Creating Embedded Linux Solutions

Course Workbook

Table of Contents

About this Workbook

The contents of this workbook are created by Adiuvo Engineering & Training, Ltd.

If you have any questions about the contents, or need assistance, please contact Adam Taylor at adam@adiuvoengineering.com.

Pre-Lab

Creating Multi Processor Solutions

Required Hardware

Ultra96V2

JTAG / USB Pod

Ultra96V2 Power Supply

SD Card with the Ultra96V2 OOB

Downloads and Installations

Step 1 – Download and install the following at least 1 day prior to the workshop. This may take a significant amount of time and drive space.

Watch the video available here to show how to configure the installation

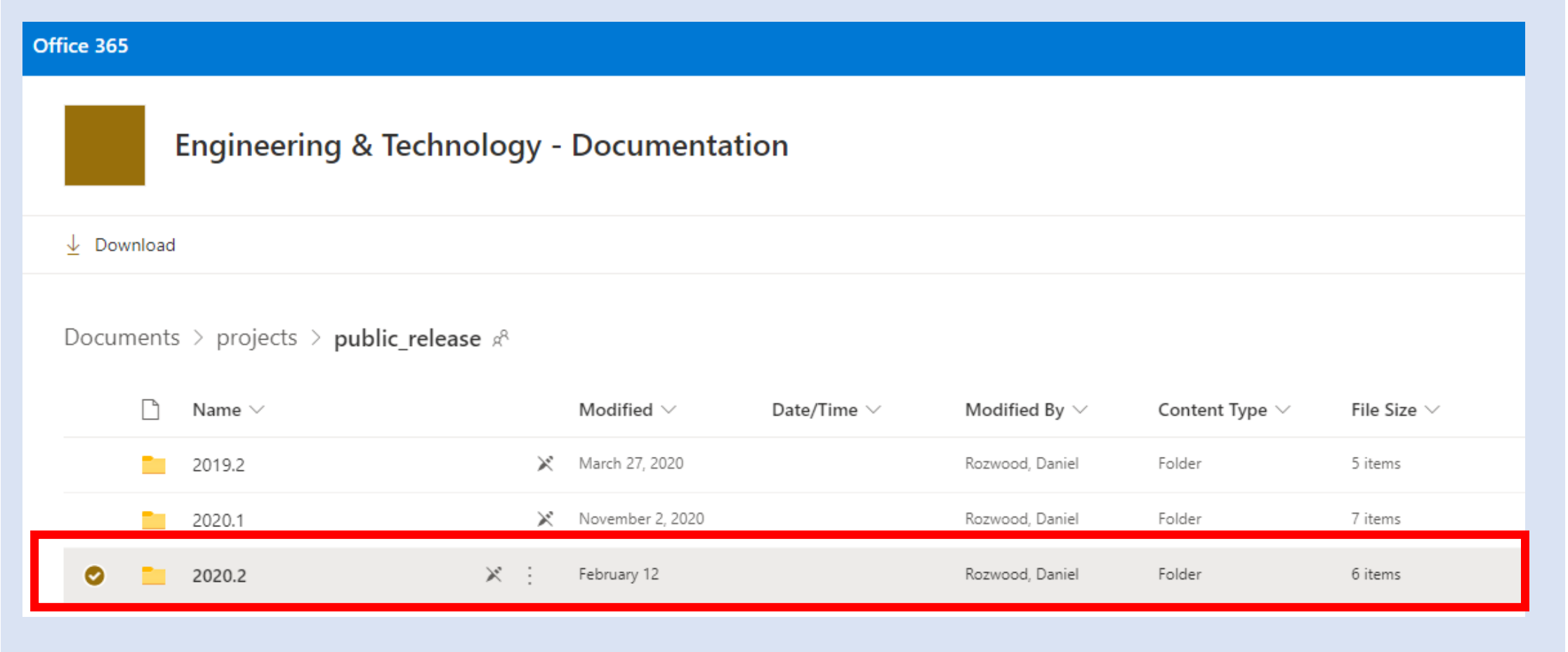
Vitis 2020.2	Download
PetaLinux 2020.2	Download

Lab 3

Creating Embedded Linux Solutions

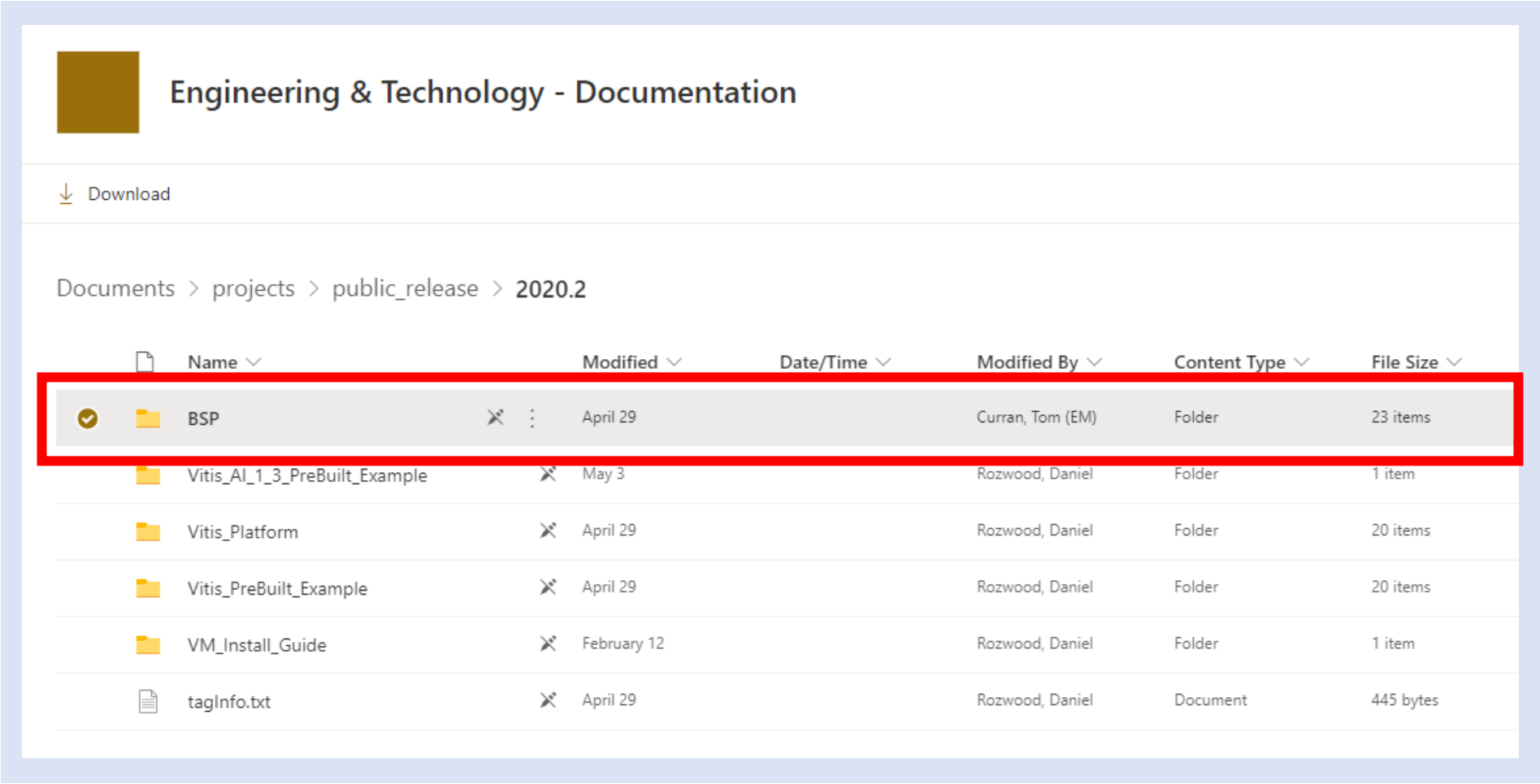
Lab 3: Creating Embedded Linux Solutions

Step 1 – Click here and select 2020.2



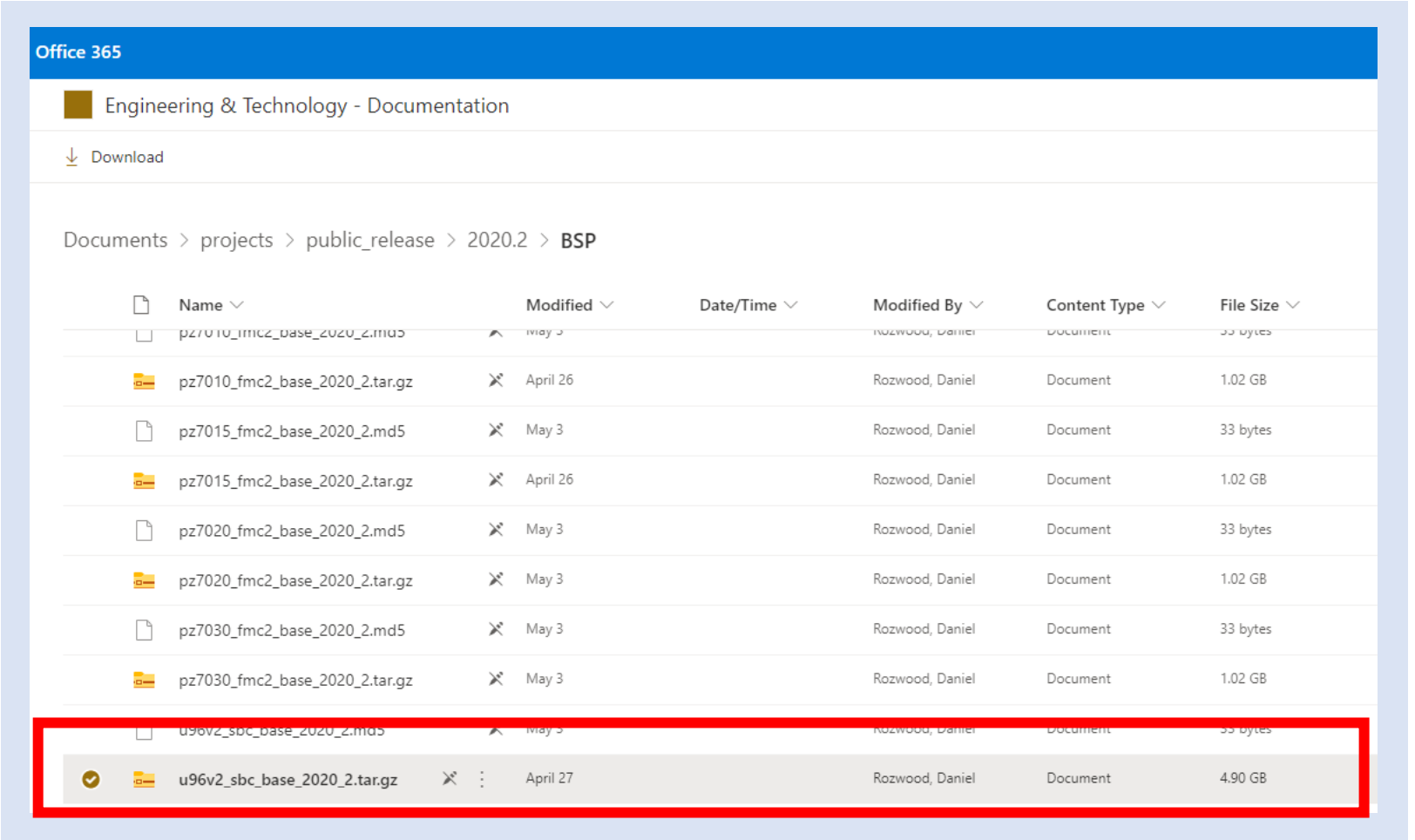
Lab 3: Creating Embedded Linux Solutions

Step 2 – Select BSP



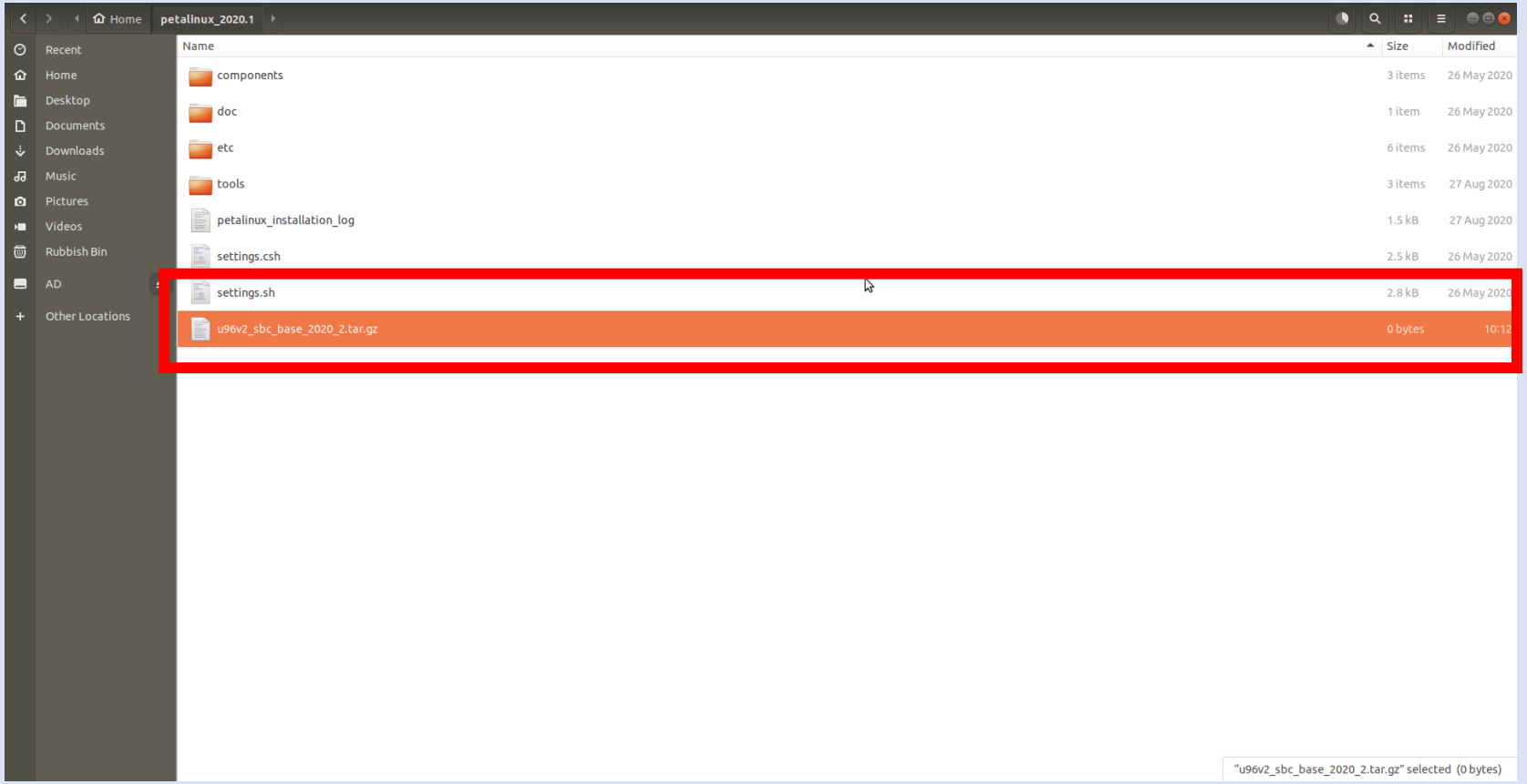
Lab 3: Creating Embedded Linux Solutions

Step 3 – Select and download the Ultra96V2 SBC Base



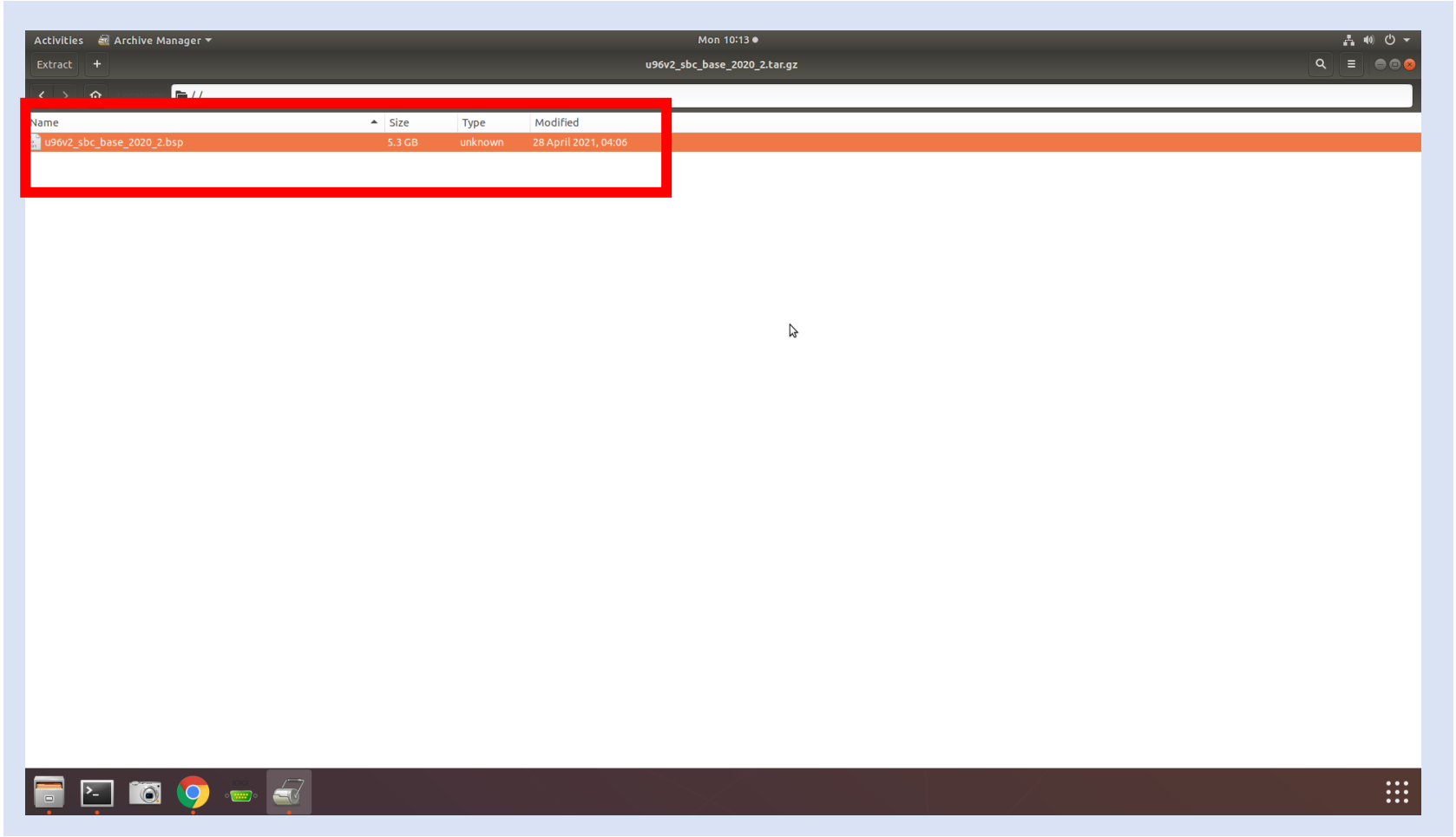
Lab 3: Creating Embedded Linux Solutions

