

# Machine Problem 1 Overview

CSCE 410/611

Robert Baykov

# Course Project

In the course project you will incrementally work to develop and implement a core operating system concepts and features using C++.

The project is divided into several parts, we will refer to these assignments as Machine Problems.

The problems will take roughly two weeks each, they are very challenging.

# Machine Problem 1 out Today

**Due : September 16 Midnight**

In the first machine problem you will setup and test out your development environment.

You are provided a simple “kernel”, which essentially prints a welcome text and goes into an infinite loop.

You are going to modify the text on the welcome message to print out your name in this simple “kernel” source code.

# Development Environment Setup

We will be using VirtualBox to run a fully configured image of Ubuntu Linux with all the development environment tools pre-installed.

The image is 64-bit, so it is required you have processor capable of 64-bit guest os emulation.

# Environment Setup

VirtualBox Download

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

If you have problems on windows, try the test build:

<https://www.virtualbox.org/download/testcase/VirtualBox-4.3.15-95713-Win.exe>

Ubuntu Image Download

<http://faculty.cs.tamu.edu/bettati/Courses/OSProjects/linux.ova>

# VirtualBox Setup

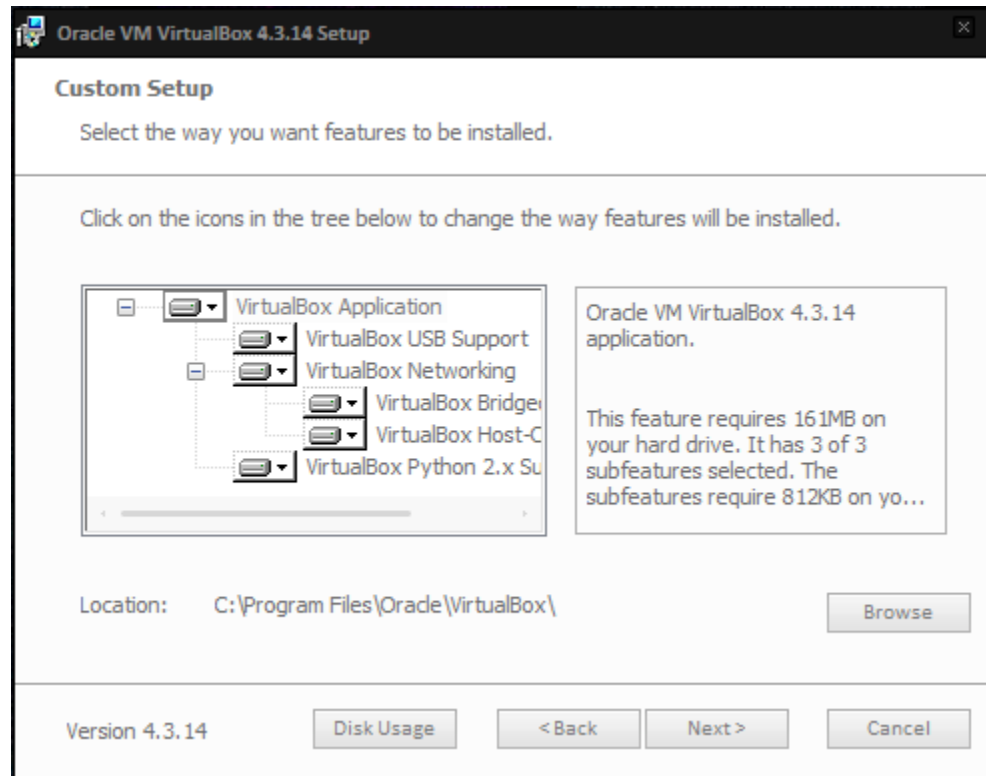
## Download VirtualBox

Here, you will find links to VirtualBox binaries and its source code.

