Implementing Linux on the

Zynq™-7000 SoC

Lab 1.2

Building U-Boot from Source Code

September 2012
Version 05

Table of Contents

Lab 1.2 Overview

The Xilinx Linux project combines the benefit of open source Linux operating system together with a customized solution geared towards developing software on its processing platform.

U-Boot is mostly used to load and boot a kernel image, but it also allows developers to change the kernel image and the root file system stored in flash.

Files can be exchanged between the target and the development workstation. This is possible through a few different methods of transfer:

- Via the network if the target has an Ethernet connection (the TFTP protocol can be used to exchange files)

- Through the serial line using kermit.

- Using an SD card formatted with a FAT32 file system.

Das U-Boot is a GPL cross-platform boot loader pioneered by project leader Wolfgang Denk and backed by an active developer and user community. U-Boot provides out-of-the-box support for hundreds of embedded boards and a wide variety of CPU architectures including ARM. U-Boot is an ideal second stage boot loader solution for Zynq. U-Boot can be easily configured to strike the right balance between a rich feature set and a small binary footprint.

It can also help to accelerate your next embedded Linux development project by using U-Boot to take care of some low-level board initializations and kernel loading, allowing you to focus on the core of your embedded application. Also, if the need arises, you have the source code and thus the potential to add support for new hardware or add a special feature.

Lab 1.2 Objectives

When you have completed Lab 1.2, you will know how to do the following:

- Retrieve U-Boot source code from Xilinx repository
- Configure U-Boot for ZedBoard target
- Build U-Boot for ZedBoard target

Experiment 1: Clone the Xilinx U-Boot Git Repository

This experiment shows how to clone the Xilinx U-Boot Git repository for Zynq. To successfully complete this lab, you will need Internet access to retrieve the repository information from the Xilinx website.

Experiment 1 General Instruction:

Clone the Xilinx U-Boot Git repository for Zynq.

Important Note: If performing this lab on an Avnet SpeedWay training laptop, the appropriate Git repository has already been cloned for you and Experiment 1 is for reference only.

If the U-Boot source already exists in the **/home/training/u-boot-xlnx/** folder on the Virtual Machine, Experiment 1 can be skipped and Experiment 2 can be started.

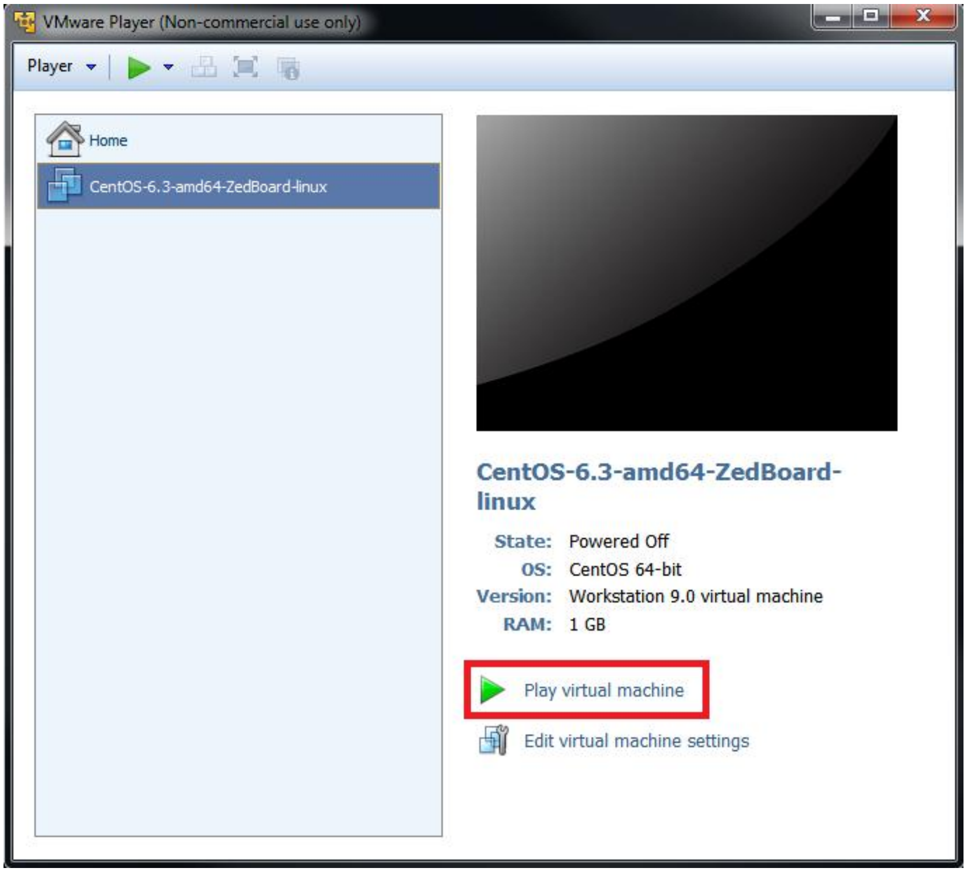
Experiment 1 Step-by-Step Instructions:

1. If the CentOS virtual machine is not already open, launch the VMware Player application from by selecting **Start → All Programs → VMware → VMware Player**. If the CentOS virtual machine is already open, skip ahead to Step 4.

