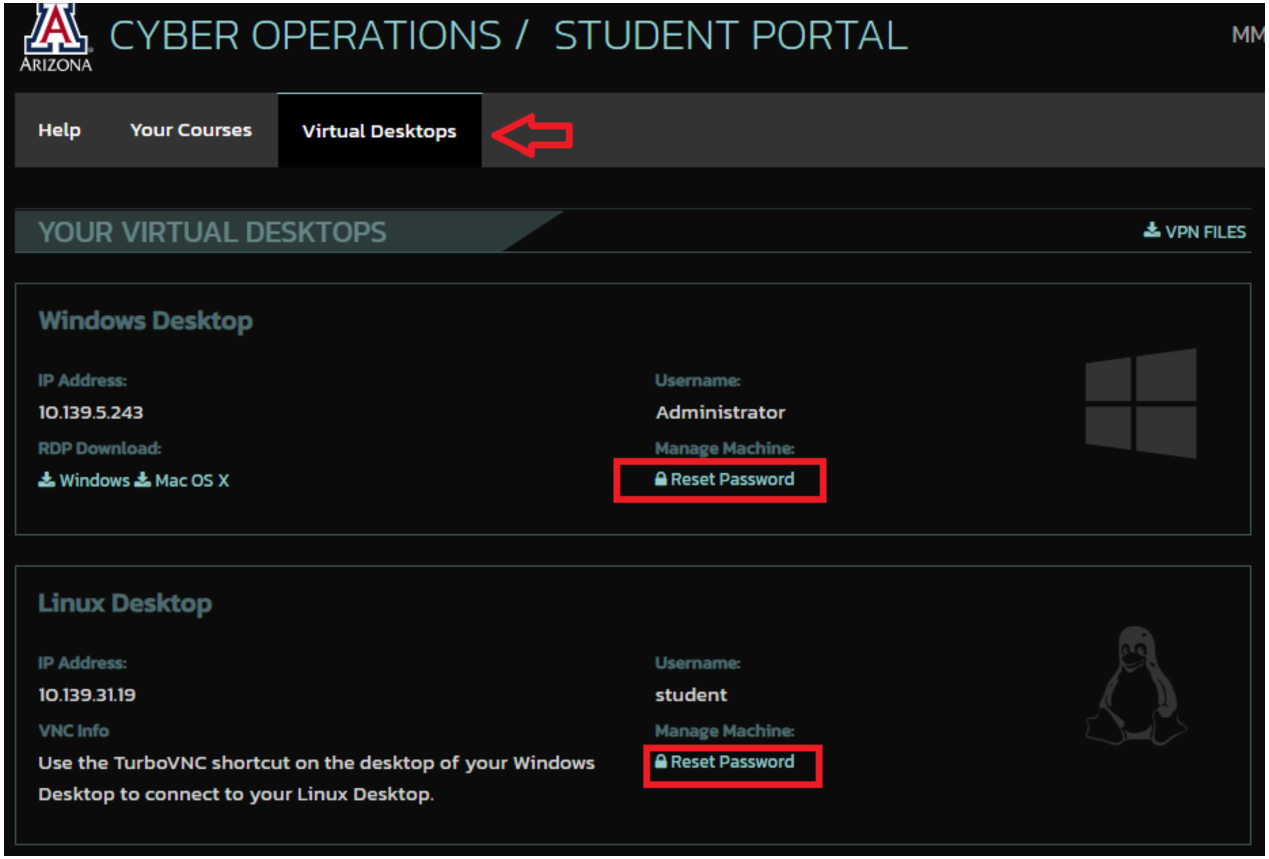


9- Connect the VPN. After connecting with OpenVPN, you can check by hovering on the “OpenVPN”icon


OpenVPN Connect -- connected to vpn.cyberapolis.com/UDP as user10113 for 0:01 using VPN IP 172.27.248.128

Access Virtual Lab/Machines

10- Start Windows Desktop VM, and choose “Rest Password” to get the password for Windows Desktop VM



YOUR VIRTUAL DESKTOPS

Windows Desktop

IP Address: 10.139.5.243

Username: Administrator

RDP Download:

Manage Machine:

Linux Desktop

IP Address: 10.139.31.19

Username: student

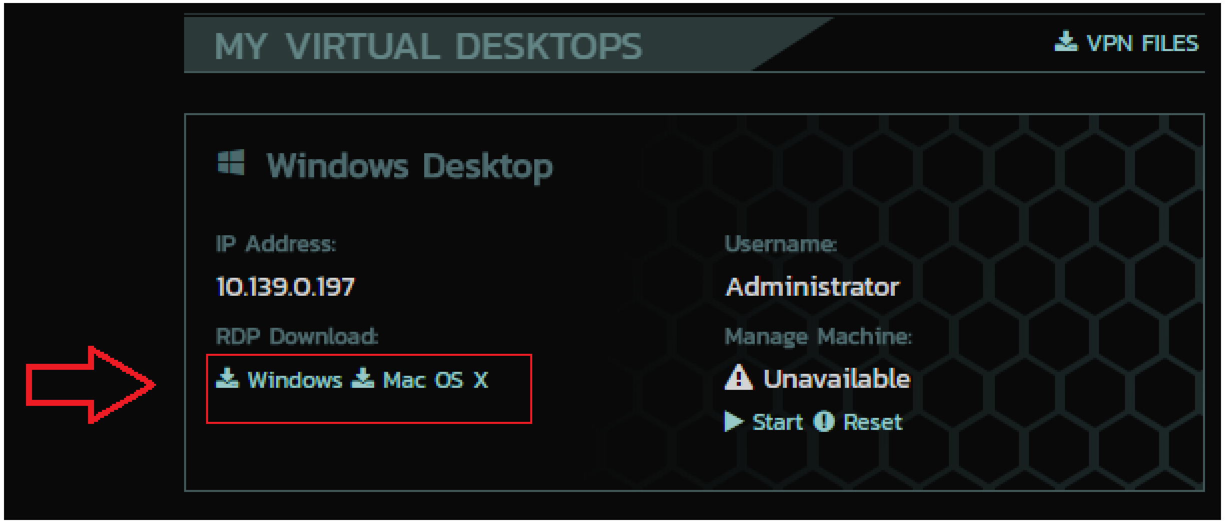
VNC Info

Use the TurboVNC shortcut on the desktop of your Windows Desktop to connect to your Linux Desktop.

