**Assignment #3 Poky Directory Tree Layout**

**Subject:**

# **2021W ESE 3005 1 [A207] EMBEDDED SYSTEMS ARCHITECTURE I**

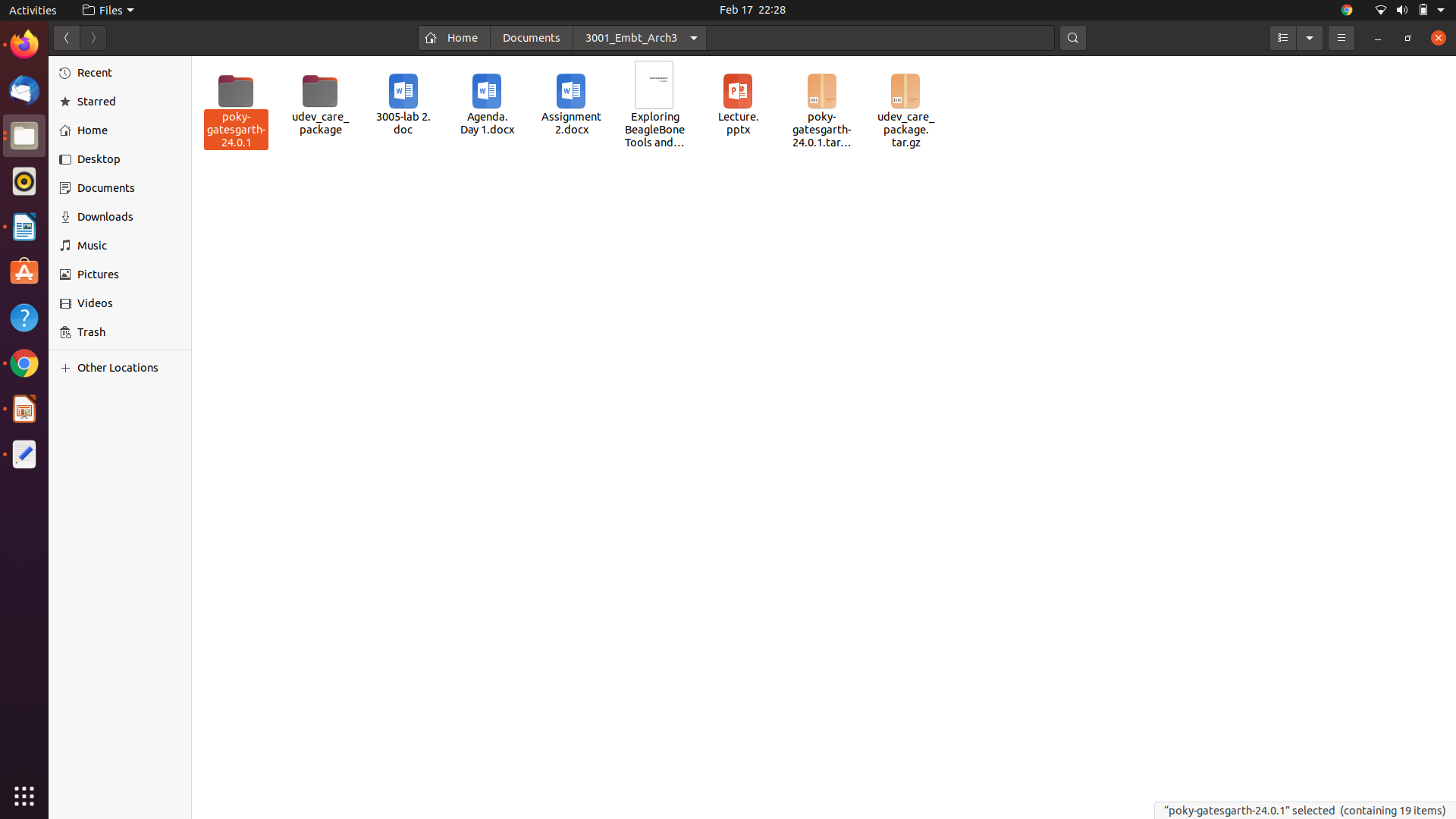
**Group members:-** Shahrukh Padaniya

Rohan Yadav

Swapnil Sevak

**1. Ensure Poky on Ubuntu has already been downloaded and extracted**

Downloaded Poky from <https://www.yoctoproject.org/software-overview/downloads/> URL and extracted.



There are many sub-directories inside the Poky.

1. Bitbake

2. Contrib

3. Documentation

4. meta

5. meta-poky

6. meta-selftest

7. meta-skeleton

8. meta-yocto-bsp

9. script

**2. Explore Documentation sub-directories**

This folder gives insight or theoretical knowledge about Poky distribution and how to use it. We can learn how to quick start our own Linux-distribution using this manual.

