Web Server Programming

CSE351 Computer Networks (Sprint 2018)

Object

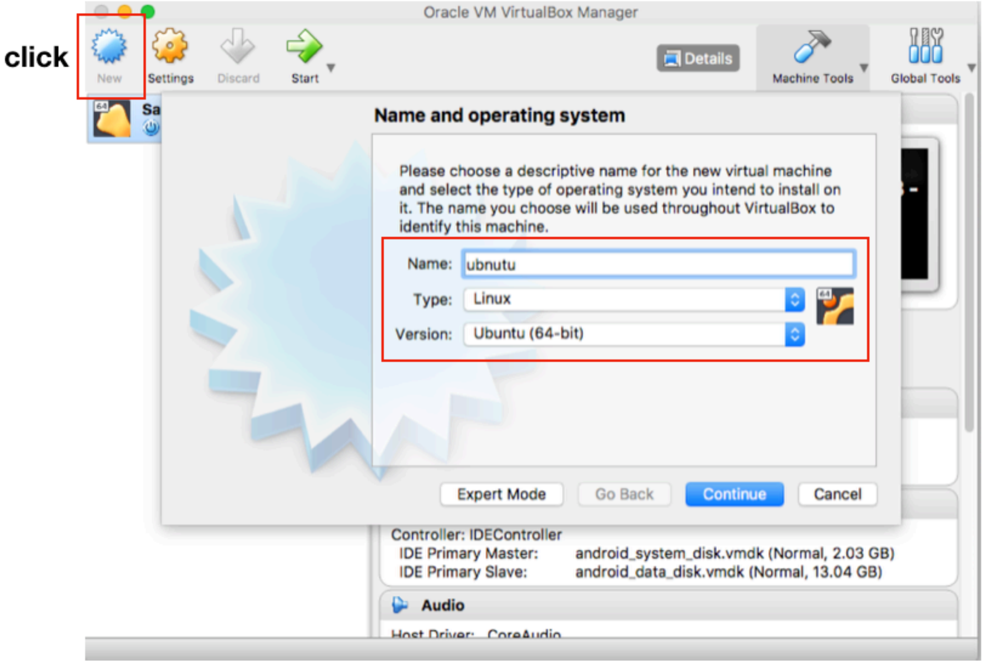
The goal of this project is to implement web server. Everyday, we frequently request information using web browser, and browser shows us proper information. However, browser is not the original source of the information. Who provides the data is web server. In this project, you should implement the server with socket programming. The server will listen to client (web browser of your smart phone), and send a webpage to the client.

Requirements

First of all, you should set up Linux developing environment using VM. So, we made a guideline for the device setting.

1. Download the VM application through below link depending on your OS.
   https://www.virtualbox.org/wiki/Downloads
2. To utilize Linux virtual machine, downloads ISO file of Ubuntu 16.04.3 LTS via this link

   https://www.ubuntu.com/download/desktop
3. Open VirtualBox and click “New” button on navigation bar, and create Ubuntu (follow default setting process)



Name and operating system

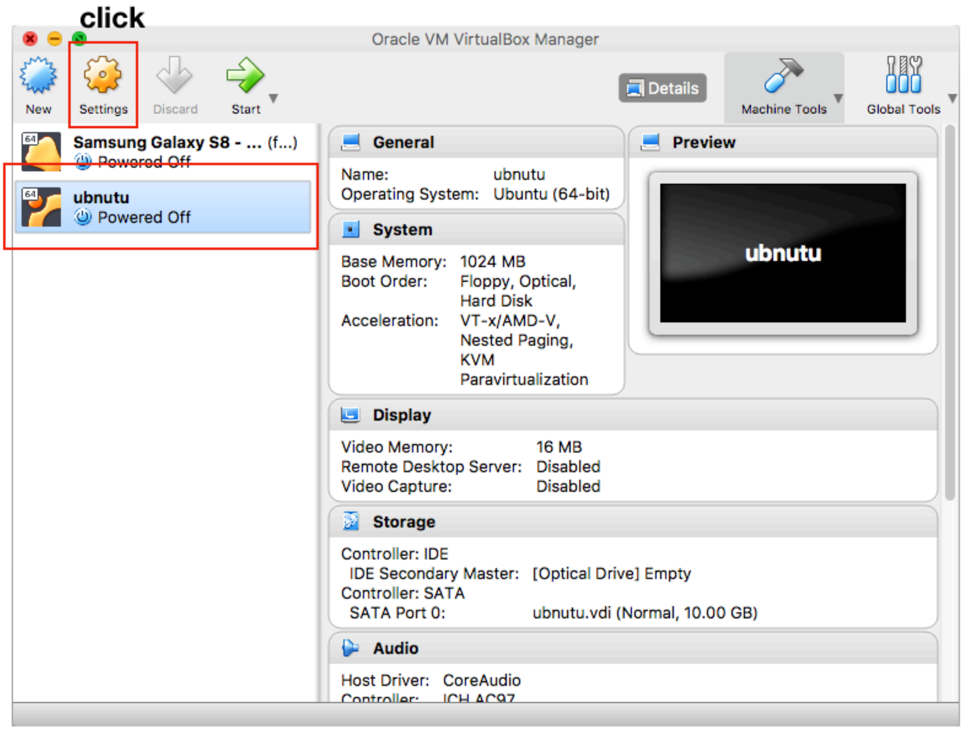
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: ubnutu
Type: Linux
Version: Ubuntu (64-bit)

