

## Linux Development Getting Started

### VB-Linux on your Laptop – Convenient Option Compared to Remote Transformer Access

First, on your Windows or Mac OS-X laptop, install Oracle Virtual-Box as follows:

First, download Ubuntu Linux distribution 12.04.1, or newer from here:

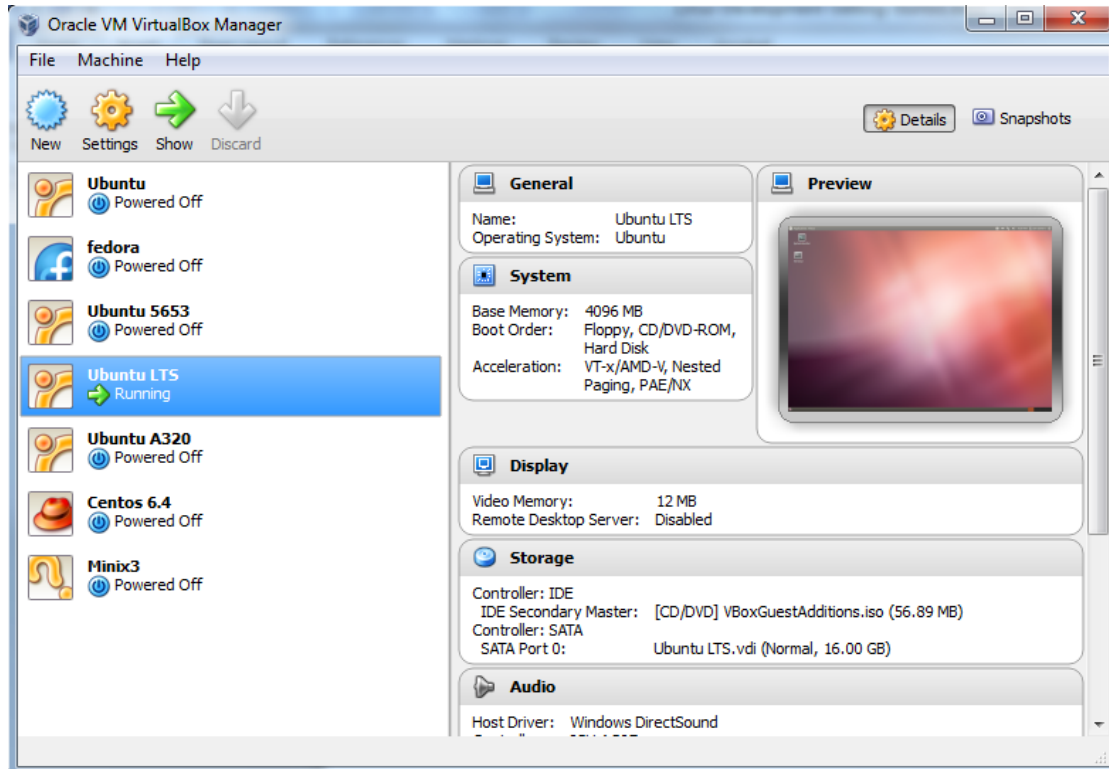
1. <http://www.ubuntu.com/download/desktop>
2. Or you can download and install the exact same version I run on VB from here - [http://math.uaa.alaska.edu/~ssiewert/linux\\_dvd\\_iso/](http://math.uaa.alaska.edu/~ssiewert/linux_dvd_iso/)

To install Ubuntu Linux on Virtual Box, then you'll also need to download Virtual Box here -

<https://www.virtualbox.org/wiki/Downloads>

If you want to do a native installation, I recommend you do a full backup first, and then either install on a second hard-disk, or on an un-used partition of your primary hard-disk, but do be careful either way. Virtual Box is fairly easy and you can most easily install the Ubuntu "guest OS" by installing VB (Virtual Box) and then installing Linux as a new machine from the DVD ISO image you downloaded from the Ubuntu download web site.

Virtual Box gives you a way to install multiple guest Operating Systems like Linux distributions, Minix OS to go with Tanenbaum's text, or any other OS you might want to run on your Windows or Mac OS-X machine that you normally use, with far less risk than a dual-boot system. The VB manager looks like this:



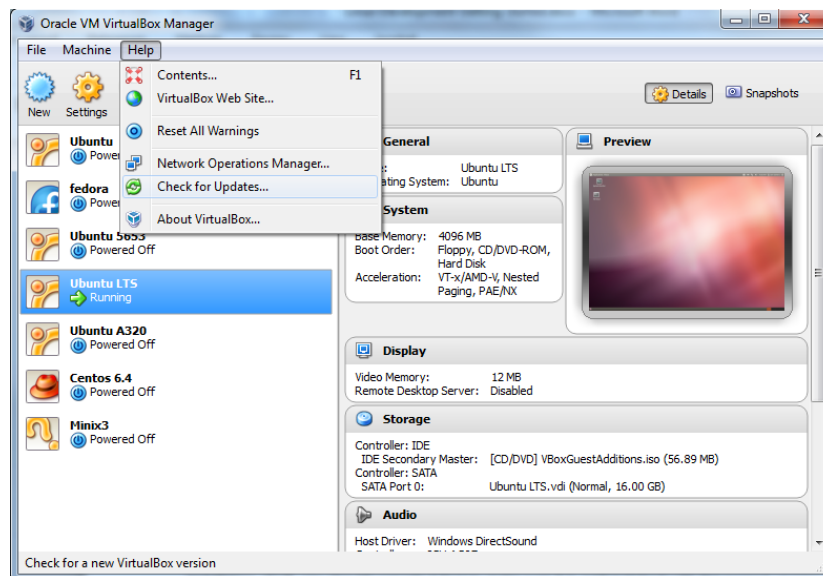
If you run into problems with a VB installation or any configuration of VB, see me during office hours.

## Virtual Box Extensions

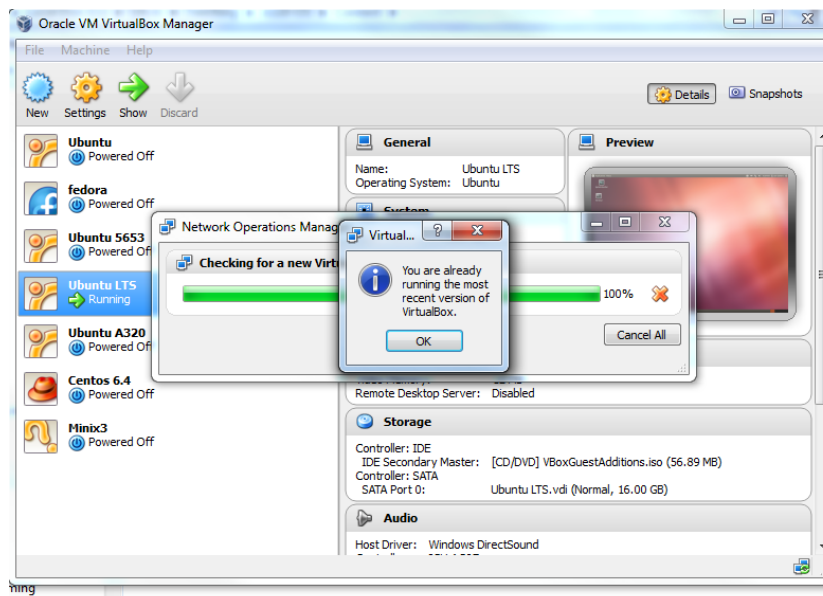
The VB Extensions can be installed after you have VB and your Linux distribution installed to provide:

- 1) Expansion of your desktop to full screen
- 2) Mounting of Windows folders inside your Linux guest OS
- 3) Cut and paste between your guest OS and Windows

So, first, install the VB Extensions per current instructions from Oracle (carefully!). If you have an older installation of VB, you should upgrade it first to make sure it is well matched with the current version of the extensions available for download (this is critical, so check carefully). It is best to download the extensions at the same time you download VB. If you already have VB installed, upgrade to the latest:

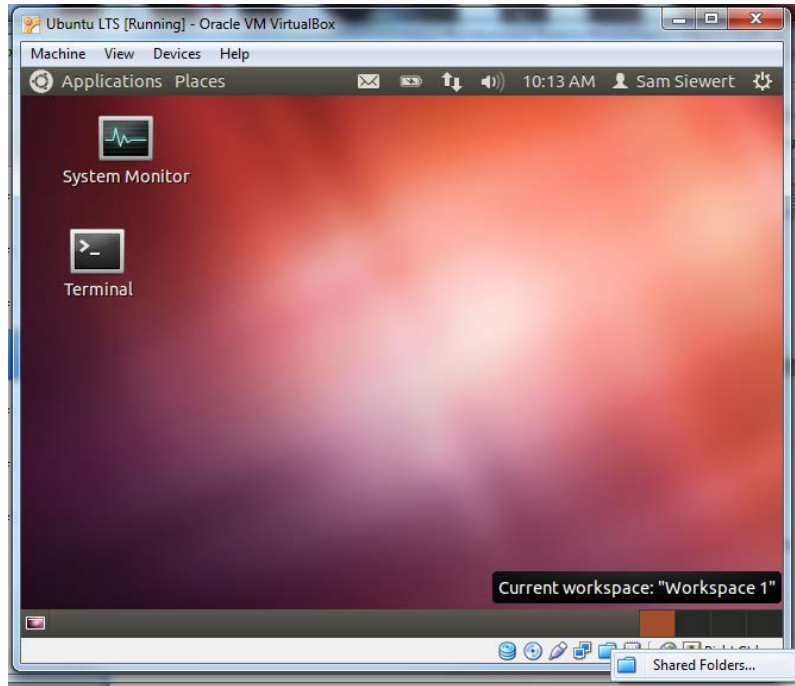


In my case, I was already up to date:

