Sensor Validation Setup - Android

Rev. A1

30 July 2014

**Sensor Validation Setup**

**For Intersil Corporation, USA**

**VVDN Contact:**

**Name:** Murali Mohan

**Email:** [murali.m@vvdntech.com](mailto:murali.m@vvdntech.com)

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev No.** | **Description** | **By** |
| 30-July-2014 | A1 | Sensor validation setup for android | VVDN |
|  |  |  |  |

***Table of Contents***

[1 Introduction 4](#_Toc394517624)

[1.1 Reference documents 4](#_Toc394517625)

[2 Software setup 5](#_Toc394517626)

[2.1 Build environment 5](#_Toc394517627)

[2.2 Downloading android source 6](#_Toc394517628)

[2.3 Building android 21](#_Toc394517629)

[2.4 Flashing Images to SD card 21](#_Toc394517630)

[2.4.1 Disable automount in Ubuntu 21](#_Toc394517631)

[2.4.2 Install Linaro image tools 21](#_Toc394517632)

[2.4.3 Flash images to SD card 22](#_Toc394517633)

[2.4.4 Install graphics libraries 22](#_Toc394517634)

[2.4.5 Restore automount in Ubuntu 22](#_Toc394517635)

# Introduction

This document describes the process of setting up an Android Build Server (would be setup at Intersil, Milpitas premises) in order to generate binaries to be flashed to SD card used to boot Pandaboard reference board. The binaries generated after integration of android sensor driver would then be used to validate the sensor features.

This document is made for the reference of

* Product managers at VVDN / Intersil and Quality Assurance Department to understand the sensor validation setup
* Engineering Team at VVDN / Intersil for recreating the sensor validation setup using this guide

## Reference documents

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Description** | **Revision** | **Date** |
| 1 | ISL29177\_AndroidDriver\_IntegrationGuide\_A1.pdf | A1 | 30 Jul 2014 |
|  |  |  |  |
|  |  |  |  |

# Software setup

## Build environment

|  |  |
| --- | --- |
| **Operating System** | Ubuntu 13.10 64-bit  Linux Kernel 3.2.0-54-generic |

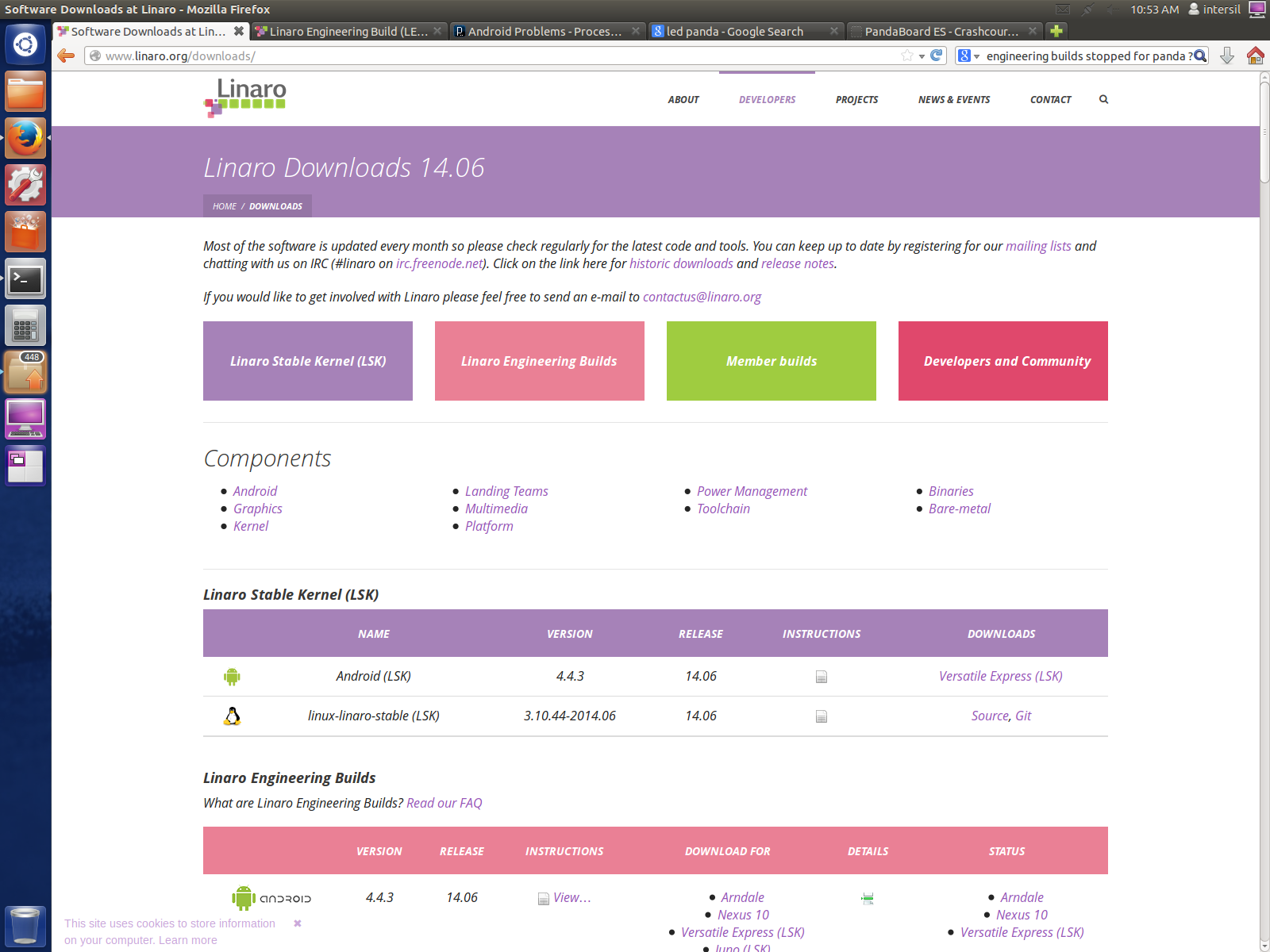
|  |
| --- |
| **IMPORTANT NOTE:** Initially the build operating system that already existed in Android build server at Intersil , Milpitas has Ubuntu 13.10 64-bit. As it is not an LTS version any future support may not be available for this operating system. Once the support period for 13.10 is over we may need to migrate to Ubuntu 12.04 (upto 2017) or 14.04 (upto 2019) versions of Ubuntu. |

## 

## Downloading android source

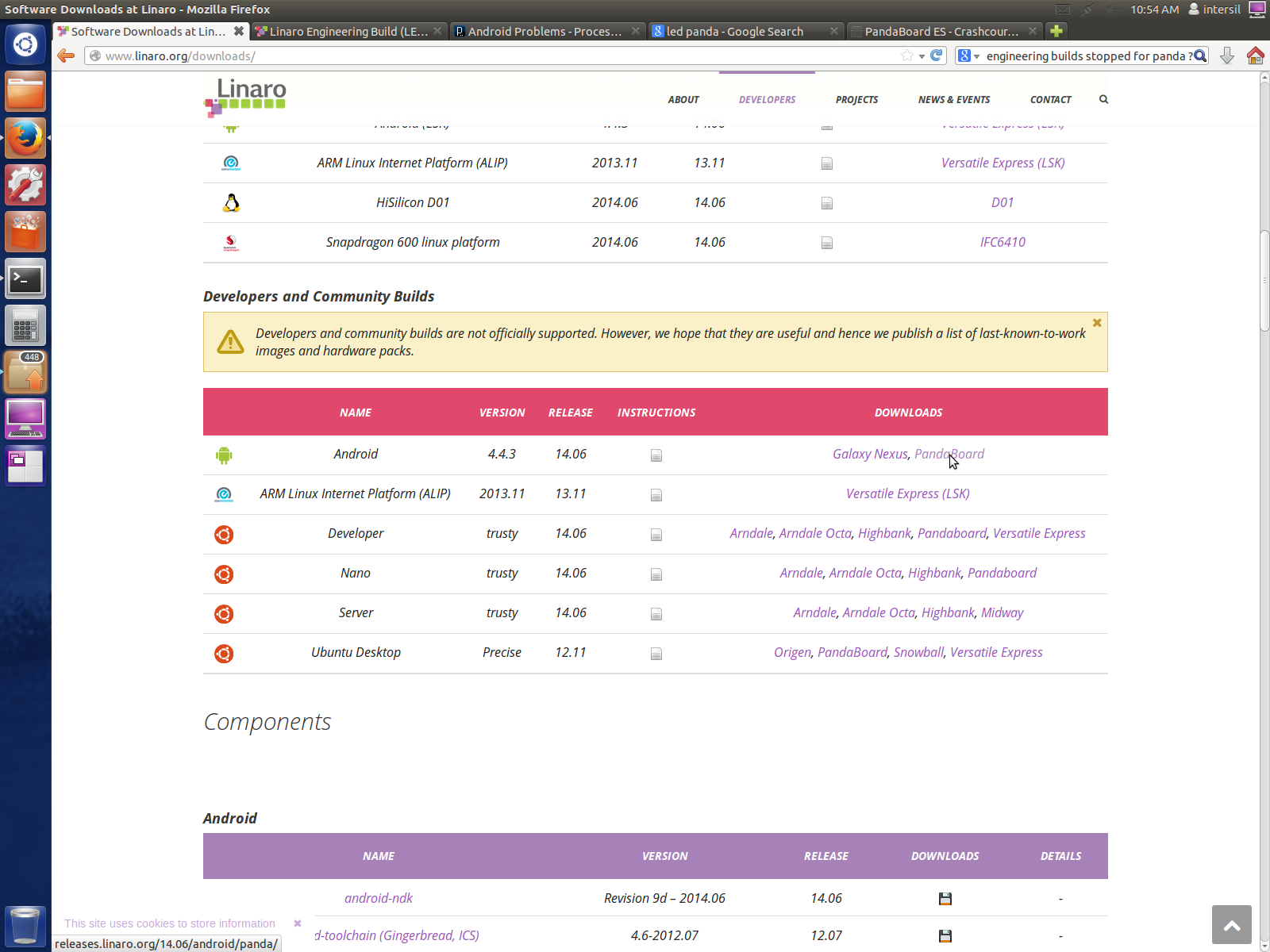
**STEP1 :** Go to the url <http://www.linaro.org/downloads> from any browser. You should see a webpage as shown below.