Figure 1 – The VMware Player Application Icon

2. Select the virtual machine named **CentOS-6.3-amd64-ZedBoard-linux** from the selections on the left and then click on the **Play virtual machine** button to the right.

   If prompted for whether the virtual machine has been copied or moved, click on the **Moved** button.

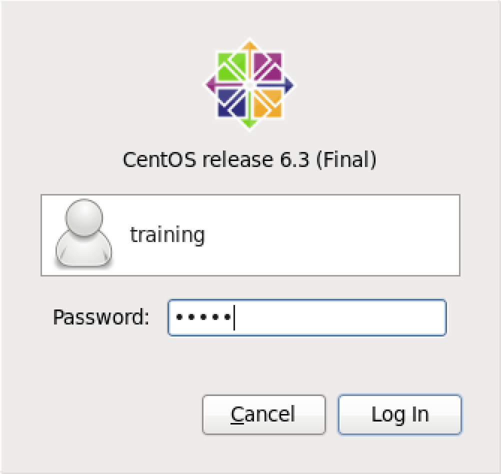

VMware Player (Non-commercial use only)
Player
Home
CentOS-6.3-amd64-ZedBoard-linux
Play virtual machine
Edit virtual machine settings

CentOS-6.3-amd64-ZedBoard-linux

State: Powered Off
OS: CentOS 64-bit
Version: Workstation 9.0 virtual machine
RAM: 1 GB

Figure 2 – The VMware Player Application

3. If prompted for a workstation login, click on the user entry **training** and enter the password **Avnet** in order to log into the system.



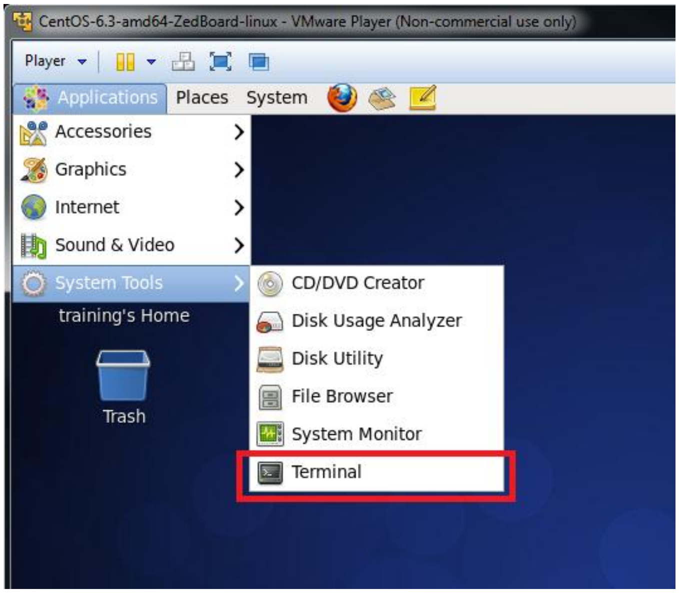
CentOS release 6.3 (Final)

training

Password: •••••

Figure 3 – The CentOS Workstation Login

4. If a terminal is not already open on the desktop of the CentOS guest operating system, open a terminal window through the **Applications→System Tools→Terminal** menu item.



CentOS-6.3-amd64-ZedBoard-linux - VMware Player (Non-commercial use only)

Figure 4 – Launching the CentOS Terminal from the Desktop

5. The Xilinx U-Boot Git repository is located at the following URL:

   git://git.xilinx.com/u-boot-xlnx.git

   To get a working copy of the codebase, clone the remote repository to your local machine. Cloning creates the repository and checks out the latest version, which is referred to as HEAD.