Controller: IDEController
IDE Primary Master: android_system_disk.vmdk (Normal, 2.03 GB)
IDE Primary Slave: android_data_disk.vmdk (Normal, 13.04 GB)

Host Driver: CoreAudio

4. Setup network of VM (go to network setting tab, and select “Bridge Adapter”)



Samsung Galaxy S8 - ... (f...)
Powered Off

ubnutu
Powered Off

General

Name: ubnutu
Operating System: Ubuntu (64-bit)

Preview

ubnutu

System

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

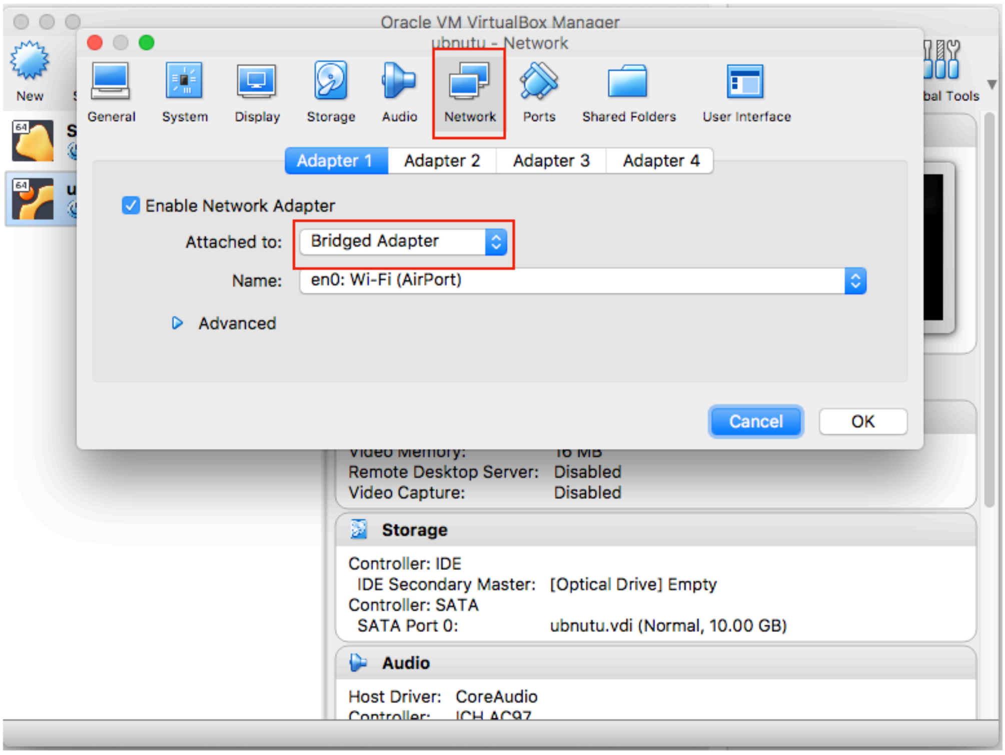
Video Memory: 16 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: ubnutu.vdi (Normal, 10.00 GB)

Audio

Host Driver: CoreAudio
Controller: ICH AC97



☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: en0: Wi-Fi (AirPort)