**Figure :** Linaro download page



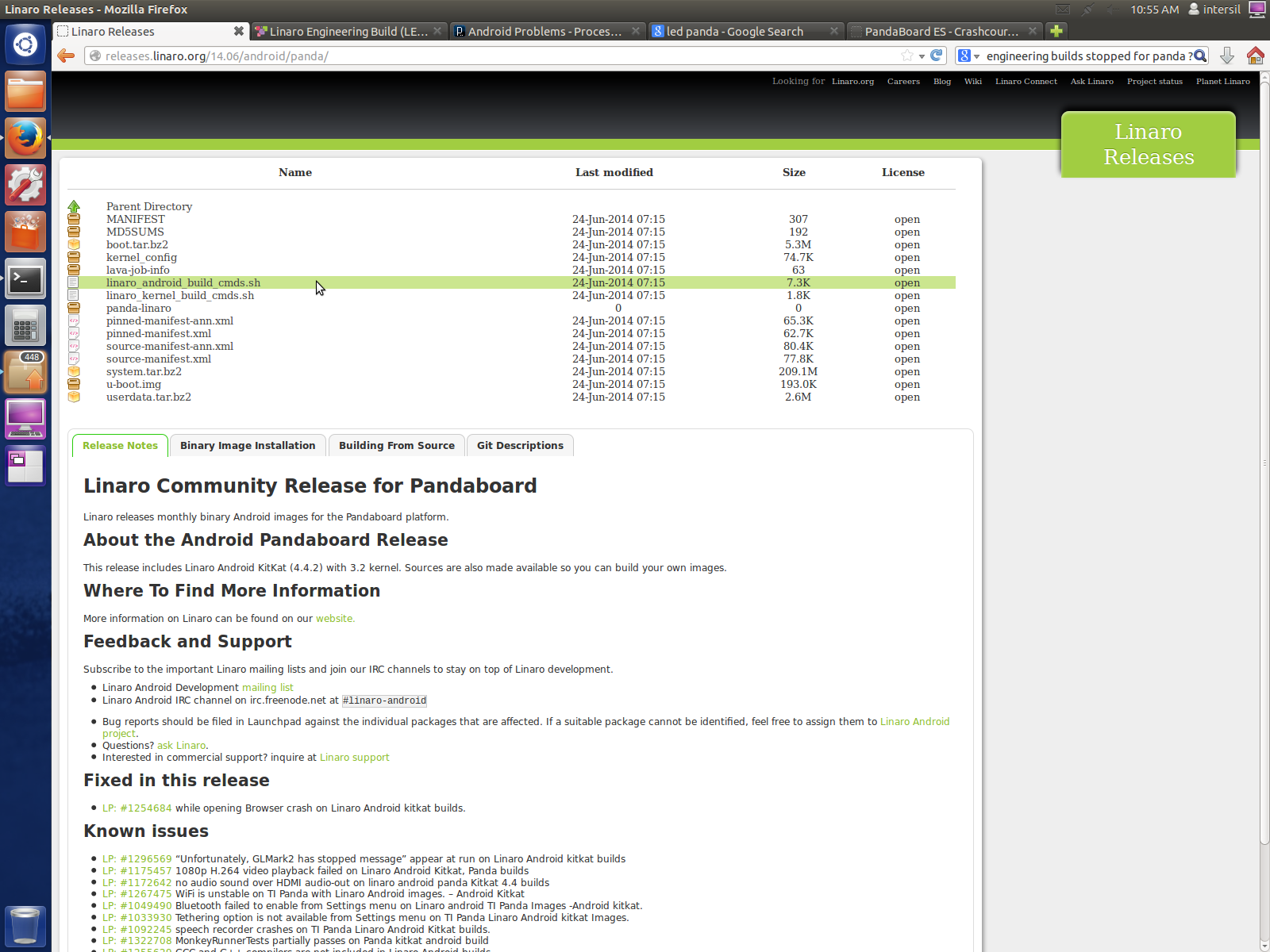
**STEP 2:** Scroll down the webpage to the “***Developer and Community Builds***” section and click on the link for Pandaboard as shown in below image. It would navigate to the webpage containing information about recent android source code released for Pandaboard by Linaro.

**Figure :** Linaro download page - Section for Pandaboard



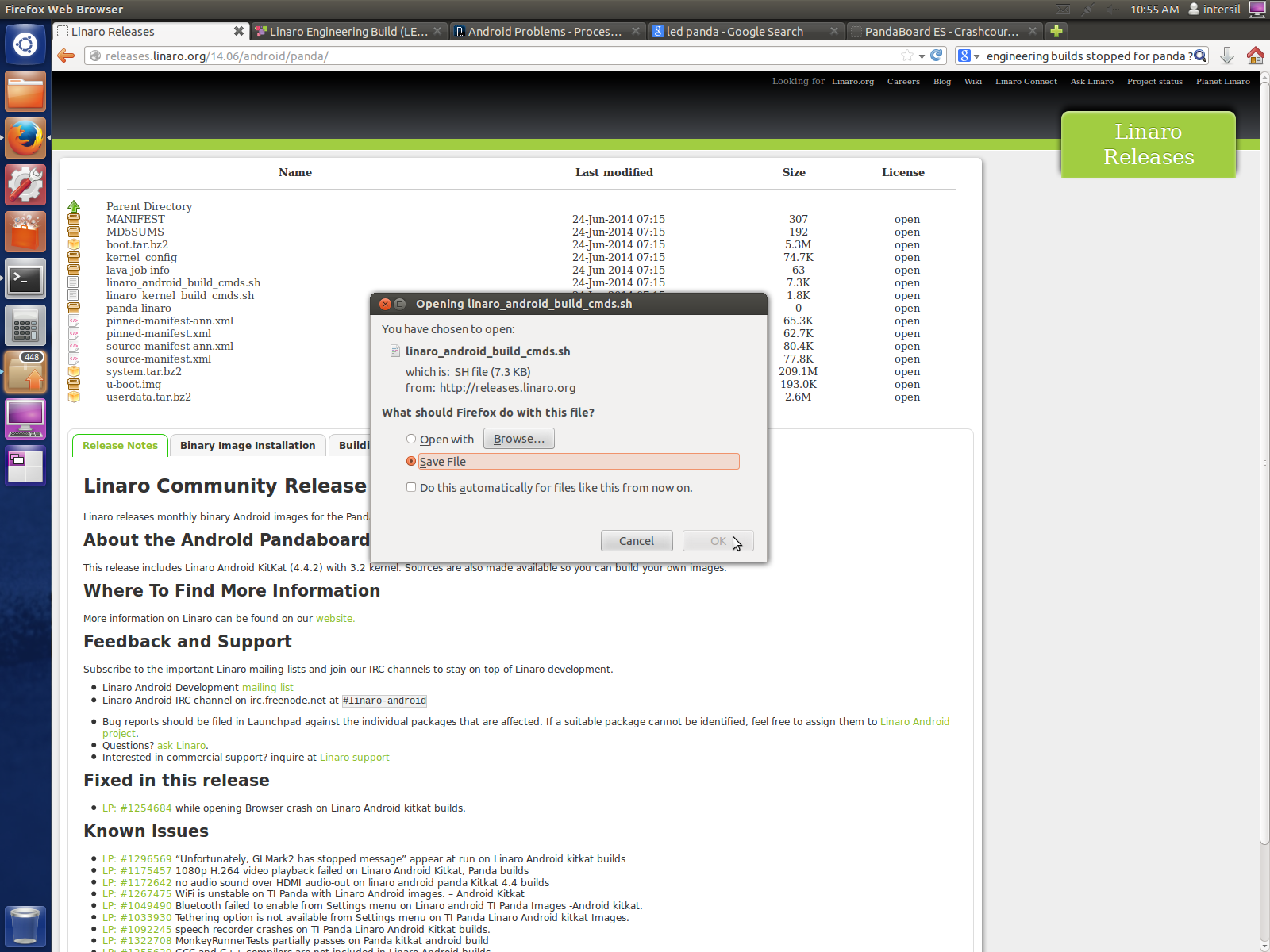
**STEP 3:** The below page has information and files related to the latest android release for Pandaboard.