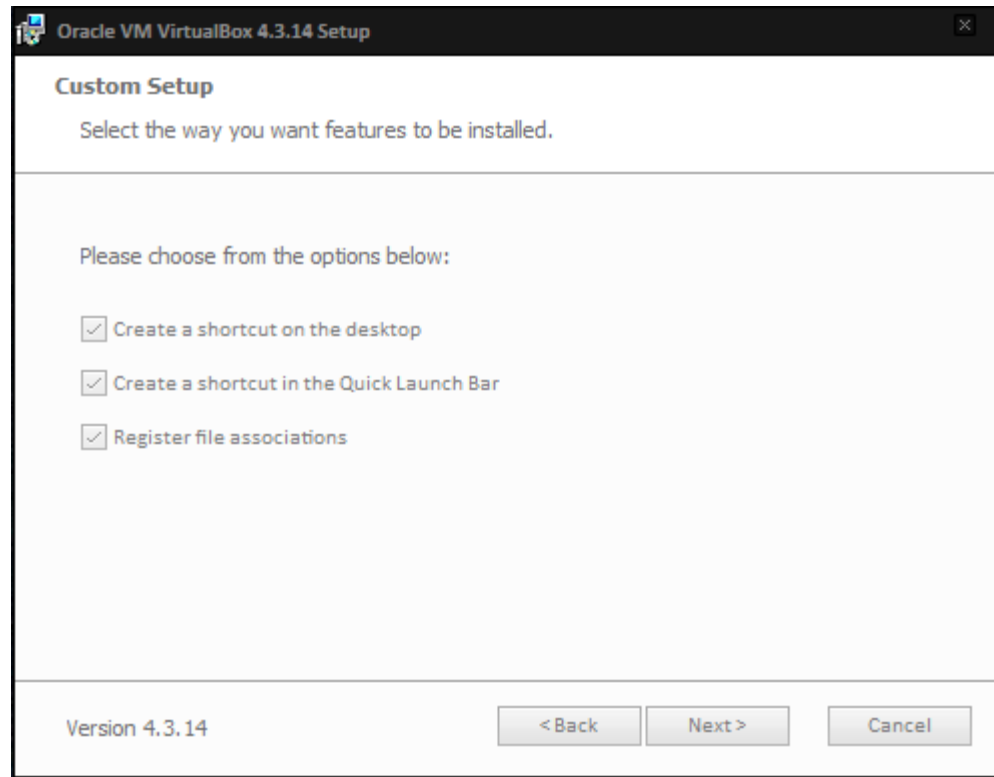
### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox platform packages.** The binaries are released under the terms of the GPL version 2.
  - **VirtualBox 4.3.14 for Windows hosts** [↗ x86/amd64](#)  
If you run into problems with the Windows package, please refer to the [↗ forum](#), it has a link to a build with some fixes and additional information. Please provide a detailed problem description if you think your case isn't covered yet.
  - **VirtualBox 4.3.14 for OS X hosts** [↗ x86/amd64](#)
  - **VirtualBox 4.3.14 for Linux hosts**
  - **VirtualBox 4.3.14 for Solaris hosts** [↗ amd64](#)