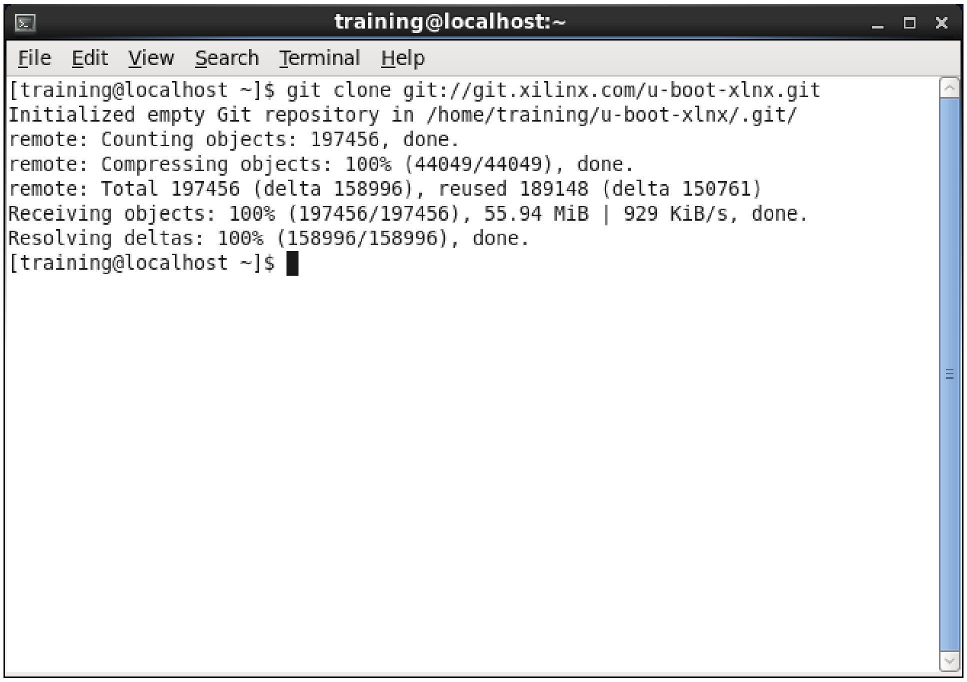
   Use the following Git command to clone the repository. Perform this step only if you are not performing this exercise as part of a Live SpeedWay course.

```
$ git clone git://git.xilinx.com/u-boot-xlnx.git
```

6. Wait until the clone operation completes, this could take 5-20 minutes depending upon your connection speed.

   The clone command sets up a few convenience items for you by:

   - Keeping the address of the original repository
   - Aliasing the address of the original repository as origin so that changes can be easily sent back (if you have authorization) to the remote repository



```
[training@localhost ~]$ git clone git://git.xilinx.com/u-boot-xlnx.git
Initialized empty Git repository in /home/training/u-boot-xlnx/.git/
remote: Counting objects: 197456, done.
remote: Compressing objects: 100% (44049/44049), done.
remote: Total 197456 (delta 158996), reused 189148 (delta 150761)
Receiving objects: 100% (197456/197456), 55.94 MiB | 929 KiB/s, done.
Resolving deltas: 100% (158996/158996), done.
[training@localhost ~]$
```

Figure 5 – Using the Git Clone Command

7. Change into the U-Boot source folder.

```
$ cd u-boot-xlnx
```

8. Checkout the code changes related to the 14.2 tools release which are kept under the **xilinx-14.2-build1** tag.

```
$ git checkout -b xilinx-14.2-build1-trd \

xilinx-14.2-build1-trd
```

Questions:

Answer the following questions:

- *How is a working copy of the Zynq U-Boot code base obtained?*

____

Experiment 2: Configuring U-Boot for the ZedBoard Target

A working copy of the U-Boot repository is now available on the local development machine and can be used to configure the source tree so that it can be built for a ZedBoard target platform.

Keep in mind that U-Boot must be configured for a target platform before being compiled. This is done using the **make config** command format:

make <BOARDNAME>_config

Where **<BOARDNAME>** is the name of the configuration file (but without the .h) found in the **include/configs/** folder.

Experiment 2 General Instruction:

Configure the U-Boot source for the ZedBoard target platform.