**Figure :** Release package for Pandaboard android

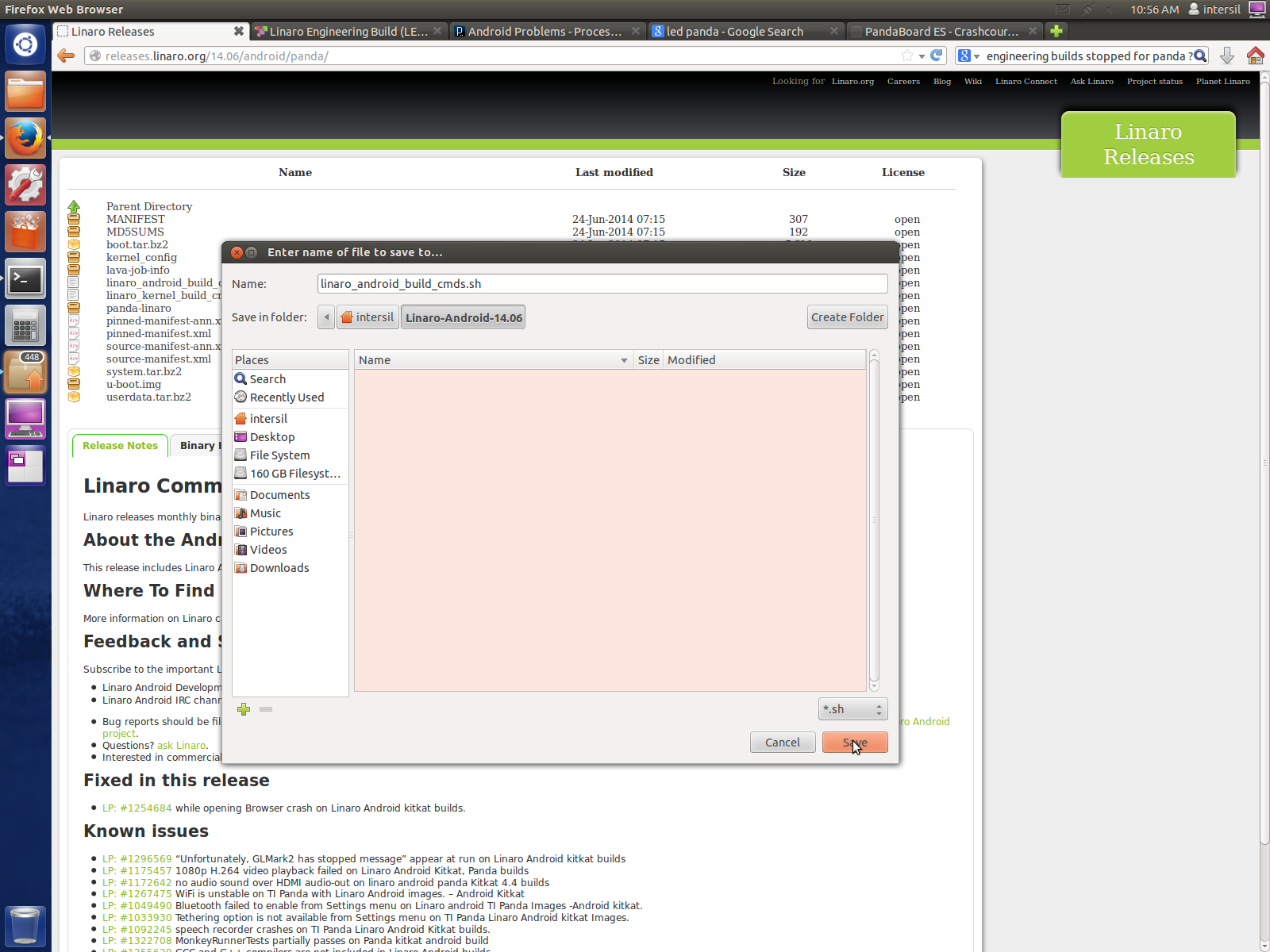


**STEP 4:** Click on linaro\_android\_build\_cmds.sh and save the script to an appropriate folder (for example Linaro-Android-14.06)

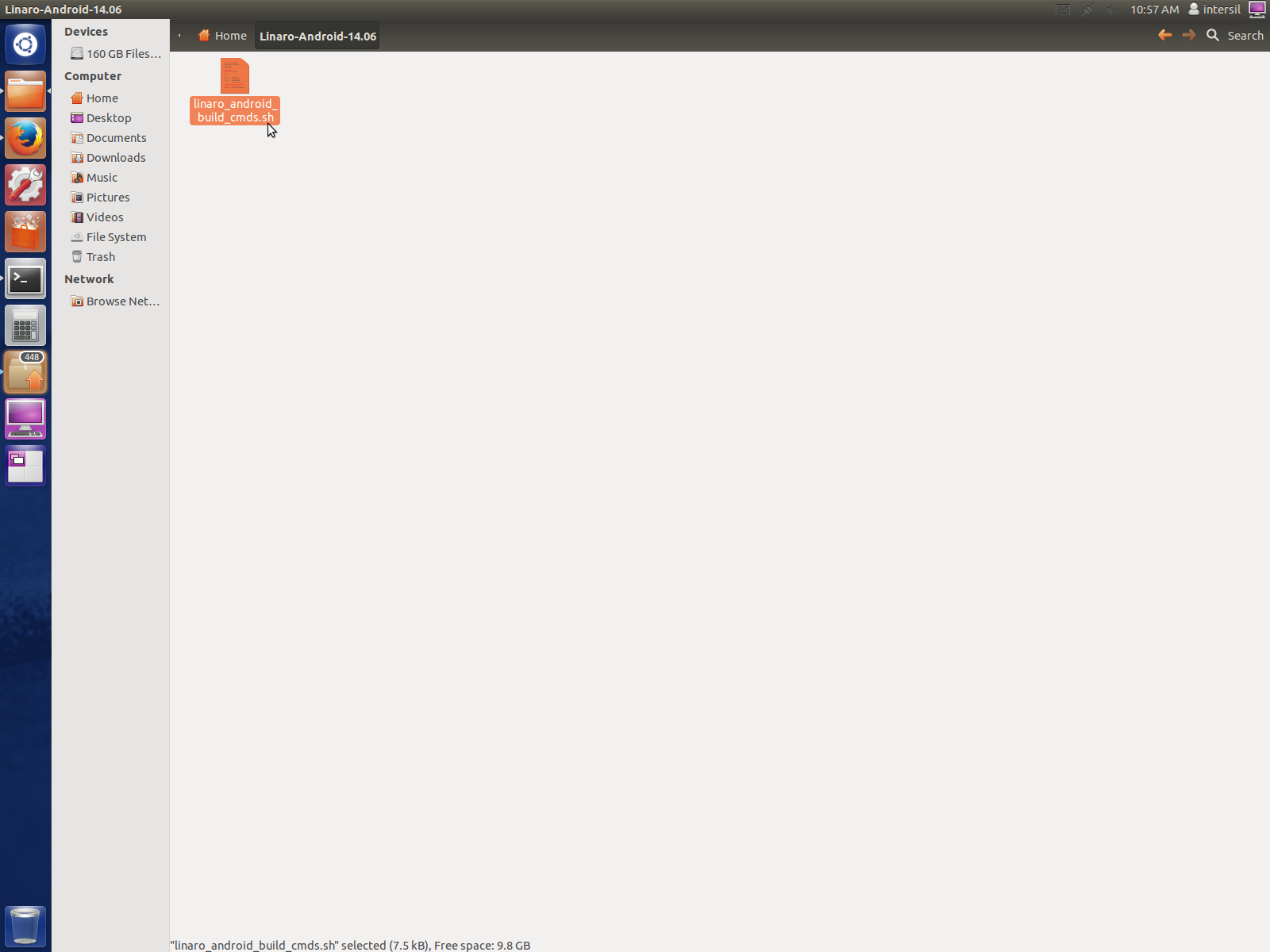
**Figure :** Prompt for saving linaro\_android\_build\_cmds.sh