Step 4 – Double click on the Ultra96V2 tar.gz to open the archive



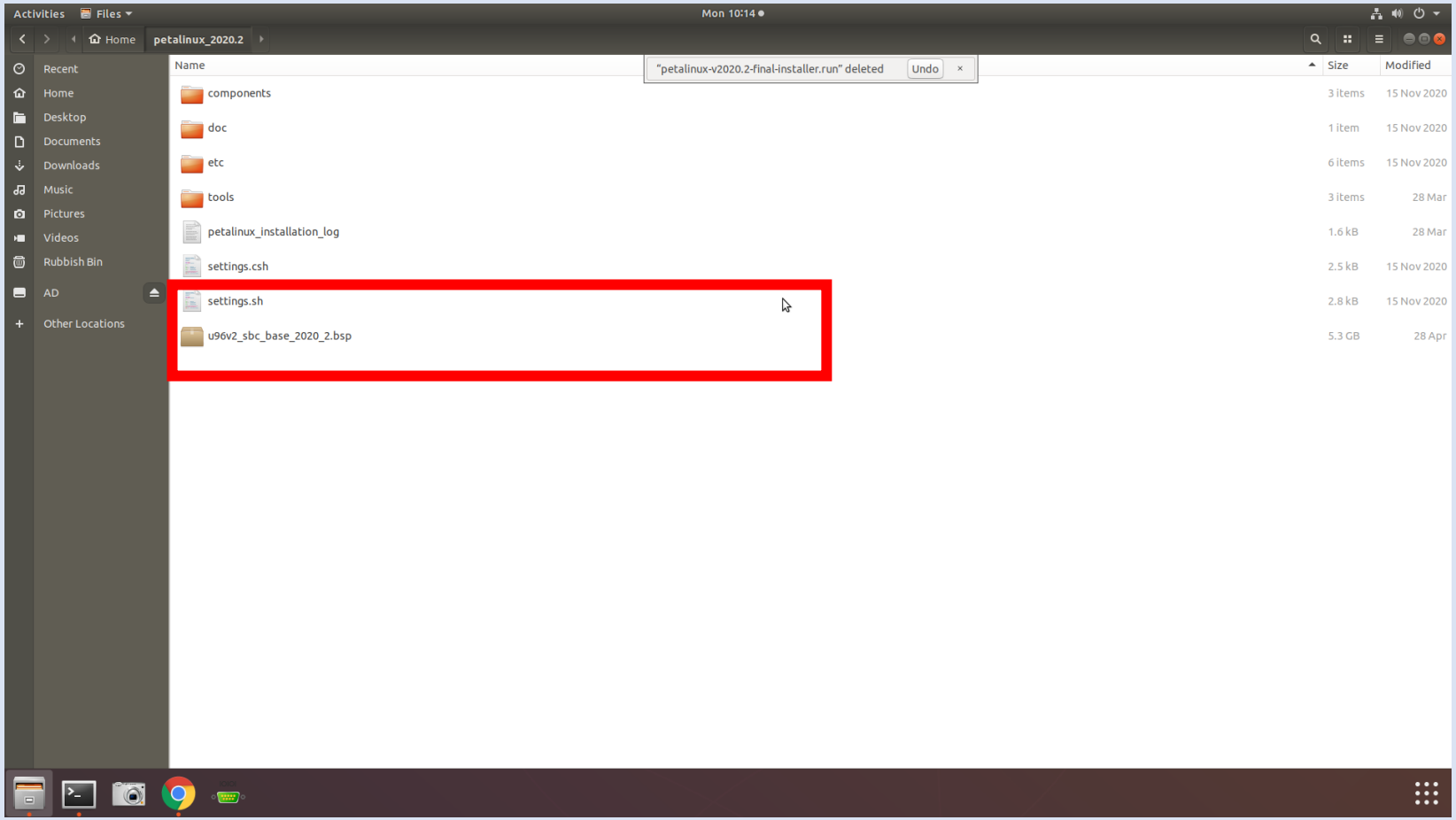
Lab 3: Creating Embedded Linux Solutions

Step 5 – Extract the BSP to the Petalinux directory



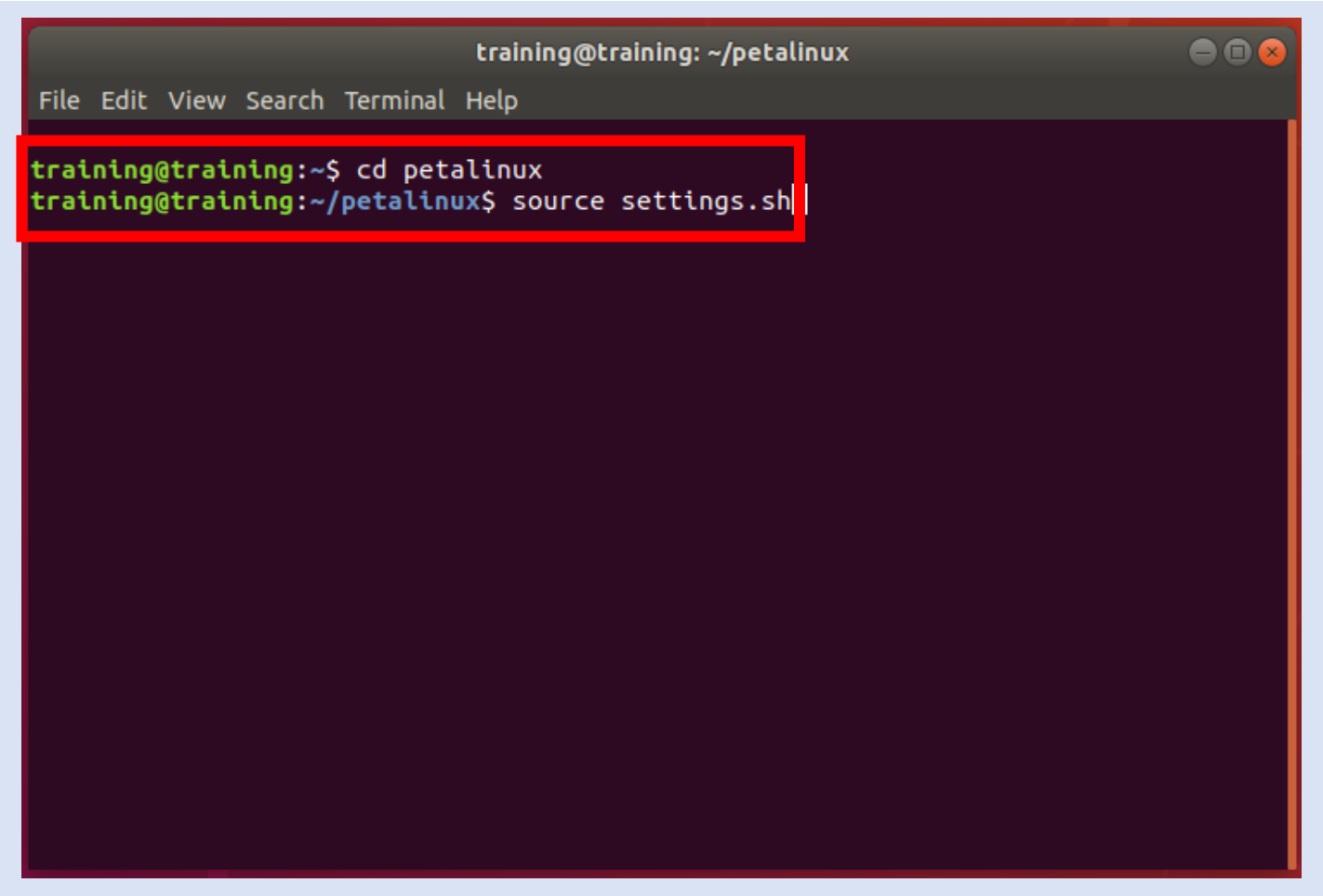
Lab 3: Creating Embedded Linux Solutions

Step 6 – Delete the compressed file, you should see the BSP in the directory



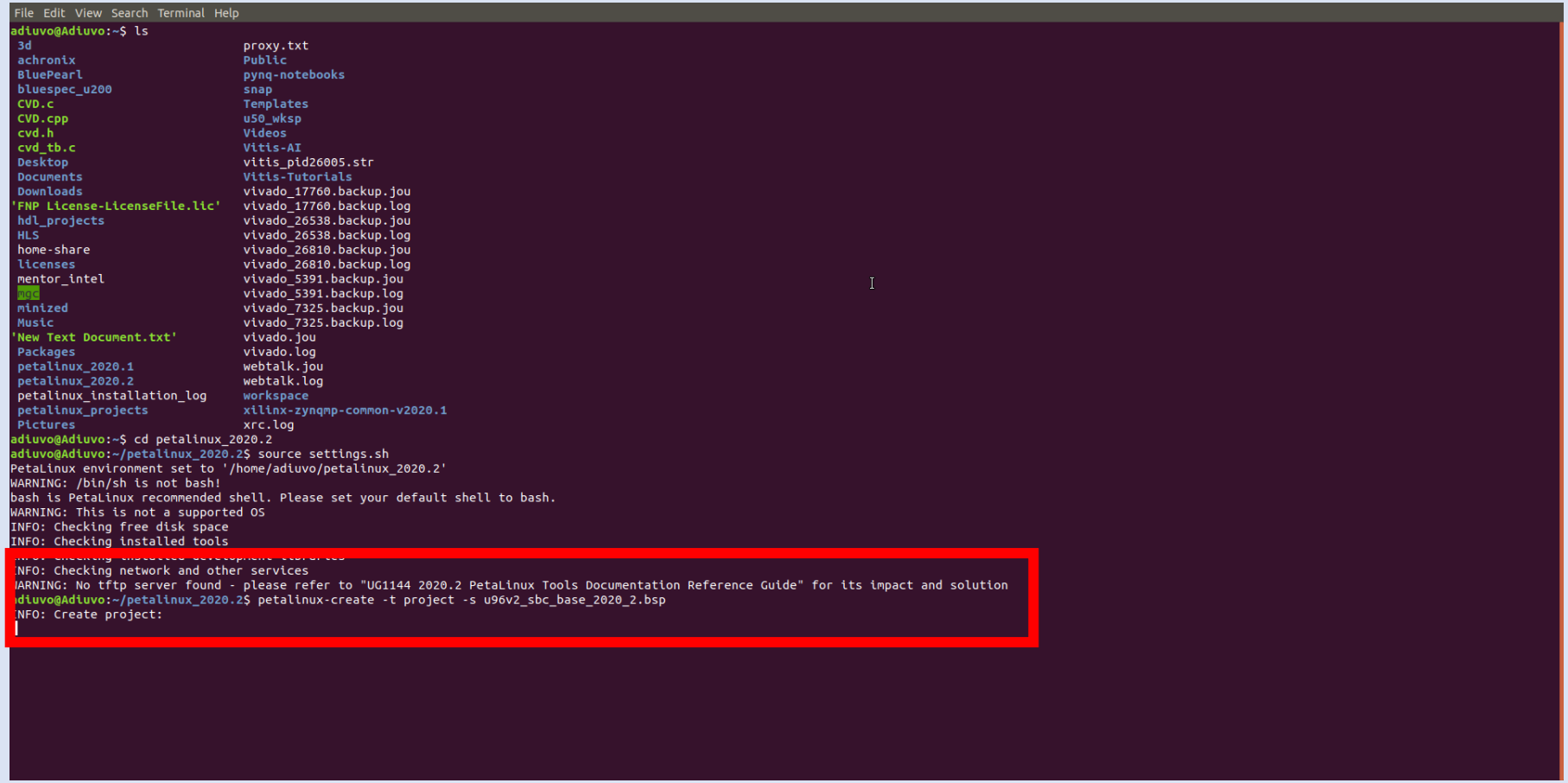
Lab 3: Creating Embedded Linux Solutions

Step 7 – In the petalinux directory, run the command *source settings.sh*



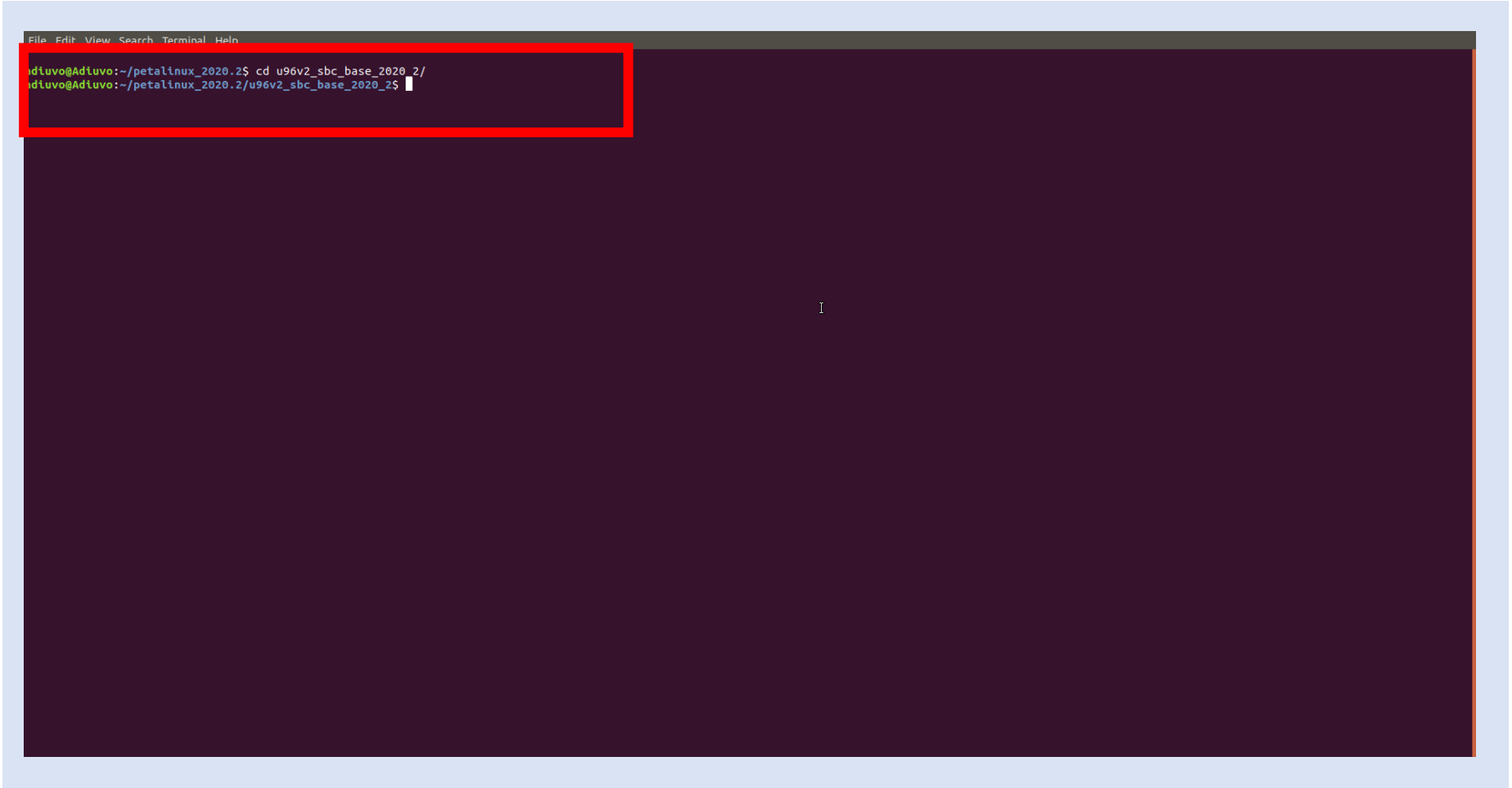
Lab 3: Creating Embedded Linux Solutions

Step 8 – Create a new project, using the command *petalinux-create –t project –s <BSP Name>*