Manage Machine:

Access Virtual Lab/Machines

11- Record the password. You need it to access the Desktop VM
12- Download the remote access desktop (RDP)



MY VIRTUAL DESKTOPS

VPN FILES

Windows Desktop

IP Address:
10.139.0.197

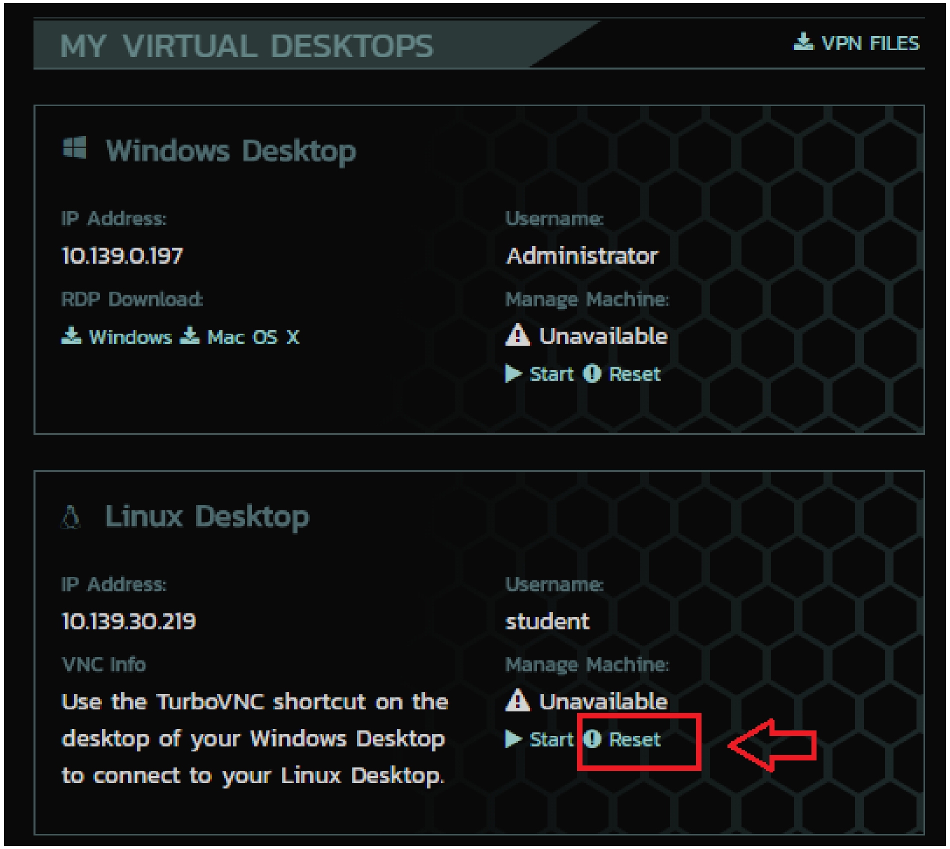
Username:
Administrator

RDP Download:
Windows Mac OS X

Manage Machine:
⚠ Unavailable
▶ Start Reset

Access Virtual Lab/Machines

13- Start Linux VM, and choose “Rest Password” to get the password for Linux VM



MY VIRTUAL DESKTOPS

VPN FILES

Windows Desktop

IP Address:
10.139.0.197

Username:
Administrator

RDP Download:
Windows Mac OS X

Manage Machine:
Unavailable
▶ Start Reset

Linux Desktop

IP Address:
10.139.30.219

Username:
student

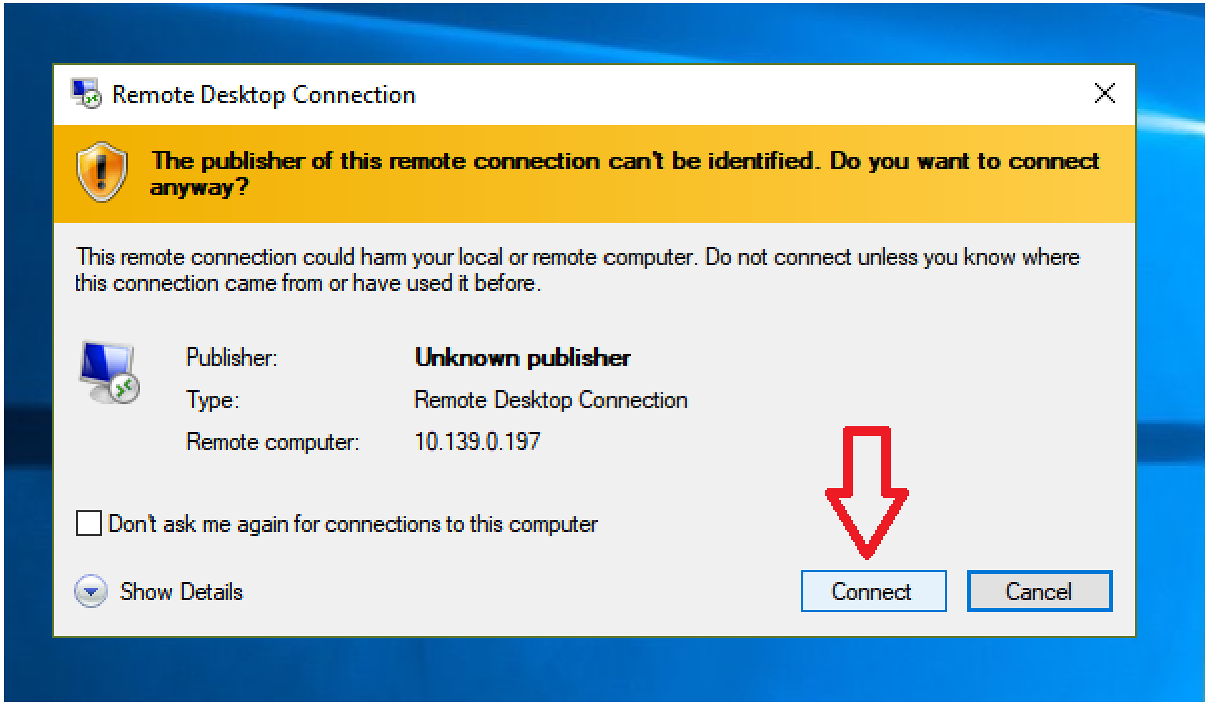
VNC Info
Use the TurboVNC shortcut on the desktop of your Windows Desktop to connect to your Linux Desktop.

Manage Machine:
Unavailable
▶ Start Reset

Access Virtual Lab/Machines

14- Make sure you connect “OpenVPN” with “

15- Open “RDP” to access Windows desktop.



Remote Desktop Connection

The publisher of this remote connection can't be identified. Do you want to connect anyway?