**Figure :** Save to location : linaro\_android\_build\_cmds.sh

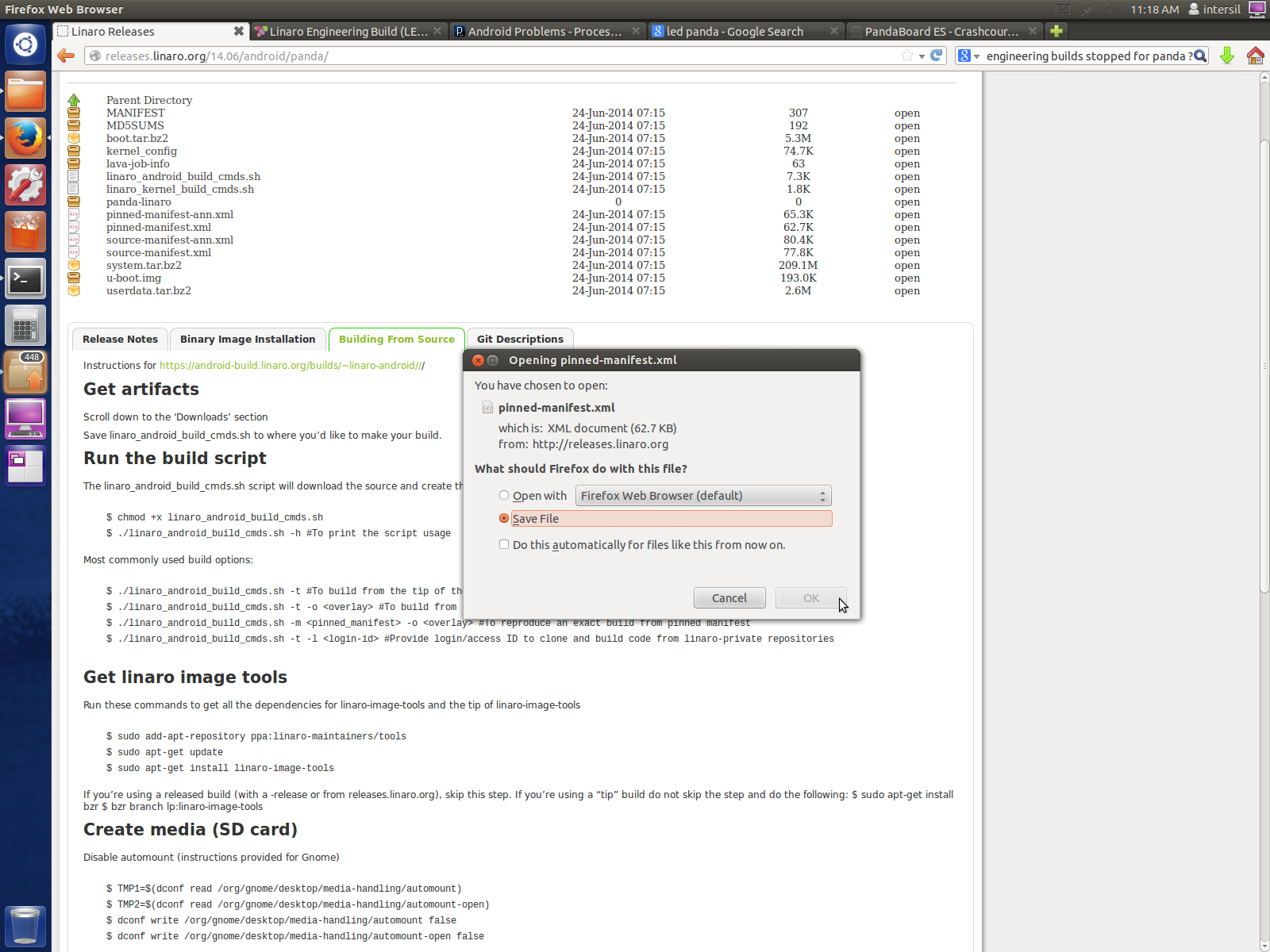


**Figure :** Verify download using File Explorer

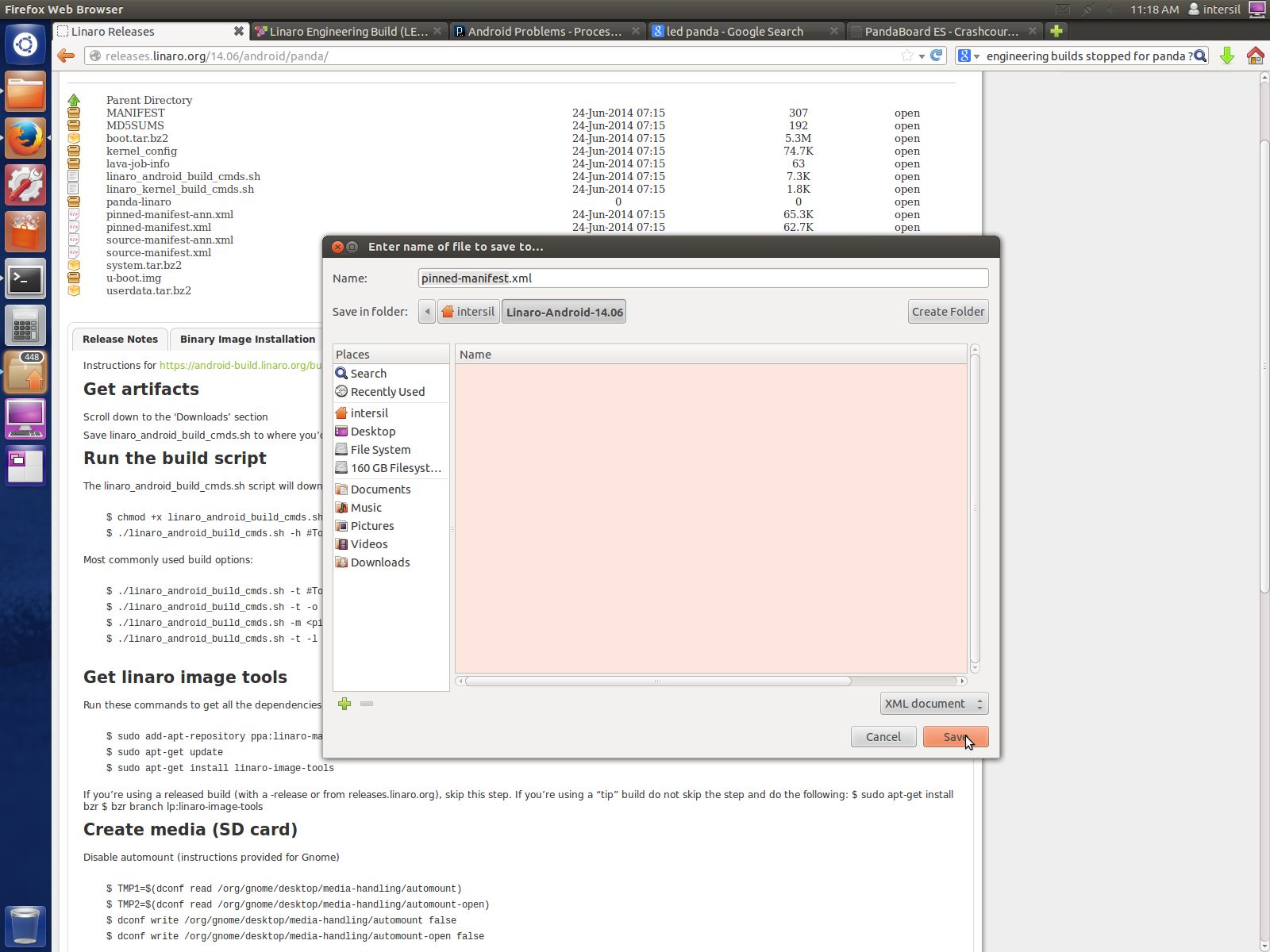


**STEP 5:** Click on pinned-manifest.xml and save the script to the same folder where the linaro\_android\_build\_cmds.sh was saved.