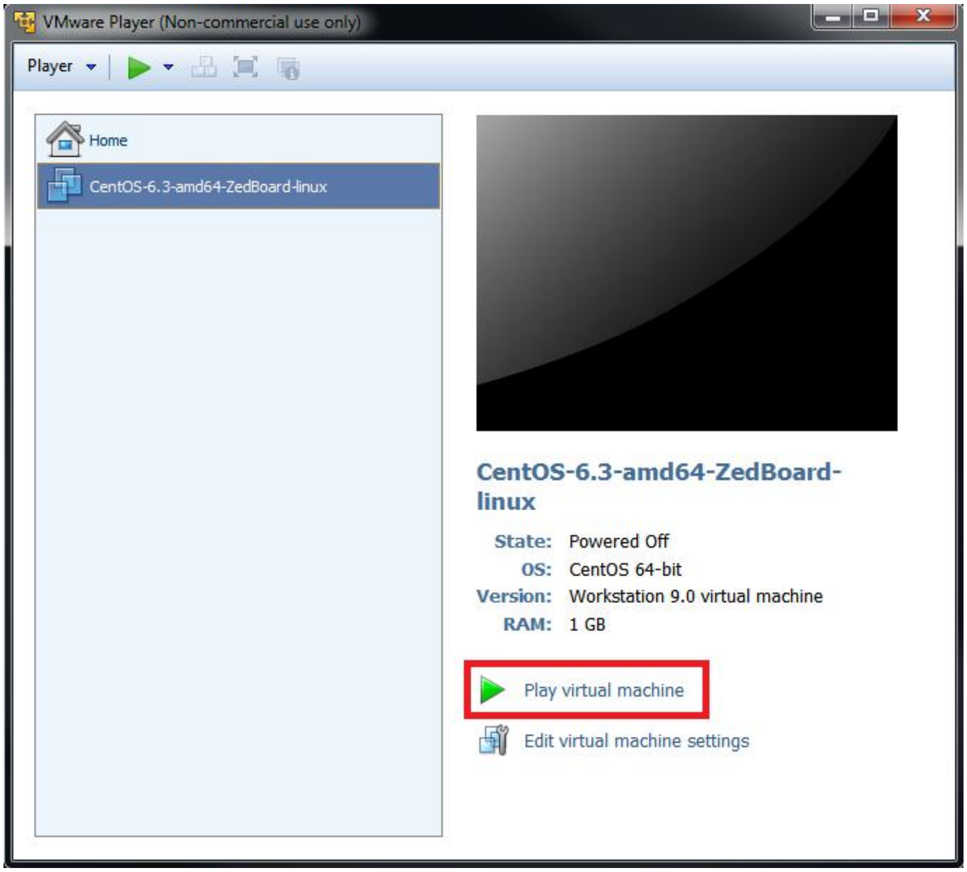
Experiment 2 Step-by-Step Instructions:

1. If the CentOS virtual machine is not already open, launch the VMware Player application from by selecting **Start → All Programs → VMware → VMware Player**. If the CentOS virtual machine is already open, skip ahead to Step 4.

Figure 6 – The VMware Player Application Icon

2. Select the virtual machine named **CentOS-6.3-amd64-ZedBoard-linux** from the selections on the left and then click on the **Play virtual machine** button to the right.

   If prompted for whether the virtual machine has been copied or moved, click on the **Moved** button.



VMware Player (Non-commercial use only)
Player
Home
CentOS-6.3-amd64-ZedBoard-linux
Play virtual machine
Edit virtual machine settings

CentOS-6.3-amd64-ZedBoard-linux

State: Powered Off
OS: CentOS 64-bit
Version: Workstation 9.0 virtual machine
RAM: 1 GB

Figure 7 – The VMware Player Application

3. If prompted for a workstation login, click on the user entry **training** and enter the password **Avnet** in order to log into the system.

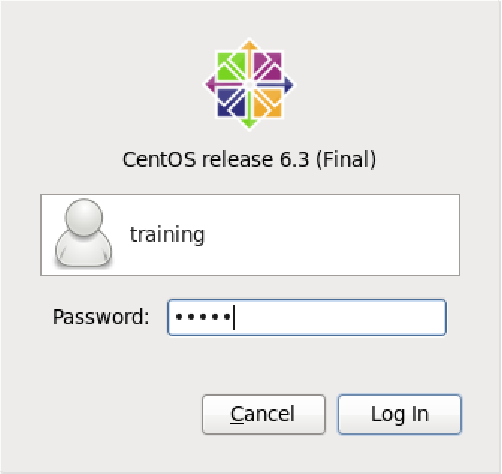


Figure 8 – The CentOS Workstation Login

4. If a terminal is not already open on the desktop of the CentOS guest operating system, open a terminal window through the **Applications→System Tools→Terminal** menu item.