This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher: **Unknown publisher**

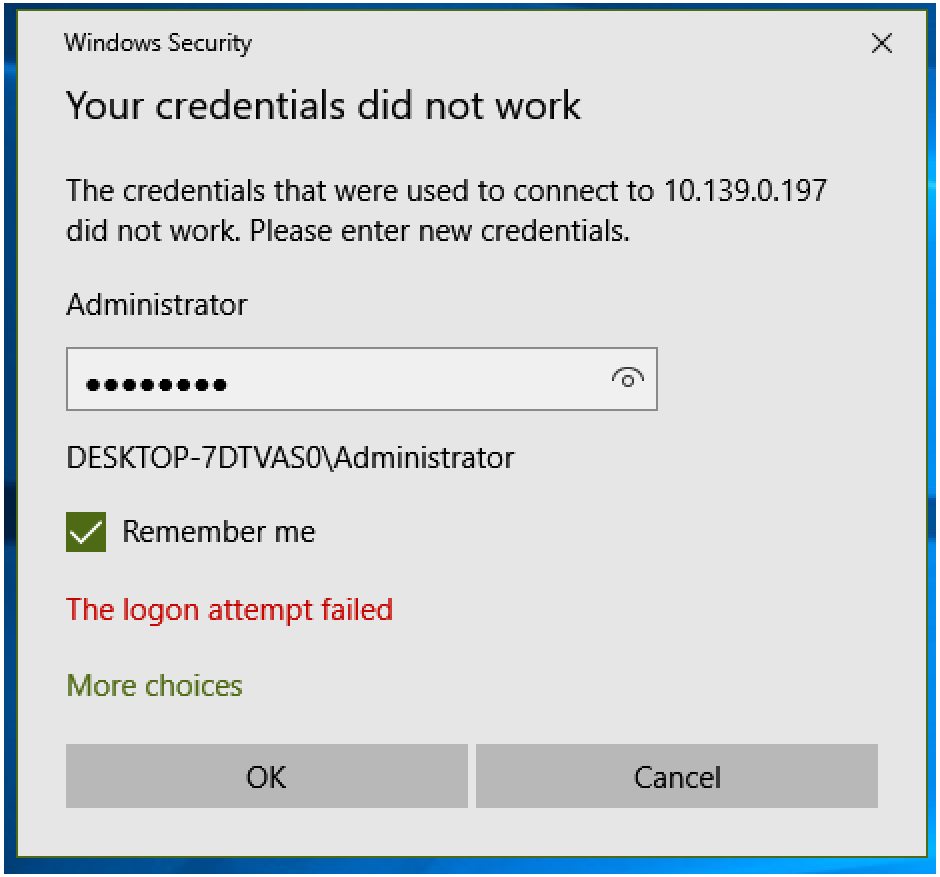
Type: Remote Desktop Connection

Remote computer: 10.139.0.197

□ Don't ask me again for connections to this computer

Access Virtual Lab/Machines

Use the correct windows desktop password



Windows Security

Your credentials did not work

The credentials that were used to connect to 10.139.0.197 did not work. Please enter new credentials.

Administrator

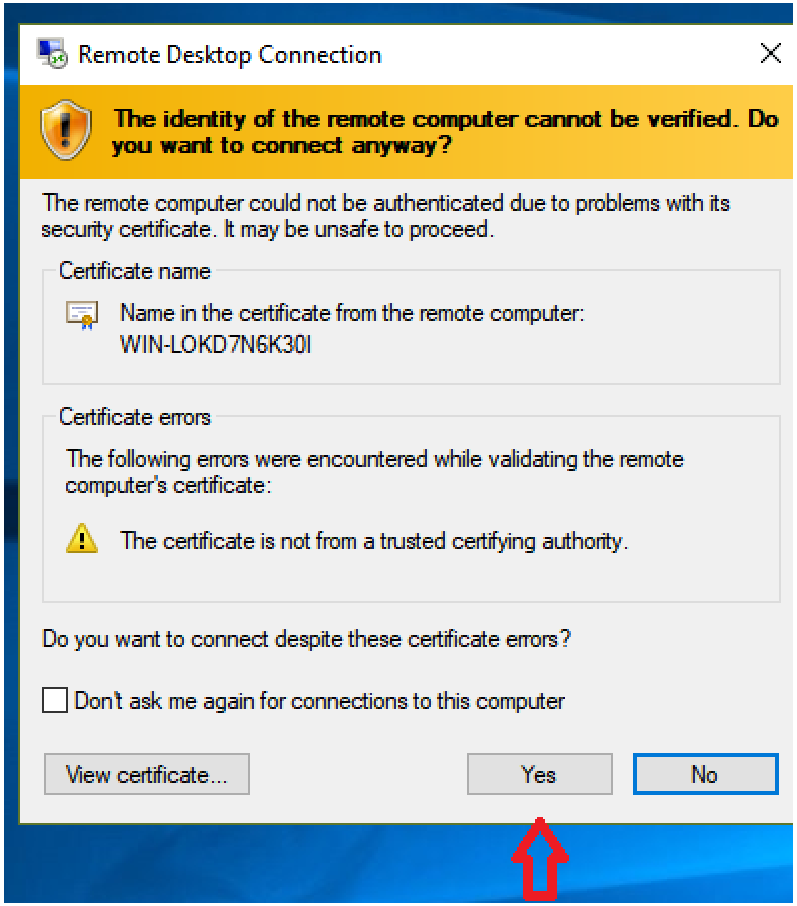
●●●●●●●●●

DESKTOP-7DTVAS0\Administrator

☑ Remember me

The logon attempt failed

More choices



Remote Desktop Connection

The identity of the remote computer cannot be verified. Do you want to connect anyway?

The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.

Certificate name

Name in the certificate from the remote computer:
WIN-LOKD7N6K30I

Certificate errors

The following errors were encountered while validating the remote computer's certificate:

⚠ The certificate is not from a trusted certifying authority.

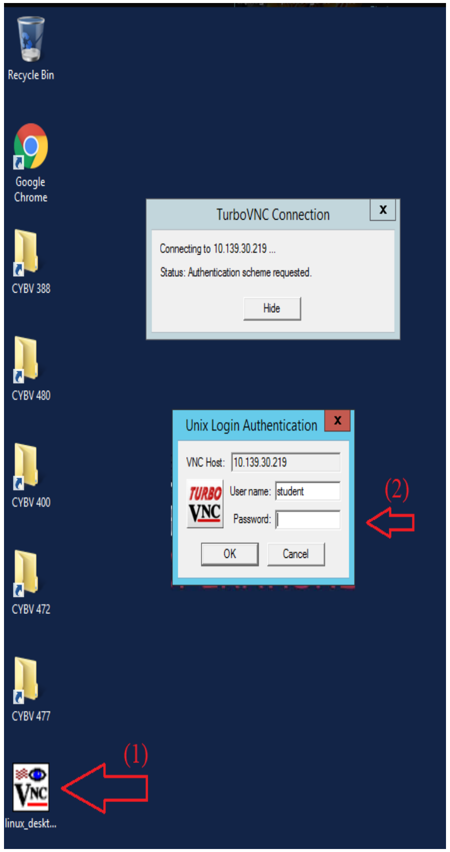
Do you want to connect despite these certificate errors?

☐ Don't ask me again for connections to this computer

Access Virtual Lab/Machines

15- Inside your desktop VM, you should see the Turbo VNC tool to connect to your Linux VM machine. Use the Linux password

PS: If the Linux VM is not available, restart it



TurboVNC Connection

Connecting to 10.139.30.219 ...

Status: Authentication scheme requested.