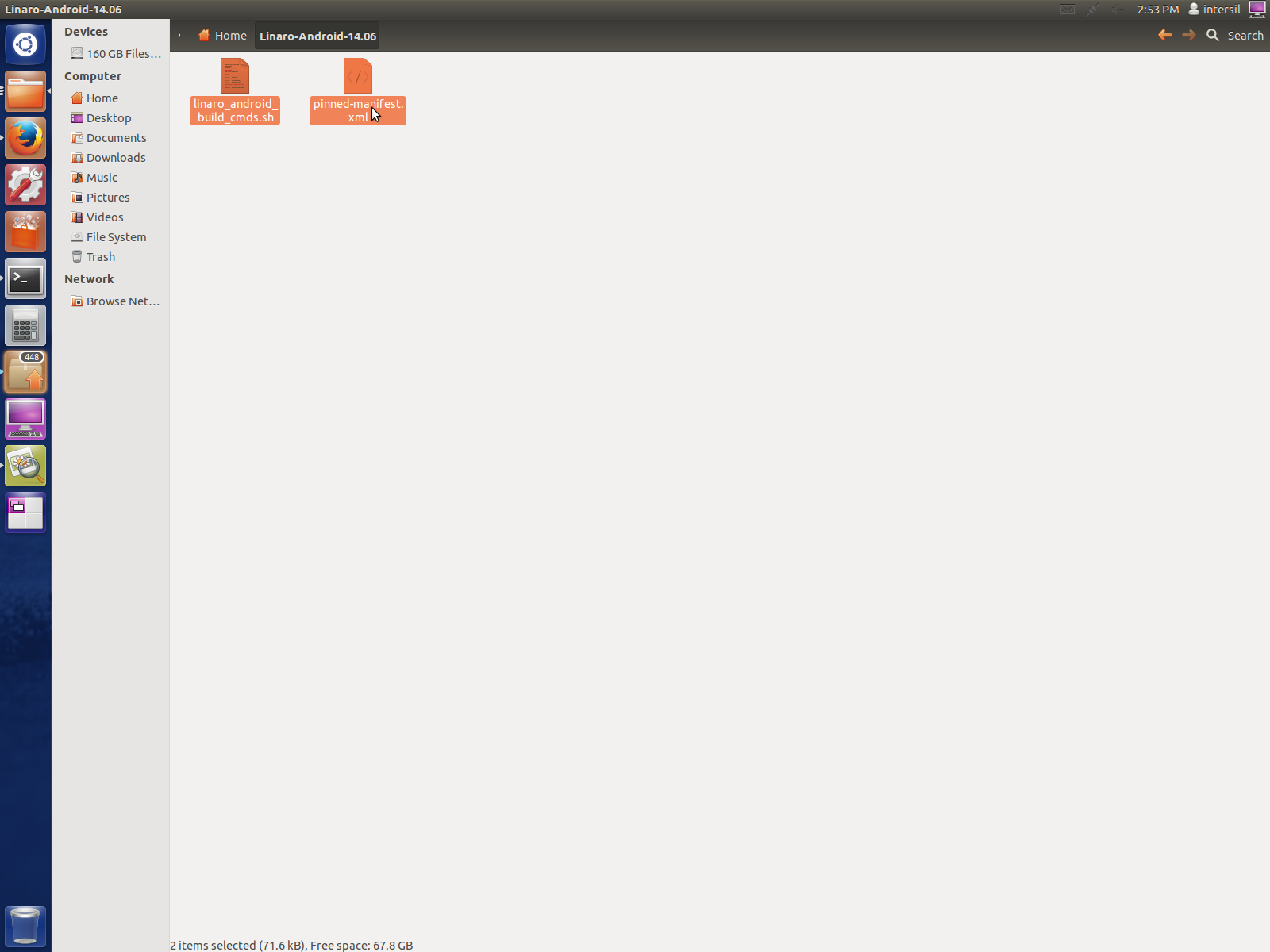
**Figure :** Prompt for saving pinned-manifest.xml



**Figure :** Save to location : pinned-manifest.xml



**Figure :** Verify download using File Explorer



At this point we have the necessary scripts to start fetching the required android source code.

**STEP6:** Open up a terminal and follow the below steps to invoke the fetching of the android source code.

1. Change directory to the path where the “linaro\_android\_build\_cmds.sh” and “pinned-manifest.xml” were downloaded

$ cd </path/to/downloaded\_scripts>

1. Make sure that the script has execution permission for the user if not use the below command

$ chmod u+x ./linaro\_android\_build\_cmds.sh

1. Ensure that you have internet connectivity in the android build server machine and then use the below command to start fetching the android source code

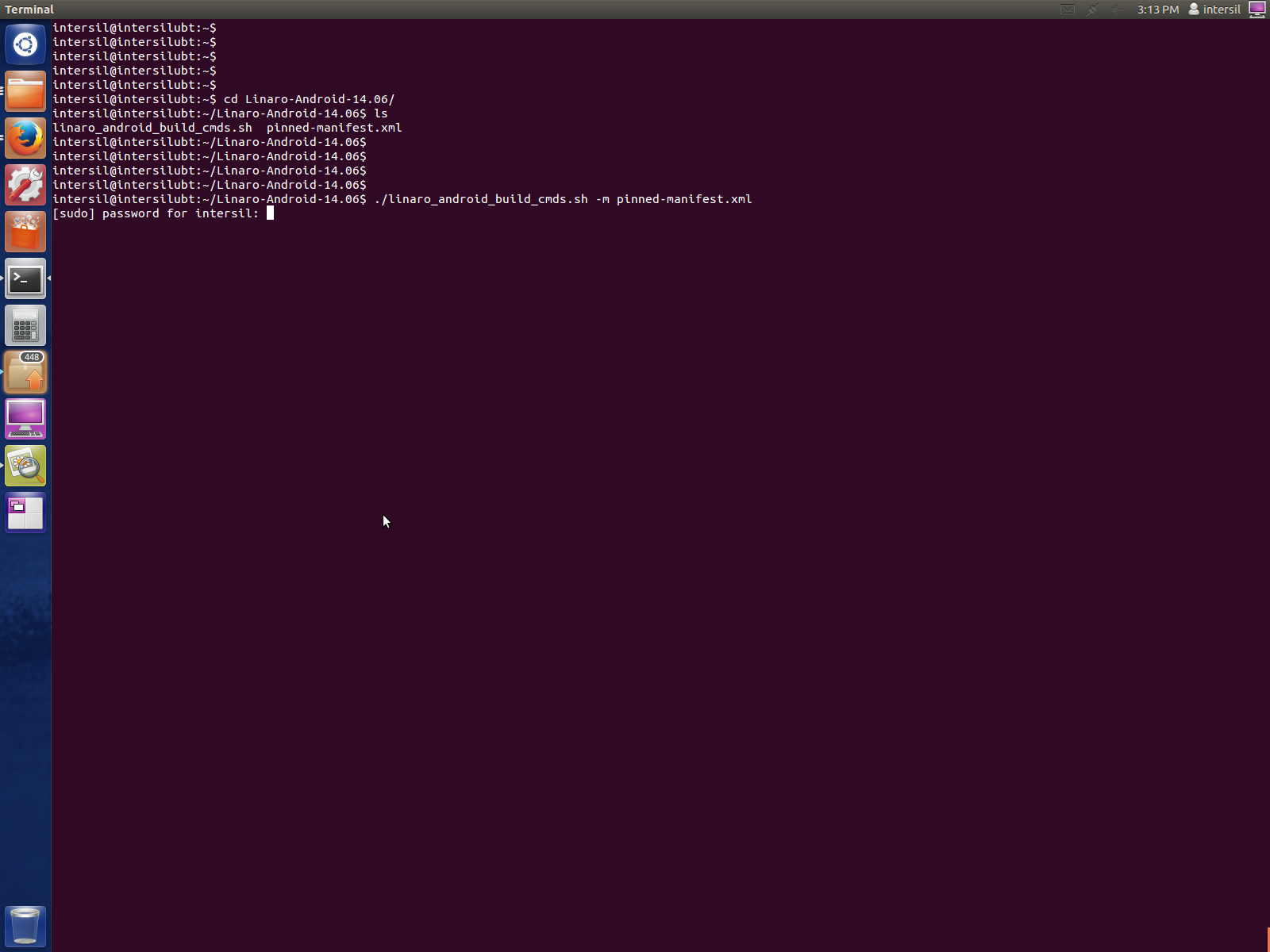
$./linaro\_android\_build\_cmds.sh –m pinned-manifest.xml