This directory holds the source for the Yocto Project documentation as well as templates and tools that allow you to generate PDF and HTML versions of the manuals. Each manual is contained in its own sub-folder; for example, the files for this reference manual reside in the ref-manual/ directory.

**3. How many meta subdirectories do you have?**

There are 17 sub-directories located inside meta and 22 importatant folders resides in it which is listed below:

1. meta/classes/

2. meta/conf/

3. meta/conf/machine

4. meta/conf/distro

5.meta/conf/machine-sdk

6. meta/files

7. meta/lib

8. meta/recipes-bsp

9. meta/recipes-connectivity

10. meta/recipes-core

11. meta/recipes-devtools

12. meta/recipes-extended

13. meta/recipes-gnome

14. meta/recipes-graphics

### 15. meta/recipes-kernel

### 16. meta/recipes-lsb4

### 17. meta/recipes-multimedia

### 18. meta/recipes-rt

### 19. meta/recipes-sato

### 20. meta/recipes-support

### 21. meta/site

### 22. meta/recipes

**4. Where are recipe file located**

Recipe files are located in meta directory. There 12 recipe files.The .bb files are usually referred to as 'recipes'. In general, a recipe contains information about a single piece of software such as from where to download the source patches (if any are needed), which special configuration options to apply, how to compile the source files, and how to package the compiled output.

* meta/recipes-bsp
* meta/recipes-connectivity
* meta/recipes-core
* meta/recipes-devtools
* meta/recipes-extended
* meta/recipes-gnome
* meta/recipes-graphics
* meta/recipes-kernel

### meta/recipes-lsb4

### meta/recipes-multimedia

### meta/recipes-rt

### meta/recipes-sato

### meta/recipes-support

**5. Where are class file are located?**

Class file also located inside meta directory.

Class files are used to abstract common functionality and share it amongst multiple recipe (.bb) files. To use a class file, you simply make sure the recipe inherits the class. In most cases, when a recipe inherits a class it is enough to enable its features. There are cases, however, where in the recipe you might need to set variables or override some default behavior.

Any Metadata usually found in a recipe can also be placed in a class file. Class files are identified by the extension .bbclass and are usually placed in a classes/ directory beneath the meta\*/ directory found in the [Source Directory](https://www.yoctoproject.org/docs/latest/ref-manual/ref-manual.html" \l "source-directory). Class files can also be pointed to by BUILDDIR (e.g. build/) in the same way as .conf files in the conf directory. Class files are searched for in BBPATH using the same method by which .conf files are searched.

This chapter discusses only the most useful and important classes. Other classes do exist within the meta/classes directory in the Source Directory. You can reference the .bbclass files directly for more information.

**6. Where is bitbake.conf located? What is this file for?**

Bitbake file located in meta/config file. It used to store variables of bitbake.

BitBake then expects to find the ``conf/bitbake.conf`` file somewhere in

the user-specified ``BBPATH``. That configuration file generally has

include directives to pull in any other metadata such as files specific

to the architecture, the machine, the local environment, and so forth.

Only variable definitions and include directives are allowed in BitBake

``.conf`` files. Some variables directly influence BitBake's behavior.

**7. Where is qemux86-64.conf? What is this file for?**

Qemux86-64 is located in meta/conf/machine file. This file is for storing environment variable of qemux86-64 emmulator.

QEMU is a generic and open source machine emulator and virtualizer." When used as a machine emulator, QEMU can run OSes and programs made for one machine (e.g. an ARM board) on a different machine (e.g. your x86 PC). By using dynamic translation, it achieves very good performance

**8. Explore scripts subdirectory**

This directory contains various integration scripts that implement extra functionality in the Yocto Project environment (e.g. QEMU scripts). The oe-init-build-env script prepends this directory to the shell's PATH environment variable.

The scripts directory has useful scripts that assist in contributing back to the Yocto Project, such as create-pull-request and send-pull-request.

**9.Where is local.conf located? What is this file for?**

There is local.cong.sample file found in meta-poky/conf directory.

Generally local.cong file generated in build/conf sub-directory after building new distribution.

This configuration file contains all the local user configurations for your build environment. The local.conf file contains documentation on the various configuration options. Any variable set here overrides any variable set elsewhere within the environment unless that variable is hard-coded within a file (e.g. by using '=' instead of '?='). Some variables are hard-coded for various reasons but such variables are relatively rare.

**10. Where is bblayers.conf located? What is this file for?**

This file is located in build/cong directory.

This configuration file defines l[ayers](../../../tmp/mozilla_shahrukh0/_top), which are directory trees, traversed (or walked) by BitBake. The bblayers.conf file uses the BBLAYERS variable to list the layers BitBake tries to find.

If bblayers.conf is not present when you start the build, the OpenEmbedded build system creates it from bblayers.conf.sample when you source the top-level build environment setup script