Unix Login Authentication

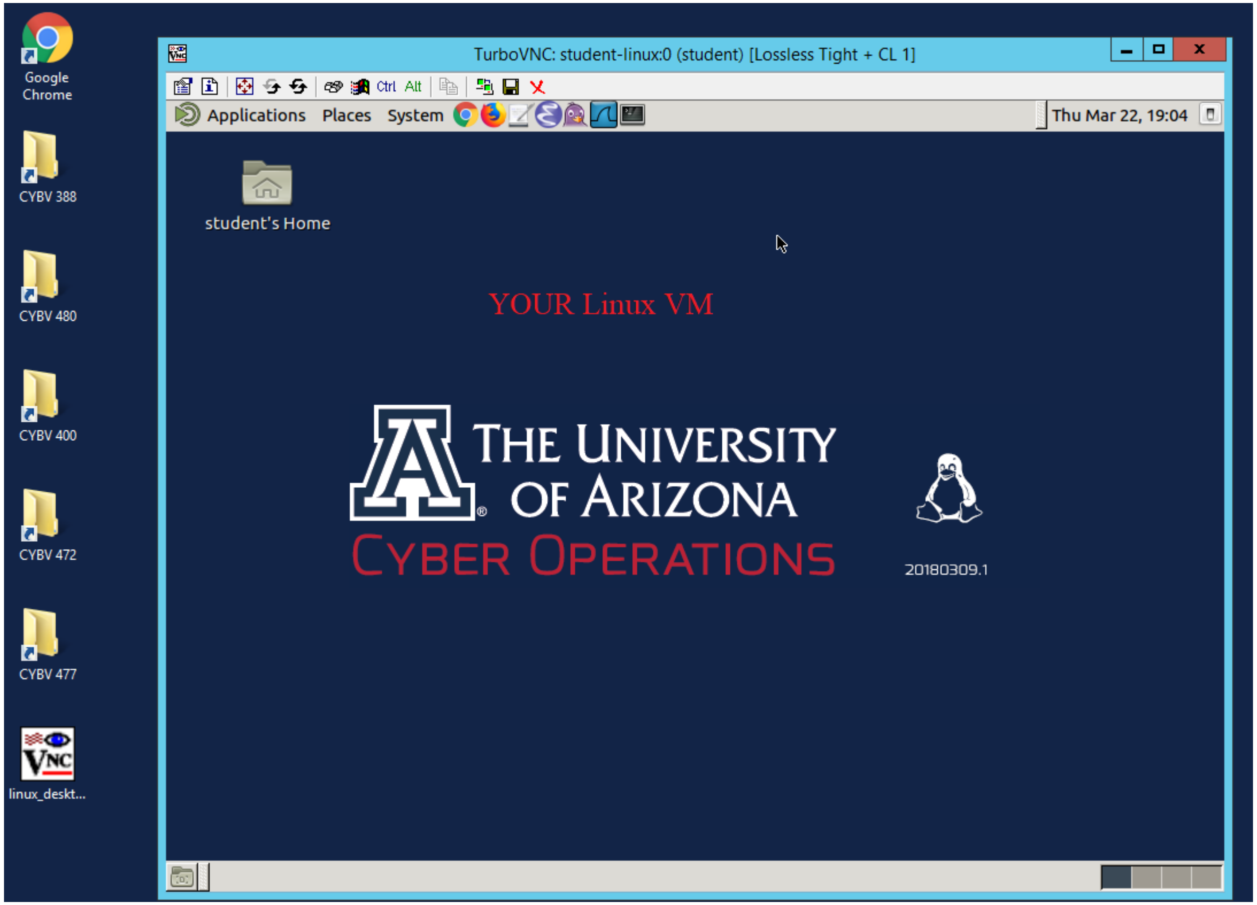
VNC Host: 10.139.30.219

User name: student

Password:

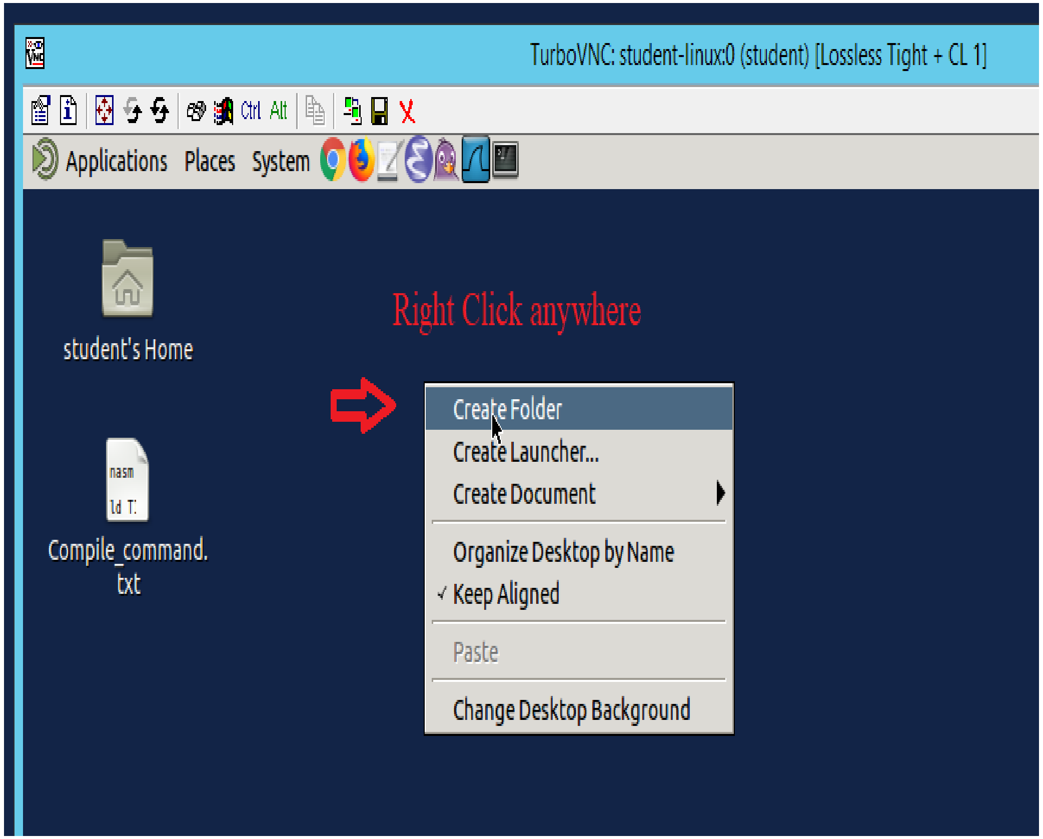
Access Virtual Lab/Machines

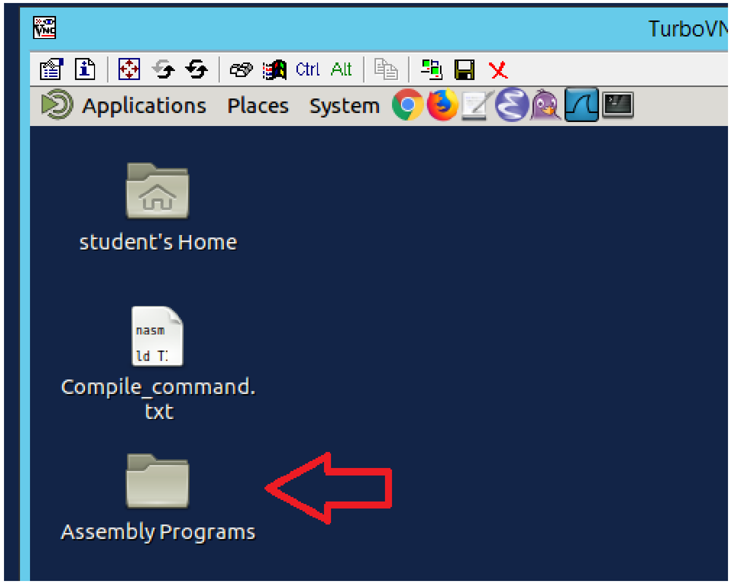
You should see now Linux VM



Access Virtual Lab/Machines

Right click to create new folder
Name the new folder as “AssemblyPrograms”

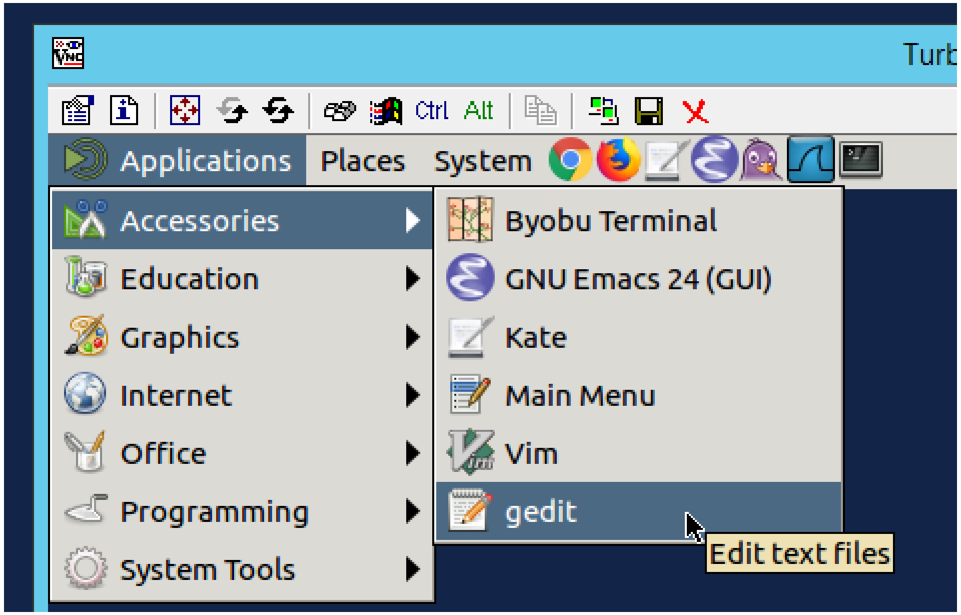




Access Virtual Lab/Machines

Open the text editor and create the first assembly program.

Type the code in the following slide



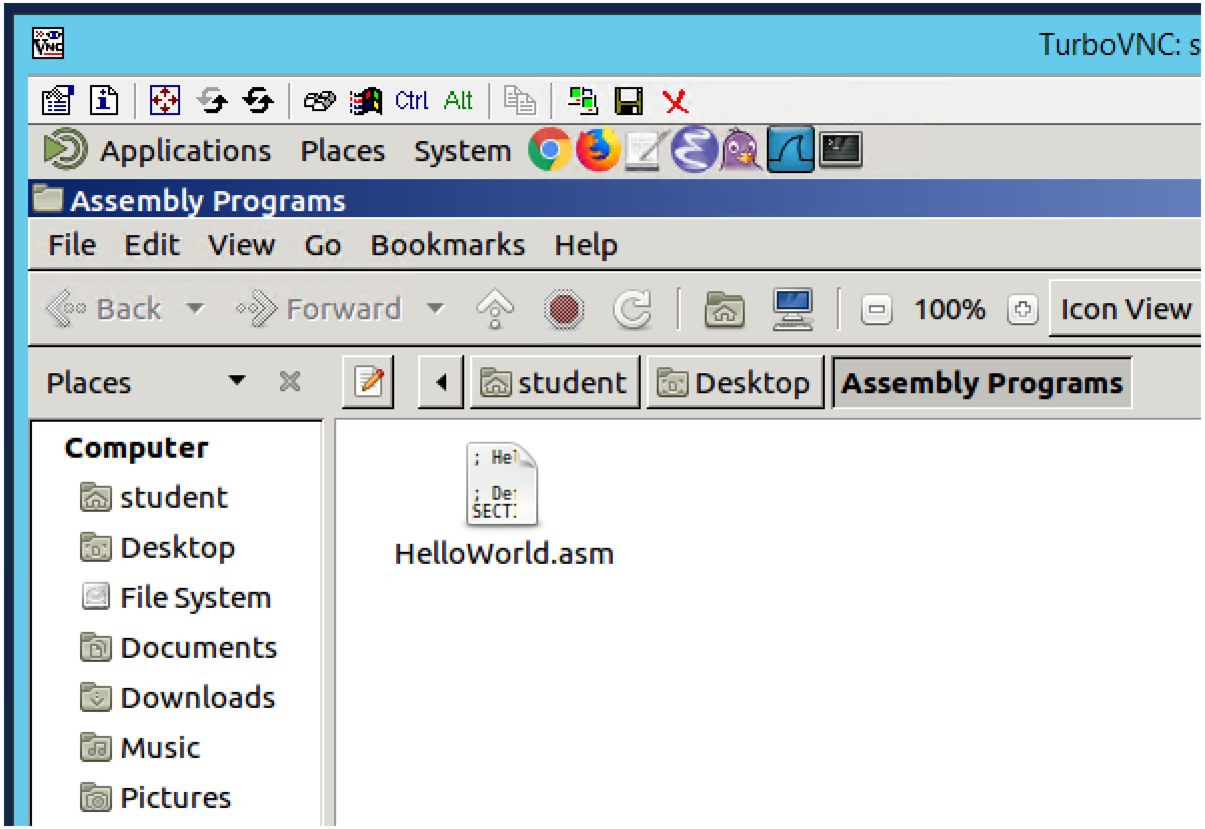
Create First Assembly Program

```
; HelloWorld.asm

; Define variables in the data section
SECTION .DATA
        msg:     db 'Hello world!',10
        msgLen:  equ $-msg

; Code goes in the text section
SECTION .TEXT
        GLOBAL _start

_start:
        mov eax,4               ;  use 'write' system call = 4
        mov ebx,1               ; file descriptor 1 = STDOUT
        mov ecx,msg             ; string to write
        mov edx,msgLen          ; length of string to write
        int 80h                 ; call the kernel

        ; Terminate program
        mov eax,1               ; 'exit' system call
        mov ebx,0               ; exit with error code 0
        int 80h                 ; call the kernel
```