**Figure :** Execute android fetch and build script



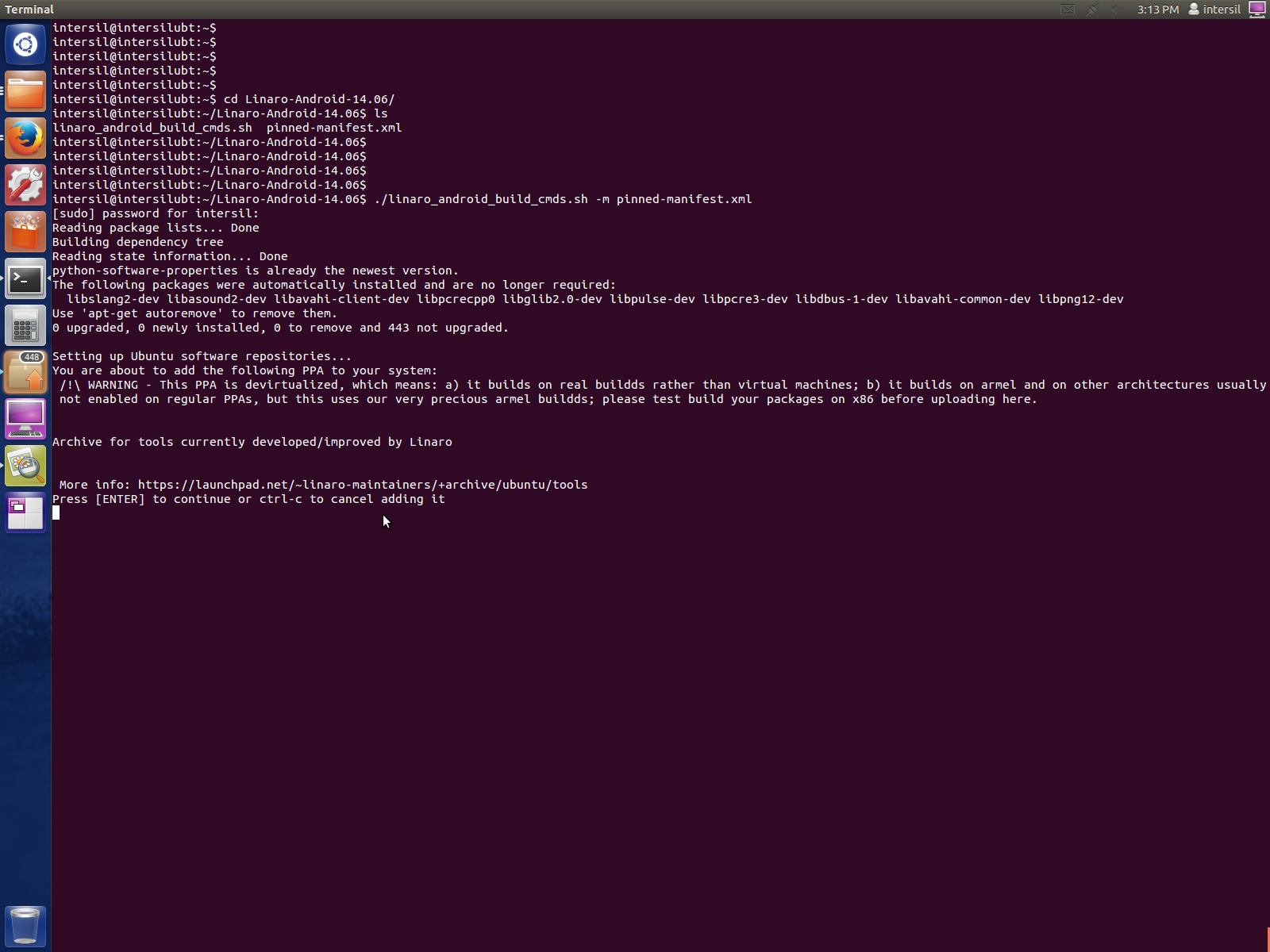
1. System would provide a password prompt, user shall enter his sudo password