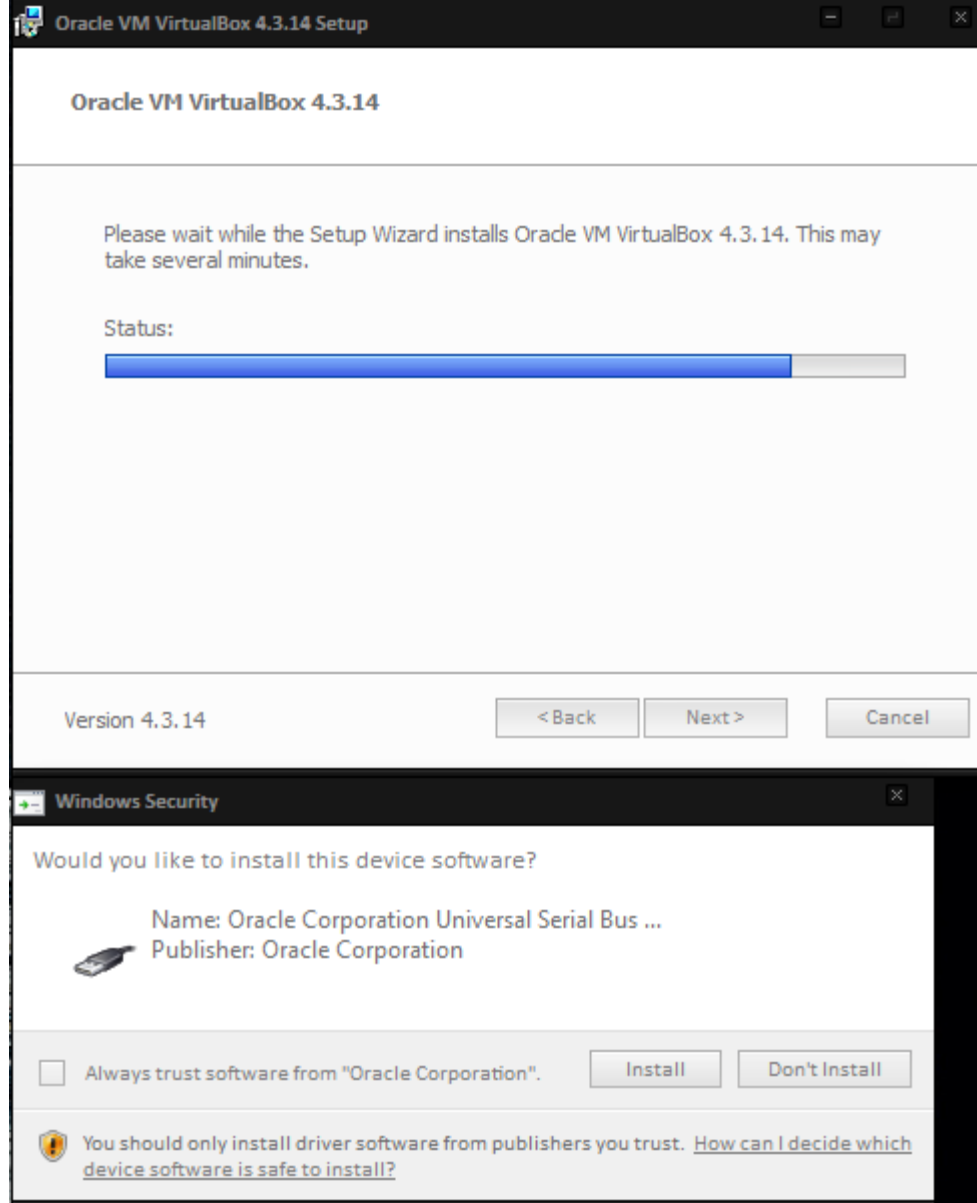


# VirtualBox Setup



# VirtualBox Setup





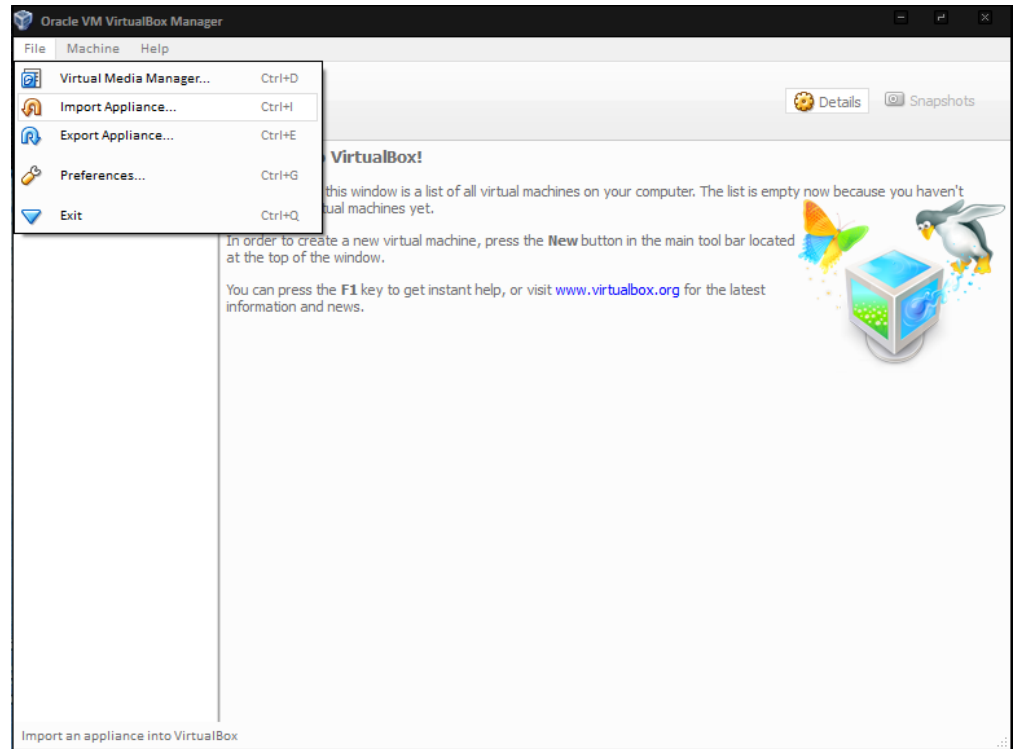
# Install All Default Components

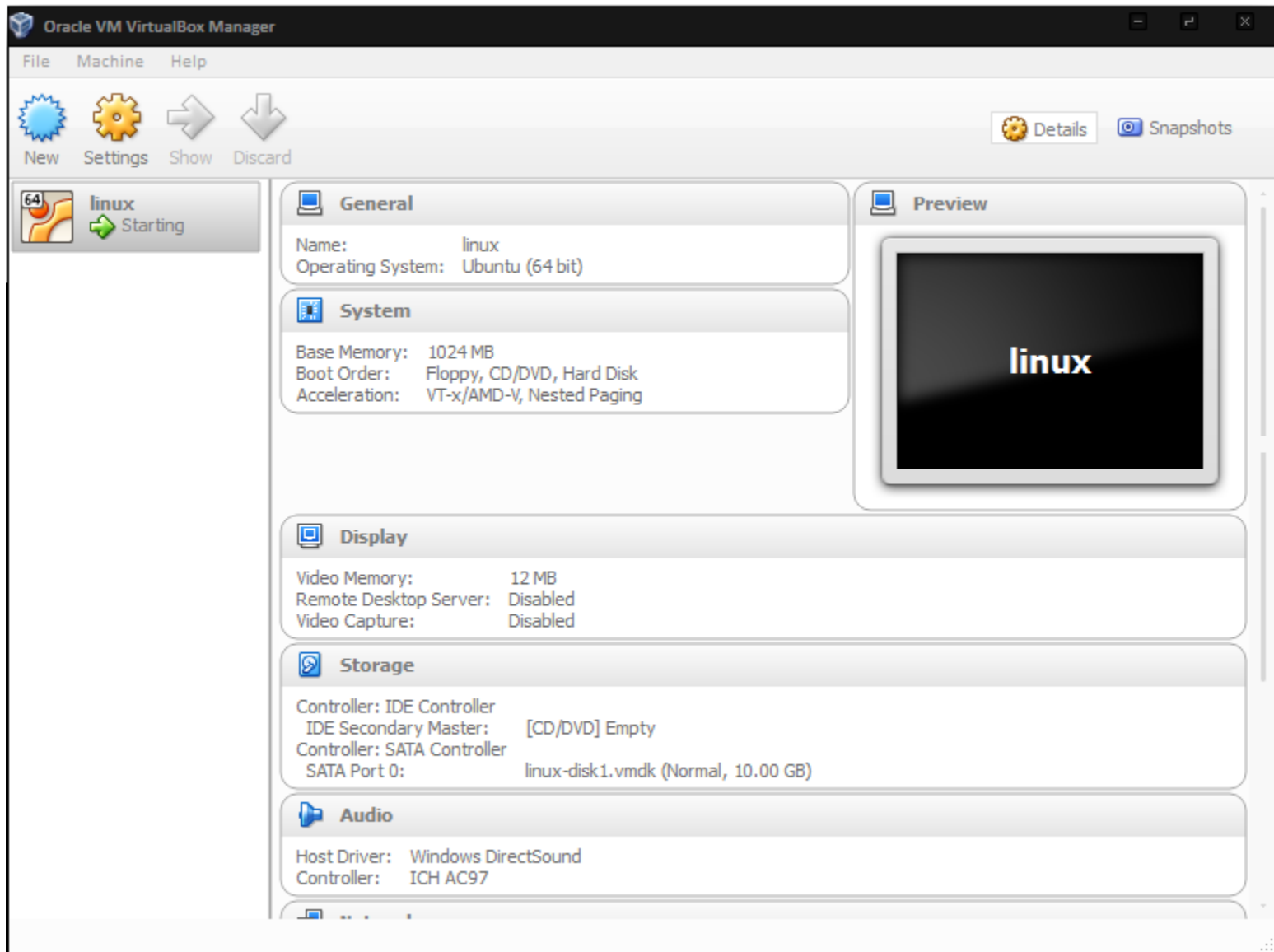
# Importing

File -> Import Appliance

Located the **linux.ova** ubuntu image you downloaded earlier.

Import with the default settings.





Start, Password: guest123

# Setting Up a Shared Folder

- In the window of a running VM, you can select "Shared folders" from the "Devices" menu, or click on the folder icon on the status bar in the bottom right corner.
- If a VM is not currently running, you can configure shared folders in each virtual machine's "Settings" dialog.
- From the command line, you can create shared folders using VBoxManage, as follows:
- `VBoxManage sharedfolder add "VM name" --name "sharename" --hostpath "C:\test"`

## Additional Instructions:

<http://www.virtualbox.org/manual/ch04.html#sharedfolders>