Access Virtual Lab/Machines

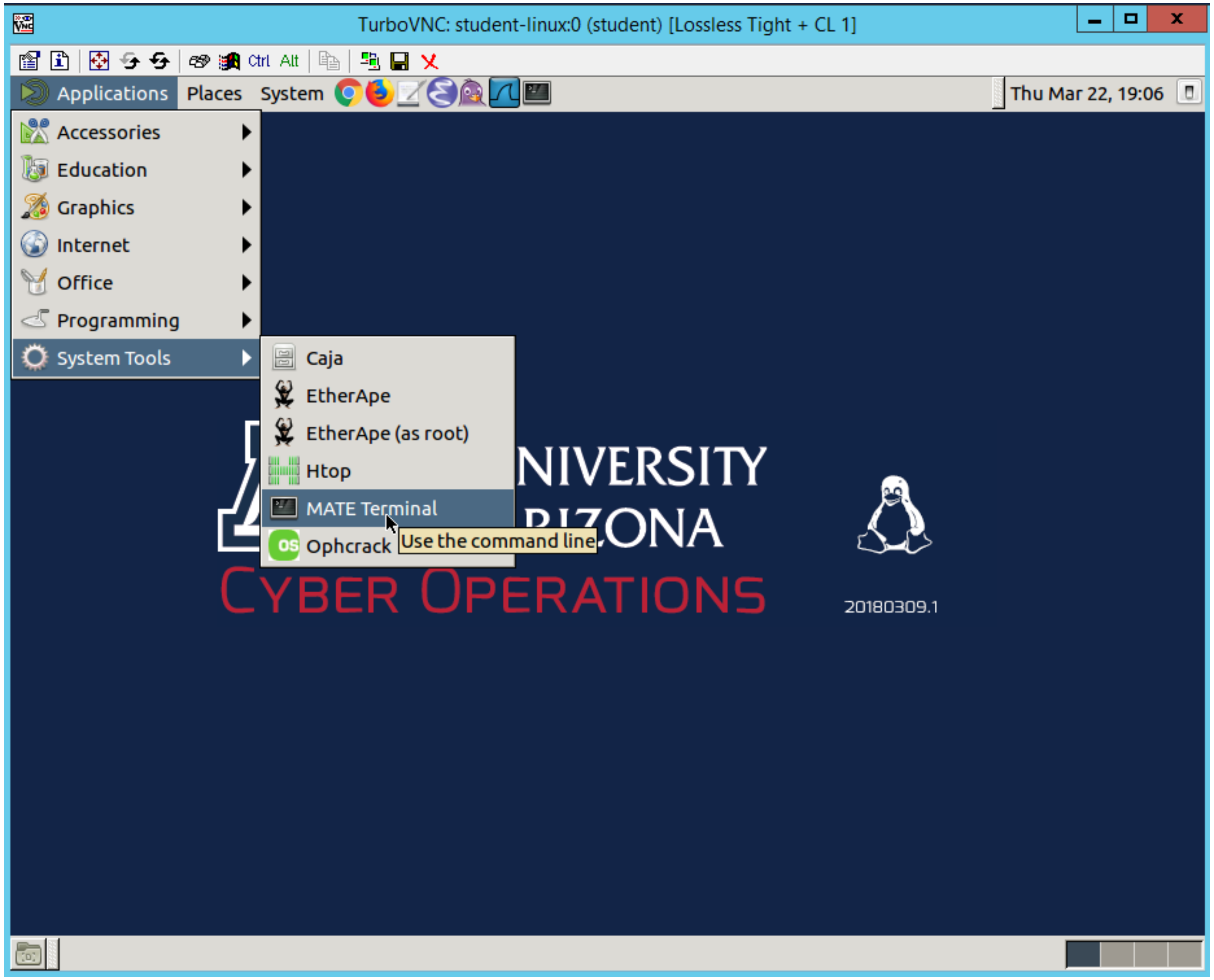
Save the file in the “AssemblyPrograms” director.

Name the file “HelloWorld.asm”



Access Virtual Lab/Machines

Open terminal as follows



Update Linux VM

Execute the following commands to resolve several issues

```
sudo apt-get update
```

```
sudo apt-get install gcc-multilib g++-multilib
```

- To compile the program:

1- Move to the directory that contains the program

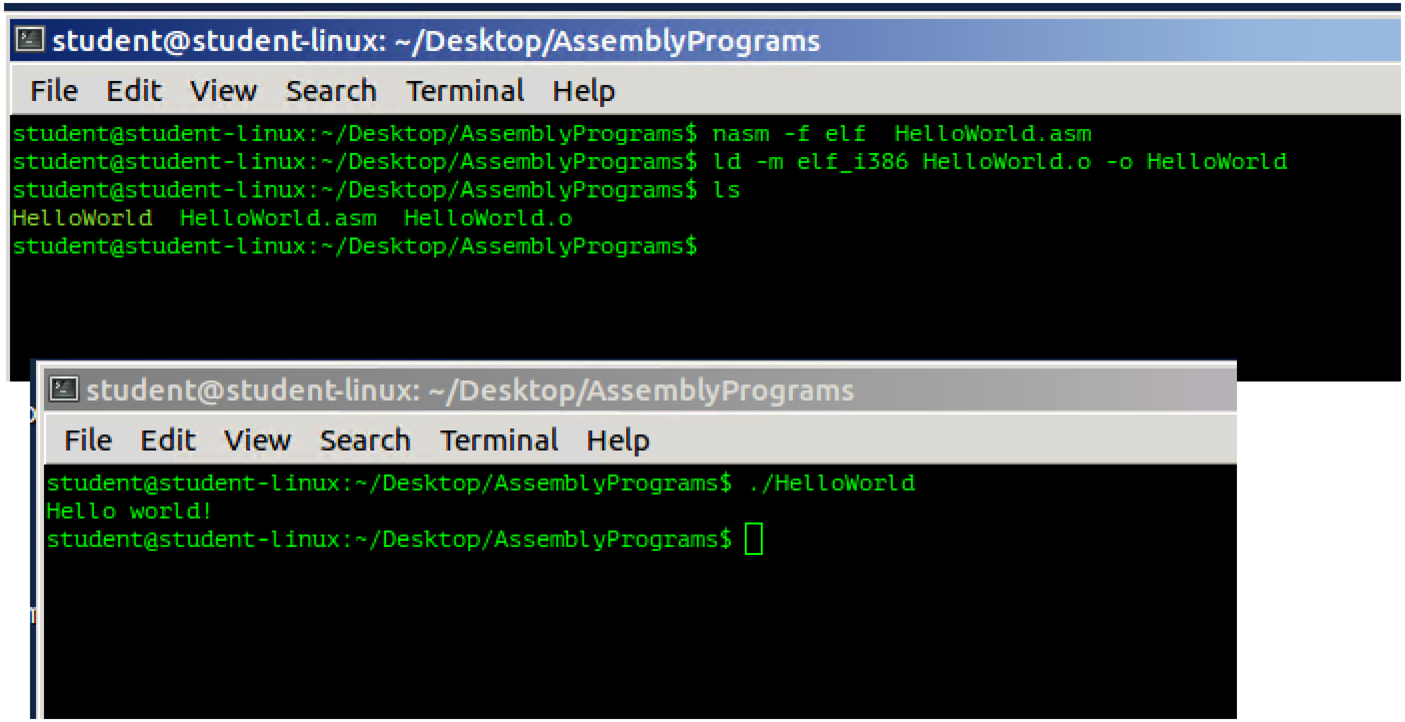
```
cd Desktop/AssemblyPrograms
```

2- Execute the following two steps

```
nasm -f elf  HelloWorld.asm
ld -m elf_i386 HelloWorld.o -o HelloWorld
```

3- To run the program by typing:

```
./HelloWorld (enter)
```



student@student-linux: ~/Desktop/AssemblyPrograms
File Edit View Search Terminal Help

```
student@student-linux:~/Desktop/AssemblyPrograms$ nasm -f elf  HelloWorld.asm
student@student-linux:~/Desktop/AssemblyPrograms$ ld -m elf_i386 HelloWorld.o -o HelloWorld
student@student-linux:~/Desktop/AssemblyPrograms$ ls
HelloWorld  HelloWorld.asm  HelloWorld.o
student@student-linux:~/Desktop/AssemblyPrograms$
```

student@student-linux: ~/Desktop/AssemblyPrograms
File Edit View Search Terminal Help

```
student@student-linux:~/Desktop/AssemblyPrograms$ ./HelloWorld
Hello world!
student@student-linux:~/Desktop/AssemblyPrograms$
```

What is Machine Language?

- High-level languages

  - Software developers normally use high level programming language (e.g. C, C++) to create software applications.

  - Converted to machine code by a compiler (e.g. Microsoft Visual Studio)

- Machine code

  Binary operation code “opcodes” that instruct the to execute instructions.

- Low-level languages (assembly language)

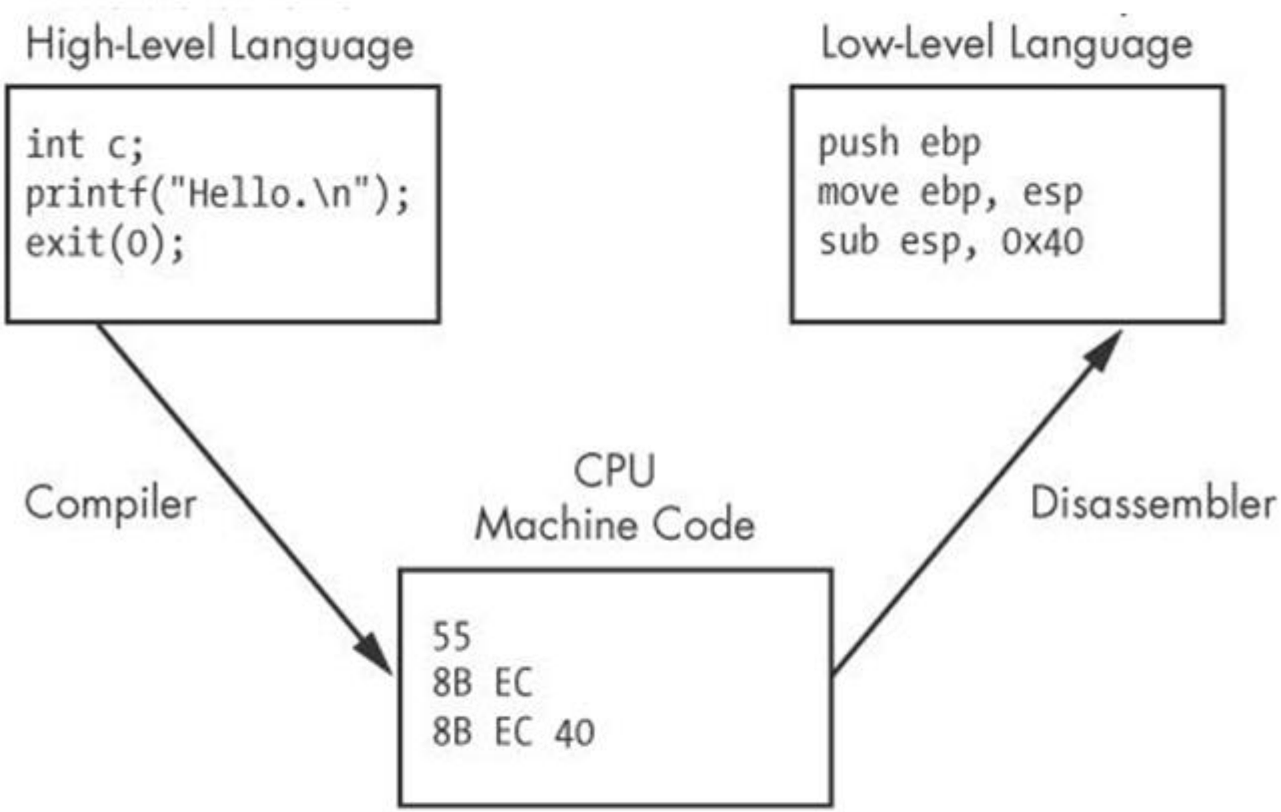
  - Human-readable version of processor's instruction set

  - Assembly language (PUSH, POP, NOP, MOV, etc.)

