

Getting Started

- Virtual machine with all tools installed, available at:

- 4 GB!

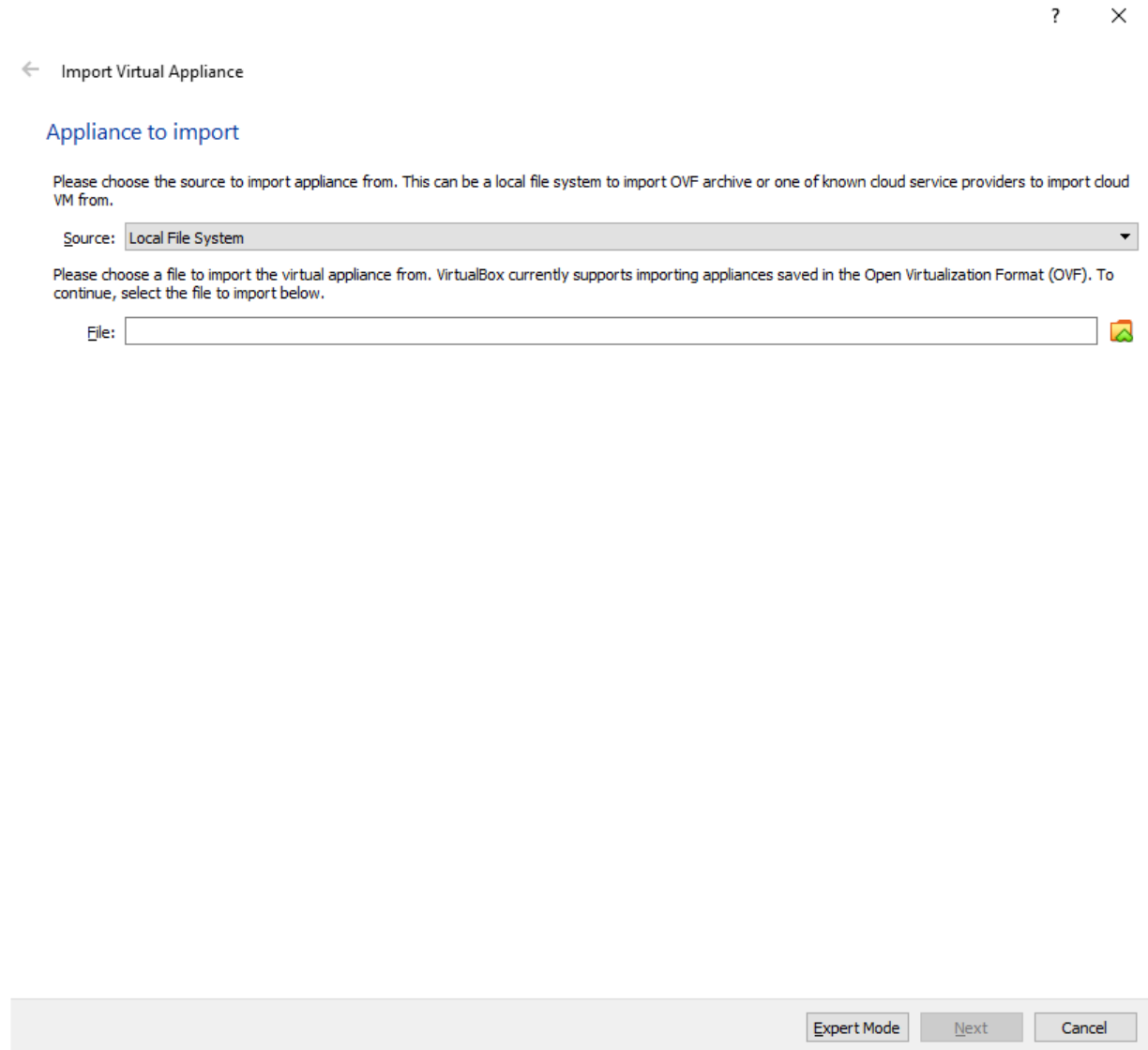
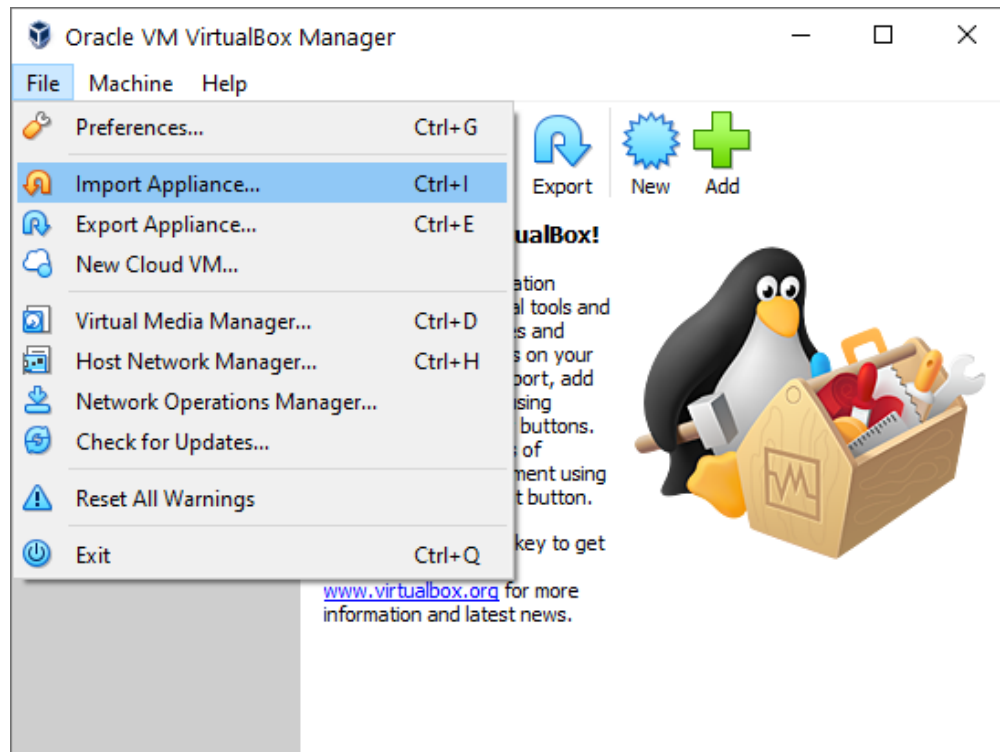
https://drive.google.com/file/d/1pIT9o1QleDkci0l_jB4Si9BTwmqmeIgF/view?usp=drive_link

