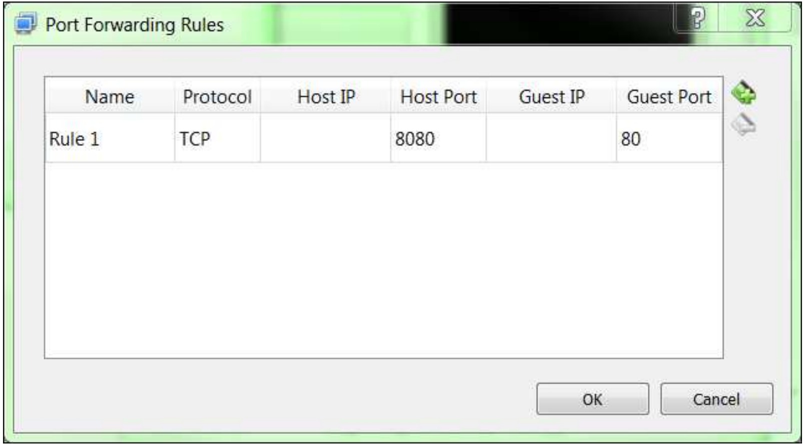
Lab 0 - cs124 Fall 2017

The objective of the lab is to build and submit to Canvas a pdf and tgz file using tools we need to create our program documentation this semester. Some of these steps will be done just once to prepare for all labs and some will be repeated for each lab. This list is a starting point, but you will need more detailed instructions on how to carry out each of the steps. In these instructions “host” computer means your real machine (laptop), “guest” computer is the virtual machine.

Virtual Box

- Copy Debian.qcow from Ohlone's usb drive to your host computer
- Rename Debian.qcow to <FirstintialLastname>.qcow on the host machine
- Copy your virtual machine (the qcow file) to your own USB drive
- Configure Virtual Box to use your qcow file and set port forwarding
  - Setup under "Networks", "Advanced", "Port Forwarding"
- Boot up Virtual Machine



Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
Rule 1	TCP		8080		80

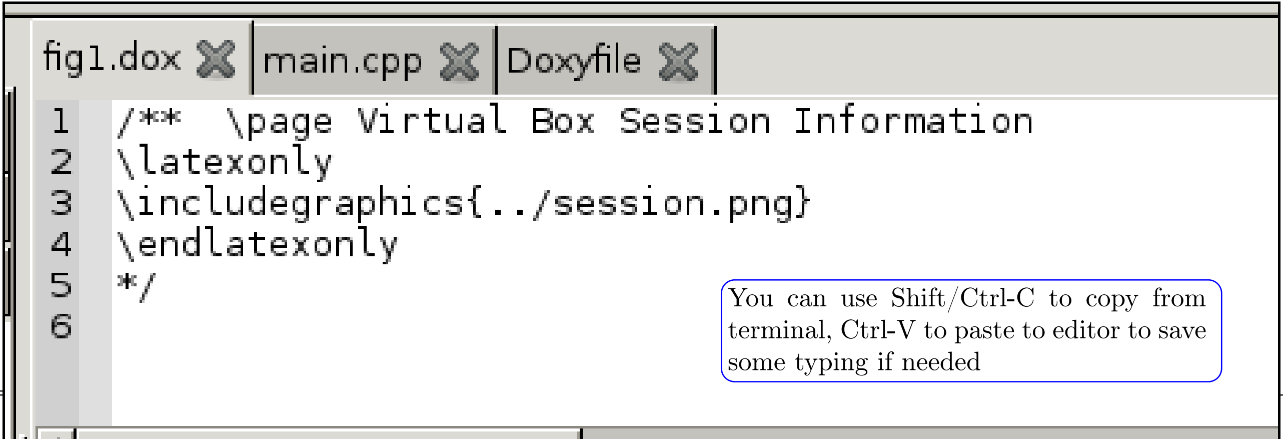
Upload file from host to guest

- Take screenshot of Session Information “Configuration Details” – make sure the Storage info is visible.
- Upload the image you created (png file) to the guest VM
  - from Host browser go to localhost:8080 and find file_upload.pl script in cgi-bin directory

Geany IDE

This is our programmer's editor • Create new directory for lab0 using **mkdir** command

- Change to lab0 directory using **cd** command
- Move the png file to lab0 directory using the **mv** command
- open the IDE program geany
  - geany &
  - Edit a dox file to add an "includegraphics" command for the png image
  - Edit a cpp file to hold your C++ source code



```
/**  \page Virtual Box Session Information
\latexonly
\includegraphics{../session.png}
\endlatexonly
*/

```

You can use Shift/Ctrl-C to copy from terminal, Ctrl-V to paste to editor to save some typing if needed

Doxygen

- Build pdf file from terminal in editor
  - Configuration file
    - doxygen -g
  - Edit Doxyfile
    - PROJECT_NAME = "Lab 0"
    - EXTRACT_ALL = YES
    - INLINE_SOURCES = YES
  - Build Documentation
    - doxygen
    - cd latex && make && cd ..
    - gv latex/refman.pdf &

gv (Ghostview) is a previewer; the pdf will look much clearer when you view it from host browser

Save work

- Compress and archive the lab directory
  - tar zcvf lab0.tgz lab0
- Download tgz and pdf files to host
- Upload tgz and pdf to Canvas, then submit

Programming Process

Programmers go through a sequence of thinking steps to design and build a program. We will create a dox page for each of those steps and write down our thoughts and plans for each one. These are the steps:

- Specification
- Analysis
- Design
- Implementation - many slides – one for each module
- Test

For most of this class, but especially at the beginning, we will do the problem-solving together. This will help us learn what each of these steps mean. The following slides outline the tasks needed to complete this first program. Notice each implementation slide describes a separate module. These are called “compilation units” in `C++` since each is compiled to object code separately, then the object code files are combined to a single executable program file by the linker.

Programming Process

- Specification
  - Description of what program does from users point of view
- Analysis
  - Inputs: List of all data your program will use
  - Process: Description of how outputs be created from inputs
  - Outputs: List of all data your program will display
- Design
  - Decisions about how you will implement the process defined during analysis
- Implementation
  - Write algorithm for expected inputs and outputs ...and describe program state (variables) in documentation comments using Latex when appropriate
  - Write the code in C++
- Test
  - Several testcases to verify that program meets requirements with expected output manually calculated for each and an image taken of program running showing the actual results.