**Figure :** Authenticate android fetch process

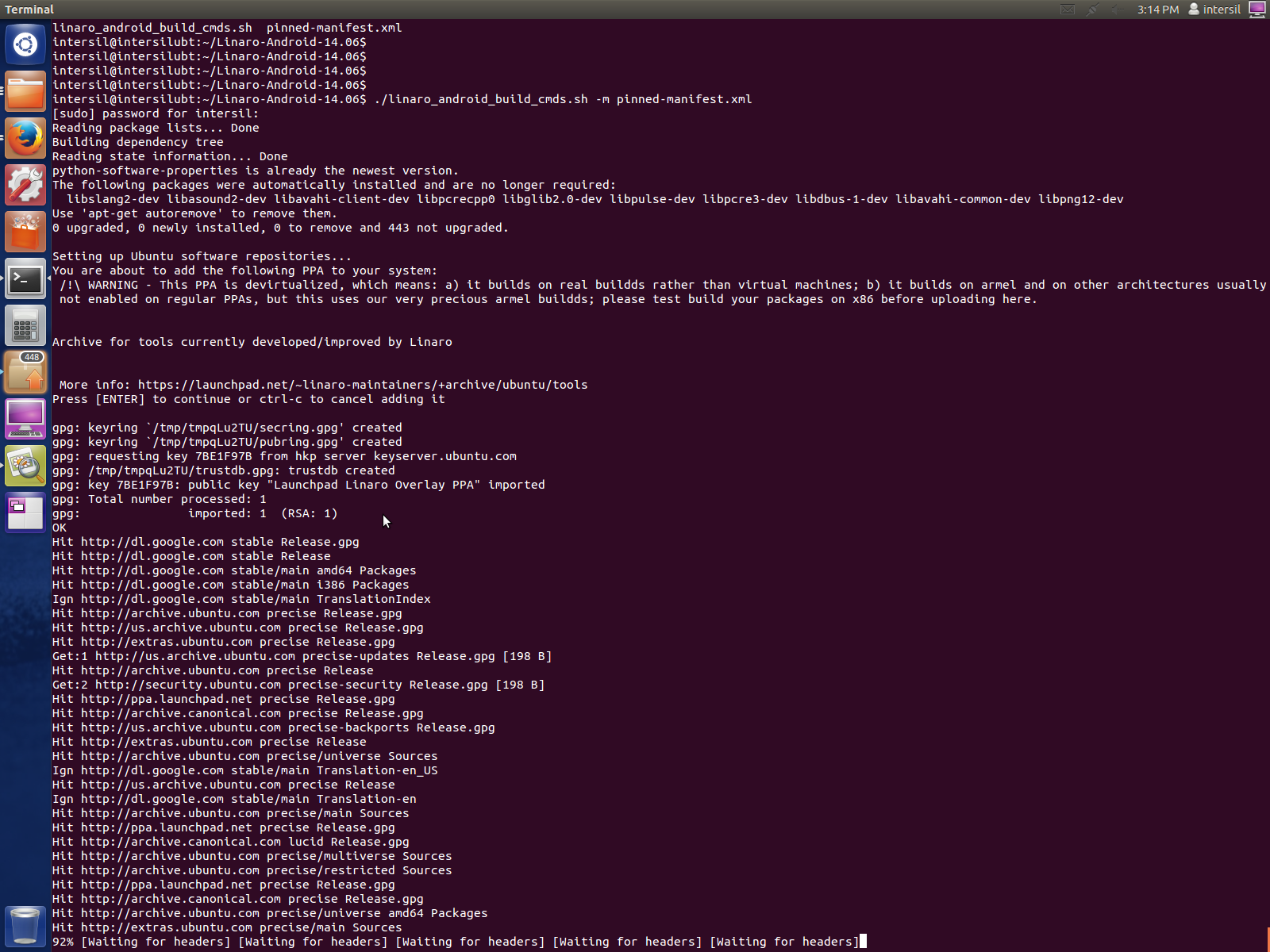


**STEP7:** Open up a terminal and follow the below steps to invoke the fetching of the android source code.

**Figure :** User response for setting up Ubuntu software repositories

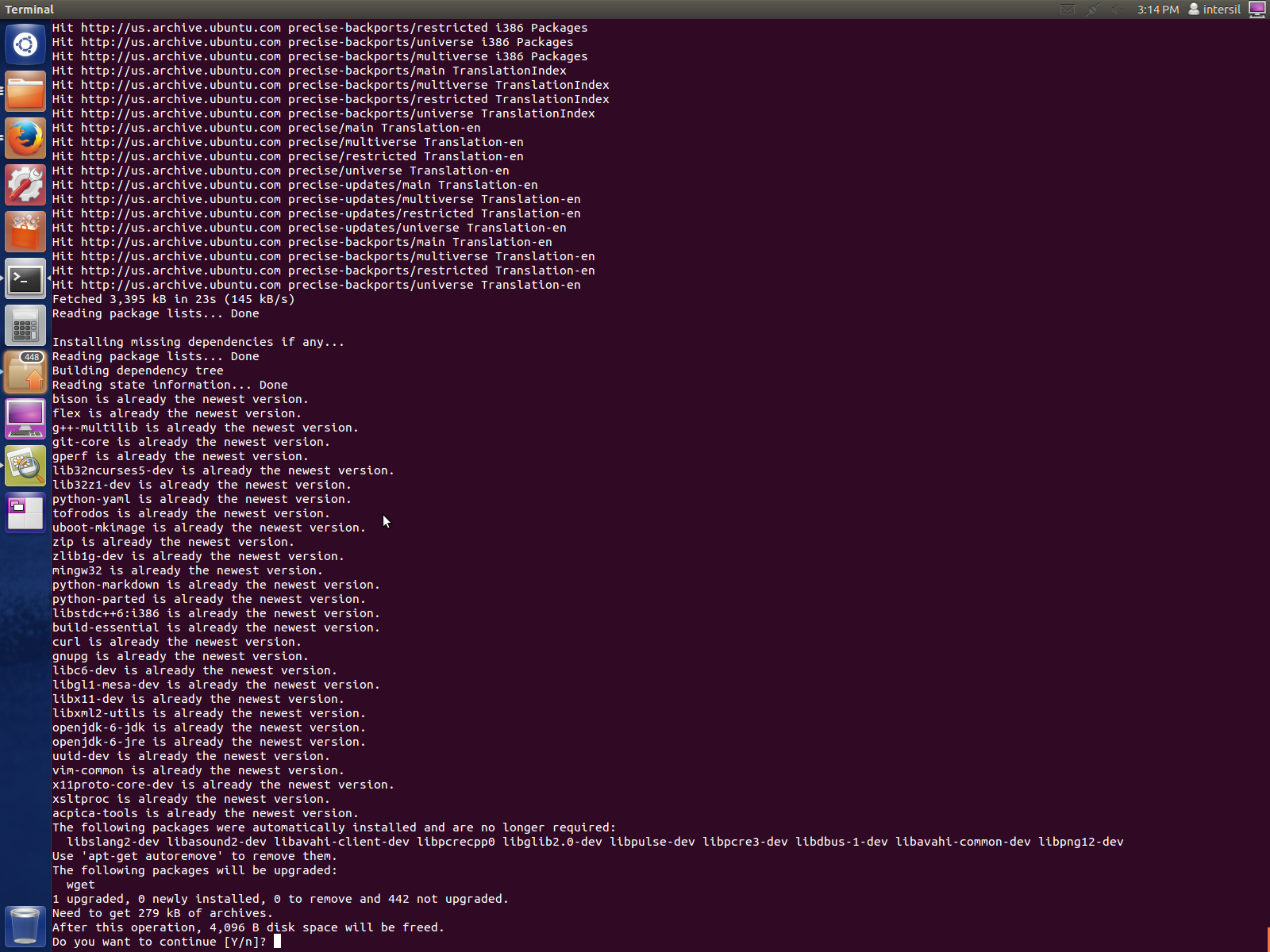


**Figure :** Ubuntu software update starts



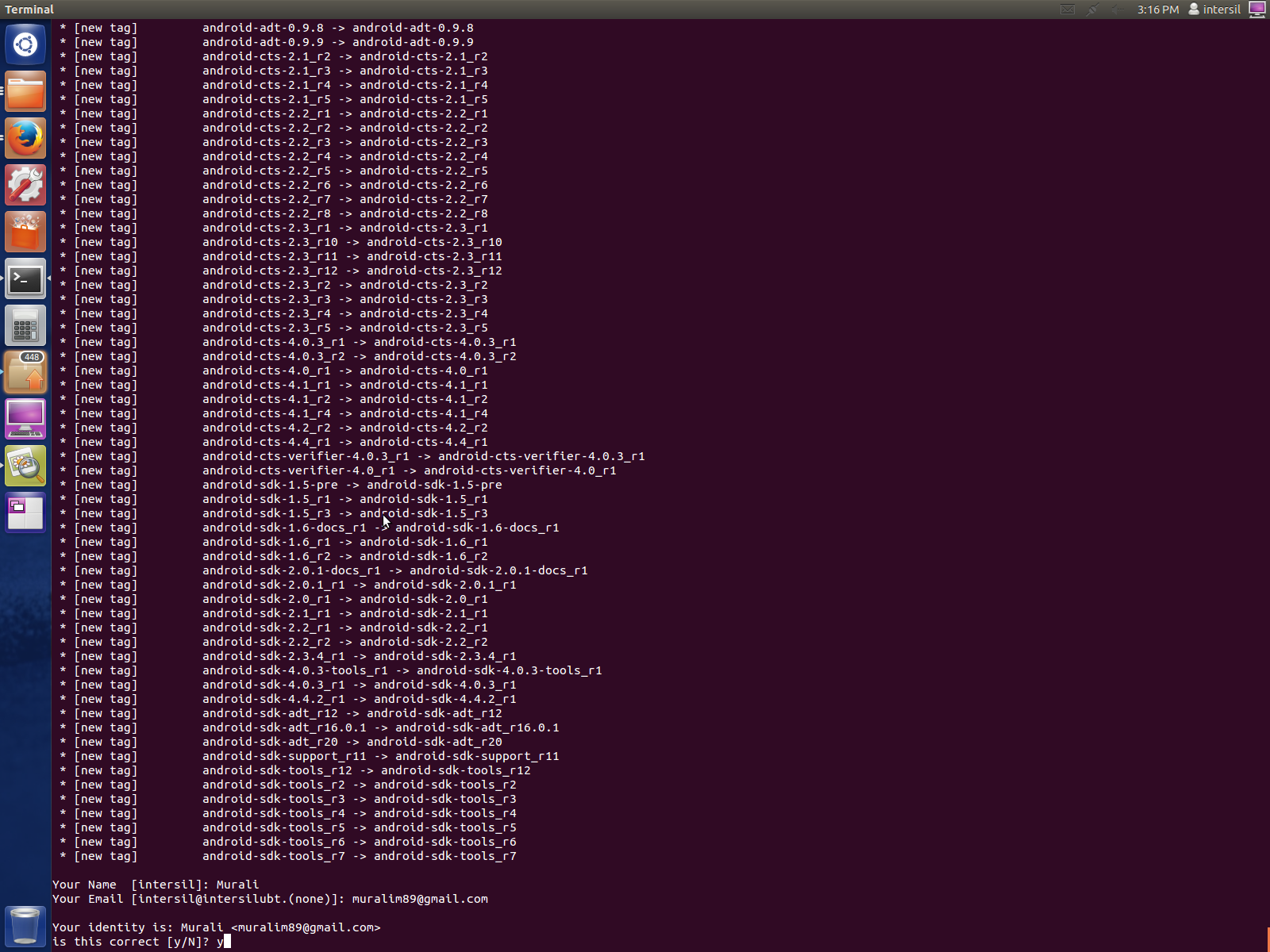
**STEP 8:** Give the confirmation as yes for fetching the Ubuntu archives

**Figure :** Confirmation for fetching Ubuntu archives



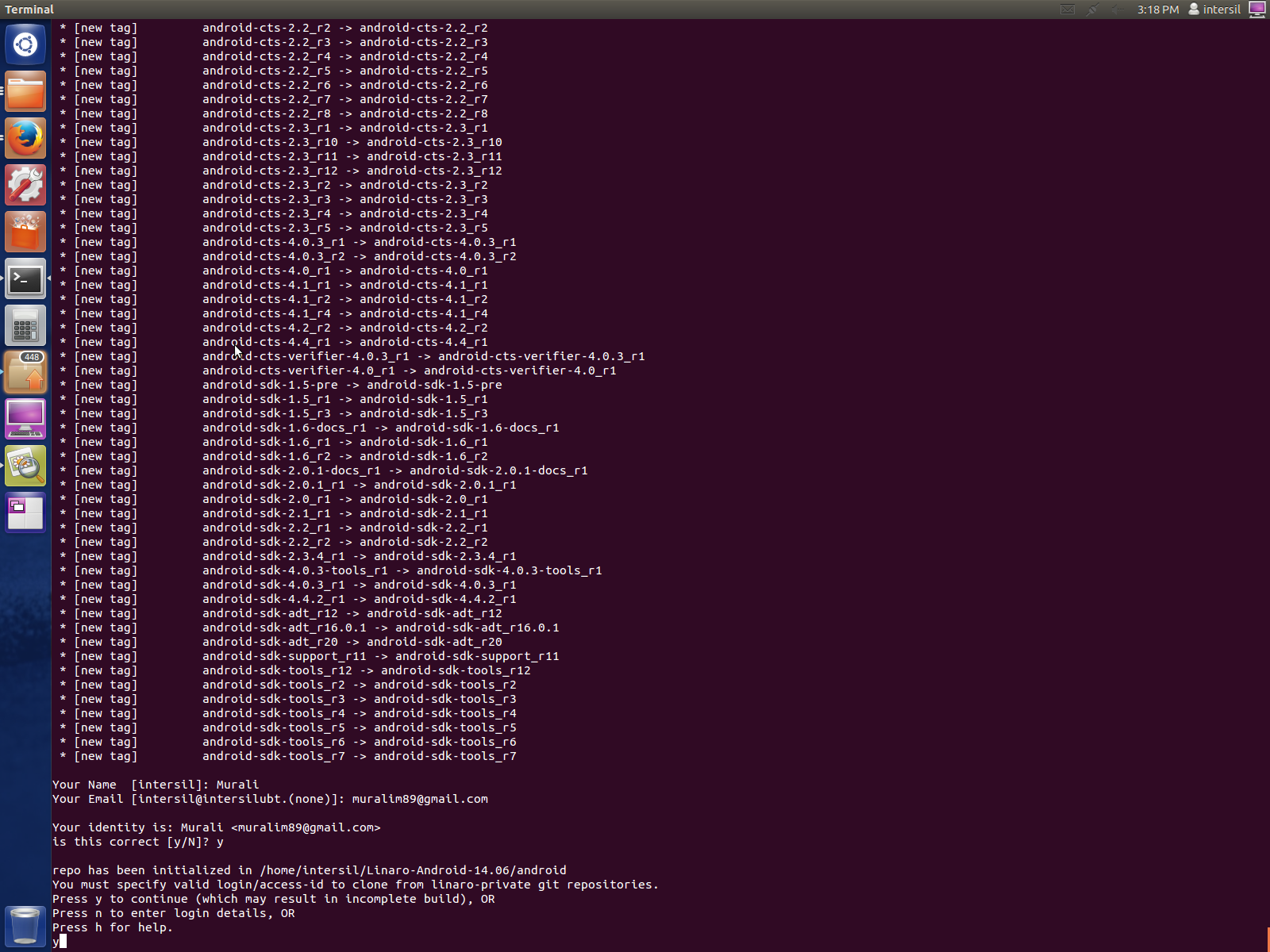
**STEP 9:** During fetch process user would be prompted to provide username and e-mail id information. Provide the same and confirm identity.