  - Disassembler converts machine code to assembly language

  - The highest-level language that can be obtained from malware executable code

Machine Languages



High-Level Language

```
int c;
printf("Hello.\n");
exit(0);
```

Compiler

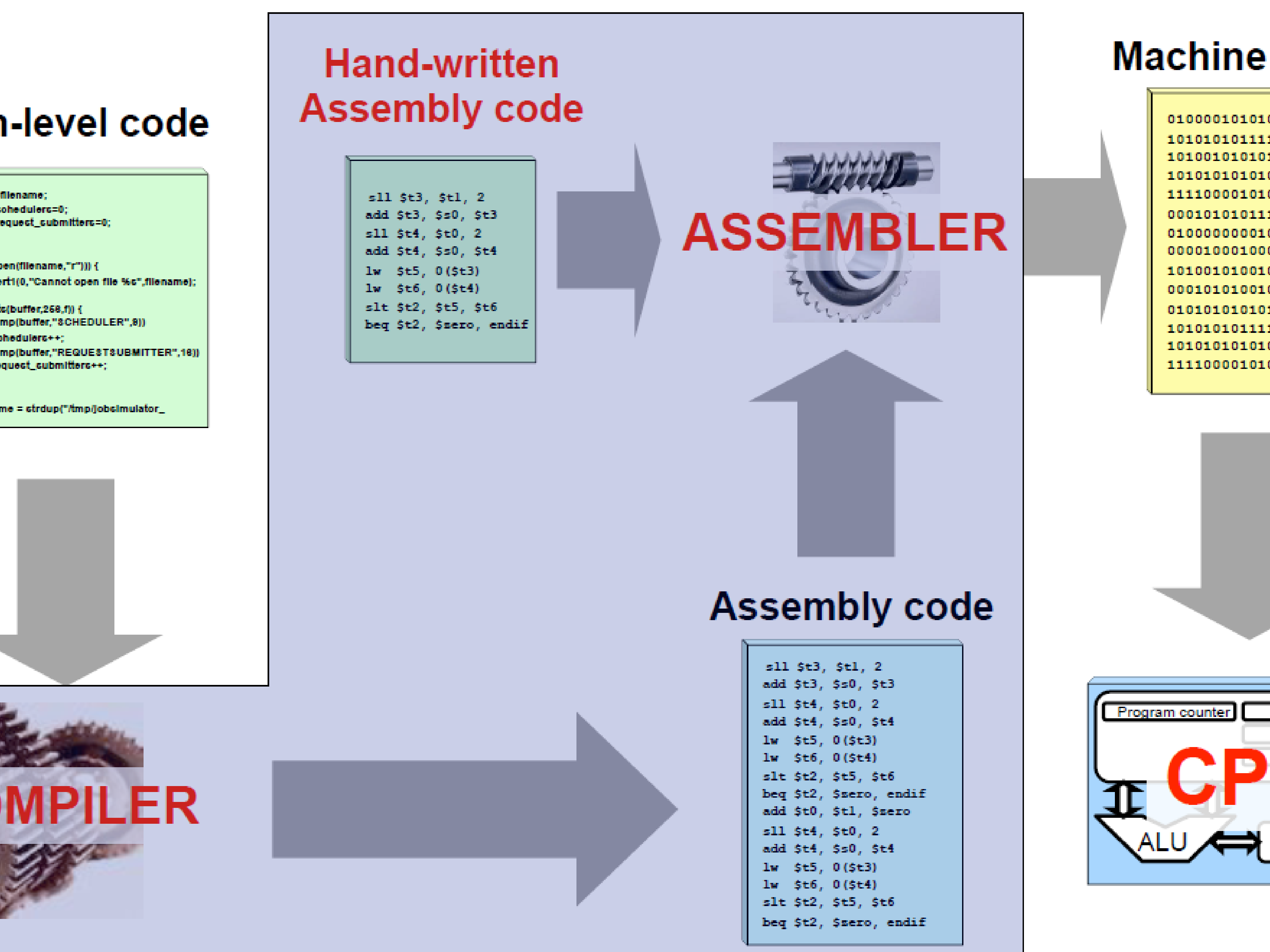
CPU
Machine Code

```
55
8B EC
8B EC 40
```

Disassembler

Low-Level Language

```
push ebp
move ebp, esp
sub esp, 0x40
```



n-level code

```
filename;
schedulers=0;
equest_submitters=0;


pen(filename,"r"))) {
ert1(0,"Cannot open file %s",filename);

s(buffer,258,f)) {
mp(buffer,"SCHEDULER",8))
chedulers++;
mp(buffer,"REQUESTSUBMITTER",16))
quest_submitters++;



me = strdup("/tmp/jobsimulator_
```

Hand-written Assembly code

```
sll $t3, $t1, 2
add $t3, $s0, $t3
sll $t4, $t0, 2
add $t4, $s0, $t4
lw  $t5, 0($t3)
lw  $t6, 0($t4)
slt $t2, $t5, $t6
beq $t2, $zero, endif
```

Machine

```
010000010101
101010101111
101001010101
101010101010
111100001010
000101010111
010000000010
000010001000
101001010010
000101010010
010101010101
101010101111
101010101010
111100001010
```

Assembly code

```
sll $t3, $t1, 2
add $t3, $s0, $t3
sll $t4, $t0, 2
add $t4, $s0, $t4
lw  $t5, 0($t3)
lw  $t6, 0($t4)
slt $t2, $t5, $t6
beq $t2, $zero, endif
add $t0, $t1, $zero
sll $t4, $t0, 2
add $t4, $s0, $t4
lw  $t5, 0($t3)
lw  $t6, 0($t4)
slt $t2, $t5, $t6
beq $t2, $zero, endif
```

Assembly Language

- Different versions for each type of processor family
- x86 – 32-bit version known as Intel IA-32 (most common)
- x64 – 64-bit, Intel and AMD64
- SPARC, PowerPC, MIPS, ARM – others
- Windows runs on x86 (32 bits) or x64 (64 bits)
- x64 machines can run x86 programs

Why should we learn Assembly Language?

- Learning assembly makes you a better programmer in high-level languages
- You may need to write assembly code for performance optimization part of larger software projects
- You need to use assembly code for embedded devices
- You need to use assembly code for device drivers

Binary number, Bits and Bytes

- Each bit is either 1 (True) or 0 (False)
- The binary number system is base 2 system since each digit could be be 0 or 1.
- Byte: 8 Bits
- Byte = a unit of storage
  - 1KB = 2^{10} = 1024 Bytes
  - 1MB = 2^{20} = 1,048,576 Bytes
  - 1GB = 2^{30} = 1,099,511,627,776 Bytes
  - Main memory (RAM) is measured in GB
  - Disk storage is measured in GB for small systems, TB (Tera Bytes = 2^{40}) for large systems

Lab 1 Assignment

- Use NASM tool at your virtual machine to write an assembly language program that display the following messages in order. Note that you should display every message is in a separate line

  I accessed my VM at the virtual lab

  This is my first assembly program using virtual lab

- Run the program and capture the screen shot to show the output
- Create a new Word or PDF file and name it “Your Name-Lab1”.
- In your Word or PDF document, include your code and briefly explain every line
- In your Word or PDF document, include the screen shot to show the output
- Submit your completed Word or PDF document in the Laboratory Assignment Lab 1 assignment link

Week 1 Assignments

- **Learning Materials**
  - Week 1 Presentation
  - Read 1-14 (Duntermann, Jeff. Assembly Language Step by Step, Programming with Linux)

- **Assignment**

1- Complete “Lab 1” by coming Sunday 11:59 PM.

Putting It All Together

You should know:

- **Understand the course’s goals**
- **Understand the course’s requirements**
- **Understand how can you access virtual lab and your virtual machines**
- **Understand how can build, compile, and run assembly language**
- **What is the assembly language?**
- **Build first assembly program**
- **Understand this week assignment**

Questions?

Lecture 1-2

Data Presentation and Number Systems