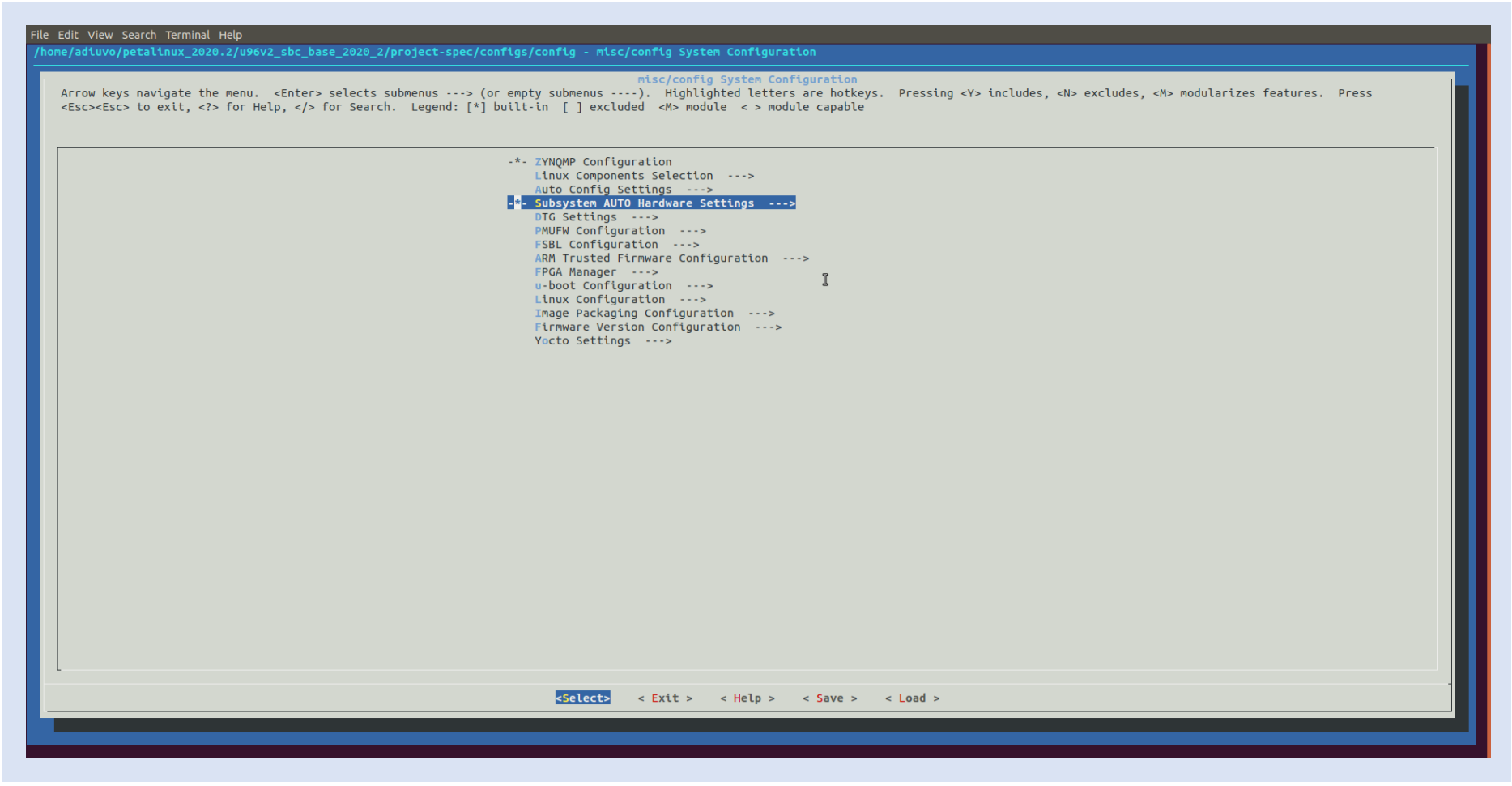
Lab 3: Creating Embedded Linux Solutions

Step 9 – Change Directory into the project directory just created



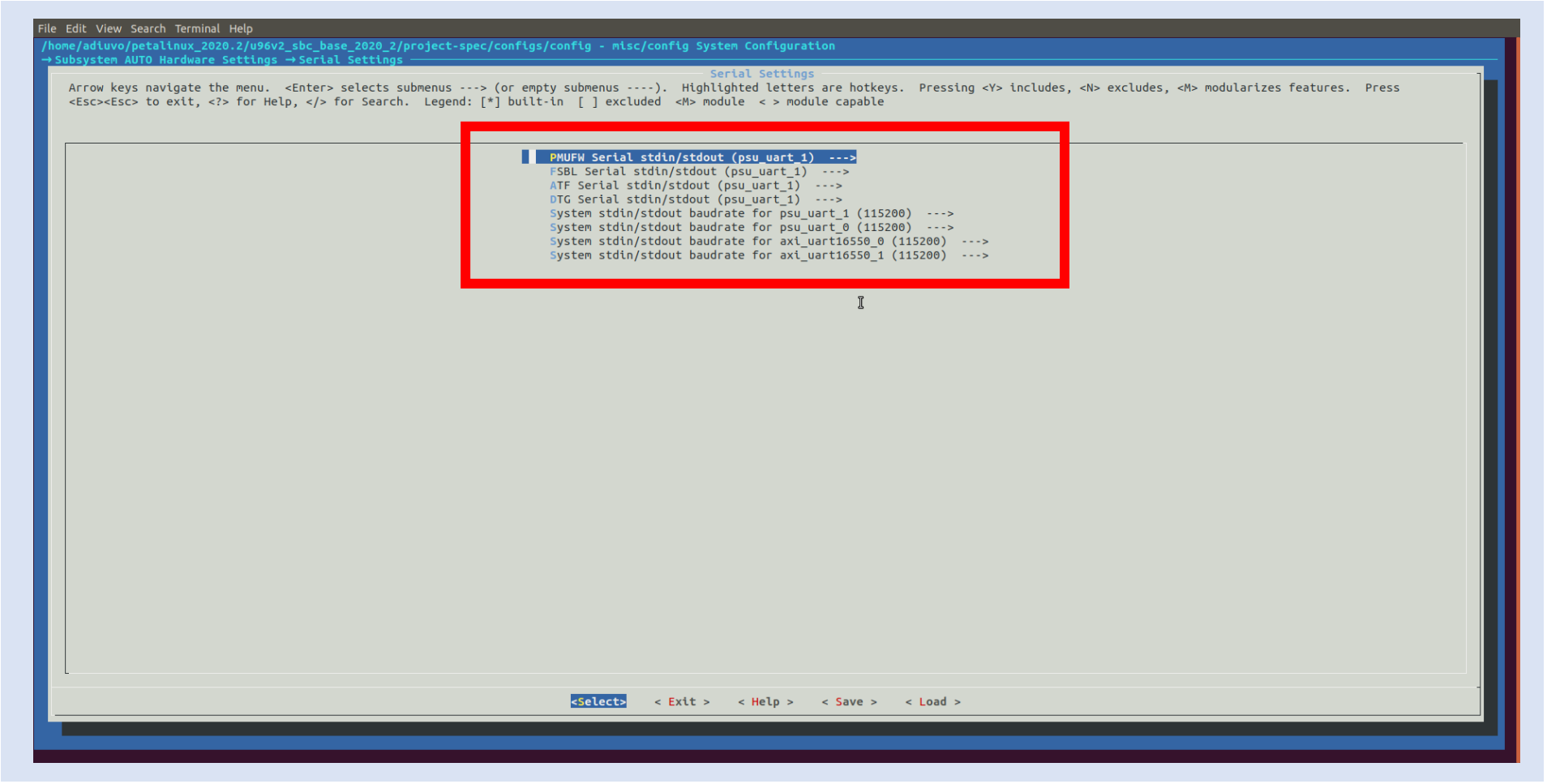
Lab 3: Creating Embedded Linux Solutions

Step 10 – In the terminal issue the command *petalinux-config* This will open the petalinux project configuration dialog. Scroll down to Subsystem AUTO Hardware Settings press enter



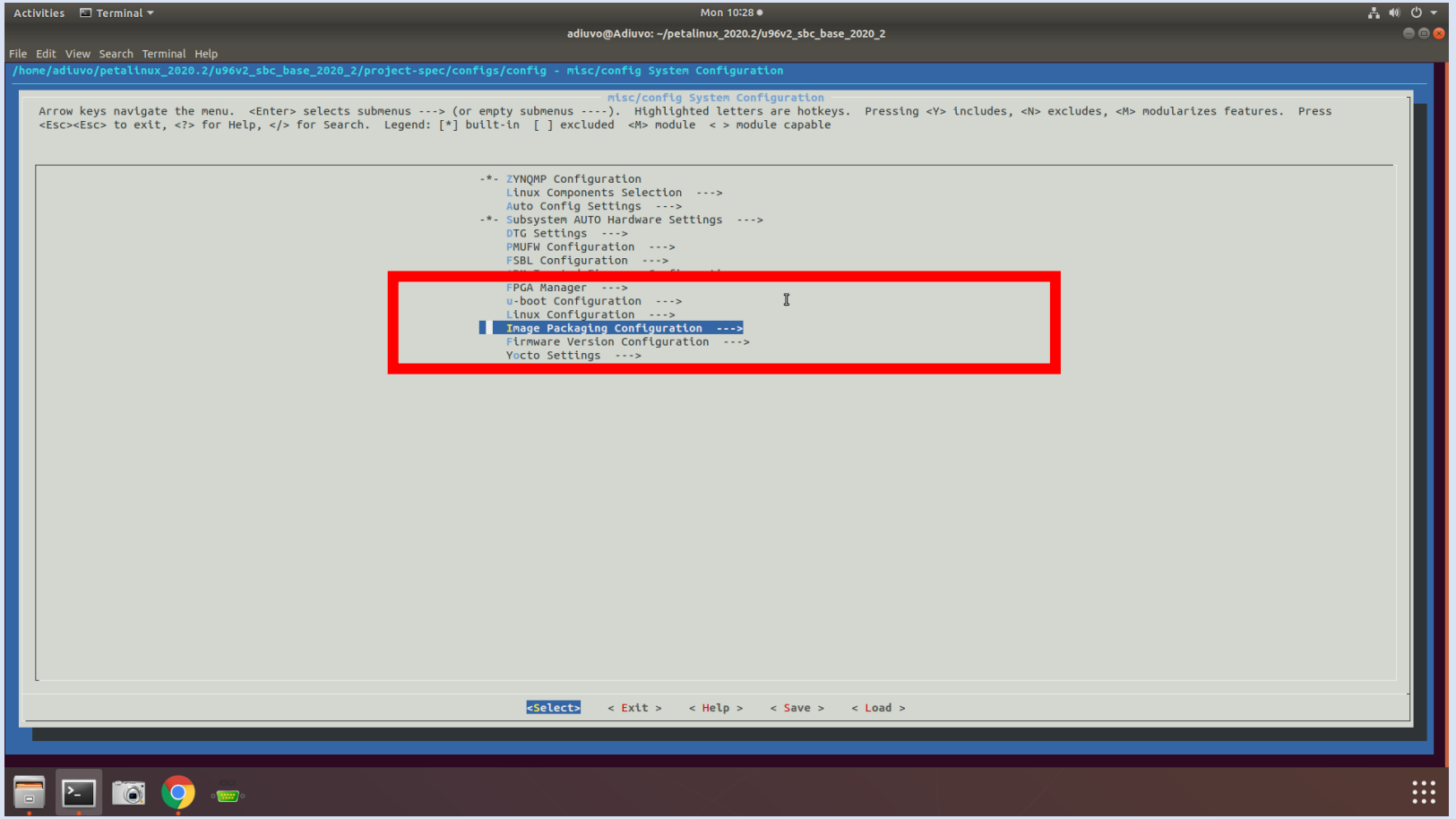
Lab 3: Creating Embedded Linux Solutions

Step 11 – Select the serial port and check that all of them are PSU_UART1



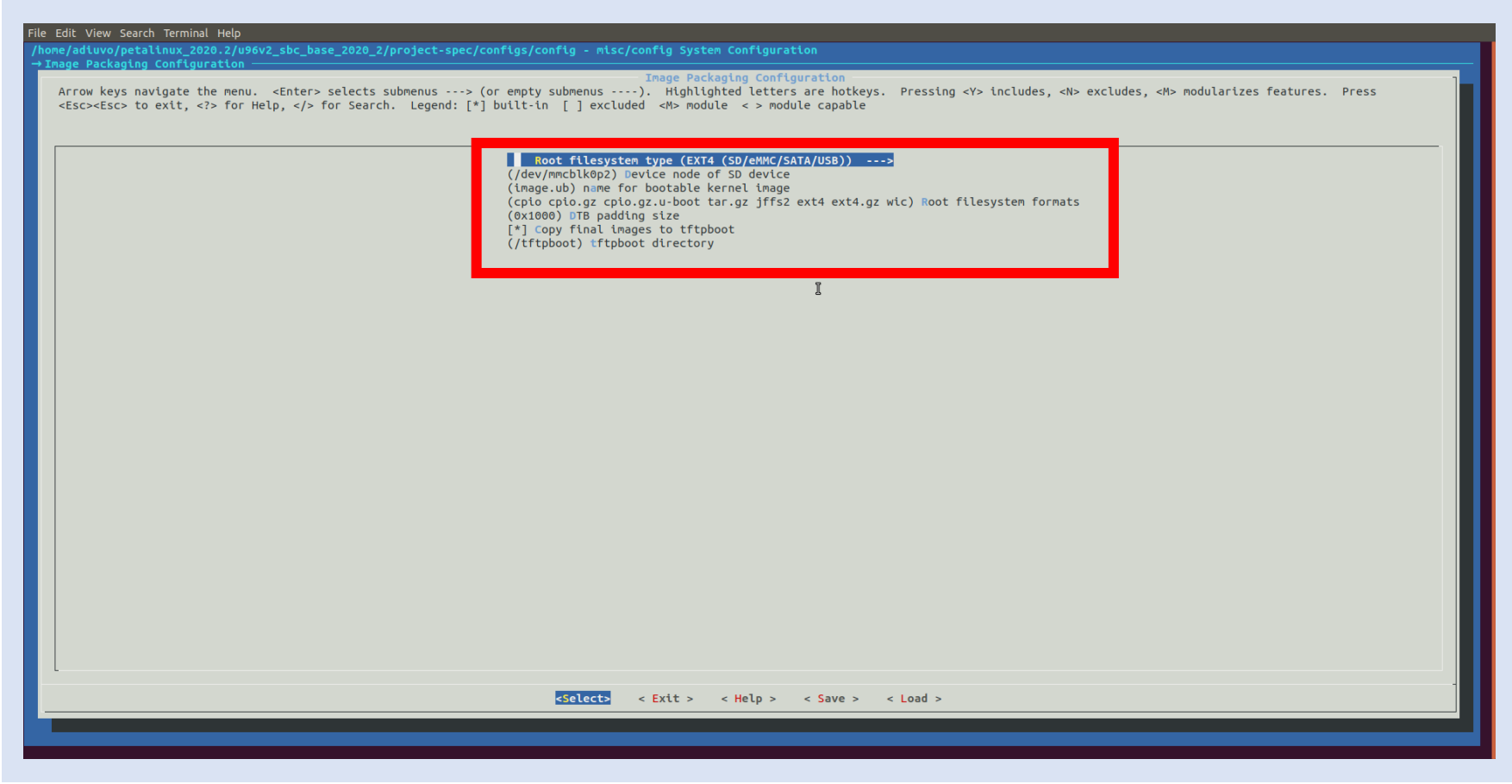
Lab 3: Creating Embedded Linux Solutions

Step 12 – Navigate back to main and select Image Packaging Configuration



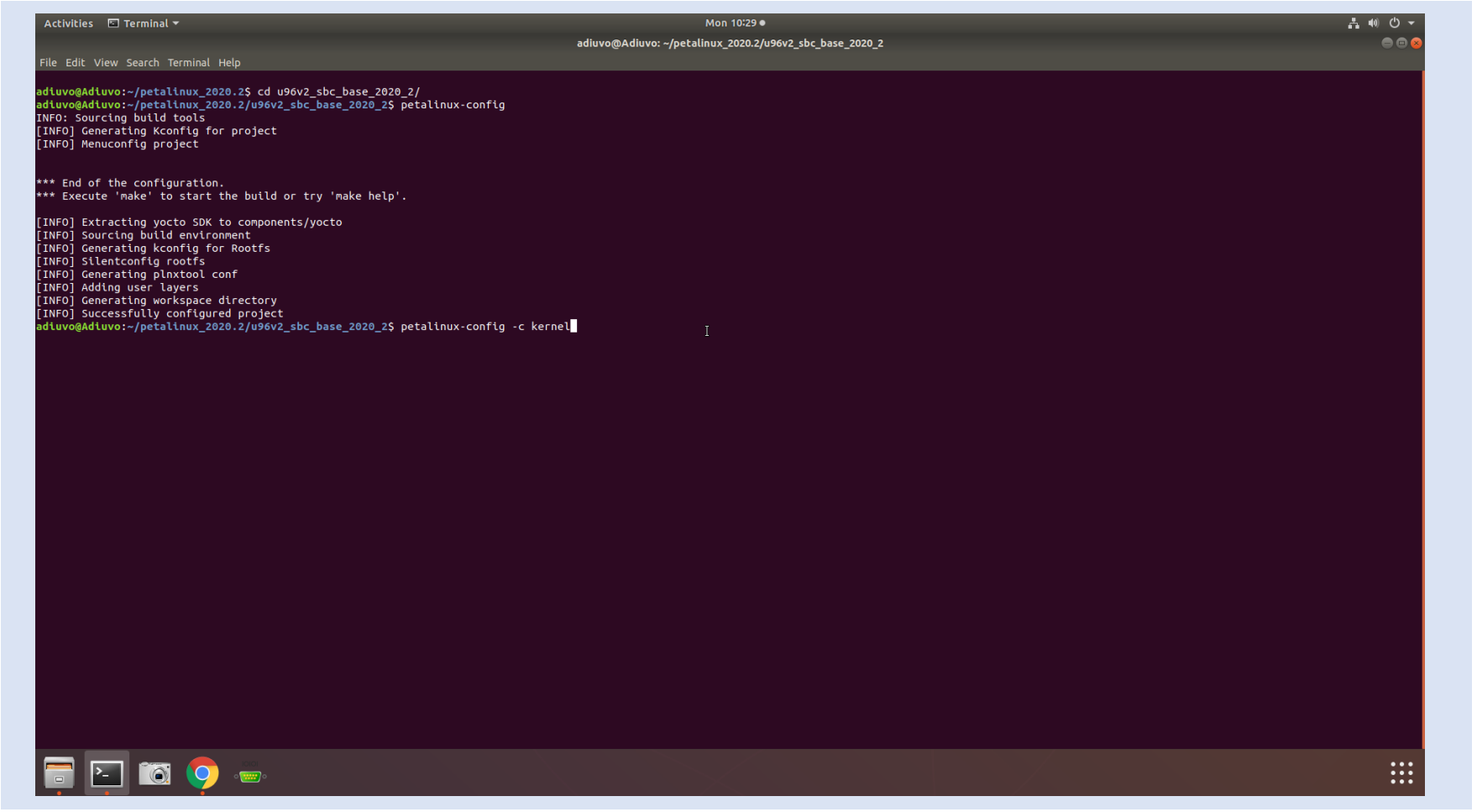
Lab 3: Creating Embedded Linux Solutions

Step 13 – Notice the SD Card is used to store the Root File System – Exit the menu and do not save any changes.