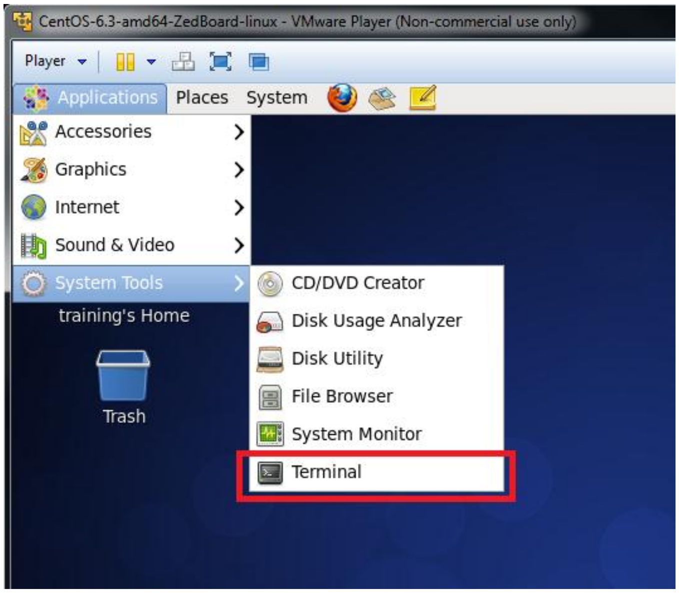


Figure 9 – Launching the CentOS Terminal from the Desktop

5. Change from the home directory into the U-Boot source directory.

```
$ cd u-boot-xlnx/
```

6. For good measure (sometimes a necessity) run a make distribution clean command against the U-Boot source code. This command will remove all intermediary files created by config as well as any intermediary files created by make and it is a good way to clean up any stale configurations.

```
$ make distclean
```

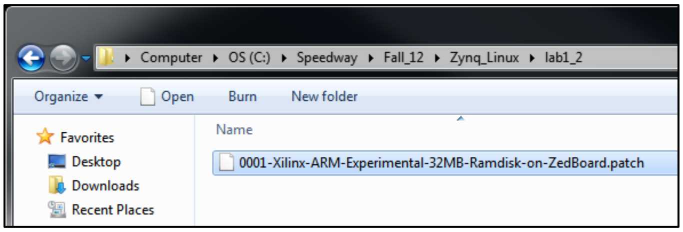
7. A ZedBoard configuration is included for the ZedBoard target and the ZedBoard specific header can be seen in the **/include/configs/zynq_zed.h** file. Configure U-Boot for the ZedBoard target by using the ZedBoard config.

```
$ make zynq_zed_config
```

8. Locate the **.patch** file in the Lab 1.2 folder on the host operating system by using Windows explorer to navigate to the following folder:

   C:\Speedway\Fall_12\Zynq_Linux\lab1_2\

   Right click on this file and select the **Copy** option which will place the selected file in the Virtual Machine clipboard.



0001-Xilinx-ARM-Experimental-32MB-Ramdisk-on-ZedBoard.patch

Figure 10 – Copying the U-Boot Patch File to Virtual Machine Clipboard

9. Open a file browser window through the **Applications→System Tools→File Browser** menu item.

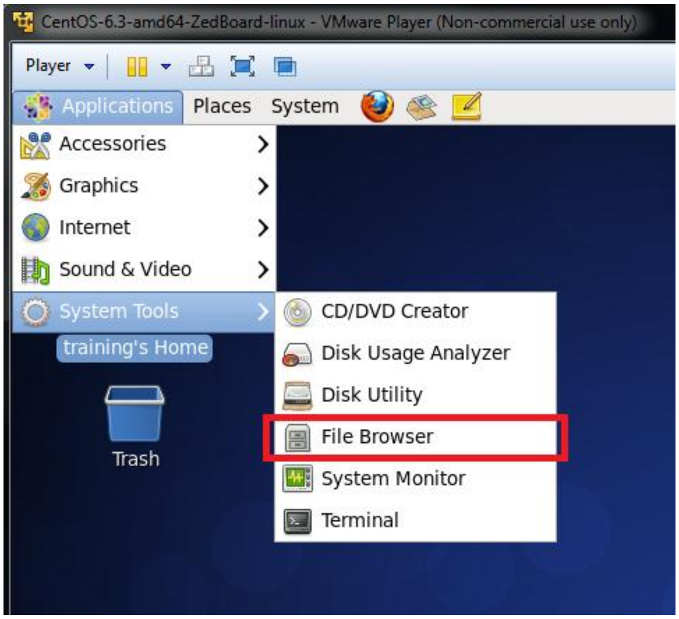


Figure 11 – CentOS File Browser

10. Using the CentOS File Browser on the guest operating system, browse to the **/home/training/u-boot-xlnx/** folder. Right click in open white space between folder icons and select the **Paste** option to paste the patch file into the Linux kernel source code tree **/home/training/u-boot-xlnx/** folder.