Advanced

Video Memory: 16 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

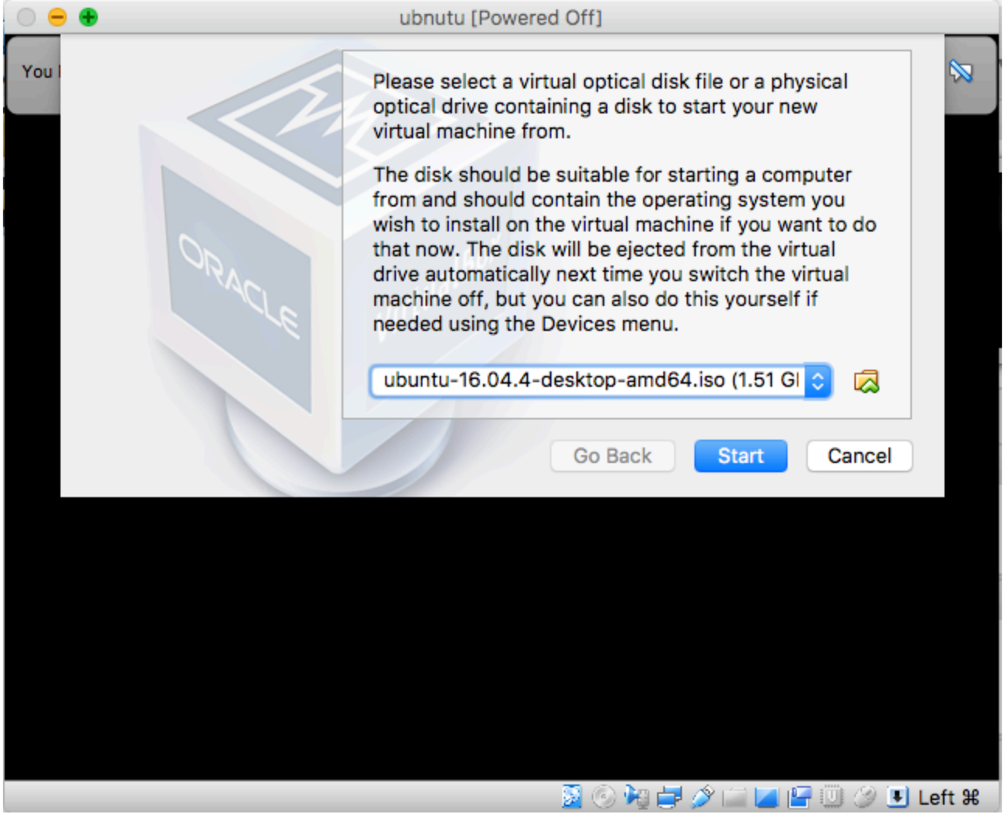
Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: ubnutu.vdi (Normal, 10.00 GB)

Audio

Host Driver: CoreAudio
Controller: ICH AC97

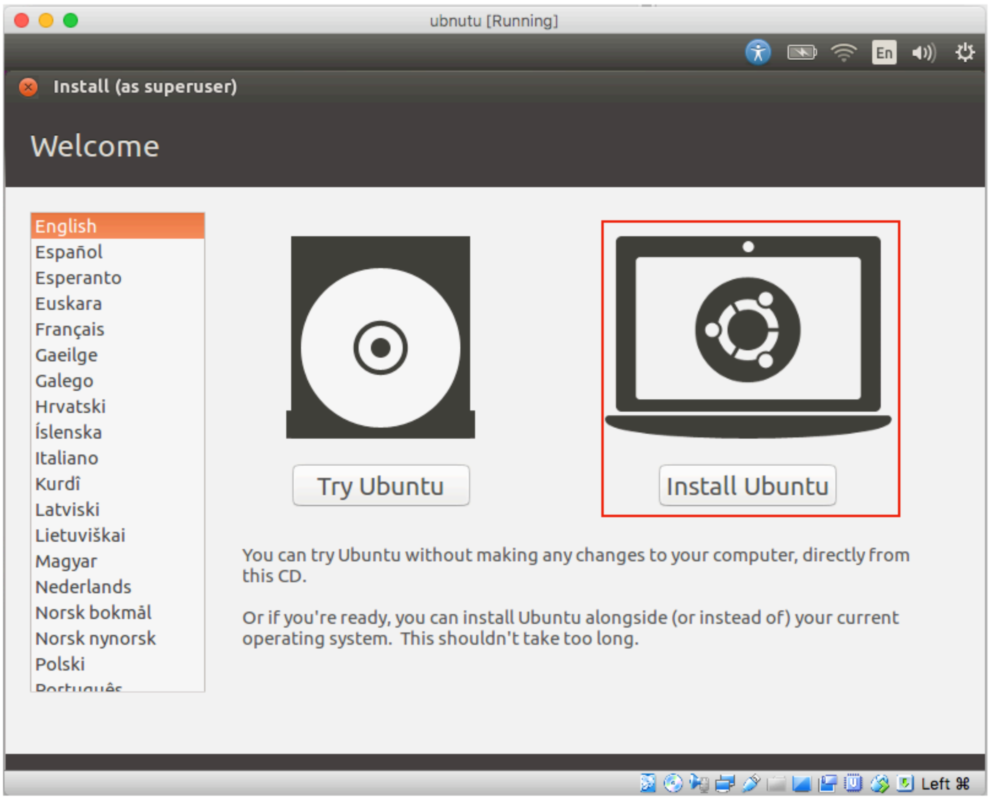
5. Launch Ubuntu, and select ios file which you installed in step 2, then follow unbuntu install process



Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

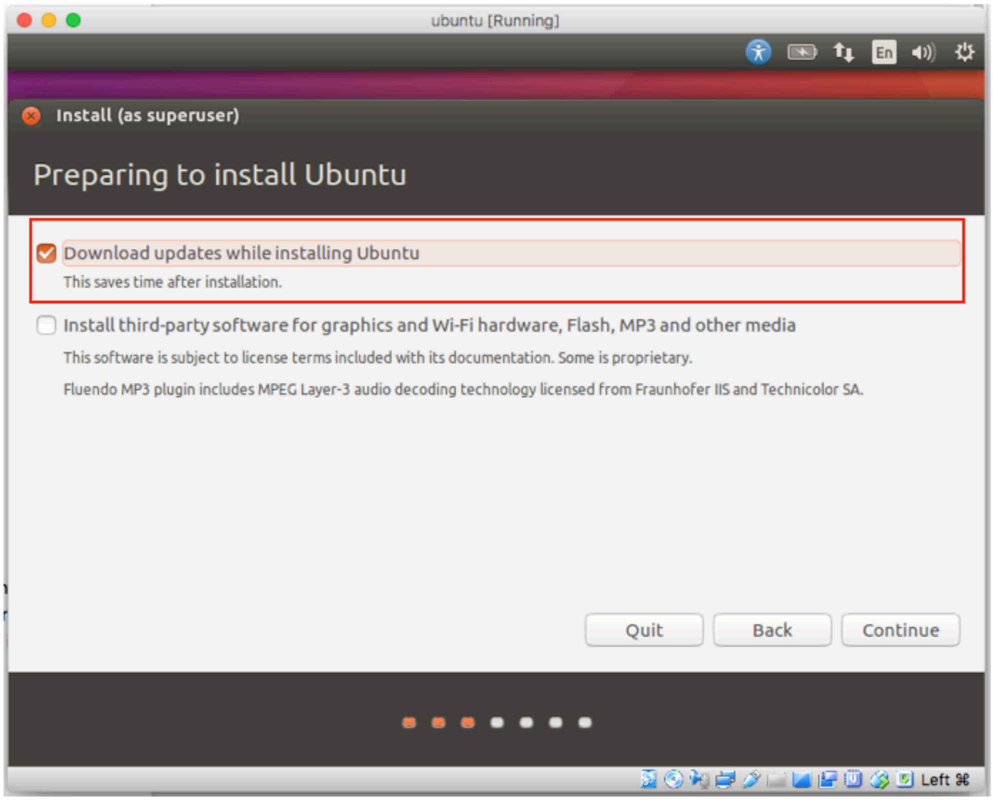
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

ubuntu-16.04.4-desktop-amd64.iso (1.51 GI



You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.



Preparing to install Ubuntu

☑ Download updates while installing Ubuntu

This saves time after installation.

□ Install third-party software for graphics and Wi-Fi hardware, Flash, MP3 and other media

This software is subject to license terms included with its documentation. Some is proprietary.

Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Technicolor SA.