- First, install Oracle Virtualbox

- Open-source virtual machine
- High performance with minimal configuration

Getting Started

- Import the downloaded VM



Getting started

← Import Virtual Appliance

Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Virtual System 1	
Name	cs152-ubuntu
Guest OS Type	Ubuntu (64-bit)
CPU	4
RAM	2048 MB
DVD	<input checked="" type="checkbox"/>
USB Controller	<input checked="" type="checkbox"/>
Sound Card	<input checked="" type="checkbox"/> ICH AC97
Network Adapter	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4
Storage Controller (IDE)	PIIX4
Storage Controller (SATA)	AHCI
Virtual Disk Image	cs152-ubuntu-disk001.vmdk
Base Folder	C:\Users\██████\VirtualBox VMs
Primary Group	/

Machine Base Folder: C:\Users\██████\VirtualBox VMs

MAC Address Policy: Include only NAT network adapter MAC addresses

Additional Options: Import hard drives as VDI

Appliance is not signed

Restore Defaults Import Cancel

Change core/memory assignment if necessary

Oracle VM VirtualBox Manager

File Machine Help

Tools

New Settings Discard **Start** Start

cs152-ubuntu Powered Off

General

Name: cs152-ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Processors: 4
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

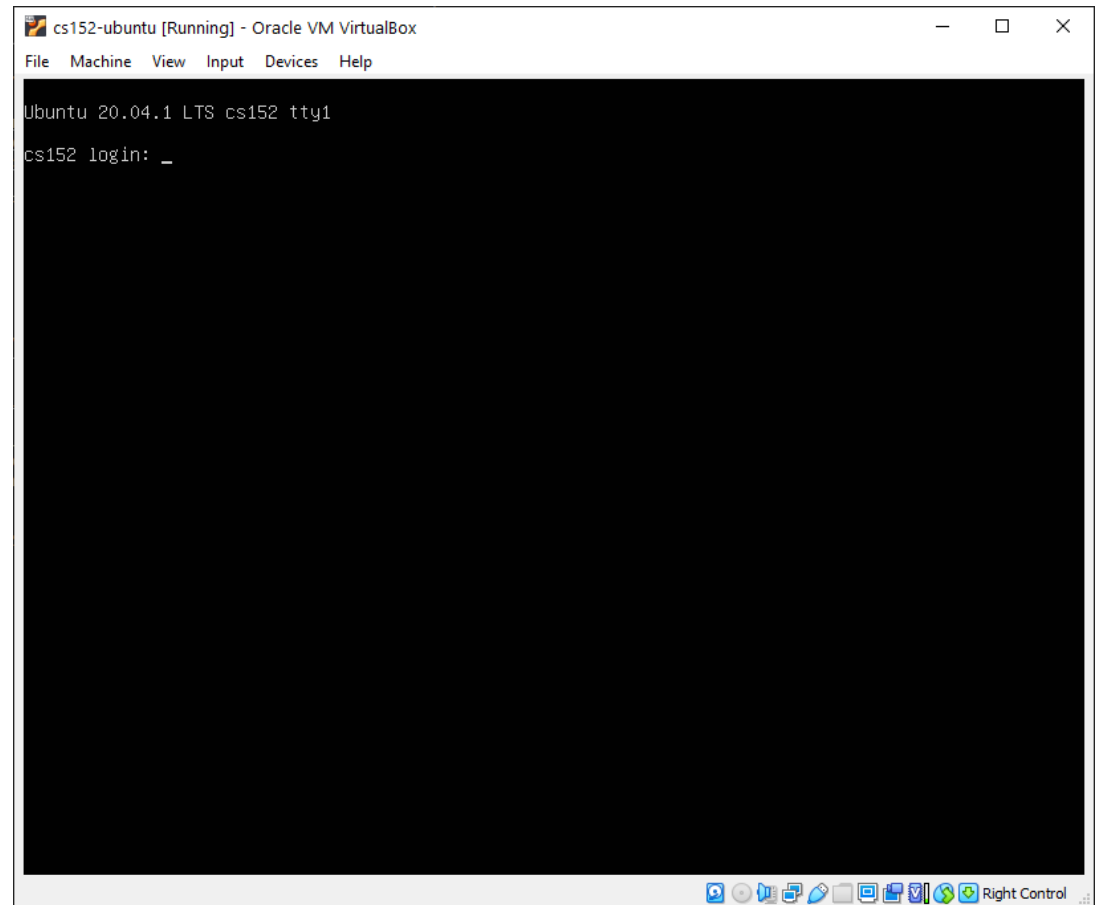
Preview

cs152-ubuntu

Getting started

- You can work in the VM window, OR
- Connect to it via a terminal
 - Putty, MobaXterm, OpenSSH, etc
- The VM forwards its
 - port 22 (ssh) to
 - 3022
 - Connect to it by ssh [cs152@127.0.0.1:3022](ssh://cs152@127.0.0.1:3022)
- Login: cs152/cs152
- Git clone https://github.com/sangwoojun/ulx3s_bsv.git

Check it out!



Trying simulation

- Ulx3s_bsv/projects/rv32i/
- Compiling and running the simulation
 - “make bsim” – Stands for “bluesim”
 - “make runsim” creates two files
 - system.log : log of processor operation
 - output.log : log of software output
- Default benchmark: Sudoku solver
 - Source: sw/minisudoku.c
 - Resulting assembly: sw/minisudoku.dump
 - Binary for processor: sw/minisudoku.bin

```
155 0000023c <solve>:  
156 23c:→ fd010113      → addi→  sp, sp, -48  
157 240:→ 02112623      → sw→  ra, 44(sp)  
158 244:→ 02812423      → sw→  s0, 40(sp)  
159 248:→ 03010413      → addi→  s0, sp, 48  
160 24c:→ fca42e23      → sw→  a0, -36(s0)  
