UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

University of Verona

A.A 2017/2018

Laboratory of Networked Embedded Systems

Lesson 2

Open Virtual Platform (OVP)

Valentina Ceoletta, Enrico Fraccaroli

May 2, 2018

Contents

1 Download & Install OVP **2**
- 1.1 Setup Open Virtual Platform (OVP) . . . 2
  - 1.1.1 Download Requried Material . . . 2
  - 1.1.2 Install the packages . . . 3

2 Setup Environment Variables **5**
- 2.1 Setting up the Linux Environment . . . 5
  - 2.1.1 32-bit Product on 64-bit Host . . . 5
  - 2.1.2 Environment . . . 5
  - 2.1.3 Specify the Simulator . . . 6
  - 2.1.4 Setup the License . . . 6

3 Run the Malta Platform **9**
- 3.1 Run a simple example . . . 9
- 3.2 Run the Malta Platform . . . 9

4 Extra **10**
- 4.1 Useful Linux Commands . . . 10
  - 4.1.1 file . . . 10
  - 4.1.2 ldd . . . 10
  - 4.1.3 make --jobs . . . 11

Chapter 1

Download & Install OVP

1.1 Setup OVP

1.1.1 Download Requried Material

Virtual Machine

If you have an Apple machine or a 64-bit machine, you must create a Linux 32-bit virtual machine to host the simulator.

Download OVP

First, register into OVP forum at:

```
http://www.ovpworld.org/forum/profile.php?mode=register
```

After the activation of you registration, login into your OVP account at:

```
http://www.ovpworld.org/forum/login.php
```

Go to the download page of OVP

```
http://www.ovpworld.org/dlp
```

Under `OVP Versions available`, select the version of OVP you have the license for:

- 20161005.0
- 20170201.0
- 20170511.0
- 20170919.0

- 20180221.0

From now on, **you** have to replace the tags `[VER]` with your version of OVP.

Download Simulator

Search the keyword `OVPsim Fast Simulator` and download `OVPsim`, which will start the download of the file:

```
OVPsim.[VER].Linux32.exe
```

Download MIPS models

Search the keyword `MIPS Models` and download `mips32.model`, which will start the download of the file:

```
mips32.model.[VER].Linux32.exe
```

Download MIPS toolchain

Search the keyword `MIPS Toolchains` and download `mips-img.toolchain`, which will start the download of the file:

```
mips-img.toolchain.[VER].Linux32.exe
```

Download MIPS Example: Malta Platform

Search the keyword `MIPS Examples` and download `Demo_Linux_MipsMalta`, which will start the download of the file:

```
Demo_Linux_MipsMalta.[VER].Linux32.exe
```

1.1.2 Install the packages

In order to use the downloaded file you first have to set the execution permissions:

```
chmod +x OVPsim.[VER].Linux32.exe
chmod +x mips32.model.[VER].Linux32.exe
chmod +x mips-img.toolchain.[VER].Linux32.exe
chmod +x Demo_Linux_MipsMalta.[VER].Linux32.exe
```

Install the simulator

Execute as sudoer the installation file:

```
sudo ./OVPsim.[VER].Linux32.exe
```

when prompted where the tool should be installed, say `no` and provide as installation directory:

```
/opt/Imperas.[VER]
```

Install MIPS models

Execute as sudoer the installation file:

```
sudo ./mips32.model.[VER].Linux32.exe
```

remember to provide the correct installation directory.

Install MIPS toolchain

Execute as sudoer the installation file:

```
sudo ./mips-img.toolchain.[VER].Linux32.exe
```

remember to provide the correct installation directory.

Install MIPS Example: Malta Platform

Execute as sudoer the installation file:

```
sudo ./Demo_Linux_MipsMalta.[VER].Linux32.exe
```

remember to provide the correct installation directory.