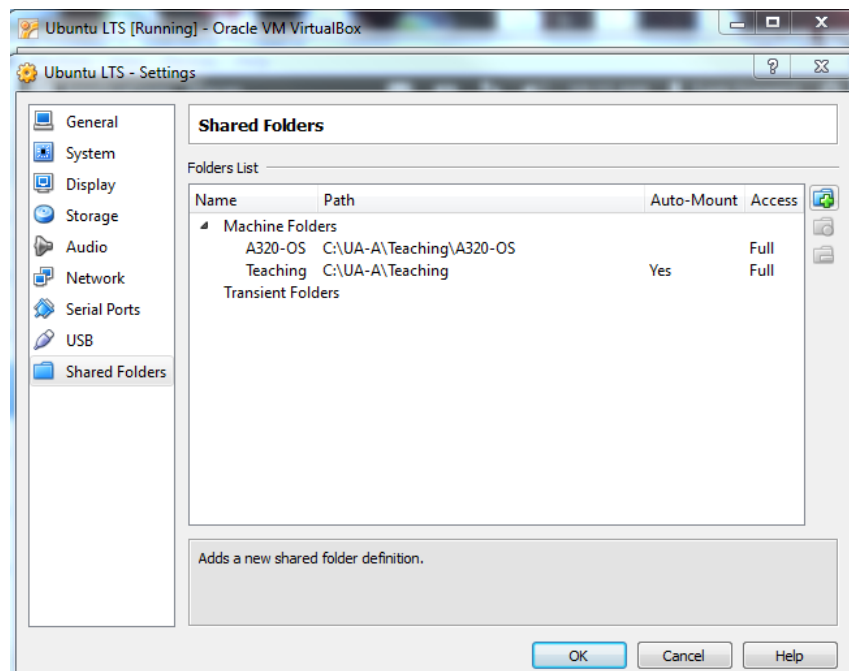


### **Sharing Files with Windows Using VB Extensions**

Once you have the latest VB manager running and you install the VB extensions to go with it, then, you can for example set up file sharing with Windows as follows. First, access the shared folder icon that you will see in the lower RIGHT corner of your guest OS VM:



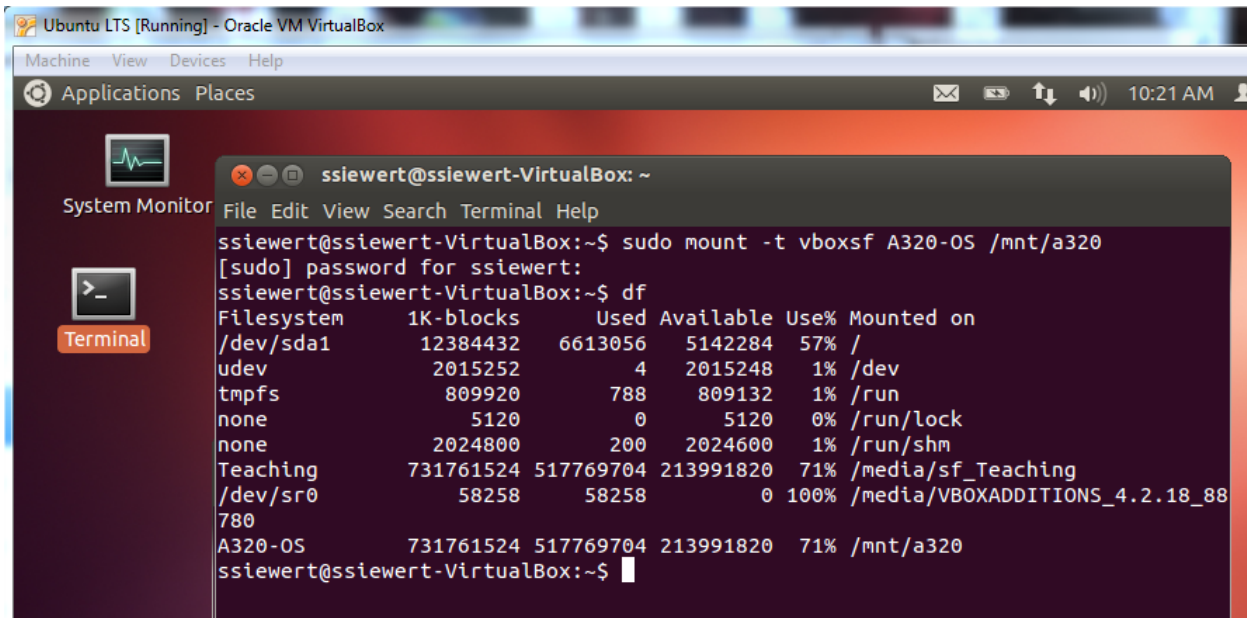
Click on the +Folder icon and browse to the Windows folder you want to share as I have done for A320-OS or Teaching:



Once you have shared the Windows folders you want, then the rest is simple. As the instructions tell you when you hover over your Shared Folders, just use the following command inside your Linux guest OS instance (e.g. for my A320-OS shared folder):

If you have not established a mount point, first do “**sudo mkdir /mnt/A320OS**”, then “**sudo mount -t vboxsf A320-OS /mnt/a320**”.

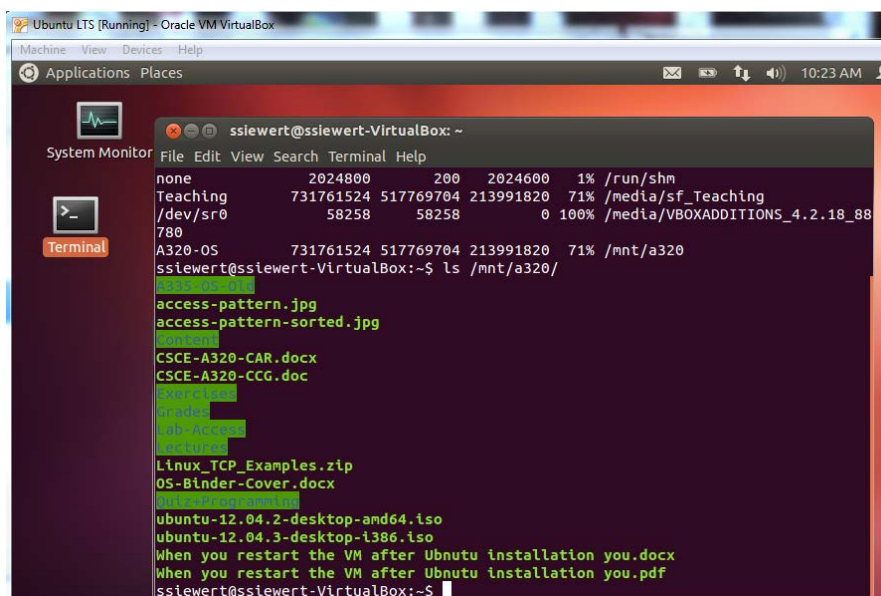
That’s it, now just access the files in Linux like any other file system. Here’s my shared folder mount as shown using the Linux df command:



The screenshot shows a terminal window titled "ssiewert@ssiewert-VirtualBox: ~". The user has executed the command `sudo mount -t vboxsf A320-OS /mnt/a320` and then `df`. The output of `df` is as follows:

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	12384432	6613056	5142284	57%	/
udev	2015252	4	2015248	1%	/dev
tmpfs	809920	788	809132	1%	/run
none	5120	0	5120	0%	/run/lock
none	2024800	200	2024600	1%	/run/shm
Teaching	731761524	517769704	213991820	71%	/media/sf_Teaching
/dev/sr0	58258	58258	0	100%	/media/VBOXADDITIONS_4.2.18_88780
A320-OS	731761524	517769704	213991820	71%	/mnt/a320

Now I can list, copy files, etc. as I would with any other Linux filesystem with this Windows shared mount:



The screenshot shows the same terminal window as before, but now the user has executed the command `ls /mnt/a320/`. The output lists the files and directories in the mounted folder:

```
access-pattern.jpg
access-pattern-sorted.jpg
content
CSCE-A320-CAR.docx
CSCE-A320-CCG.doc
exploitlab
exploitlab
lab-answers
lab-answers
Linux_TCP_Examples.zip
OS-Binder-Cover.docx
ubuntu12.04.2-desktop-amd64.iso
ubuntu-12.04.2-desktop-amd64.iso
ubuntu-12.04.3-desktop-i386.iso
When you restart the VM after Ubuntu installation you.docx
When you restart the VM after Ubuntu installation you.pdf
```