161 250:→ fcb42c23      → sw→  a1, -40(s0)  
162 254:→ fd842703      → lw→  a4, -40(s0)  
163 258:→ 00f00793      → addi→  a5, zero, 15  
164 25c:→ 00e7d663      → bge→a5, a4, 268 <solve+0x2c>
```

Example simulation execution

system.log

```
1 [0x00000000:0x0000] Fetching instruction count 0x0000
2 sent all data 4116
3 Processor starting
4 [0x000020d2:0x0000] decoding 0x00002137
5 [0x000020d3:0x0000] Executing
6 [0x000020d4:0x0000] Writeback writing 00002000 to 2
7 [0x000020d5:0x0004] Fetching instruction count 0x0001
8 [0x000020d9:0x0004] decoding 0x33c000ef
9 [0x000020da:0x0004] Executing

⋮

69943 [0x00021302:0x0498] Writeback writing 0000049c to 0
69944 [0x00021303:0x0008] Fetching instruction count 0x40d4
69945 [0x00021307:0x0008] decoding 0x00000000
69946 [0x00021308:0x0008] Executing
69947 Reached unsupported instruction
69948 Total Clock Cycles = 135944
69949 Total Instruction Count = 16596
69950 Dumping the state of the processor
69951 pc = 0x00000008
69952 Quitting simulation.
```

Cycle

PC

output.log

```
1 0304
2 0020
3 4030
4 0002
5
6 2314
7 1423
8 4231
9 3142
```

Question

Solution

Performance numbers!

IPC = $16,596 / 135,944 \approx 0.122$

Trying synthesis

- Synthesis to hardware
 - “make | tee build.log”
 - Log file is long!
- Example log files from synthesis:
 - Look for “Device utilisation” [sic]:

```
Info: Device utilisation:  
Info: →      TRELIS_SLICE:  4982/41820    11%
```

- Look for “Max frequency” :

```
Info: Max frequency for clock '$glbnet$CLK_clk_25mhz$TRELIS_IO_IN': 69.80 MHz (PASS at 25.00 MHz)
```

- Look for “Critical path report for clock”:

```
Info: Critical path report for clock '$glbnet$CLK_clk_25mhz$TRELIS_IO_IN' (posedge -> posedge):  
Info: curr total  
Info: 0.5 0.5 Source main_proc.imemRespQ.data0_reg_TRELIS_FF_Q_30_DI_PFUMX_Z_SLICE.Q0  
Info: 1.5 2.0 Net main_proc.imemRespQ_D_OUT[1] budget 5.041000 ns (33,27) -> (33,28)
```

Measuring performance

- From the simulation, we can measure the clock cycles to completion
- From synthesis, we can measure the clock speed
- $(\text{cycle count}) / (\text{clock frequency}) = \text{time to completion!}$

- In our previous example, $135,944 \text{ cycles} / 69.80 \text{ MHz} = 0.0019\text{s}$
 - Is this good?
 - We can do MUCH better!