**Figure :** Enter personal details

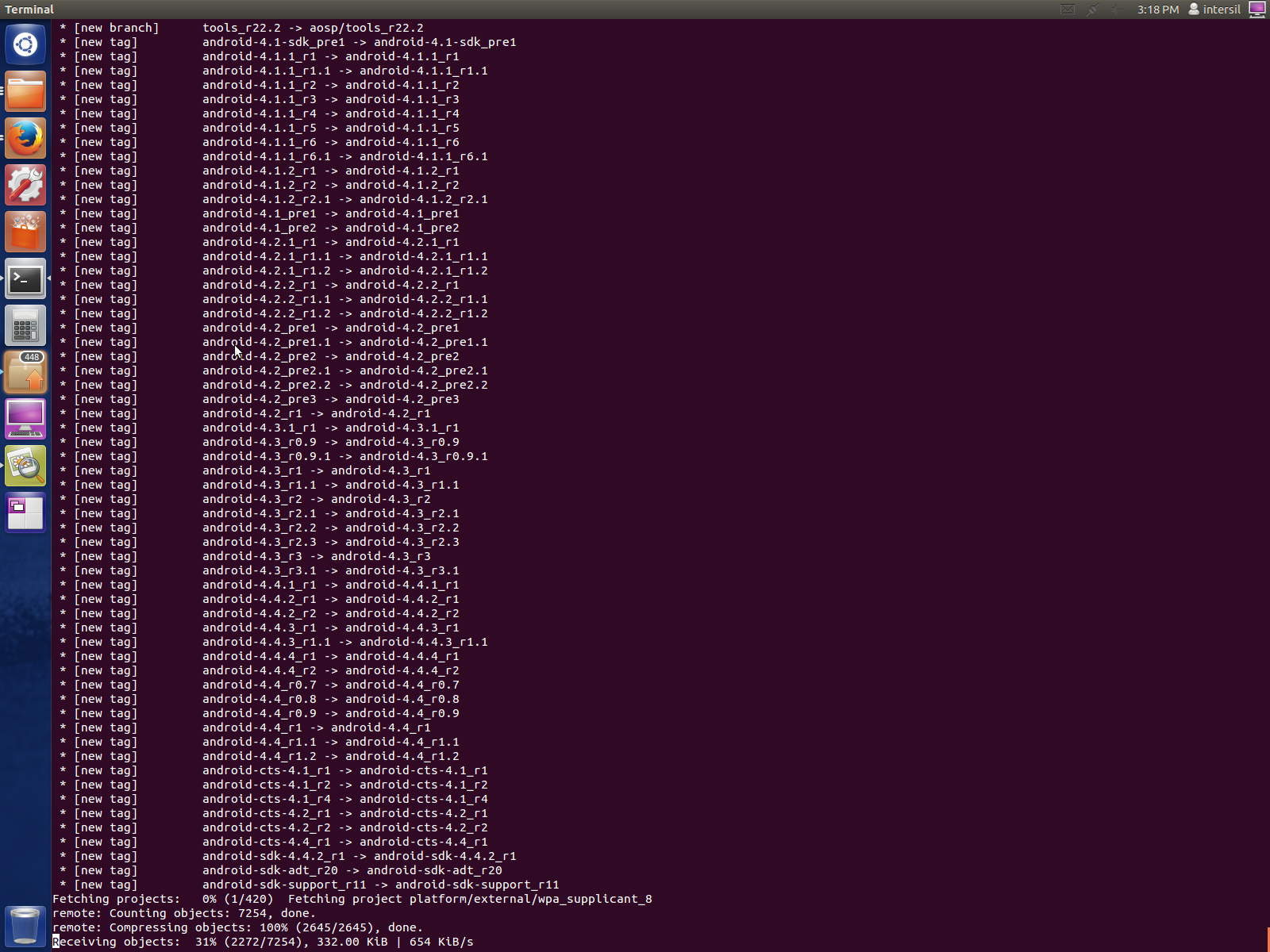


**STEP 10:** When prompted with a question as shown in below figure, enter ‘y’ as option and proceed.

**Figure :** User confirmation for Linaro private repositories



**Figure :** Android source code fetch starts



After this step the repository is synced 100% and the source code is available in a directory named android in the current directory.

|  |
| --- |
| **IMPORTANT NOTE:**   1. If the android source code was download in a single go without any interruption the script “linaro\_android\_build\_cmds.sh” would automatically start the build process too. But if the download stops due to some reason following instructions in the following section to manually start the android build. 2. If the download stops due to some reason follow the below instructions    1. Change to directory named android in current directory   $ cd android   * 1. Use the below command to resume syncing the android repository   $ ./repo sync |

## Building android

To build the android source code provided by Linaro use the below command.

Once repositories are fetched inside a directory named android in the current repository. Use the below commands

$ cd android

$ . build/envsetup.sh

$ lunch pandaboard-eng

$ make boottarball systemtarball userdatatarball

The above commands will start the android build and the process would continue until the build is complete

Once the build is complete the following files would be generated in the below path

**Android/out/target/product/pandaboard/**

1. Boot.tar.bz2
2. System.tar.bz2
3. Userdata.tar.bz2

## Flashing Images to SD card

Follow the below instructions in sequence in order to flash the SD card with android image binaries

### Disable automount in Ubuntu

To disable automount use the below commands

$ TMP1=$(dconf read /org/gnome/desktop/media-handling/automount)

$ TMP2=$(dconf read /org/gnome/desktop/media-handling/automount-open)

$ dconf write /org/gnome/desktop/media-handling/automount false

$ dconf write /org/gnome/desktop/media-handling/automount-open false

### Install Linaro image tools

In order to flash the SD card with the android binaries generated from the build process, we require to install the following package

* Linaro-image-tools

To install this package use the below commands

$ sudo add-apt-repository ppa:linaro-maintainers/tools

$ sudo apt-get update

$ sudo apt-get install linaro-image-tools

### Flash images to SD card

Insert a formatted SD card into the host machine. Once inserted use the below command to get to know the device node associated with SD card. Device node is a representation of SD card on Linux system.

$ dmesg

Look for a line that looks like the following at the end of the log

[288582.790722] sdc: sdc1 sdc2 sdc3 sdc4 < sdc5 sdc6 >

The most recent log after inserting the SD card would indicate the correct node name (like sdb, sdc or sdb etc.)

|  |
| --- |
| **IMPORTANT NOTE:**   1. In some system the node name would be something similar to **mmcblkx** where x is the device number 2. Choosing a wrong device may cause loss of data in system. So to be safe it is advised to verify the size of the device using the partition manager present in Ubuntu. 3. If partition manager is not installed use below command to install one (gparted partition manager)   $ sudo apt-get install gparted |

Run linaro image tools

$ linaro-android-media-create --mmc /dev/sdc --dev panda --boot boot.tar.bz2 --system system.tar.bz2 --userdata userdata.tar.bz2

### Install graphics libraries

Use the below commands in order to flash the graphic libraries to the SD card

$ wget http://people.linaro.org/~vishalbhoj/install-binaries-4.0.4.sh

$ chmod a+x install-binaries-4.0.4.sh

$ ./install-binaries-4.0.4.sh

### Restore automount in Ubuntu

Use the below commands to restore automount in Ubuntu

$ dconf write /org/gnome/desktop/media-handling/automount $TMP1

$ dconf write /org/gnome/desktop/media-handling/automount-open $TMP2

Now plug the SD card into pandaboard setup and , boot it to view android screen on HDMI or DVI display.