Chapter 2

Setup Environment Variables

2.1 Setting up the Linux Environment

A script is provided to install a function to setup the environment in which to run the Imperas tools. The script is provided in the bin directory below the Imperas installation directory. This script should be sourced in a shell to provide the function ***setupImperas***, which is then executed passing the ***full*** path to the Imperas directory as the argument.

```
source [installDir]/Imperas.[VER]/bin/setup.sh
```

```
setupImperas [installDir]/Imperas.[VER]
```

2.1.1 Environment

The following environment variables are used by the OVP simulator or Imperas professional tools:

IMPERAS_HOME	Points to the root of the tools installation
IMPERAS_UNAME	Is set to the Host OS type, Linux
IMPERAS_ARCH	Is set to the Host architecture, Linux32
IMPERAS_SHRSUF	Is set to the suffix for shared libraries, so
IMPERAS_VLNV	Points to the root of the compiled library in the Imperas installation.
IMPERAS_RUNTIME	Specifies which simulator, Imperas (CpuManager) or OVPsim, to load at runtime.

The PATH should include `$IMPERAS_HOME/bin/$IMPERAS_ARCH`

The `LD_LIBRARY_PATH` should include `$IMPERAS_HOME/bin/$IMPERAS_ARCH` and `$IMPERAS_HOME/lib/$IMPERAS_ARCH/External/lib`

These are all the variables which are automatically exported by `setupImperas`.

2.1.2 Specify the Simulator

Export `OVPsim` as simulator

```
export IMPERAS_RUNTIME=OVPsim
```

2.1.3 Setup the License

The OVP and Imperas products are licensed using FLEXlm. In order to execute the tools you will need a license file. A license file is bound to an individual computer through that computer's host ID. To obtain a license file you must provide the host ID and host name of the machine that will execute the simulator.

Obtaining the computer hostname and Host ID

To determine the host ID of a computer use the `lmhostid` command of the `lmutil` utility program (lmutil on Linux or lmutil.exe on Windows). If you have already gone through the installation process then the lmutil utility will be in the directory `$IMPERAS_HOME/bin/$IMPERAS_ARCH` which will already be on your executable path. From a Linux shell or a Windows Command Prompt/MSYS window do the following:

```
$ lmutil lmhostid
 - Copyright (c) 1989-2006 Macrovision Europe Ltd. and/or Macrovision
ion. All Rights Reserved.
lm host ID of this machine is "002486571114"
```

The host name of the computer can be obtained with the hostname command. From a Linux shell or a Windows Command Prompt/MSYS window do the following:

```
hostname
```

OVPsim FLEXlm License keys

OVPsim FLEXlm license keys may be requested directly from `http://www.ovpworld.org/licensekey.php` by registered OVP users.

Update license information

Open the license file contained inside the OVP installation dir:

```
sudo gedit /opt/Imperas.[VER]/OVPsim.lic
```

Copy your license information into the file.

NOTE about license problems

You may see an error such as:

```
Invalid host.
The hostid of this system does not match the hostid
specified in the license file.
Feature: IMP_OVPSIM_20100528.0
Hostid: 112233445566
License path: /home/Imperas.20130630/OVPsim.lic
FLEXnet Licensing error:-9,57. System Error: 19 "(null)"
For further information, refer to the FLEXnet Licensing End User Guide,
available at "www.macrovision.com".
```

or

```
$ lmutil lmhostid
lmhostid - Copyright (c) 1989-2006 Macrovision Europe Ltd. and/or Macrovision
Corporation. All Rights Reserved.
The FLEXlm host ID of this machine is "000000000000"
```

This problem occurs when the FlexLM license system checks the `hostid` in the license file against the `hostid` of the license server. If the network interface is enabled on a different device with respect to `eth0`, the FlexLM license system does not read a valid MAC address and a `NULL` host Id is reported.