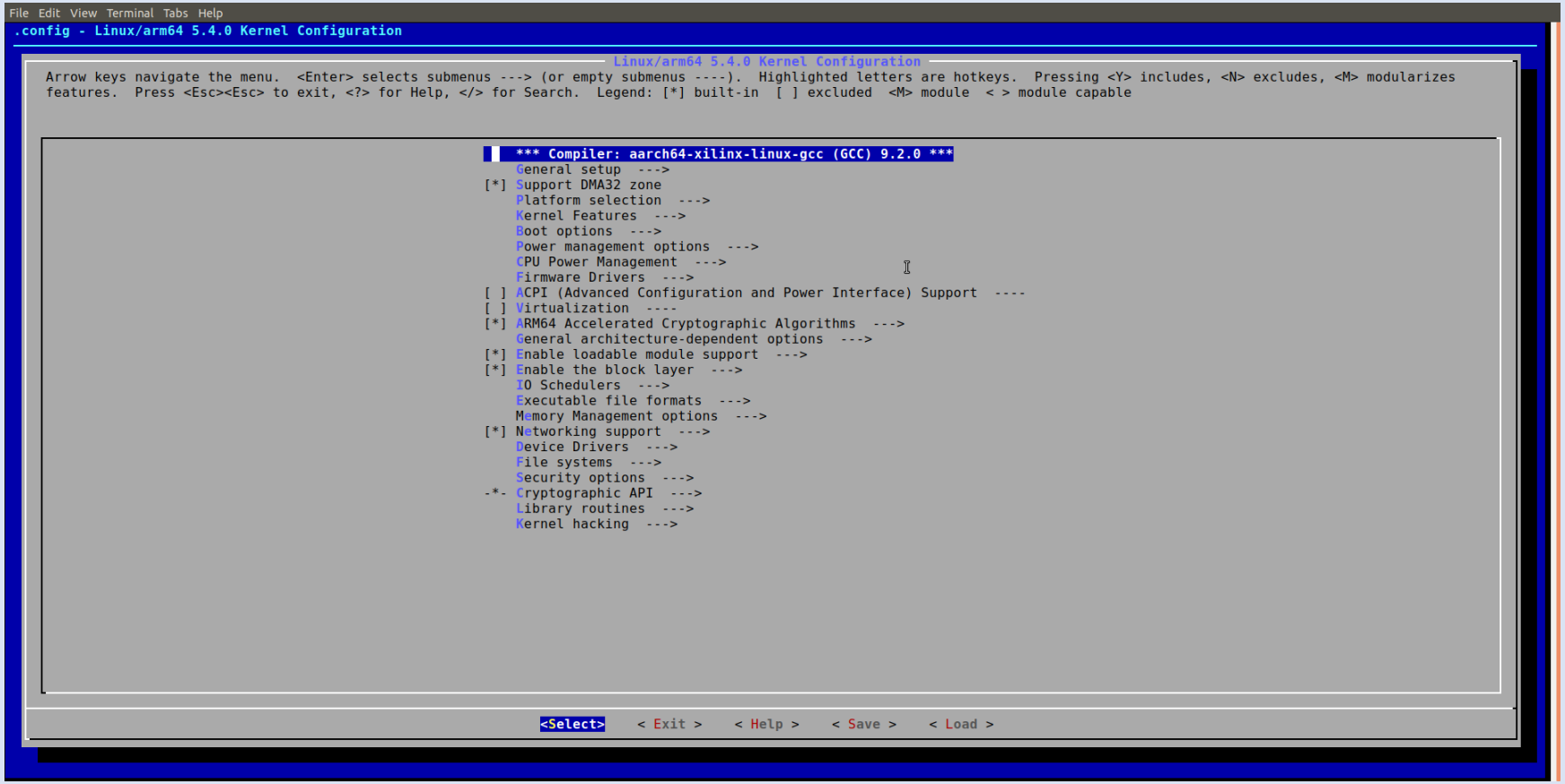
Lab 3: Creating Embedded Linux Solutions

Step 14 – Enter the command *petalinux-config –c kernel*



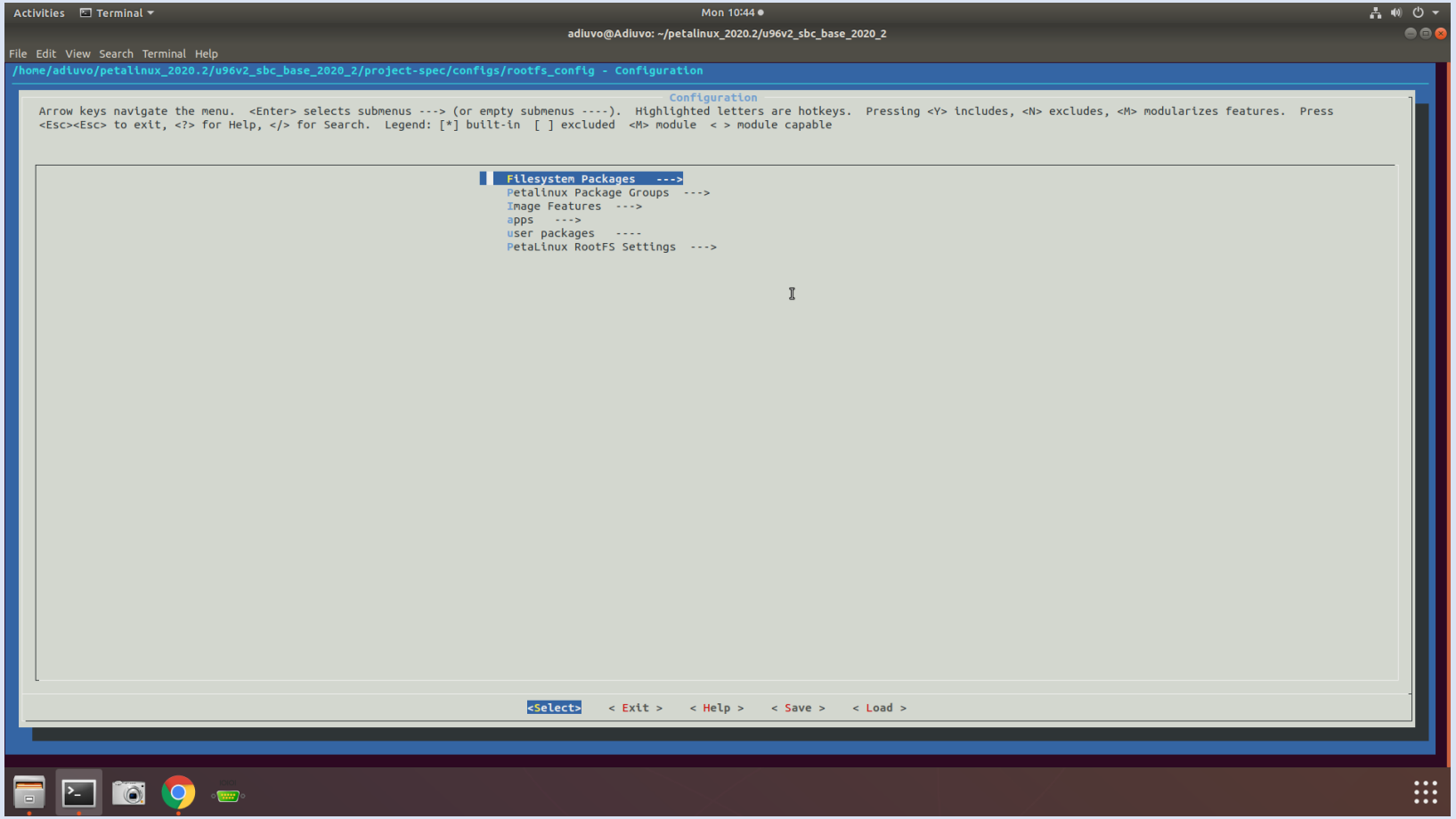
Lab 3: Creating Embedded Linux Solutions

Step 15 – This is the Kernel Configuration, explore the settings and close the menu without making any changes.



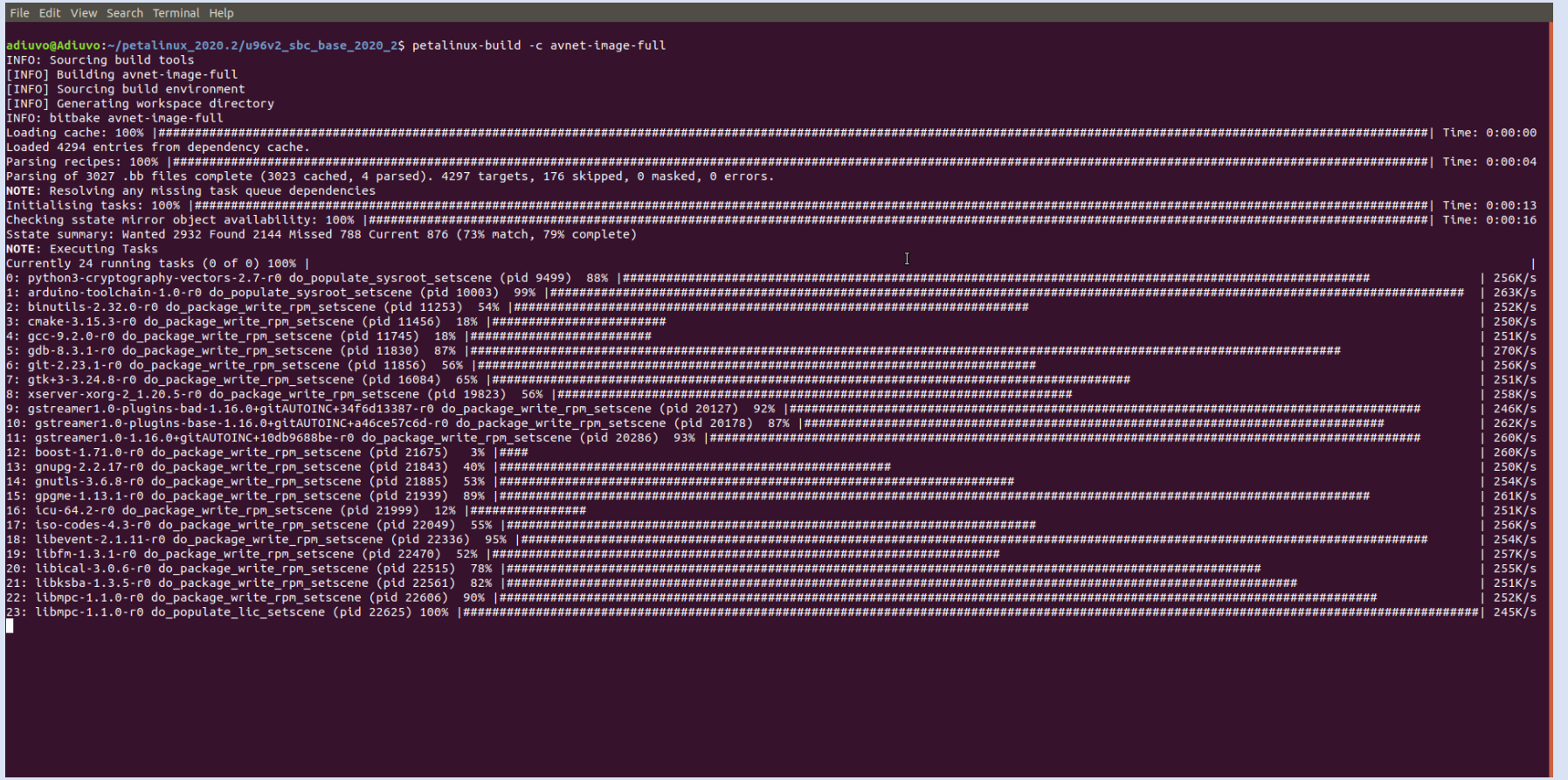
Lab 3: Creating Embedded Linux Solutions

Step 16 – In the terminal enter the command *petalinux-config –c rootfs* This is the Kernel Configuration, explore the settings and close the menu without making any changes.



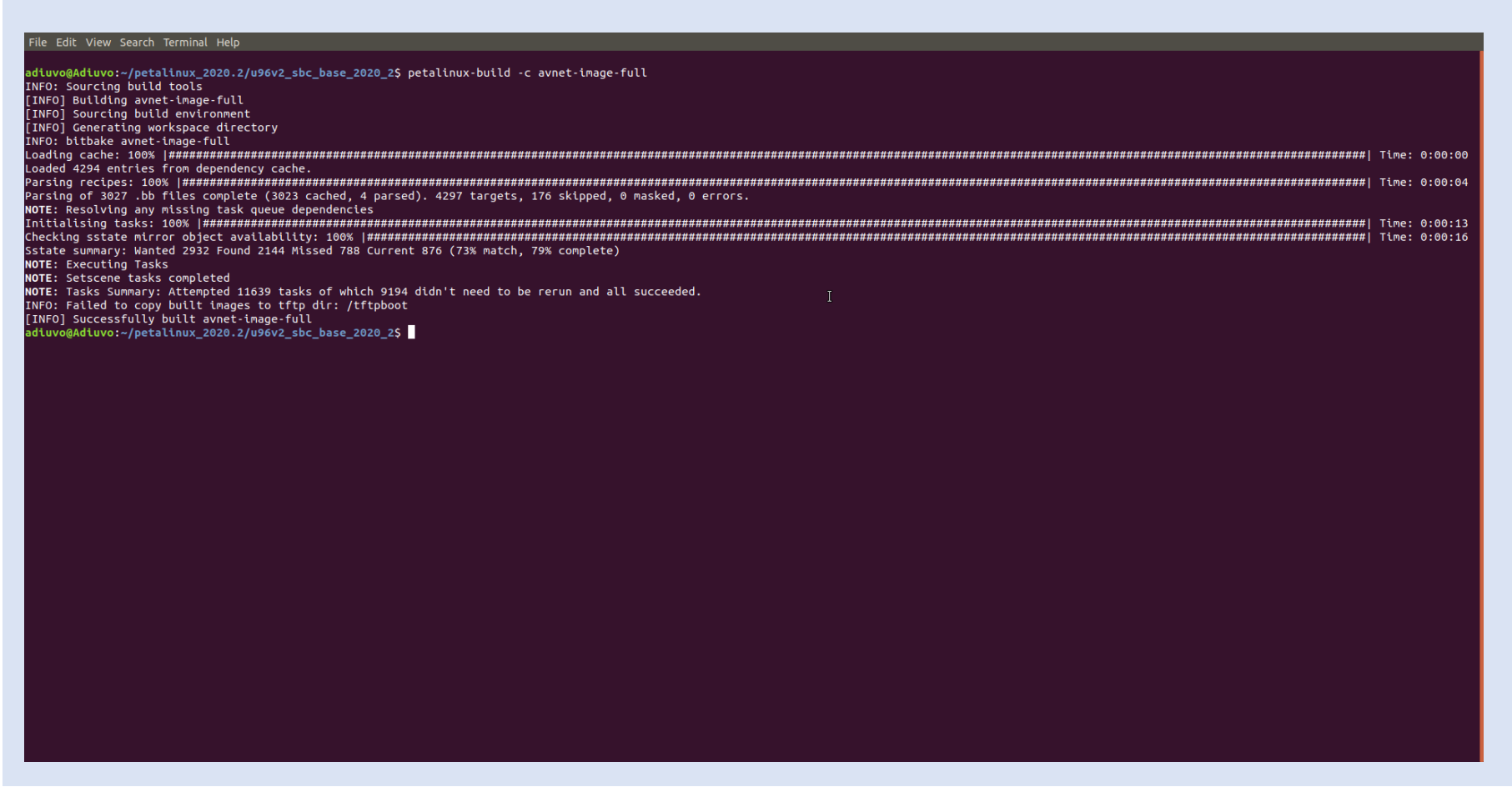
Lab 3: Creating Embedded Linux Solutions

Step 17 – Build the image by typing *petalinux-build –c avnet-image-full*



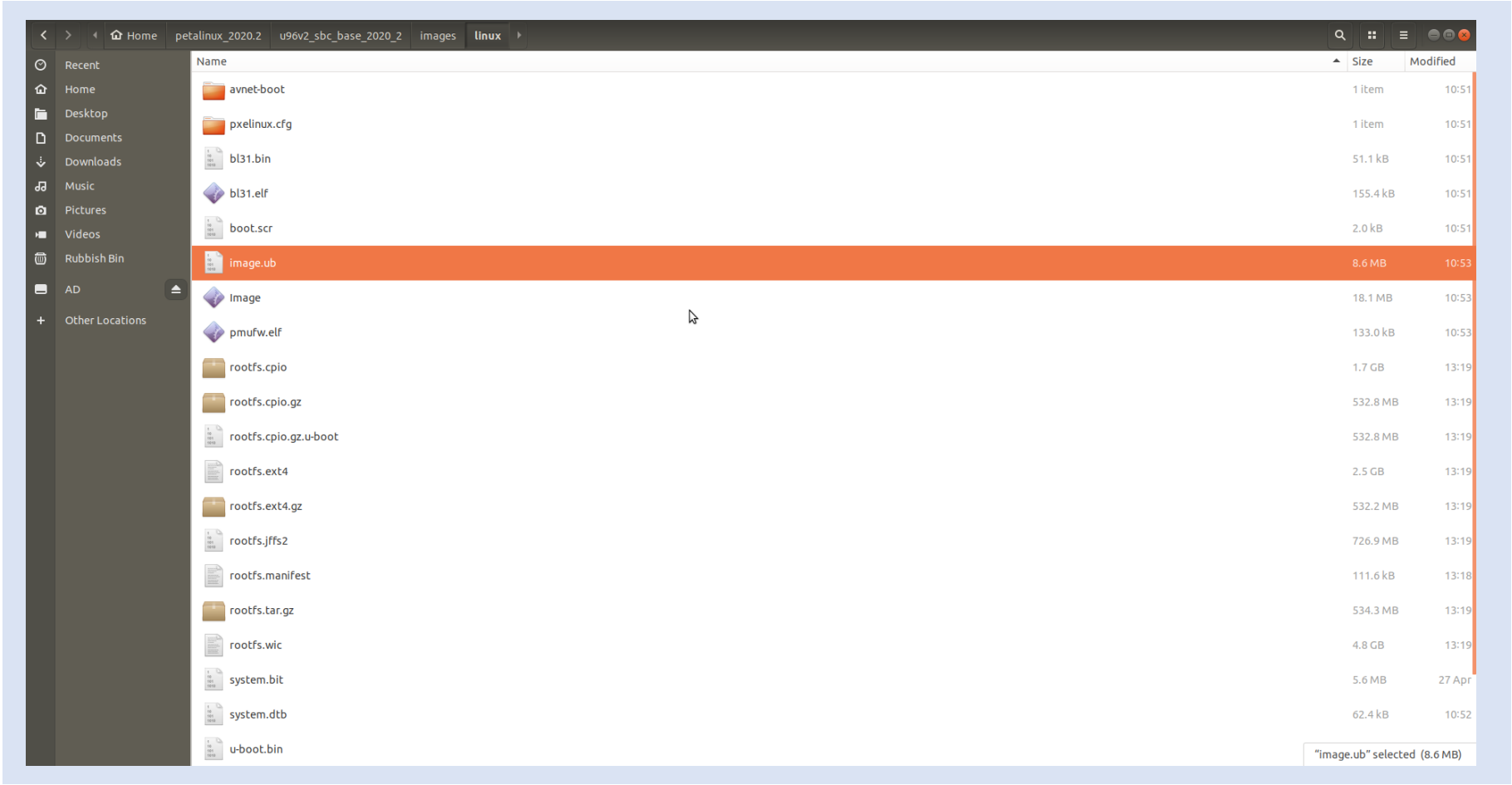
Lab 3: Creating Embedded Linux Solutions

Step 18 – Wait for the image to complete its build.