## Development Tools for VB-Linux

If you're new to Linux and/or just don't like command line type development, here's some recommendations for the course:

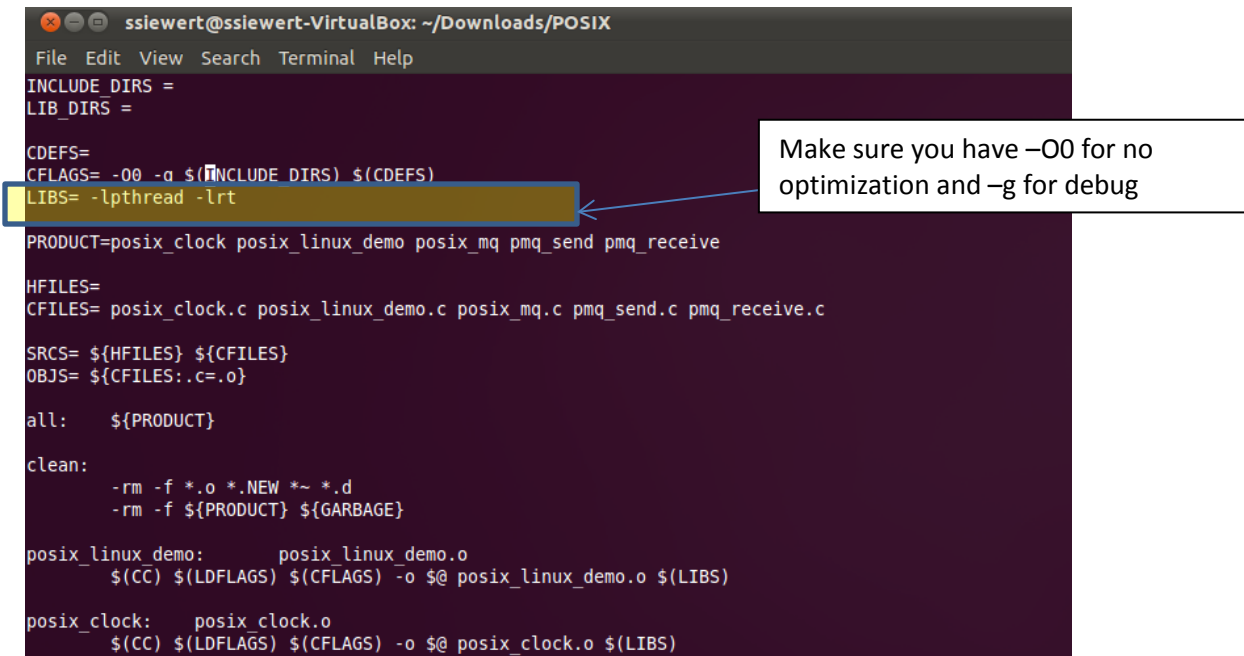
1. On VB-Linux – try the Eclipse C development environment for a full IDE and/or use **nano**, **make**, and **ddd** (**sudo apt-get install ddd**), or better yet, for full compatibility with Beagle, use **nano**, **make**, and **nemiver** (**sudo apt-get install nemiver**).
2. For a less integrated, but smaller install – use **nano**, **make**, and **nemiver** (**sudo apt-get install nemiver**).

Before you get going with debugging tools, make sure you have read “[Basic Makefile by Example for Linux](#)”, another tutorial I wrote. The key build parameters you need to set for simple and effective debugging are:

If you're new to Linux and/or just don't like command line type development, here's some recommendations for the course on using Makefiles:

1. First, always start with an example – I have many posted [here](#):
2. **Make sure flags are set for no optimization (-O0) and debug symbols (-g)** until you have your code working, then you can turn off debug symbols and turn on optimization if you want/need to do so.

Here's a Makefile that can be executed by simply running “make” in the same directory in which it appears (as long as the name of the file is Makefile) that I will annotate with notes describing in a series of shots of the same file viewed on my system:



```
ssiewert@ssiewert-VirtualBox: ~/Downloads/POSIX
File Edit View Search Terminal Help
INCLUDE_DIRS =
LIB_DIRS =

CDEFS=
CFLAGS= -O0 -g $(INCLUDE_DIRS) $(CDEFS)
LIBS= -lpthread -lrt

PRODUCT=posix_clock posix_linux_demo posix_mq pmq_send pmq_receive

HFILES=
CFILES= posix_clock.c posix_linux_demo.c posix_mq.c pmq_send.c pmq_receive.c

SRCS= ${HFILES} ${CFILES}
OBJS= ${CFILES:.c=.o}

all:    ${PRODUCT}

clean:
    -rm -f *.o *.NEW *~ *.d
    -rm -f ${PRODUCT} ${GARBAGE}

posix_linux_demo:    posix_linux_demo.o
    $(CC) $(LDFLAGS) $(CFLAGS) -o $@ posix_linux_demo.o $(LIBS)

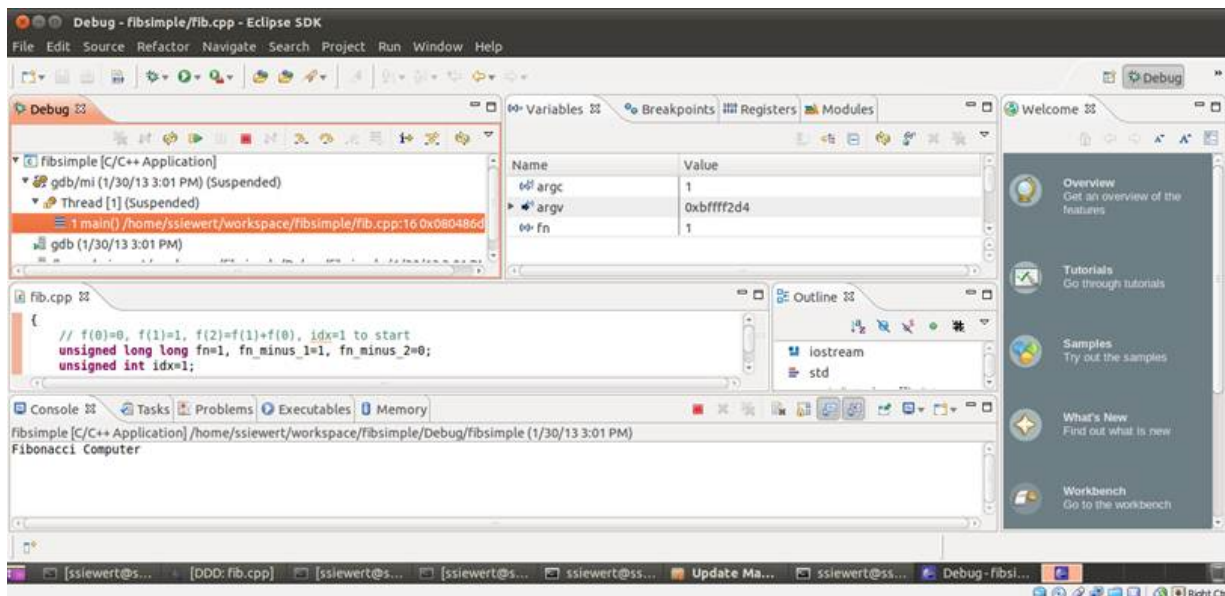
posix_clock:    posix_clock.o
    $(CC) $(LDFLAGS) $(CFLAGS) -o $@ posix_clock.o $(LIBS)
```

Here are the details on how to get going:

For those new to Linux, if you want an IDE that is similar to Visual Studio, you can install eclipse-cdt (on VB-Linux ONLY) as follows:

***sudo apt-get install eclipse-cdt***

After it installs, run it (type in ***eclipse***) you can import the fibsimple for example (File, Import ..., type file in filter, select File System, browse to downloaded fibonacci code, Finish) and you should see:



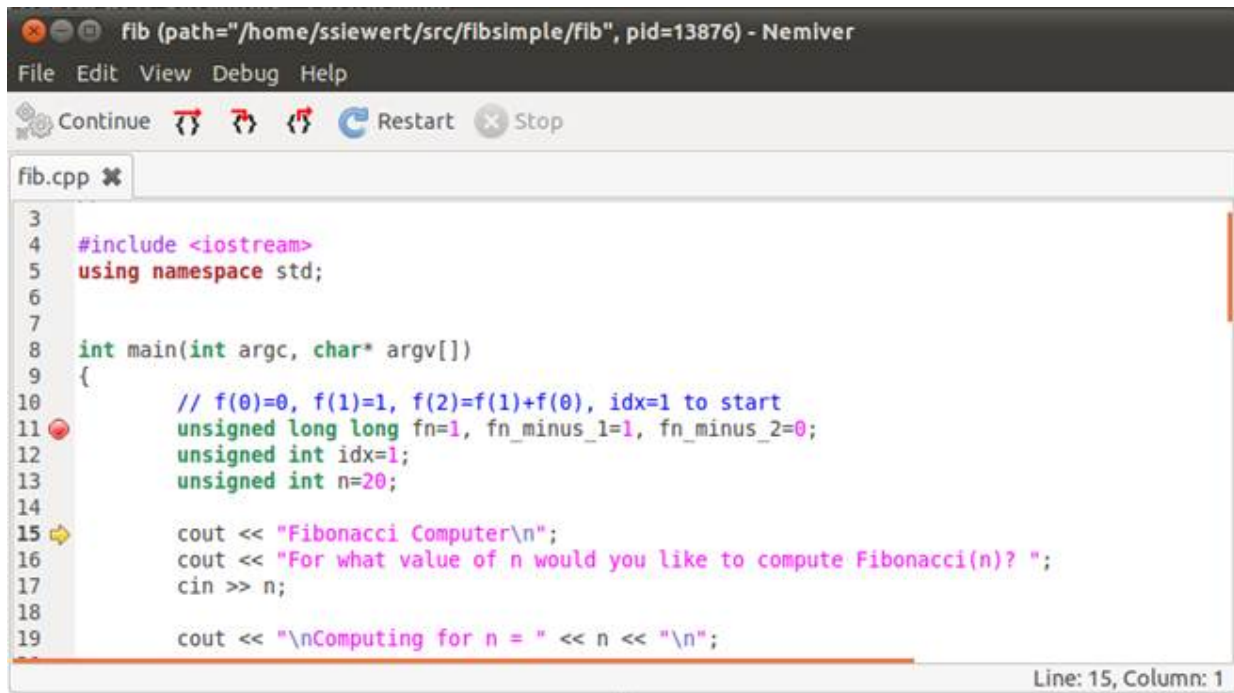
You can set breakpoints, do builds, edit (with an editor that is more like VS), so you may find this more friendly than nano and Makefile with an external debugger, but we will also learn how to use an external debugger.

So, I recommend that you also install nmiver on BOTH your Beagle xM and VB-Linux with:

***sudo apt-get install nemiver***

This is a debugger only and you can load and run code with ***nemiver fib*** for example after your code is built using ***make***. It runs nicely on smaller Linux systems (E.g. an Andorid-class Linux system such as the Beagle xM or Raspberry Pi, but also on VB-Linux).





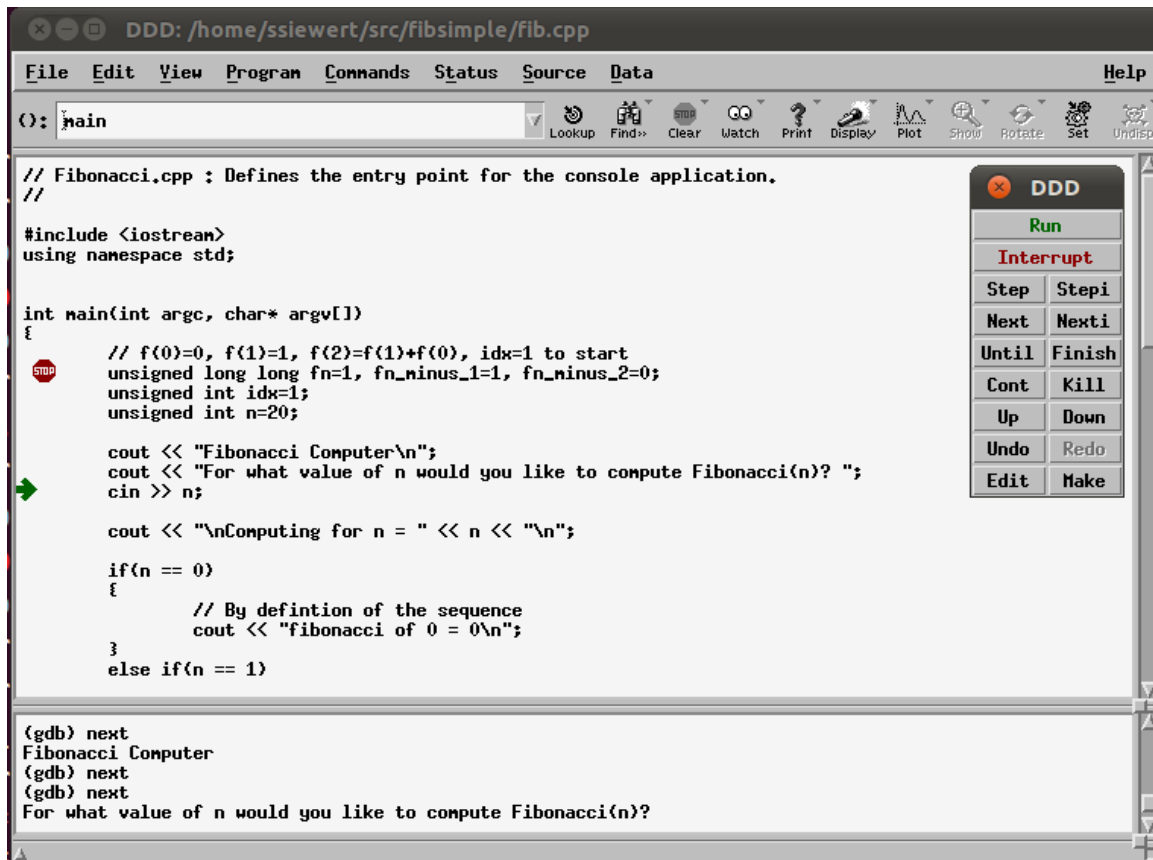
The screenshot shows the Nemiver IDE window titled "fib (path=\"/home/ssiewert/src/fibsimple/fib\", pid=13876) - Nemiver". The menu bar includes File, Edit, View, Debug, and Help. Below the menu bar are buttons for Continue, Restart, and Stop. The main editor area displays the code for "fib.cpp". The code is a C++ program that calculates Fibonacci numbers. It includes the <iostream> header and uses the std namespace. The main function takes argc and argv as arguments. It initializes variables for the Fibonacci sequence: fn=1, fn\_minus\_1=1, and fn\_minus\_2=0. It also sets idx=1 and n=20. The program outputs "Fibonacci Computer" and prompts the user for a value of n. It then outputs the computed Fibonacci value for n. The status bar at the bottom right indicates "Line: 15, Column: 1".

```
3
4 #include <iostream>
5 using namespace std;
6
7
8 int main(int argc, char* argv[])
9 {
10     // f(0)=0, f(1)=1, f(2)=f(1)+f(0), idx=1 to start
11     unsigned long long fn=1, fn_minus_1=1, fn_minus_2=0;
12     unsigned int idx=1;
13     unsigned int n=20;
14
15     cout << "Fibonacci Computer\n";
16     cout << "For what value of n would you like to compute Fibonacci(n)? ";
17     cin >> n;
18
19     cout << "\nComputing for n = " << n << "\n";
```

Hopefully this helps all out a bit more with some nice development and debug tools that go beyond command line.

There are other stand-alone debuggers that run nicely on VB-Linux if you're interested ("ddd" and "xxgdb"), but nemiver runs well on both VB-Linux and small scale systems.

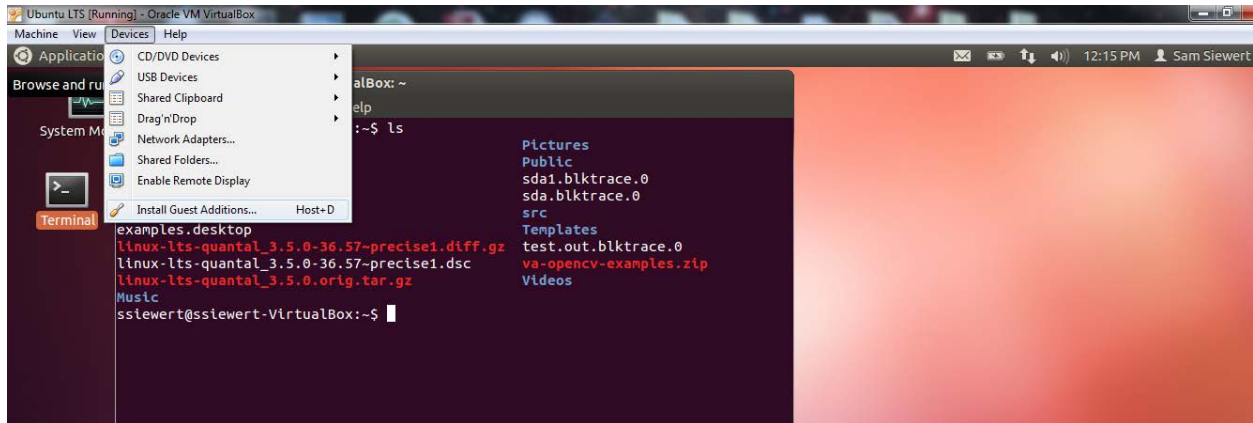
In case you're curious or want to use it on VB-Linux, here's what it looks like:



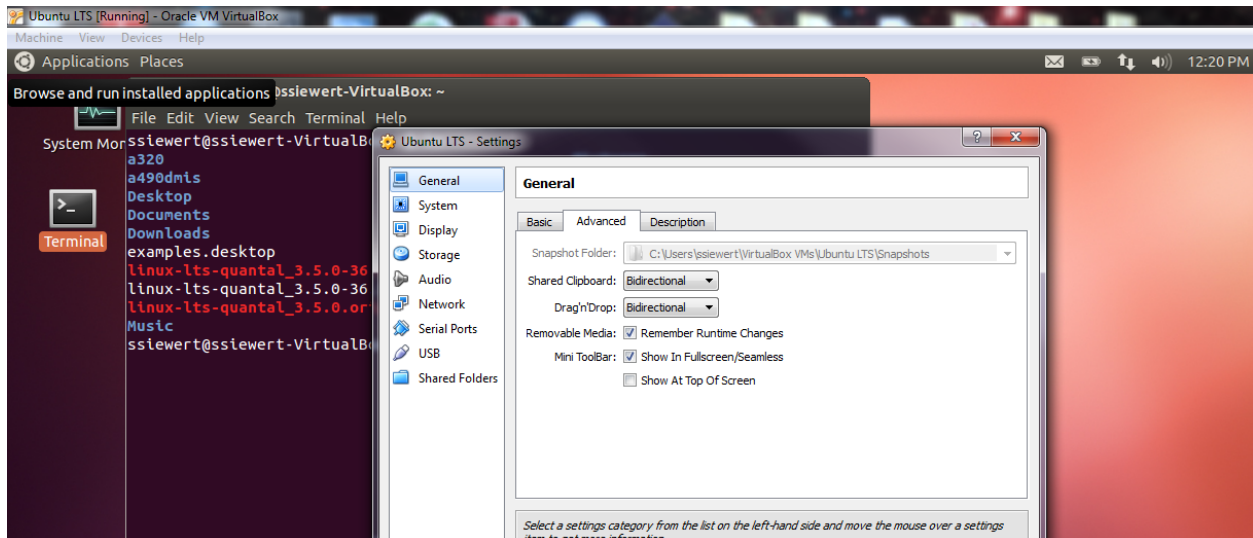


## Some helpful suggestions for configuring your VB-Linux VM:

Once you have Linux installed (e.g. Ubuntu 12.04 LTS) on your VM in Virtual Box, I would suggest installing the Guest additions as follows (make sure you have downloaded them with version that matches your VB - <https://www.virtualbox.org/wiki/Downloads>):

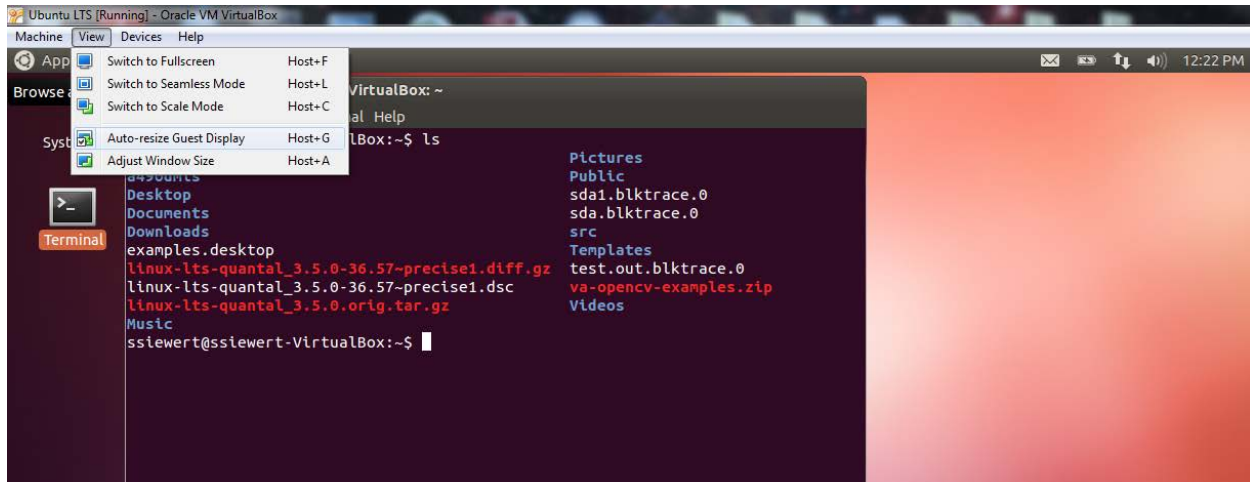


Once the guest additions have installed, you can set up a number of useful VB to host features. First, how about **cut and paste** (use Machine, Settings... and then set up bidirectional clipboard):

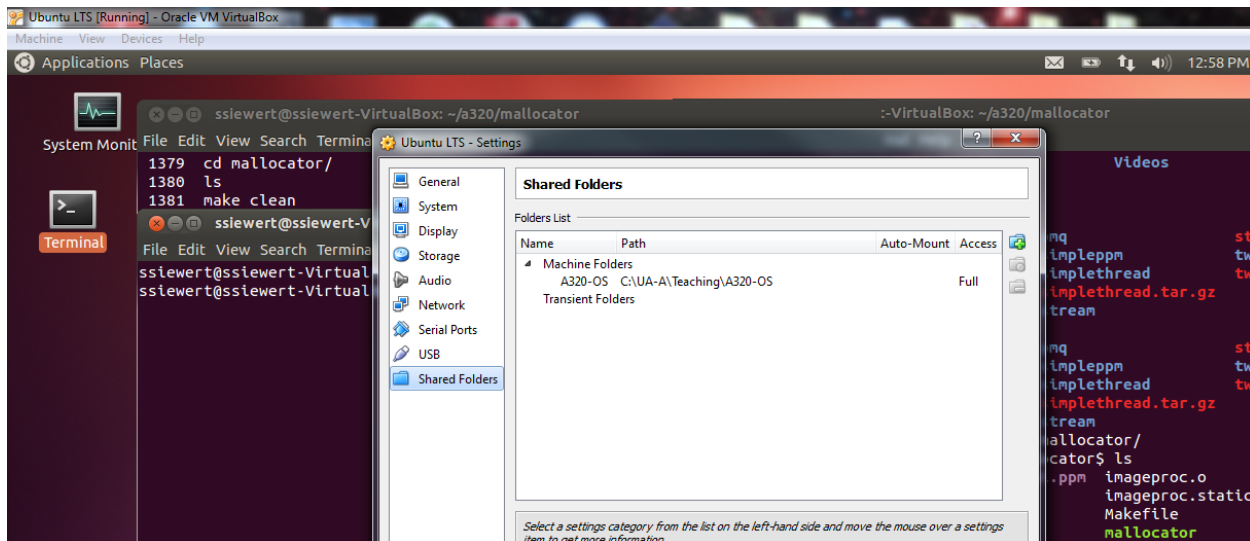


Now you can copy text from shell output and paste it into your reports as well as doing screen-dumps. Screen-dumps are just fine, but sometimes you may want to cut and paste from an external web browser or any number of host tools into your guest OS as well.

Without Guest Additions, you can't resize your VB window and the guest OS desktop. So, second, it's real nice if your **VB window and desktop can be resized:**



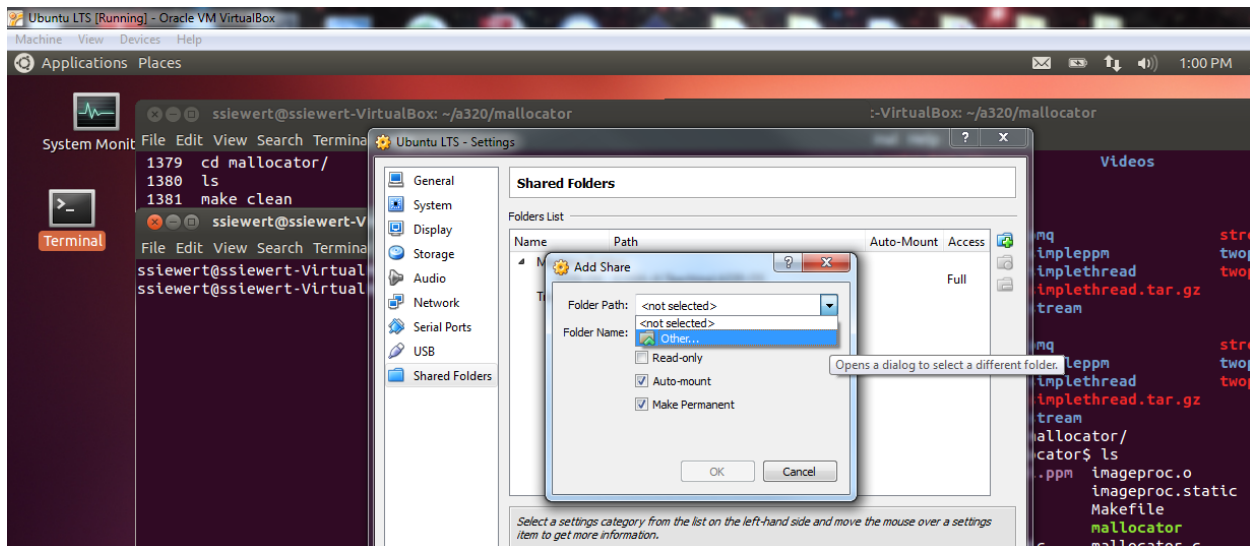
One final option I use quite a bit is the ability to mount a Windows file-system inside my guest OS. You can do this as follows (note I set up C:\UA-A\Teaching\A320-OS already):