# MP1 Getting Started

Download the supplementary code from the course website to the VirtualBox image, either directly or by adding it to a share (if you made one)

Machine Problem Handout ( u: **courses** p: **leyk**)

<http://courses.cs.tamu.edu/teresa/csce410/pdf/410-14c-P1.pdf>

Supplementary Code

<http://courses.cs.tamu.edu/teresa/csce410/code/410-14c-P1-supp-code.zip>

# MP1 Getting Started

# unzip [410-14c-P1-supp-code.zip](#)

## Contents:



— BIOS-bochs-latest	— bochsrc.bxrc	— console.C	
— console.H	— copykernel.bat	— copykernel.sh	— dev_kernel_grub.img
— <b>kernel.C</b>	— linker.ld	— makefile.linux	— makefile.linux64
— makefile.win	— README.TXT	— start.asm	— utils.C
— utils.H	— VGABIOS-lgpl-latest		

0 directories, 17 files

# Assignment

You are to modify the given “kernel” to print out your name on the welcome screen. For this, you modify the provided file `kernel.C`

You then compile the source to generate the kernel executable `kernel.bin` (using `make` or manually `g++`).

Copy the kernel onto the provided `.img` file and test using Bochs emulator.

For detailed instructions refer to the README file

# Machine Problem 1 Out

**Due : September 16 Midnight**

Hand in on <https://csnet.cs.tamu.edu>

- You are to hand in one file, with name `mp1.zip`, which contains a single file, named `mp1.img`. The latter is the floppy image file that you obtain by copying (replacing) the file `kernel.bin` on the provided `.img` file (shell script included in files)
- Grading of these MPs is a very tedious chore. These handin instructions are meant to mitigate the difficulty of grading, and to ensure that the grader does not overlook any of your efforts.
- Failure to follow the handling instructions will result in lost points.



# Grading Criteria

Code Compiles: 40% (This basically means that you get at least 40% for each machine problem)

Feature Completeness and functional correctness: 20%

Documentation : 10%

Efficiency : 10%

Report 20%

Grand Total: 100%

# Office Hours

MW 12:00-02:00

HRBB 219 (Open Access Lab) :

Or by appointment HRBB 502A (Office)

Questions?