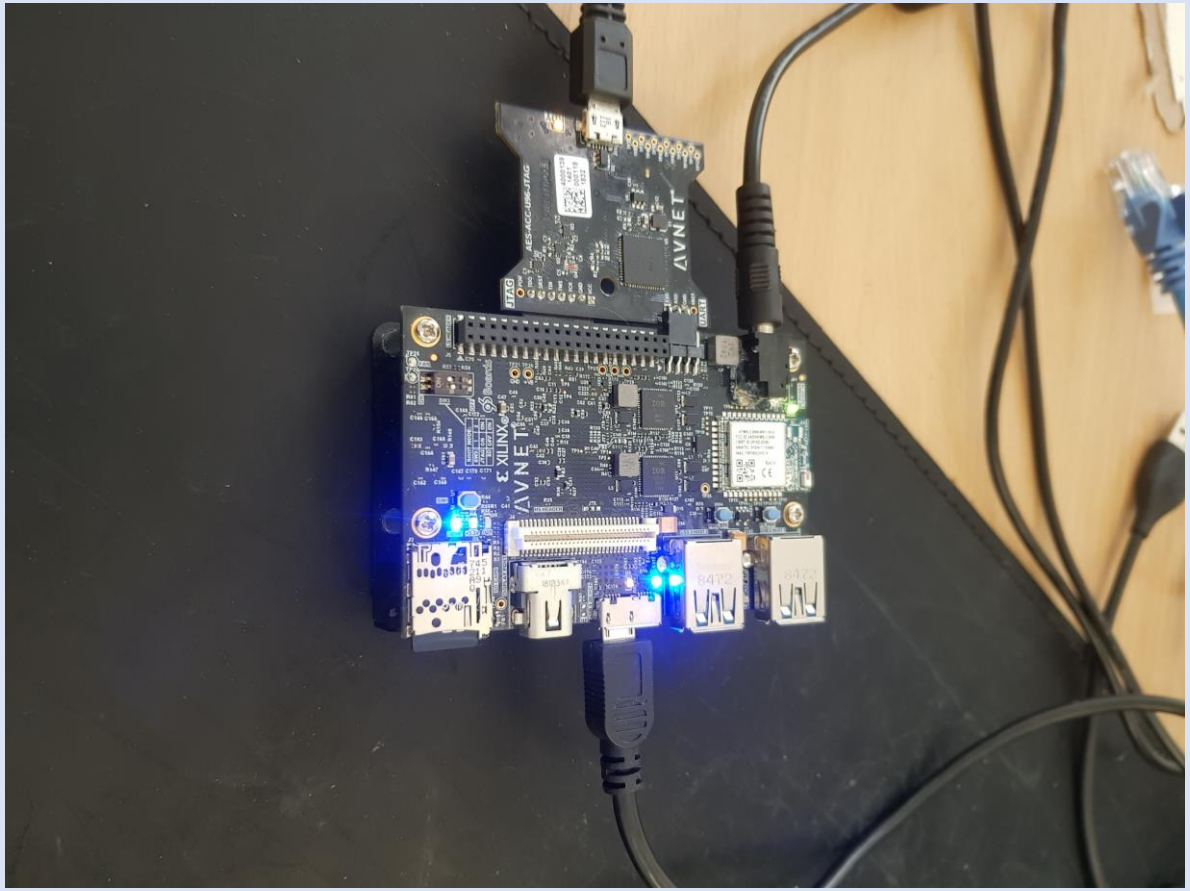
Lab 3: Creating Embedded Linux Solutions

Step 19 – From the <project>/images/linux directory copy image.ub to the SD Card boot partition



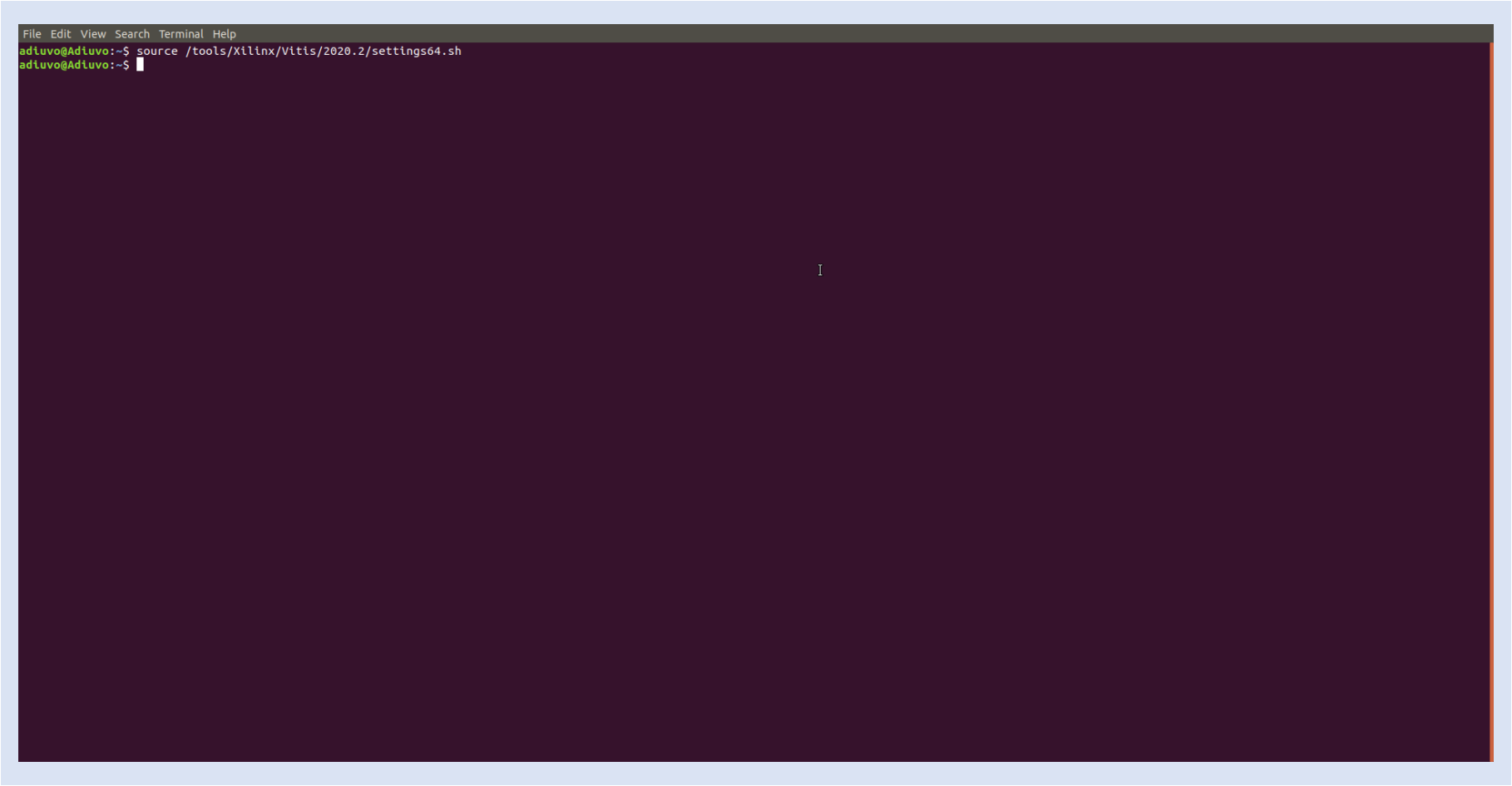
Lab 3: Creating Embedded Linux Solutions

Step 20 – Insert the SD Card into the Ultra96V2 and connect the JTAG pod, connect a USB from the USB connector on the front to your development machine and power on



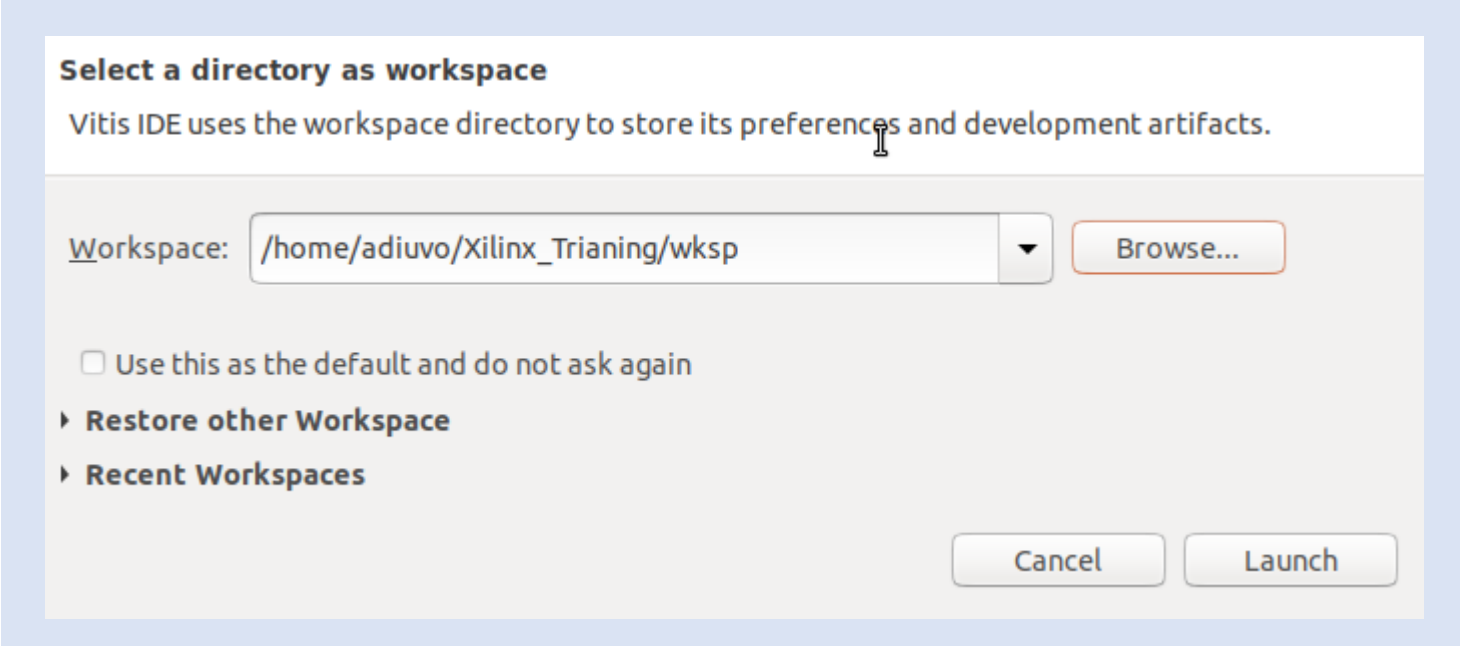
Lab 3: Creating Embedded Linux Solutions

Step 21 – On the linux development machine / virtual machine. Open a terminal and type *Source /opt/tools/Xilinx/Vitis/2020.2/settings64.sh* This assumes installation path is /opt/



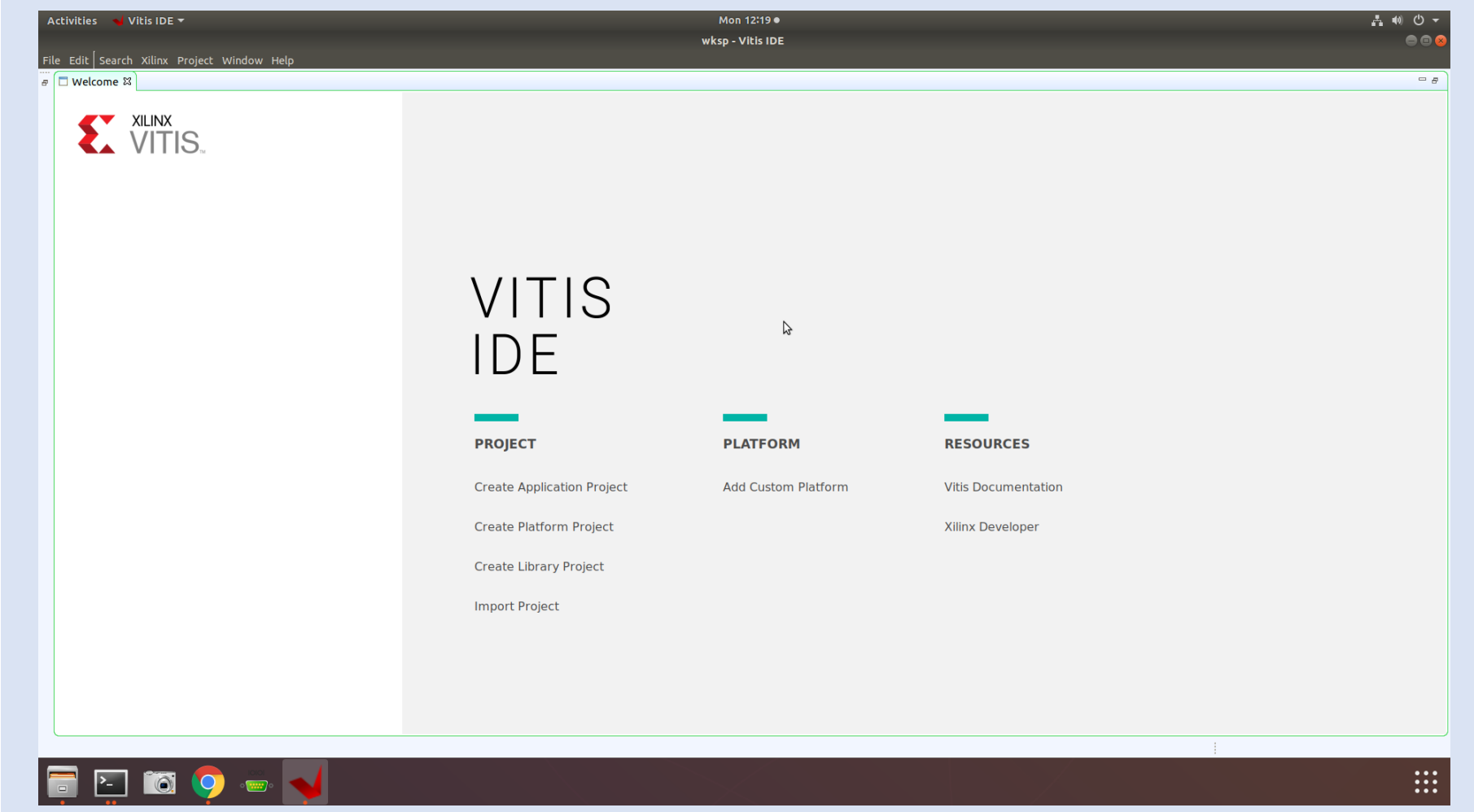
Lab 3: Creating Embedded Linux Solutions

Step 22 – Vitis will start and ask for the workspace, create a new directory using browse to store the projects.