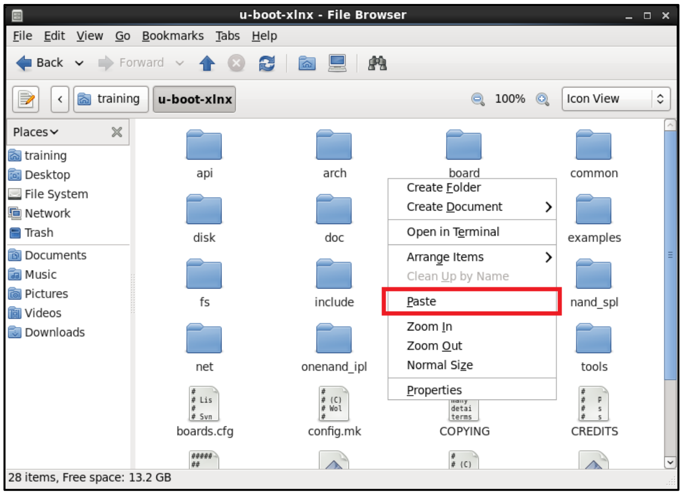


Figure 12 – U-Boot Patch Pasted to the Guest Machine

11. Using the CentOS terminal, apply the patch to the U-Boot tree.

    Hint: For long filenames such as the one in the following command, a portion of the filename can be typed (i.e. 0001-Xilinx) and the TAB key will run the autocomplete feature of the command interpreter to complete the remainder of the filename.

```
$ git apply 0001-Xilinx-ARM-Experimental-32MB-Ramdisk-\

on-ZedBoard.patch
```

Questions:

Answer the following questions:

- *Which folder contains the U-Boot target board configuration headers?*

____

Experiment 3: Building U-Boot from the Configured Source

Now that the source tree has been configured for the ZedBoard target platform, it can be built using the cross toolchain.

Experiment 3 General Instruction:

Build U-Boot for the ZedBoard target platform using the cross toolchain and copy the U-Boot executable to the Lab1.3 folder under the host operating system.

Experiment 3 Step-by-Step Instructions:

1. In the previous experiment, the U-Boot source tree was configured for a ZedBoard target platform.

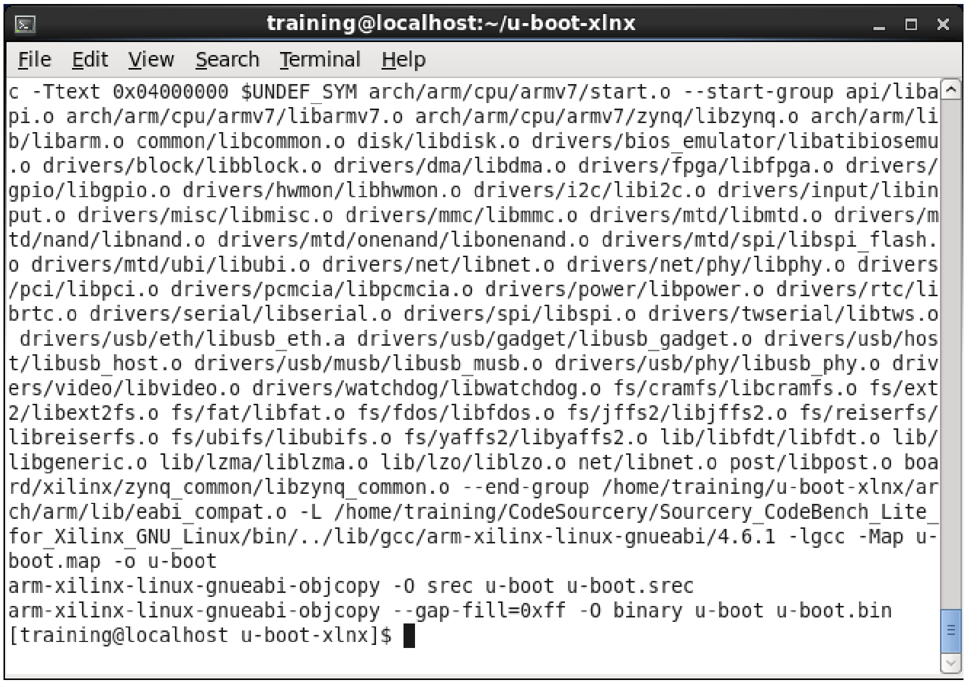
   First, make sure the **/home/training/u-boot-xlnx/** folder is the current working directory by using the pwd command. If the working directory shown is not the **/home/training/u-boot-xlnx/** folder change directories to that folder.

```
$ pwd
```

2. Once the working directory is the **/home/training/u-boot-xlnx/** folder, build the U-Boot source with the make command.