The following URL explains the full procedure:
`http://unix.stackexchange.com/questions/81834/how-can-i-change-the-default-ens33-network-device-to-old-eth0-on-fedora-19`

The minimum set of stages that were required when performed are:

1. Edit `/etc/default/grub`. Note you need root privileges to edit this file.
2. At the end of `GRUB_CMDLINE_LINUX` line append `net.ifnames=0 biosdevname=0`
3. Save the file
4. type the command `grub2-mkconfig -o /boot/grub2/grub.cfg`
5. type the command `reboot`

Chapter 3

Run the Malta Platform

3.1 Run a simple example

To verify the installation after an OVPsim or an Imperas install one of the Demos that are downloaded in both installations can be executed. These Demos will work with both the OVPsim and the Imperas Simulators.

For example, execute the Fibonacci benchmark on a single core MIPS 74Kf platform.
Change to the directory (the following assumes a Linux or MSYS shell is used).

```
cd ${IMPERAS_HOME}/Demo/Processors/MIPS/Classic/74Kf/single_core
```

```
./Run_Fibonacci.sh
```

3.2 Run the Malta Platform

In order to use the Malta files you need to copy the folder containing the platform inside your home:

```
cp -rvf /opt/Imperas.[VER]/Demo/Platforms/Linux_MipsMalta ${HOME}/
```

Move inside the copied directory:

```
cd ${HOME}/Linux_MipsMalta
```

To simply simulate the platform, move inside the `harness` directory:

```
cd harness
```

Execute the script which will start the simulation

```
./RUN_MIPSMalta.sh
```

Chapter 4

Extra

4.1 Useful Linux Commands

4.1.1 file

Move inside the lib-linux directory inside the SystemC root directory:

```
cd ${SYSTEMC_HOME}/lib-linux
```

Then execute the file command on the library

```
file libsystemc-2.3.1.so
```

this should output something like this:

```
libsystemc-2.3.1.so:
    ELF 32-bit LSB shared object,
    Intel 80386, version 1 (GNU/Linux),
    dynamically linked,
    BuildID[sha1]=...,
    not stripped
```

As you can see you have just cross-compile the SystemC library to 32bit.

4.1.2 ldd

This command allows you to see the shared object dependencies. `ldd` prints the shared objects (shared libraries) required by each program or shared object specified on the command line. An example of its use and output is the following:

```
ldd /bin/ls
    linux-vdso.so.1                          (0x00007ffcc3563000)
    libselinux.so.1 => /lib64/libselinux.so.1 (0x00007f87e5459000)
    libcap.so.2     => /lib64/libcap.so.2     (0x00007f87e5254000)
    libc.so.6       => /lib64/libc.so.6       (0x00007f87e4e92000)
    libpcre.so.1    => /lib64/libpcre.so.1    (0x00007f87e4c22000)
```

```
    libdl.so.2      => /lib64/libdl.so.2      (0x00007f87e4a1e000)
    libattr.so.1    => /lib64/libattr.so.1    (0x00007f87e4817000)
    libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f87e45fa000)
    /lib64/ld-linux-x86-64.so.2               (0x00005574bf12e000)
```

Be aware that in some circumstances (*e.g.*, where the program specifies an ELF interpreter other than ld-linux.so), some versions of `ldd` may attempt to obtain the dependency information **by attempting to directly execute the program** (which may lead to the execution of whatever code is defined in the program's ELF interpreter, and perhaps to execution of the program itself).

Thus, you should never employ ldd on an untrusted executable, since this may result in the execution of arbitrary code.

A safer alternative when dealing with untrusted executables is:

```
objdump -p /bin/ls | grep NEEDED
    NEEDED    libselinux.so.1
    NEEDED    libc.so.6
```

4.1.3 make --jobs

The `-j` or `-jobs` option tells make to execute many recipes simultaneously. If the `-j` option is followed by an `integer`, this is the number of recipes to execute at once; this is called the number of job slots. If there is nothing looking like an integer after the `-j` option, *there is no limit on the number of job slots* (*i.e.*, **BE CAREFUL!**). The default number of job slots is one, which means serial execution (one thing at a time).

That's all folks