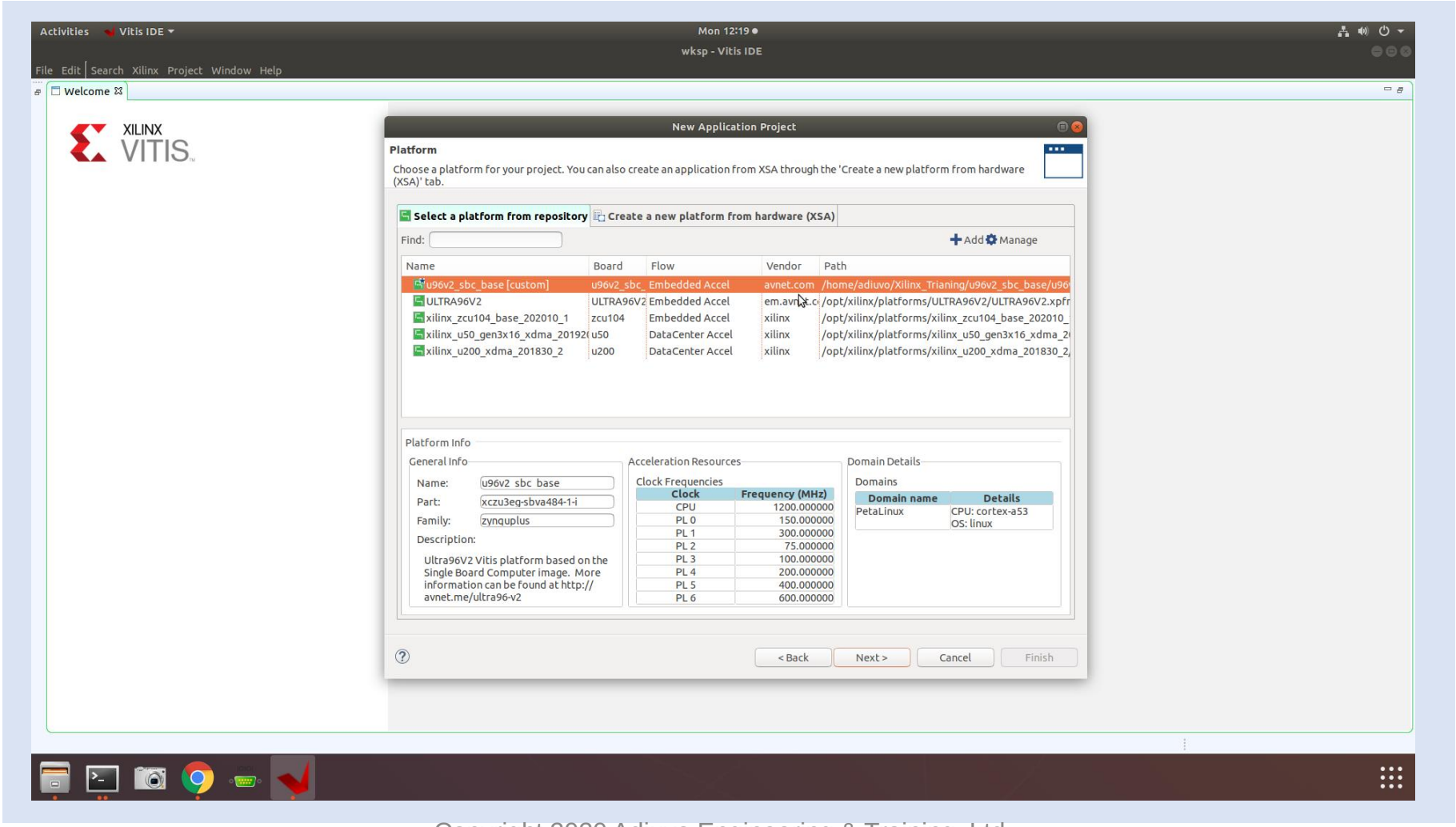
Lab 3: Creating Embedded Linux Solutions

Step 23 – Select Create Application Project



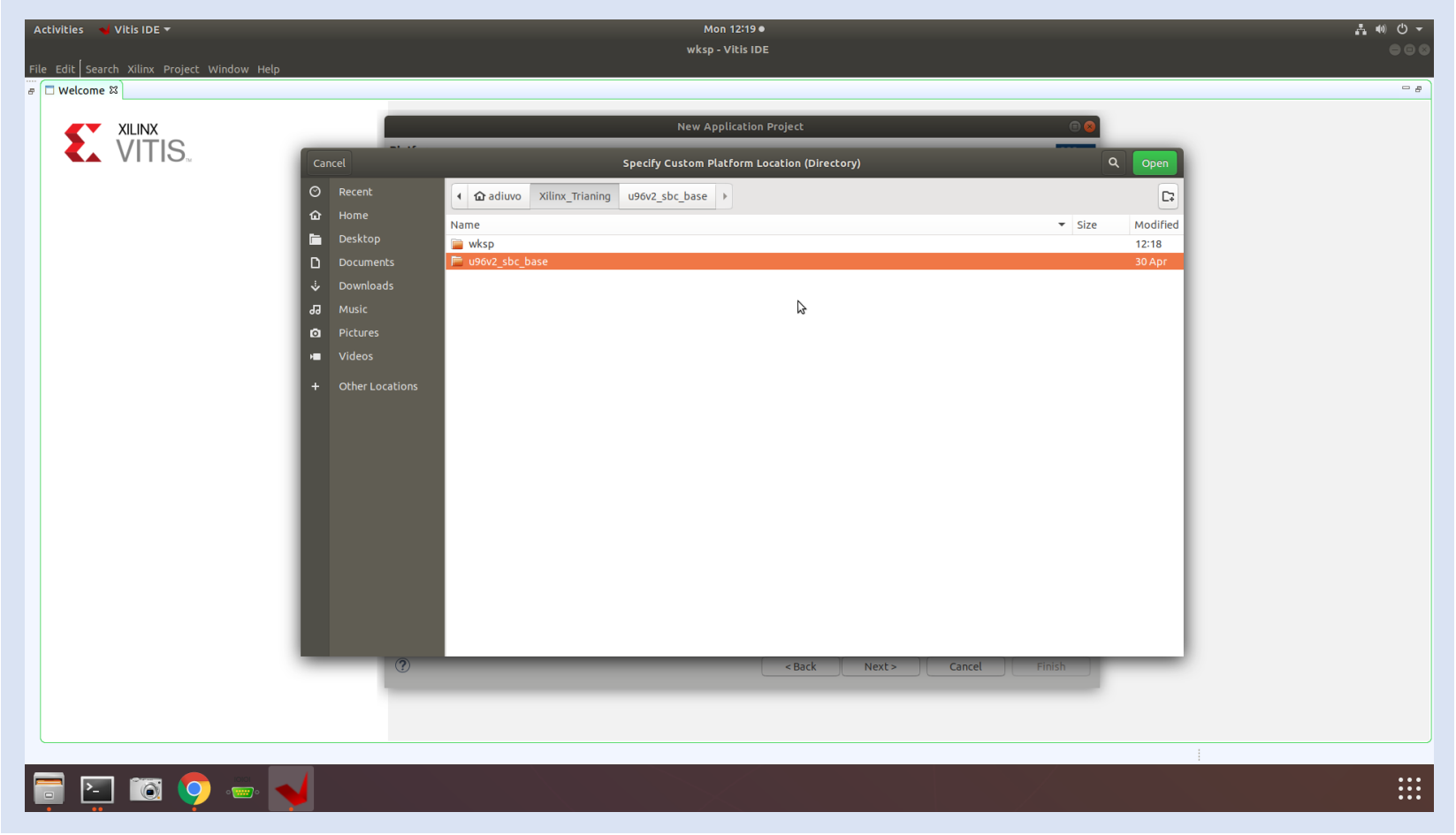
Lab 3: Creating Embedded Linux Solutions

Step 24 – On the Platform page select + Add and point it to the extracted U96 we downloaded earlier



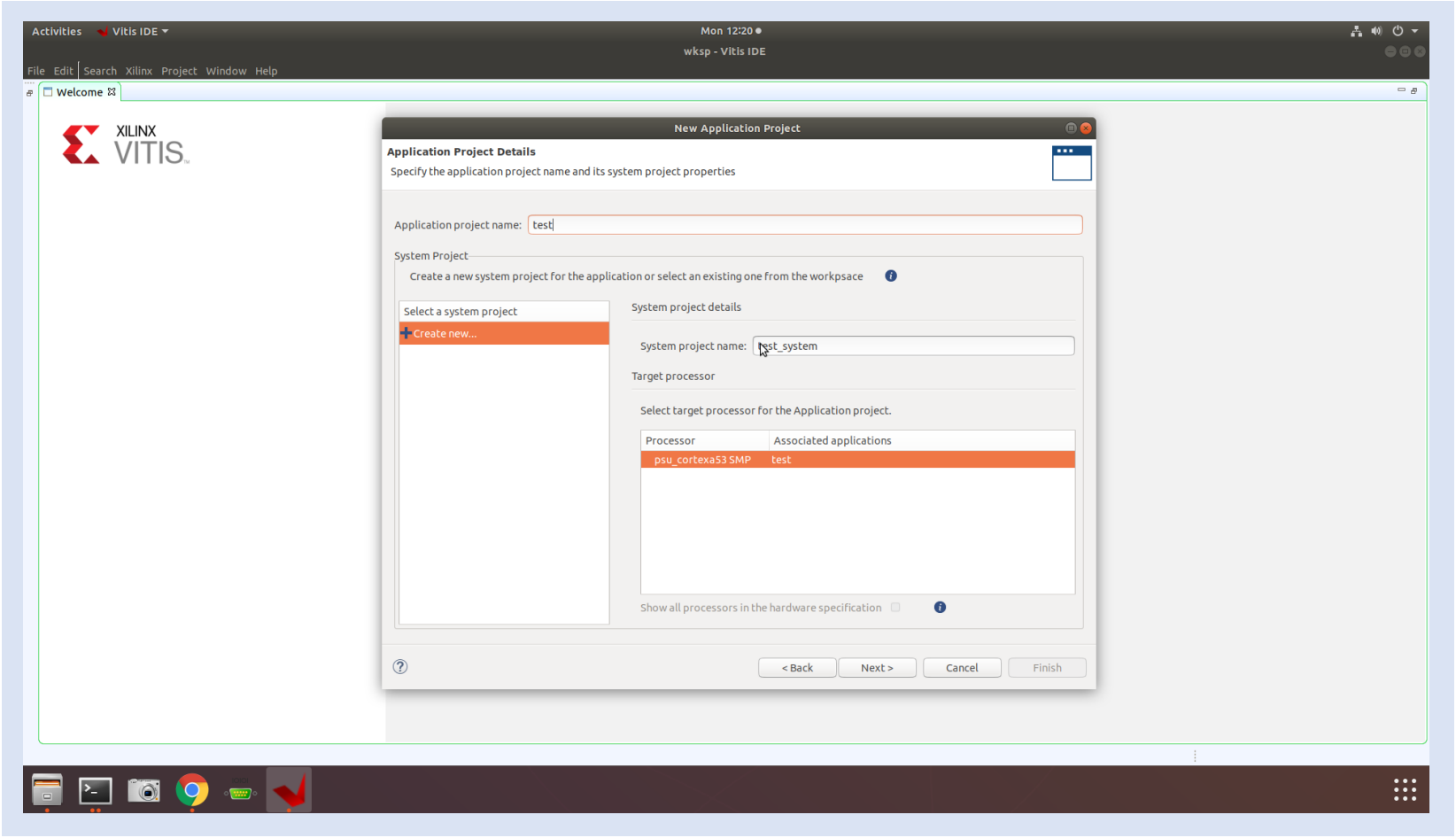
Lab 3: Creating Embedded Linux Solutions

Step 25 – Select the Top-Level directory of the Ultra96V2 Platform – Select that for use click next



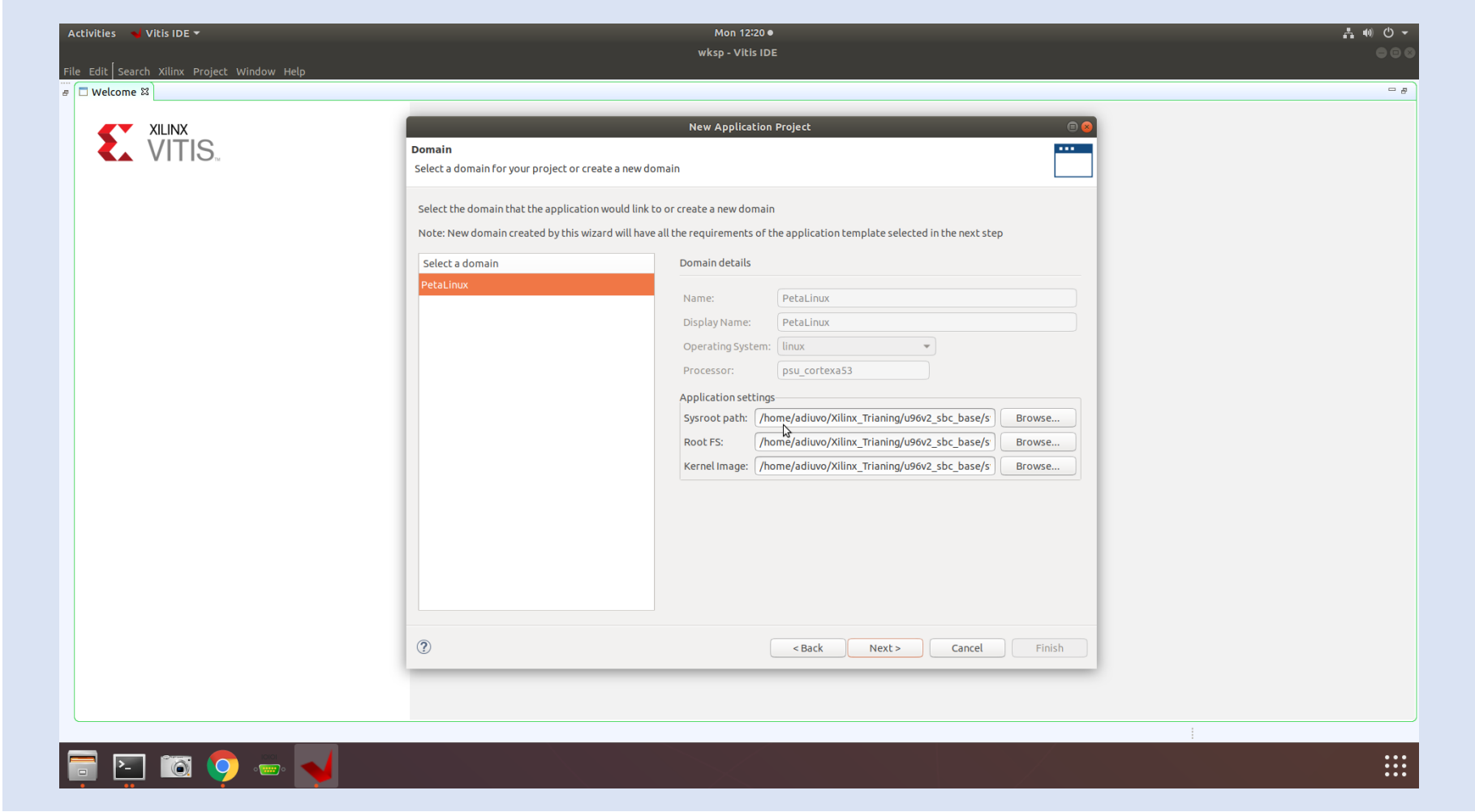
Lab 3: Creating Embedded Linux Solutions

Step 26 – Enter a name for the project



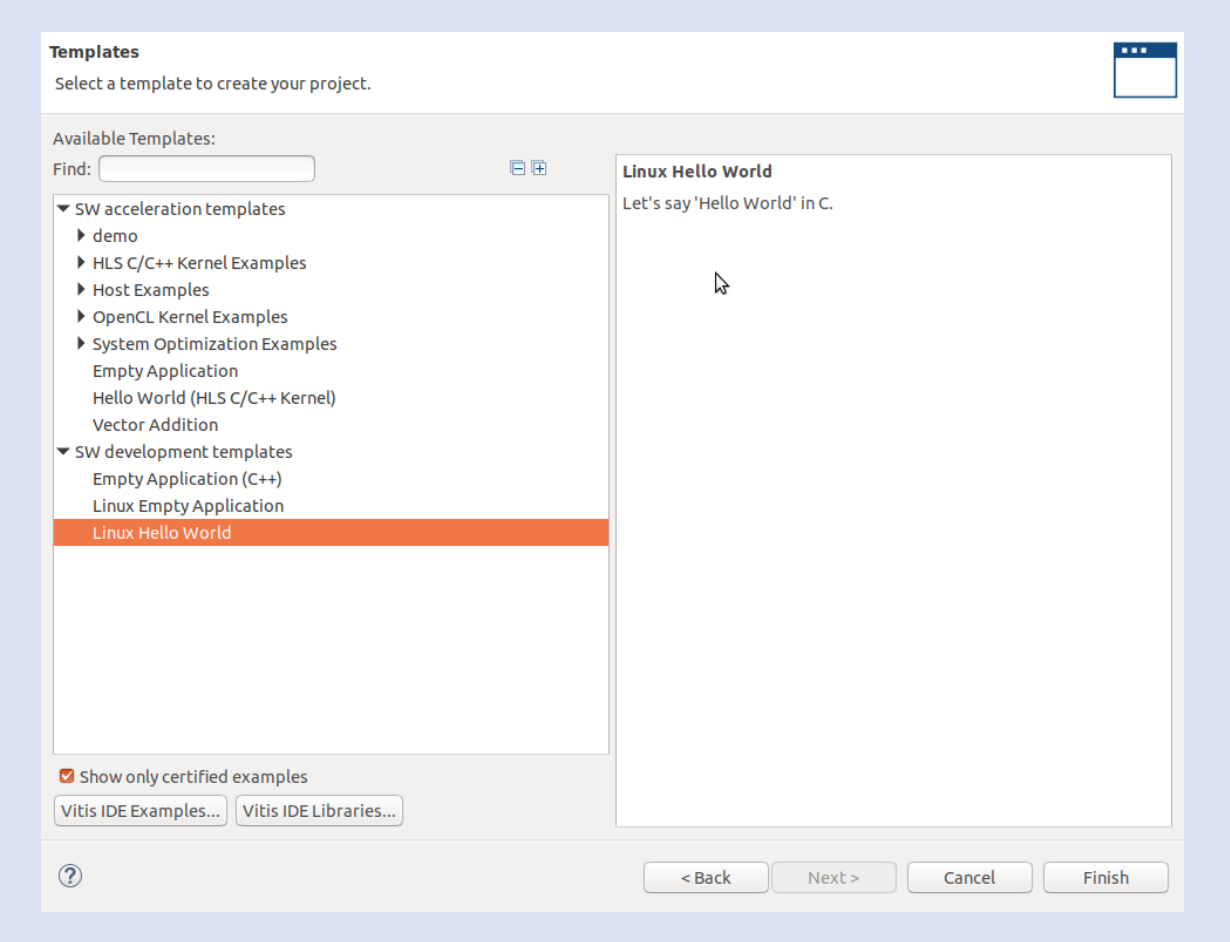
Lab 3: Creating Embedded Linux Solutions