   The build process should take anywhere from 1 to 10 minutes to complete and should complete successfully. If the build is successful and the console output looks similar to that shown in Figure 13, proceed to Step 4.

```
$ make
```



training@localhost:~/u-boot-xlnx
File Edit View Search Terminal Help

```
c -Ttext 0x04000000 $UNDEF_SYM arch/arm/cpu/armv7/start.o --start-group api/liba
pi.o arch/arm/cpu/armv7/libarmv7.o arch/arm/cpu/armv7/zynq/libzynq.o arch/arm/li
b/libarm.o common/libcommon.o disk/libdisk.o drivers/bios_emulator/libatibiosemu
.o drivers/block/libblock.o drivers/dma/libdma.o drivers/fpga/libfpga.o drivers/
gpio/libgpio.o drivers/hwmon/libhwmon.o drivers/i2c/libi2c.o drivers/input/libin
put.o drivers/misc/libmisc.o drivers/mmc/libmmc.o drivers/mtd/libmtd.o drivers/m
td/nand/libnand.o drivers/mtd/onenand/libonenand.o drivers/mtd/spi/libspi_flash.
o drivers/mtd/ubi/libubi.o drivers/net/libnet.o drivers/net/phy/libphy.o drivers
/pci/libpci.o drivers/pcmcia/libpcmcia.o drivers/power/libpower.o drivers/rtc/li
brtc.o drivers/serial/libserial.o drivers/spi/libspi.o drivers/twserial/libtws.o
 drivers/usb/eth/libusb_eth.a drivers/usb/gadget/libusb_gadget.o drivers/usb/hos
t/libusb_host.o drivers/usb/musb/libusb_musb.o drivers/usb/phy/libusb_phy.o driv
ers/video/libvideo.o drivers/watchdog/libwatchdog.o fs/cramfs/libcramfs.o fs/ext
2/libext2fs.o fs/fat/libfat.o fs/fdos/libfdos.o fs/jffs2/libjffs2.o fs/reiserfs/
libreiserfs.o fs/ubifs/libubifs.o fs/yaffs2/libyaffs2.o lib/libfdt/libfdt.o lib/
libgeneric.o lib/lzma/liblzma.o lib/lzo/liblzo.o net/libnet.o post/libpost.o boa
rd/xilinx/zynq_common/libzynq_common.o --end-group /home/training/u-boot-xlnx/ar
ch/arm/lib/eabi_compat.o -L /home/training/CodeSourcery/Sourcery_CodeBench_Lite_
for_Xilinx_GNU_Linux/bin/../lib/gcc/arm-xilinx-linux-gnueabi/4.6.1 -lgcc -Map u-
boot.map -o u-boot
arm-xilinx-linux-gnueabi-objcopy -O srec u-boot u-boot.srec
arm-xilinx-linux-gnueabi-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
[training@localhost u-boot-xlnx]$ 
```

Figure 13 – U-Boot Build Completed

3. If you run into strange errors regarding problems locating your cross toolchain or errors with segmentations faults during the build, perform this step to correct the problem, otherwise skip ahead to Step 4.

   The terminal **PATH** environment variable may not be set correctly. One way to resolve this is to pick up the updated user profile using the source command from the **/home/training/** folder.

   Return to the U-Boot source folder, then clean the source tree with **make distclean** and go back to Step 2 to rebuild. This should help resolve any segmentation faults encountered with the toolchain.

```
$ cd ~

$ source .bash_profile

$ cd u-boot-xlnx/

$ make distclean

$ make zynq_zed_config
```

   The terminal PATH environmental variable tells the terminal shell which directories to search for executable files in response to commands issued by a user. This environment variable increases both the convenience and the safety of command line based operating systems and is widely considered to be the single most important environmental variable. The PATH environment variable can be checked by echoing it to the command line. Check it to be sure that it contains a path to the Sourcery CodeBench installed /bin folder.

```
$ echo $PATH
```

4. Once the build has completed successfully, open a file browser window through the **Applications→System Tools→File Browser** menu item.