For this existing shared folder, I set it up to be manually mounted as follows:

```
ssiewert@ssiewert-VirtualBox:~$ sudo mount -t vboxsf A320-OS /mnt/a320
[sudo] password for ssiewert:
ssiewert@ssiewert-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1       12384432  4889280   6866060  42% /
udev            2015456      4    2015452   1% /dev
tmpfs           809920      780    809140   1% /run
none            5120        0      5120    0% /run/lock
none           2024800     200    2024600   1% /run/shm
/dev/sr0        58258      58258        0 100%
/media/VBOXADDITIONS_4.2.18_88780
A320-OS         731761524 490036428 241725096 67% /mnt/a320
ssiewert@ssiewert-VirtualBox:~$
```

But, let me add C:\UA-A\Teaching now using the +Folder icon ...



For first time mounting, you may need to do manually:

```
ssiewert@ssiewert-VirtualBox:~$ sudo mkdir /mnt/Teaching
ssiewert@ssiewert-VirtualBox:~$ sudo mount -t vboxsf Teaching
/mnt/Teaching
ssiewert@ssiewert-VirtualBox:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda1       12384432   4889300   6866040   42% /
udev            2015456         4   2015452    1% /dev
tmpfs           809920         780    809140    1% /run
none            5120           0      5120     0% /run/lock
none           2024800        200   2024600    1% /run/shm
/dev/sr0         58258       58258         0 100%
/media/VBOXADDITIONS_4.2.18_88780
A320-OS         731761524 490036940 241724584   67% /mnt/a320
Teaching        731761524 490036940 241724584   67% /mnt/Teaching
ssiewert@ssiewert-VirtualBox:~$
```

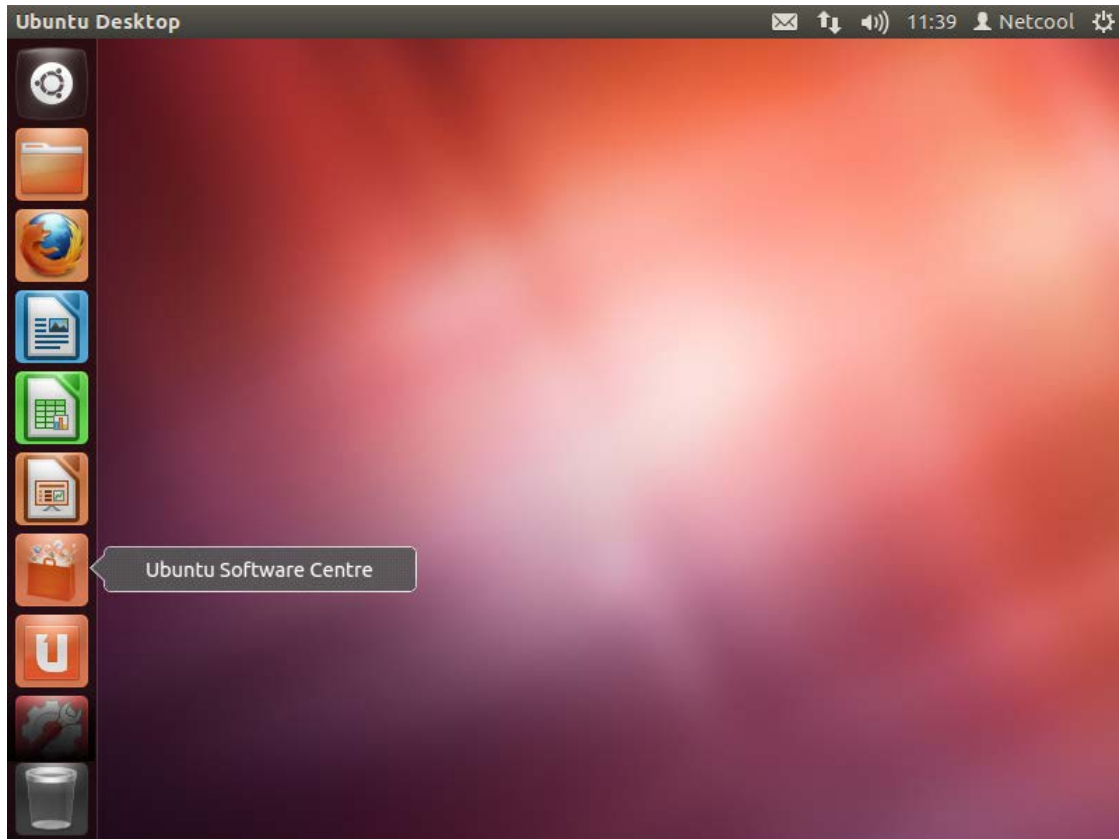
Finally, depending on what style of window manager you prefer, you may want the Gnome sidebar or not. I don't like it, so I use classic Gnome, which can be set up as follows on the next page. If you like the side-bar, then please do use it as it installs.

[Yulei.Liu.AU](#) | Apr 26 2012 | Tags: [gnome](#) [12.04](#) [precise](#) [ubuntu](#) [classic](#) | 60,520 Visits

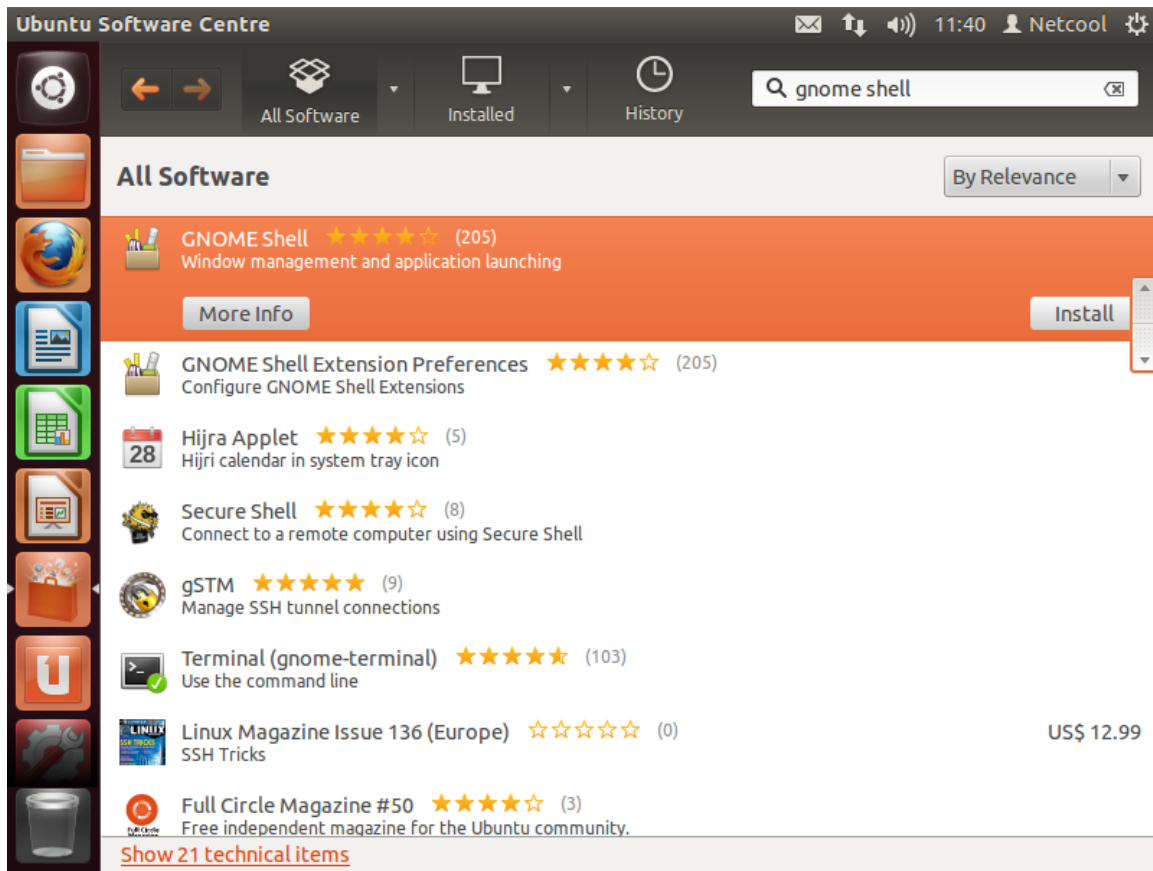
Ubuntu 12.04 LTS is released, unfortunately Unity is still the default interface, if you want to go back to classic Gnome as I do, below is how to:

Gnome session package name has changed from 11.04 to 12.04, the new package name is gnome-shell.