Step 27 – Click Next on the Domain page



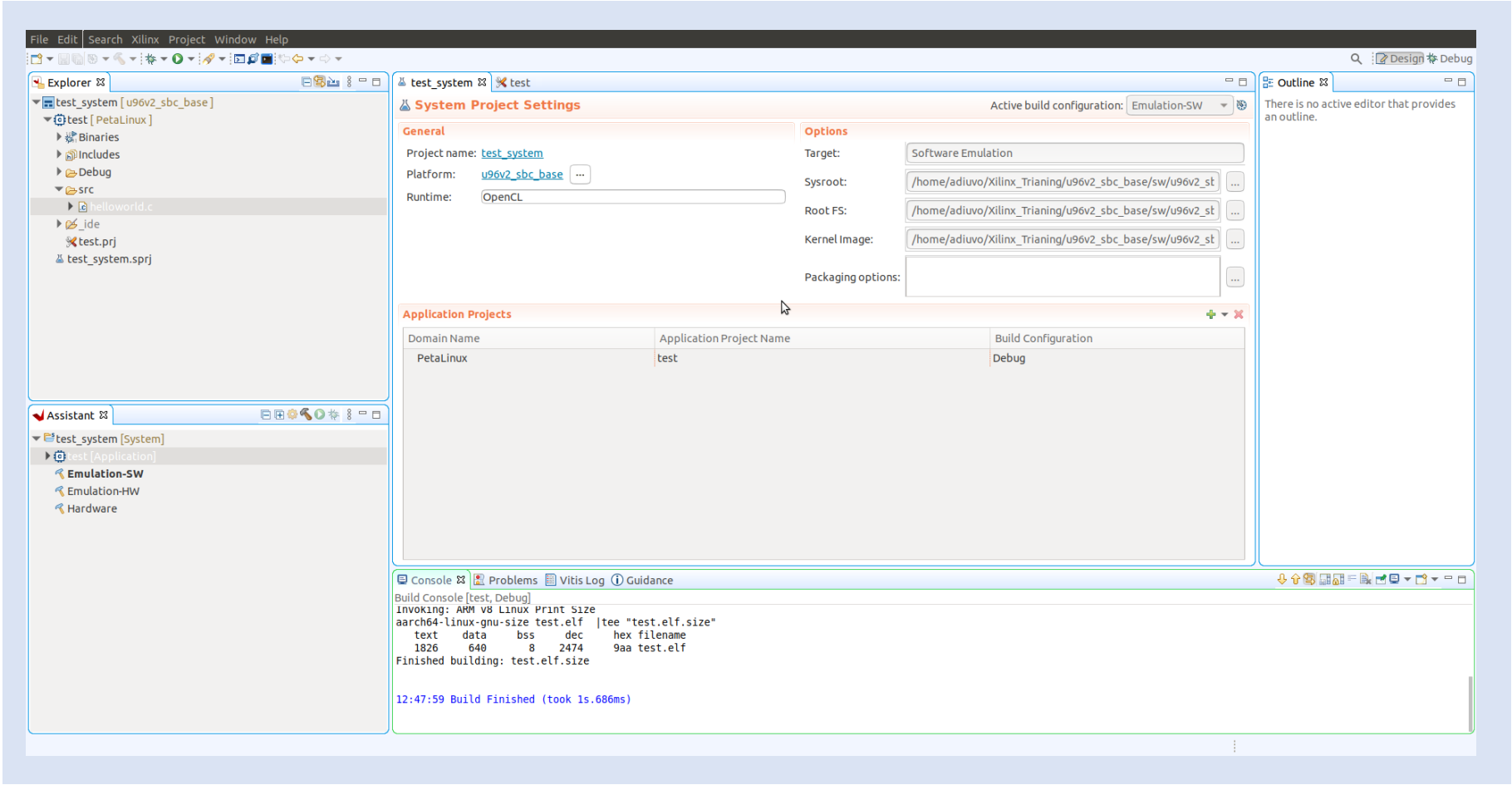
Lab 3: Creating Embedded Linux Solutions

Step 28 – Select the Linux Hello World application and click finish



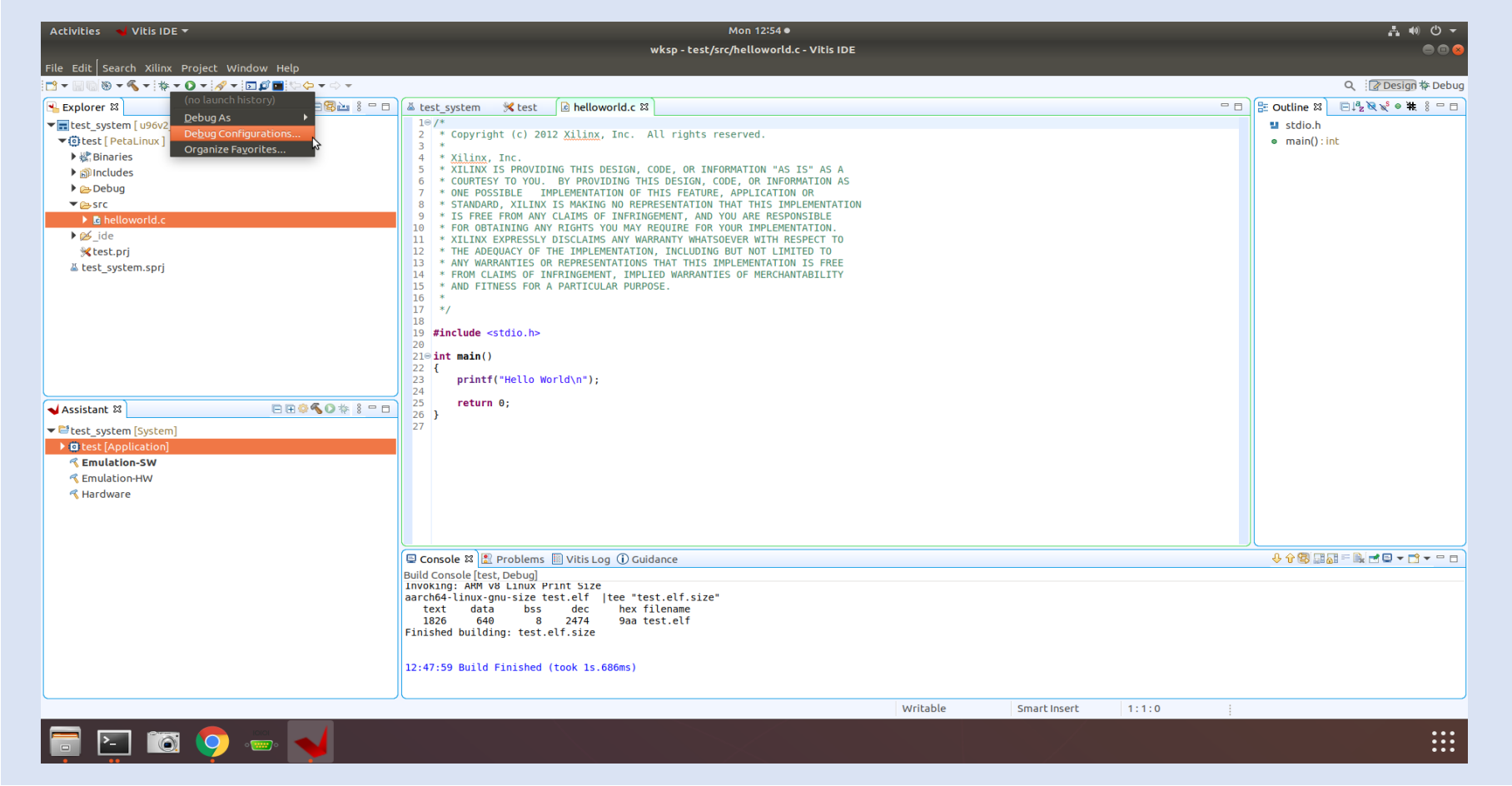
Lab 3: Creating Embedded Linux Solutions

Step 29 – Click on the Hammer to build the application



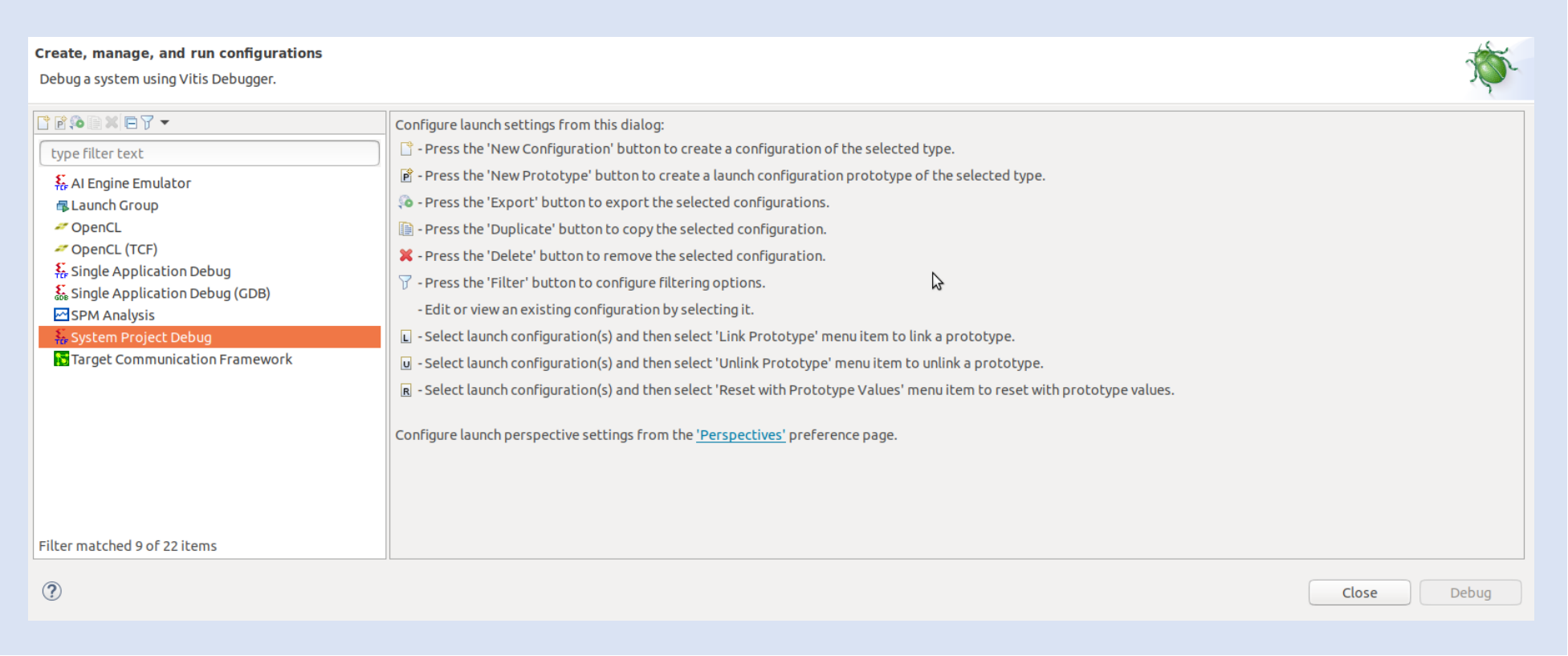
Lab 3: Creating Embedded Linux Solutions

Step 30 – Select the arrow by the green debug icon and select debug configuration



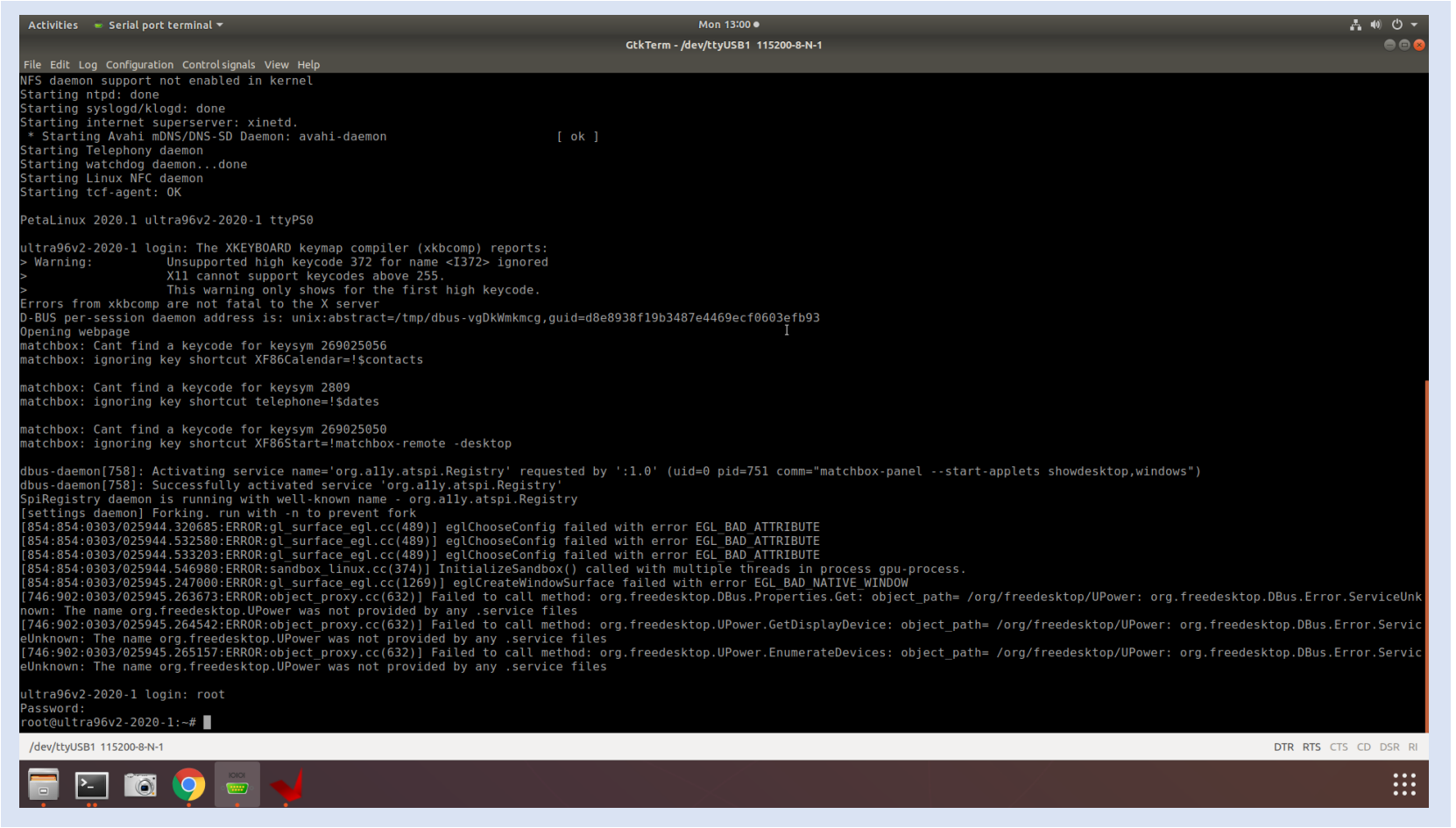
Lab 3: Creating Embedded Linux Solutions

Step 31 – Select System Project Debug and click new



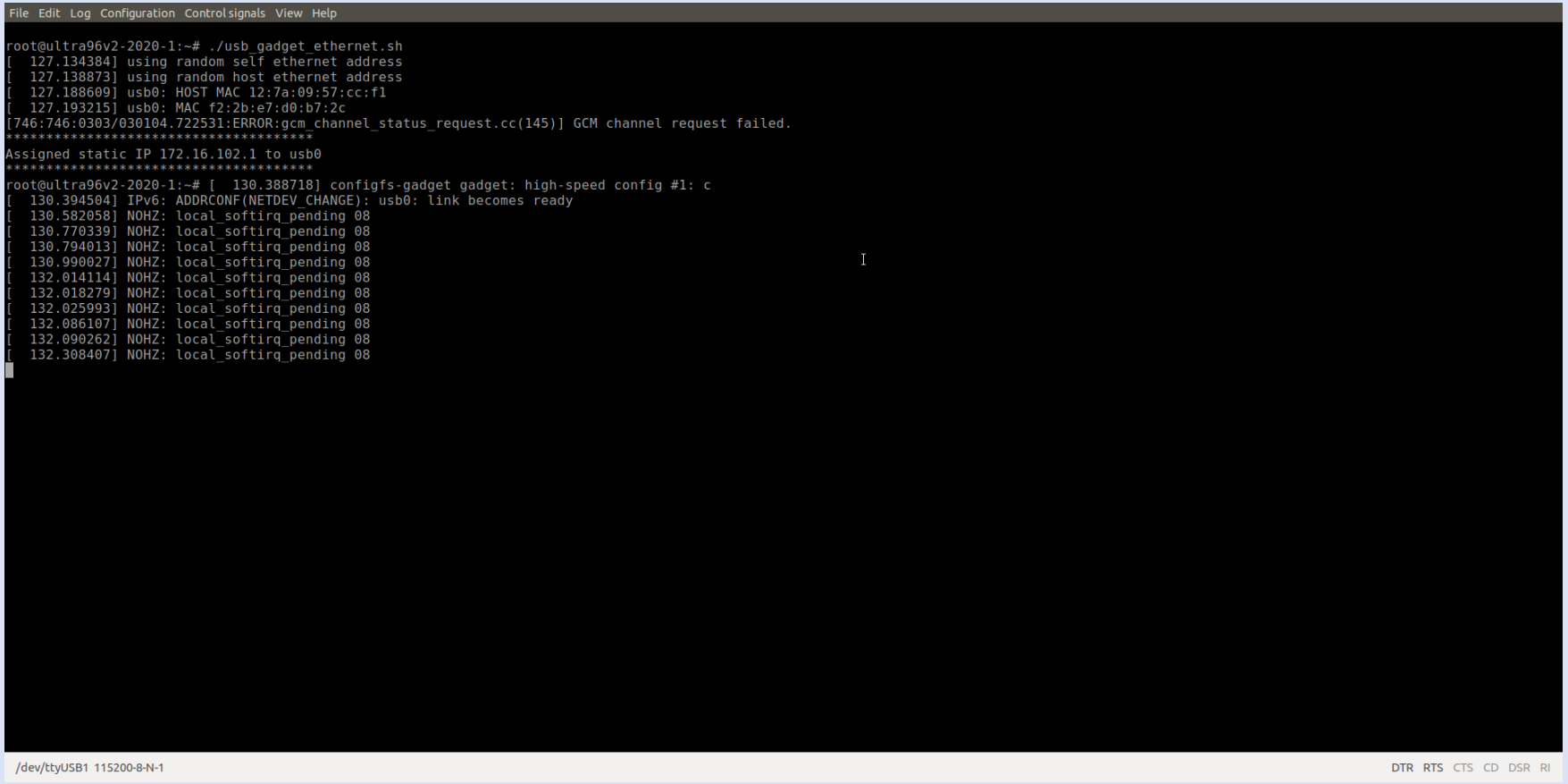
Lab 3: Creating Embedded Linux Solutions