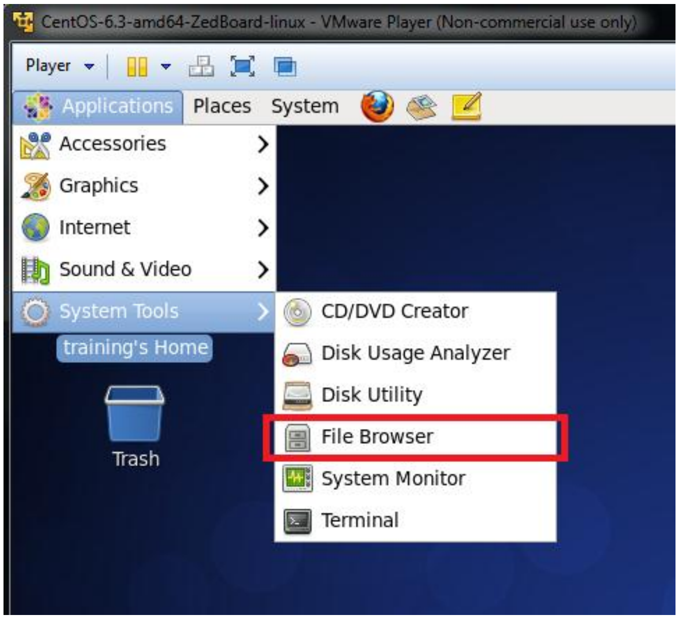
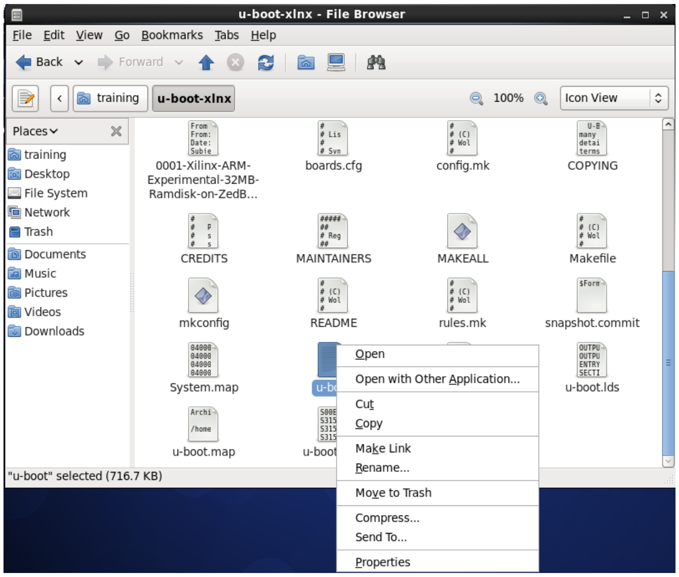


Figure 14 – CentOS File Browser

5. Locate the file **/home/training/u-boot-xlnx/u-boot** which is the target executable file in the Extended Linker Format (ELF) needed to execute on Zynq. Right click on this file and select the **Copy** option which will place the selected file in the Virtual Machine clipboard.



"u-boot" selected (716.7 KB)

Figure 15 – Copying U-Boot Executable to Virtual Machine Clipboard

6. Paste the U-Boot executable into the Lab1.3 folder under the host operating system by using Windows Explorer to navigate to the following folder:

   C:\Speedway\Fall_12\Zynq_Linux\lab1_3

   Then paste the **u-boot** file into this folder. Rename the **u-boot** file to **u-boot.elf** so that it has the appropriate ELF extension needed for the SDK tools to see it.

   This U-Boot executable file will be used in the next lab session.

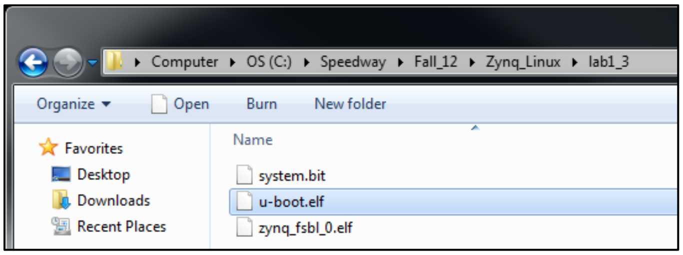


Figure 16 – U-Boot Executable Copied to the Host Machine and Renamed

Questions:

Answer the following questions:

- *Which U-Boot target file has the ELF format that is needed for execution on the Zynq platform?*

____

Exploring Further

If you have additional time and would like to investigate more…

- Take a look through the ZedBoard configuration header /include/configs/zynq_zed.h and determine which U-Boot commands and macros are used to boot Linux from the SD card. Change something recognizable in the config file then rebuild u-boot and copy back to the host operating system.

This concludes Lab 1.2.

Revision History

Date	Version	Revision
06 Sep 12	00	Initial Draft
28 Sep 12	01	Revised Draft
18 Oct 12	02	Course Release
14 Jan 13	05	ZedBoard.org Training Course Release

Resources

http://www.zedboard.org

http://www.xilinx.com/zynq

http://www.xilinx.com/planahead

http://git-scm.com/

http://www.denx.de/wiki/U-Boot

Answers

Experiment 1

- *How is a working copy of the Zynq U-Boot code base obtained?*

A working copy is obtained using the Git clone command against the repository URL.

Experiment 2

- *Which folder contains the U-Boot target board configuration headers?*

/home/training/u-boot-xlnx/include/configs/

Experiment 3

- *Which U-Boot target file has the ELF format that is needed for execution on the Zynq platform?*

u-boot