First, start Ubuntu software center as below:



then type "gnome shell" in the search bar, then install GNOME Shell.



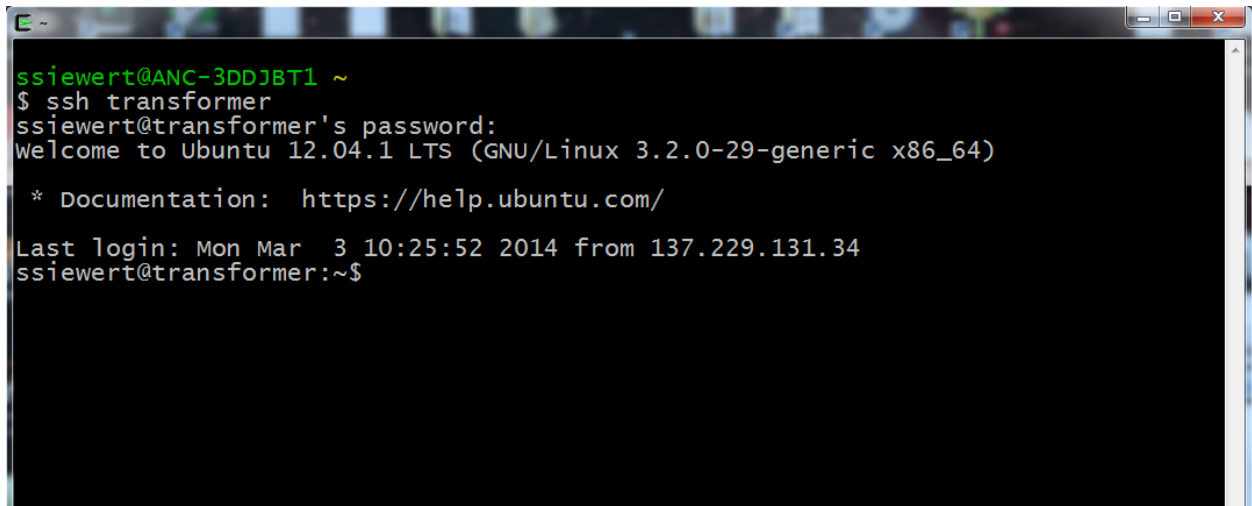
Logout then login, you can now select Gnome Classic in the login screen:

- [Back](#)
- [GNOME](#)
- [GNOME Classic](#)
- [GNOME Classic \(No effects\)](#)
- [Ubuntu](#)
- [Ubuntu 2D](#)

## Ubuntu 12.04 LTS on transformer.uaa.alaska.edu

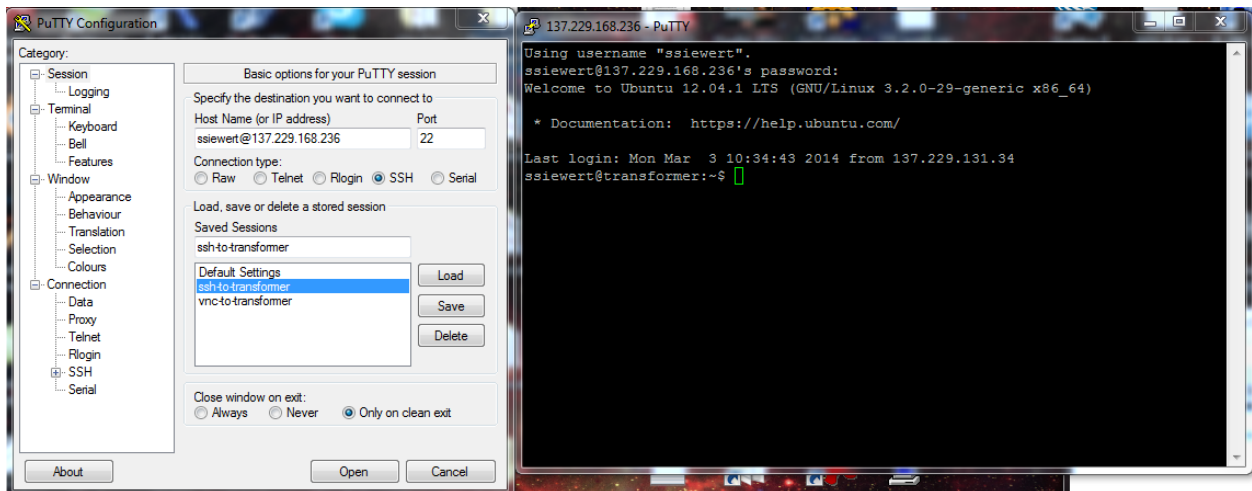
You can simply [ssh](#), [sftp](#), or [scp](#) to transformer.uaa.alaska.edu. If you use Windows, one of the best tools to use for this is Cygwin - <http://cygwin.com/install.html> (make sure you select SSH or OpenSSH for installation when you configure the download), but easier than this if you just want a simple terminal is to use Putty for windows - <http://www.putty.org/>

Assuming you use Cygwin, a simple SSH login will look like this:



```
ssiewert@ANC-3DDJBT1 ~  
$ ssh transformer  
ssiewert@transformer's password:  
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
Last login: Mon Mar  3 10:25:52 2014 from 137.229.131.34  
ssiewert@transformer:~$
```

Assuming you use Putty, a simple SSH login will look like this:

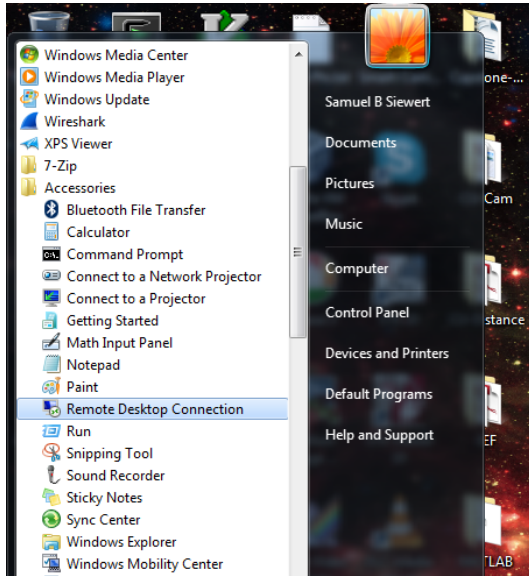




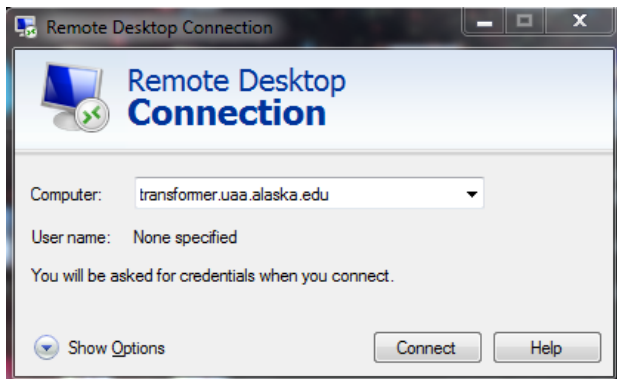
## For a Remote Desktop on Windows

There are two options: 1) Windows Remote Desktop to XRDP (running on transformer) or 2) TightVNC client connection over and SSH tunnel to VNC server running on transformer.

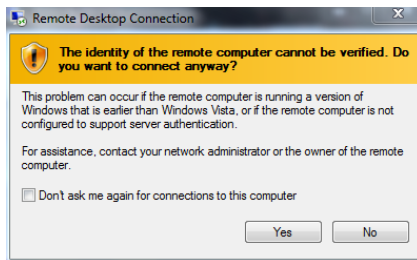
## The simplest option is to use Windows Remote Desktop



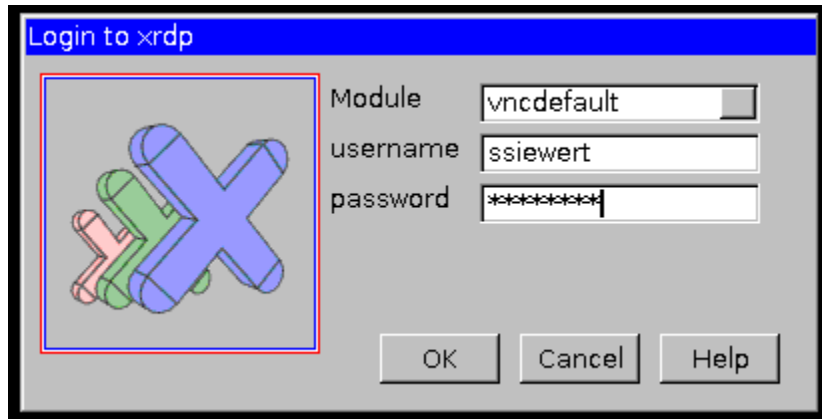
You should be prompted for the remote connection, so enter transformer:



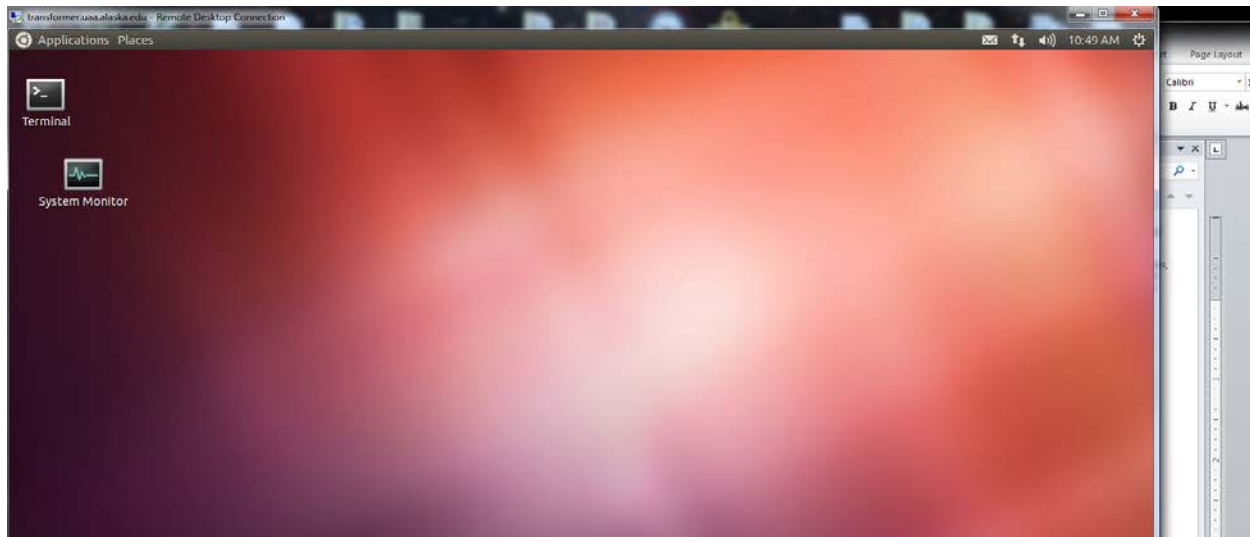
You may be prompted to ok lack of server authentication (just say OK):



Next, you'll be asked to enter login credentials:



After this, you should see a desktop (make sure you have a good configuration in your .vnc directory):



**Copy over ~ssiewert/.vnc**

Edit xstartup:

```
#!/bin/sh
```

```
# Uncomment the following two lines for normal desktop:
```

```
unset SESSION_MANAGER
```

```
#exec /etc/X11/xinit/xinitrc
```

```
gnome-session --session=gnome-classic &
```

```
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
```

```
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources
```

```
xsetroot -solid grey
```

```
vncconfig -iconic &
```

```
#x-terminal-emulator -geometry 1280x1024+10+10 -ls -title "$VNCDESKTOP  
Desktop" &
```

```
#x-window-manager &
```

**On a Windows machine use SSH to setup a tunnel (get help to start tunnel server)**

`ssh -L 5907:localhost:5907 ssiewert@transformer.uaa.alaska.edu`

**Copy over ~ssiewert/.vnc**

Edit xstartup:

```
#!/bin/sh
```

```
# Uncomment the following two lines for normal desktop:
```

```
unset SESSION_MANAGER
```

```
#exec /etc/X11/xinit/xinitrc
```

```
gnome-session --session=gnome-classic &
```

```
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
```

```
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources
```

```
xsetroot -solid grey
```

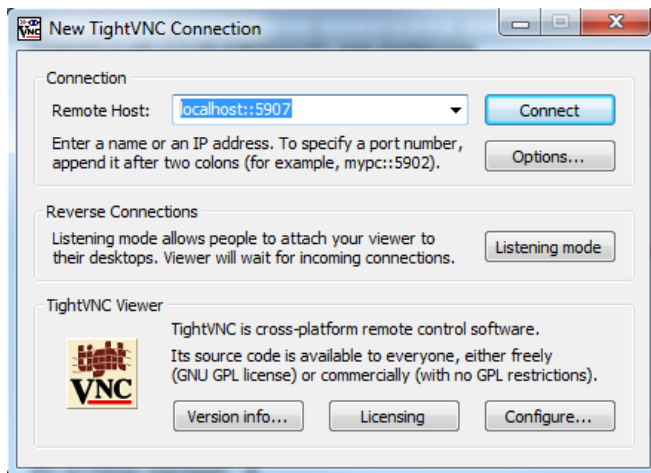
```
vncconfig -iconic &
```

```
#x-terminal-emulator -geometry 1280x1024+10+10 -ls -title "SVNCDESKTOP  
Desktop" &
```

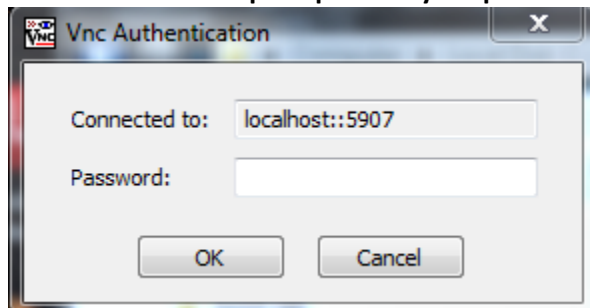
```
#x-window-manager &
```

**Install tightVNC on your Windows machine:** <http://www.tightvnc.com/download.php>

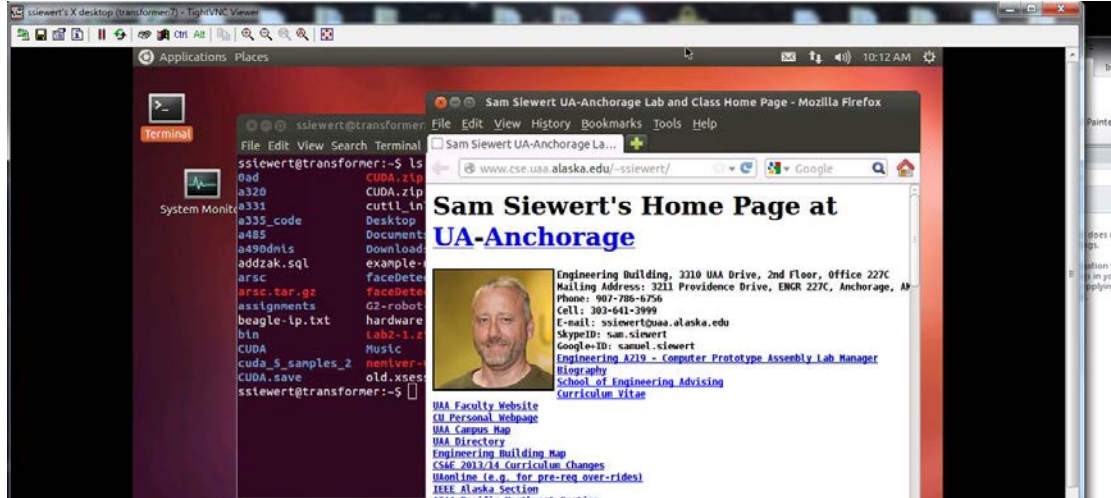
**While you are SSH'd into transformer, run VNC client to localhost::5907 to connect to your desktop.**



**You should then be prompted for your password for the SSH tunnel (enter normal password):**



If all goes well, you should get a remote desktop like this:



### Setting up server to start automatically (/etc/init.d/vncserver)

After adding a new user to /etc/init.d/vncserver and assigning a DISPLAY, do:

**sudo /etc/init.d/vncserver restart**

```
#!/bin/bash
PATH="/usr/bin/"
export USER="ssiewert"
DISPLAY="7"
DEPTH="16"
GEOMETRY="1024x768"
OPTIONS="-depth ${DEPTH} -geometry ${GEOMETRY} :${DISPLAY} -localhost"
. /lib/lsb/init-functions

case "$1" in
start)
log_action_begin_msg "Starting vncserver for user '${USER}' on
localhost:${DISPLAY}"
su ${USER} -c "/usr/bin/vncserver ${OPTIONS}"
;;

stop)
log_action_begin_msg "Stopping vncserver for user '${USER}' on
localhost:${DISPLAY}"
su ${USER} -c "/usr/bin/vncserver -kill :${DISPLAY}"
;;

restart)
$0 stop
$0 start
;;
esac
exit 0
```

## **References:**

1. <http://www.hiddentao.com/archives/2013/09/17/setting-up-tightvnc-on-ubuntu-12-04/>
2. <http://askubuntu.com/questions/130110/tightvncserver-on-ubuntu-12-04-server-with-ubuntu-desktop-installed-no-unity>
3. <http://www.havetheknowhow.com/Configure-the-server/Install-VNC.html>
4. <http://www.liberiangeek.net/2012/05/connect-to-ubuntu-12-04-precise-pangolin-via-windows-remote-desktop/>