Step 32 – In the Terminal window (115200,1,N,8) log in password and username are root



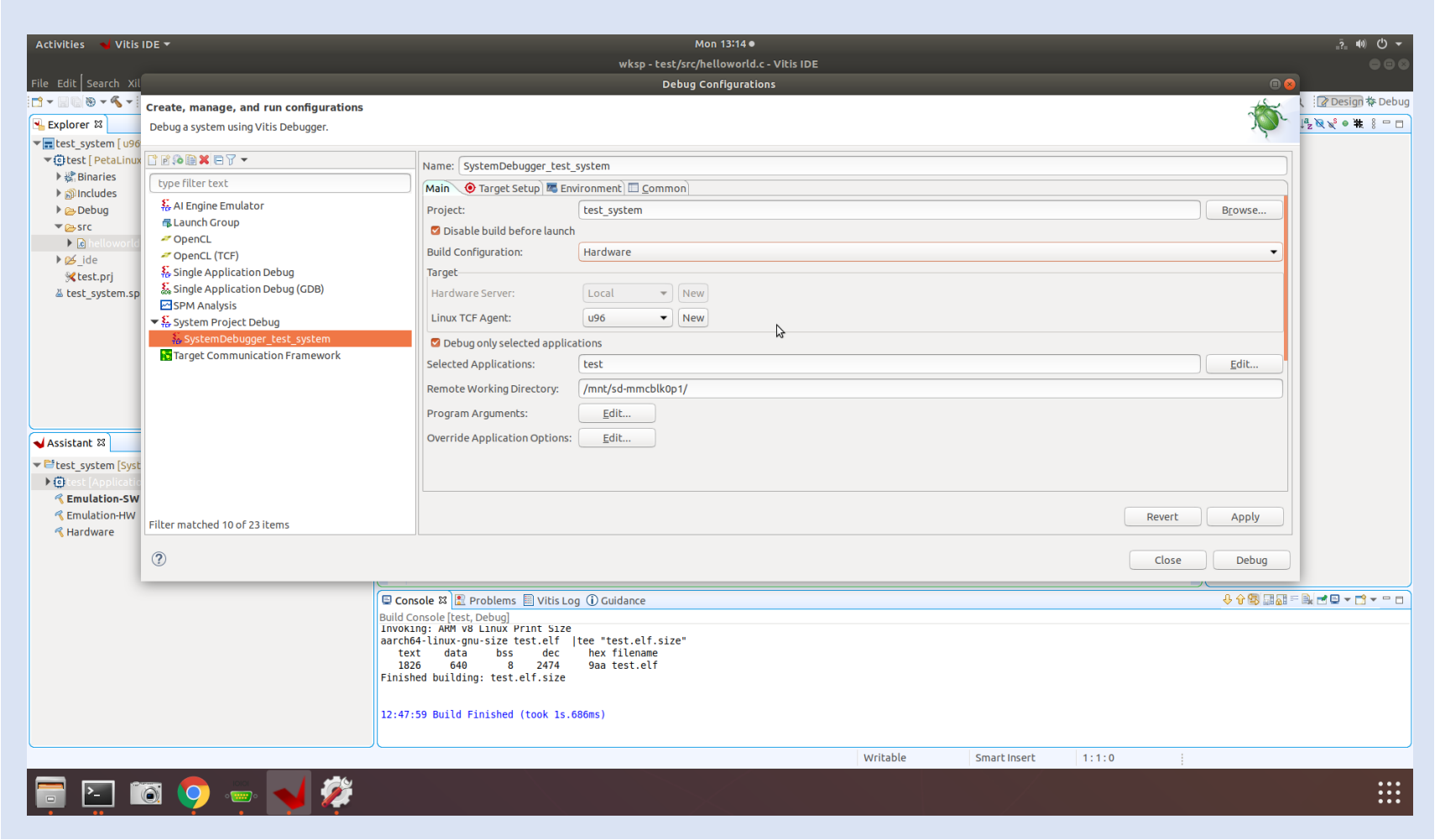
Lab 3: Creating Embedded Linux Solutions

Step 33 – Enter the command *./usb_gadget_ethernet.sh* you may need to use VI to edit IP address to align with your network. By default, it is 192.168.3.1



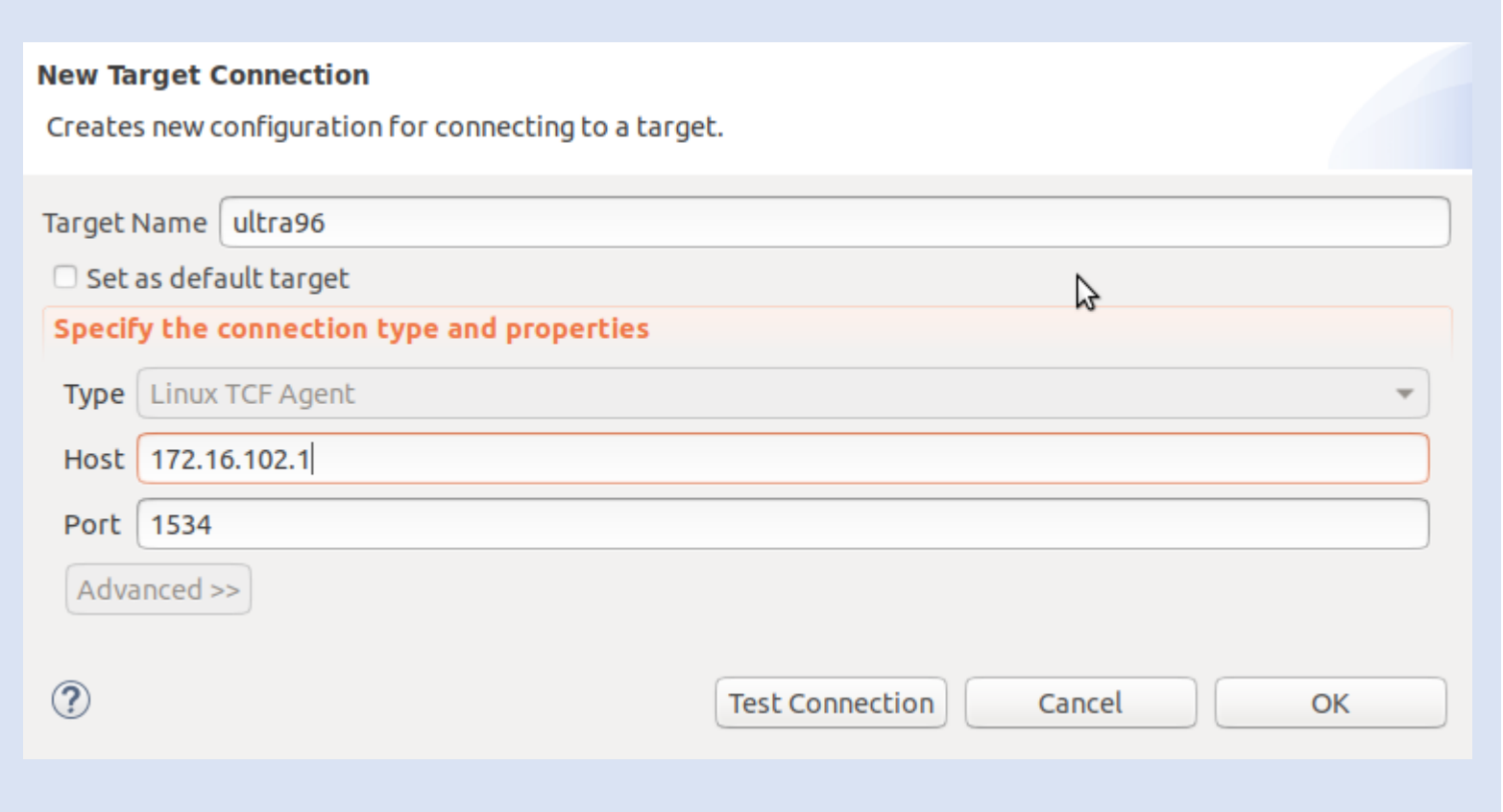
Lab 3: Creating Embedded Linux Solutions

Step 34 – Change the build configuration to Hardware and for Linux TCF Agent select new



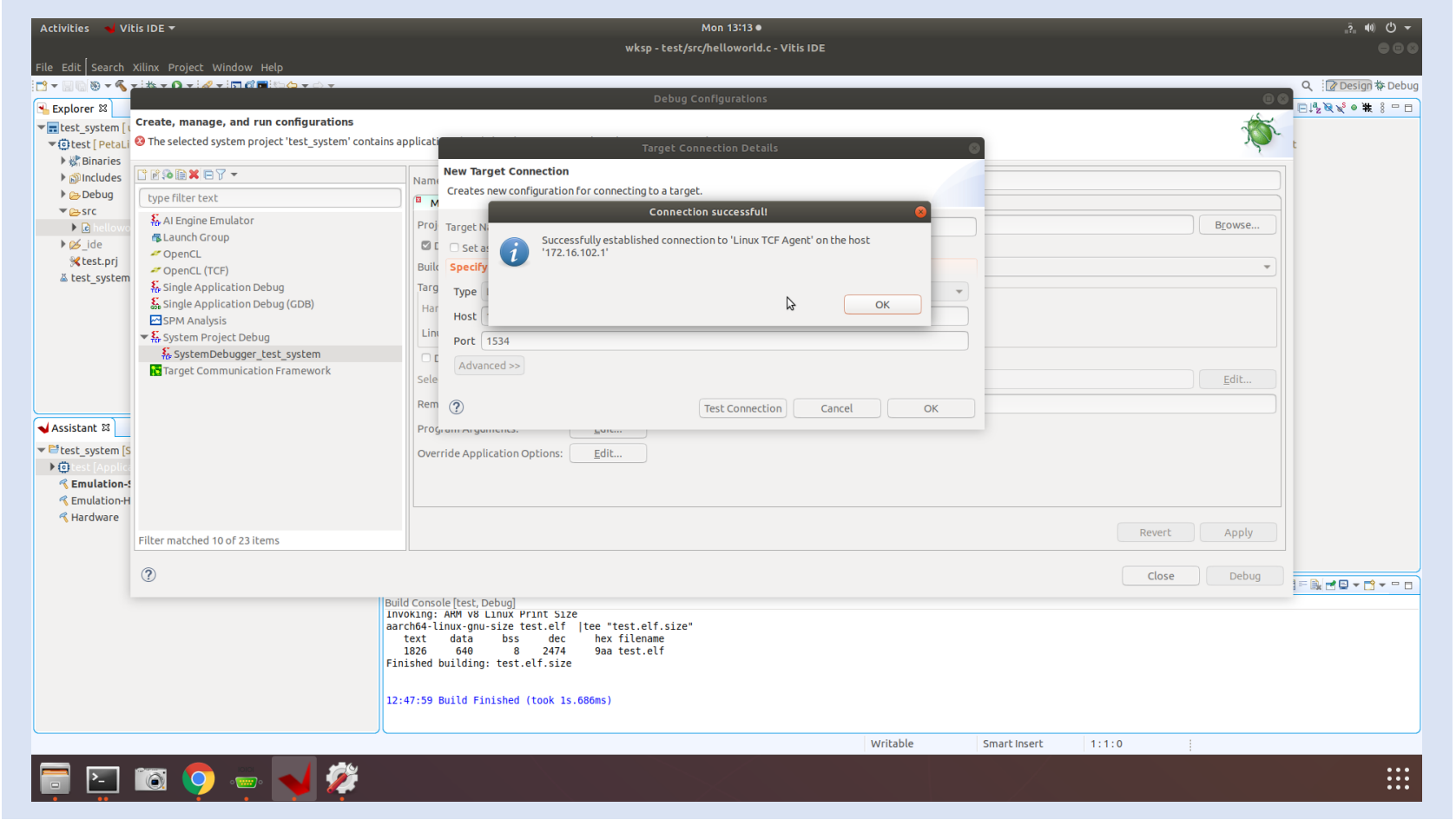
Lab 3: Creating Embedded Linux Solutions

Step 35 – Enter a name, and enter the USB Gadget Ethernet IP address – Click test connection to verify the board can be seen.



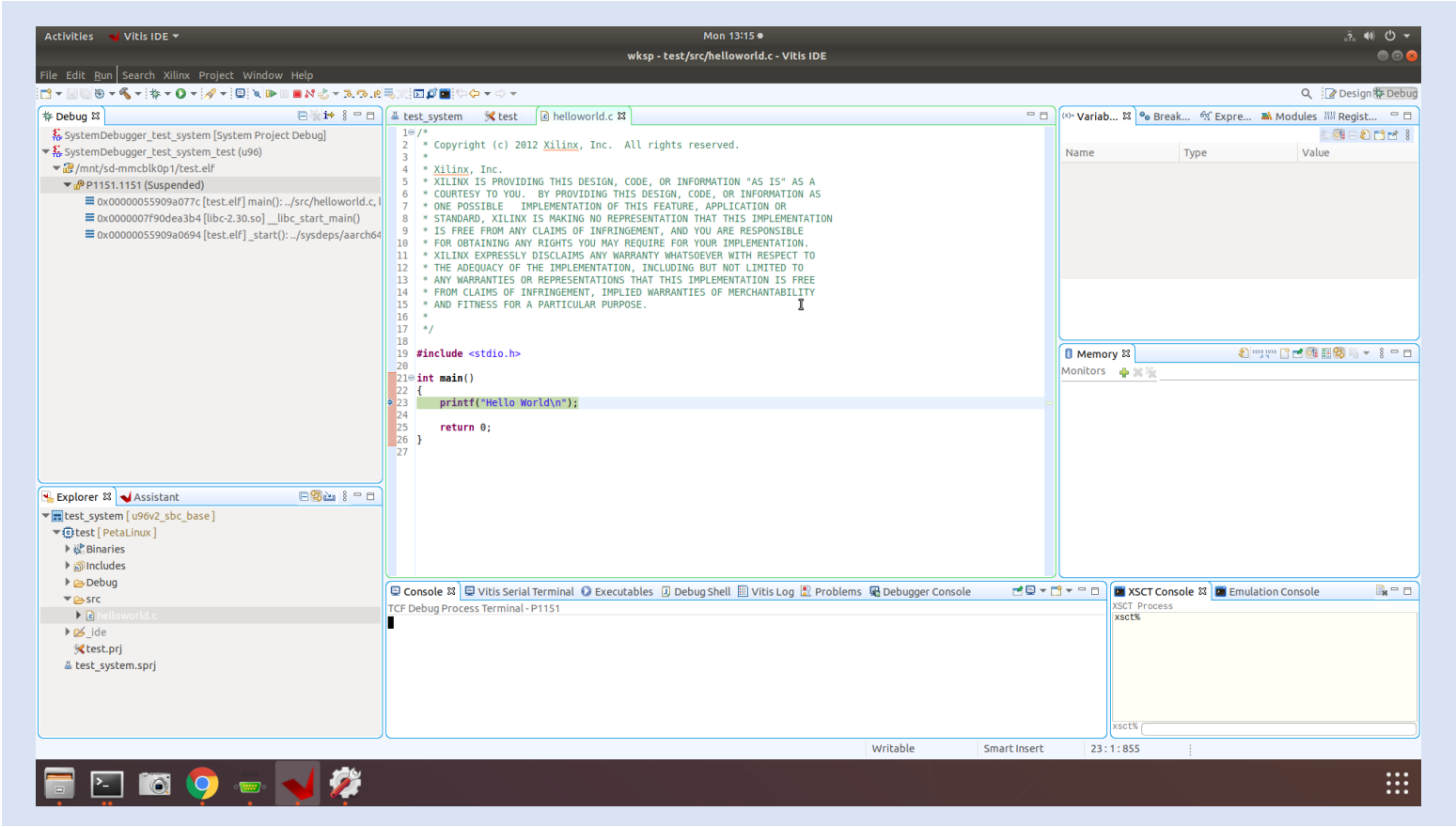
Lab 3: Creating Embedded Linux Solutions

Step 36 – Click OK when the connection is successful, click OK on the target connection and the click debug



Lab 3: Creating Embedded Linux Solutions

Step 37 – The application will be downloaded to the target and paused for execution, press the run button



Lab 3: Creating Embedded Linux Solutions

Step 38 – Click on the run button and you will see the hello world appear in the console.