6. After Ubuntu is Installed, open a new terminal and install vim using below command.

   ```
   sudo apt-get install vim
   ```

If you are done with these steps, you are ready to start this project.

Program Specification

Skeleton code is provided in assignment tab of BB. Copy the code to your VM, and modify it to implement your own web server. Your web server should send index.html file to the client in HyperText Transfer Protocol (HTTP).
Check the message format of HTTP (https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

Basically, you should fill up empty parts in skeleton code to run this code successfully. This code is consisted of two functions, main(), respond()

main() function let socket bound to address and port (using bind() function), and listen on the socket (using listen() function). Then, it accept connection from client (using accept() function), and give index.html to the client in HTTP message format. Server should continuously accept other clients after it finishes response.

respond() function reads index.html file, combine it with HTTP header, and send the file to client.

Way of implementation and grading

There are three ways to implement web server.

1. Single Threaded Server

   Single threaded server just
   1) Wait for client request
   2) Process the request
   3) Repeat this cycle.

   All of these steps are executed in single thread

2. Multi Threaded Server

   Different from single-threaded server, multi threaded server
   1) wait for client request
   2) when it accept client connection, create new thread, and wait for another clients
   3) new thread process the request

   => Requests of clients can be processed concurrently

3. Thread Pooled Server

   Web server creates threads very frequently, so it can consume lots of computing resources to create new threads. To solve this problem, thread-pooled server creates threads in advance, wake the thread and give task to the thread whenever it needed.
   https://en.wikipedia.org/wiki/Thread_pool

   The ThreadPool.h is provided in the skeleton code, so you can use this to implement thread-pooled server

http://tutorials.jenkov.com/java-multithreaded-servers/index.html

This link can be helpful to understand these ways

If you implement
Thread Pooled Server : Grade will be 120%

Multi Threaded Server : 100%

Single Threaded Server : 70%

Compile and Usage

File name:

server.cpp

Compile:

```
g++ -std=c++11 -o server server.cpp –lpthread
```

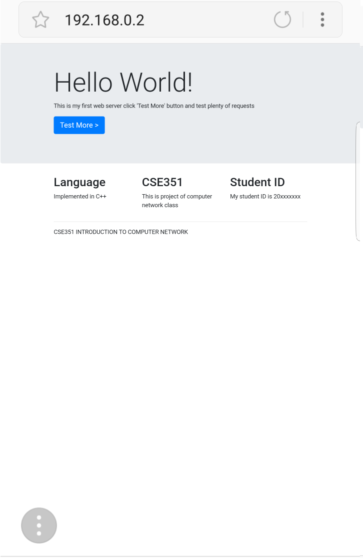
or you can just compile with Makefile

Usage:

1) Check ip address of your VM (type in “ifconfig” in terminal)
2) Compile your code (make)
3) Run server (./server)
4) Open web browser in your smart phone
5) Open link “http://IP:5000” (ex – http://192.168.0.1:5000)

Output Specification

Before you submit code, change index.html file to include your student ID
Click “Test More” button and check bunch of image loading works well.



192.168.0.2

Hello World!

This is my first web server click 'Test More' button and test plenty of requests

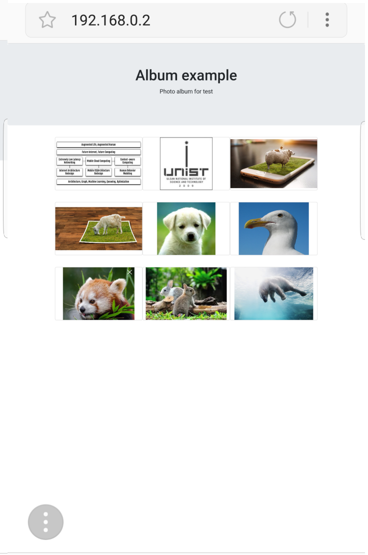
Test More >

Language
Implemented in C++

CSE351
This is project of computer network class

Student ID
My student ID is 20xxxxxxx

CSE351 INTRODUCTION TO COMPUTER NETWORK



192.168.0.2

Album example

Photo album for test

Submission

- ✓ Submit your codes: jongyunlee@unist.ac.kr
- ✓ The email should have the title in the format of “[CSE351] webserver_<name>_<student id>”
  - ➢ You can submit only one file (in *.zip or *.tar.gz only)
  - ➢ Make the directory name “webserver_<student id>”