

Lab Submission-Week 1

ESE 3005: EMBEDDED SYSTEMS ARCHITECTURE II

Lambton College in Toronto

Instructor: Takis Zourntos

STUDENT NAME & ID:

Vishal Hasrajani(C0761544)

Parth Patel(C0764929)

Goutham Reddy Alugubelly(C0747981)

Ratnajahnavi rebbapragada(C0762196)

INTRODUCTION:

To get the Yocto working by following the Yocto mega manual and to build “core-image-sato” operating systems. Also, to display it that it runs on an emulated machine using qemuarm.

We must check the requirements for installing Yocto, the system should be having an updated software along with 50GB of space.

DESCRIPTION:

YOCTO PROJECT QUICK BUILD

In this process we are going to build a typical image using the Yocto-project. We will configure a build for a specific hardware and also to build a reference embedded operating systems called Poky.

To build the host packages we need to install essential host packages based on ubuntu distribution by running the following command.

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
  build-essential chrpath socat cpio python3 python3-pip python3-pexpect \
  xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev \
  pylint3 xterm
```

Then, we need to use Git to Clone Poky, after the setup on machine we need to get a copy of poky repository on your build host.

To clone the poky repository run the below command on the machine.

```
$ git clone git://git.yoctoproject.org/poky
```

To initialize the build, we need to follow the below steps.

Initializing the build environment

```
$cd ~/poky
```

In this poky directory, we have to run oe-init-build-env environment script for building yocto project environment on host machine.

```
$source oe-init-build-env
```

Now a local conf file will be created in conf subdirectory folder with some default values. Then we can edit it by selecting any machine.

Now to build an OS image for this process, we need to run,

`$bitbake core-image-sato`

Now we need to simulate our image using qemu and then exit from qemu.

OUTPUTS:

```
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx: ~/poky/build
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~$ cd poky
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky$ source oe-init-build-env

### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
  core-image-minimal
  core-image-sato
  meta-toolchain
  meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemu86'

Other commonly useful commands are:
  - 'devtool' and 'recipetool' handle common recipe tasks
  - 'bitbake-layers' handles common layer tasks
  - 'oe-pkgdata-util' handles common target package tasks

goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky/build$ bitbake core-image-sato
Loading cache: 100% |#####| Time: 0:00:00
Loaded 1380 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.44.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "arm-poky-linux-gnueabi"
MACHINE              = "qemuarm"
DISTRO               = "poky"
DISTRO_VERSION       = "3.0+snapshot-20201022"
TUNE_FEATURES        = "arm armv7ve vfp thumb neon callconvention-hard"
TARGET_FPU           = "hard"
meta
meta-poky
meta-yocto-bsp       = "master:34535f3e0ca6f6e37e6457fc880dfbfff64d9298"

Initialising tasks: 100% |#####| Time: 0:00:03
State summary: Wanted 0 Found 0 Missed 0 Current 2653 (0% match, 100% complete)
NOTE: Executing Tasks
NOTE: Setscene tasks completed
NOTE: Tasks Summary: Attempted 6549 tasks of which 6549 didn't need to be rerun and all succeeded.
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky/build$ runqemu
runqemu - INFO - Running bitbake -e ...
runqemu - INFO - Continuing with the following parameters:
KERNEL: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/zImage--5.2.28+git0+dd6019025c_ed43b791f2-r0-qemuarm-20200221144700.bin]
MACHINE: [qemuarm]
FSTYPE: [ext4]
ROOTFS: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/core-image-sato-qemuarm-20200221144700.rootfs.ext4]
CONFFILE: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/core-image-sato-qemuarm-20200221144700.qemuboot.conf]

runqemu - INFO - Setting up tap interface under sudo
[sudo] password for goutham:
runqemu - INFO - Network configuration: 192.168.7.2:192.168.7.1:255.255.255.0
```

```
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx: ~/poky/build
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~$ cd poky
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky$ source oe-init-build-env

### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
  core-image-minimal
  core-image-sato
  meta-toolchain
  meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemu86'

Other commonly useful commands are:
  - 'devtool' and 'recipetool' handle common recipe tasks
  - 'bitbake-layers' handles common layer tasks
  - 'oe-pkgdata-util' handles common target package tasks

goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky/build$ bitbake core-image-sato
Loading cache: 100% |#####| Time: 0:00:00
Loaded 1380 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.44.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "arm-poky-linux-gnueabi"
MACHINE              = "qemuarm"
DISTRO               = "poky"
DISTRO_VERSION       = "3.0+snapshot-20201022"
TUNE_FEATURES        = "arm armv7ve vfp thumb neon callconvention-hard"
TARGET_FPU           = "hard"
meta
meta-poky
meta-yocto-bsp       = "master:34535f3e0ca6f6e37e6457fc880dfbfff64d9298"

Initialising tasks: 100% |#####| Time: 0:00:03
State summary: Wanted 0 Found 0 Missed 0 Current 2653 (0% match, 100% complete)
NOTE: Executing Tasks
NOTE: Setscene tasks completed
NOTE: Tasks Summary: Attempted 6549 tasks of which 6549 didn't need to be rerun and all succeeded.
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky/build$ runqemu
runqemu - INFO - Running bitbake -e ...
runqemu - INFO - Continuing with the following parameters:
KERNEL: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/zImage--5.2.28+git0+dd6019025c_ed43b791f2-r0-qemuarm-20200221144700.bin]
MACHINE: [qemuarm]
FSTYPE: [ext4]
ROOTFS: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/core-image-sato-qemuarm-20200221144700.rootfs.ext4]
CONFFILE: [/home/goutham/poky/build/tmp/deploy/images/qemuarm/core-image-sato-qemuarm-20200221144700.qemuboot.conf]

runqemu - INFO - Setting up tap interface under sudo
[sudo] password for goutham:
runqemu - INFO - Network configuration: 192.168.7.2:192.168.7.1:255.255.255.0
runqemu - INFO - Running /home/goutham/poky/build/tmp/work/x86_64-linux/qemu-helper-native/1.0-r1/recipe-sysroot-native/usr/bin/qemu-system-arm -device virtio-net-device,netdev=net0,nic=52:54:00:12:34:02
-netdev tap,id=net0,ifname=tap0,script=no,downscript=no -drive id=disk0,file=/home/goutham/poky/build/tmp/deploy/images/qemuarm/core-image-sato-qemuarm-20200221144700.rootfs.ext4,if=none,format=raw -device
virtio-blk-device,drive=disk0 -show-cursor -device VGA,edid=on -device qemu-scsi -device usb-tablet -device usb-kbd -object rng-random,filename=/dev/urandom,id=rng0 -device virtio-rng-pci,rng=rng0 -n
hine virt -cpu cortex-a15 -m 512 -serial non-vc -serial null -kernel /home/goutham/poky/build/tmp/deploy/images/qemuarm/zImage--5.2.28+git0+dd6019025c_ed43b791f2-r0-qemuarm-20200221144700.bin -append 'roo
t=/dev/vda rw mem=512M ip=192.168.7.2:192.168.7.1:255.255.255.0 console=ttyAMA0 '

runqemu - INFO - Cleaning up
Set 'tap0' nonpersistent
goutham@goutham-HP-Pavilion-Laptop-15-cs3xxx:~/poky/build$
```

QEMU OUTPUTS:

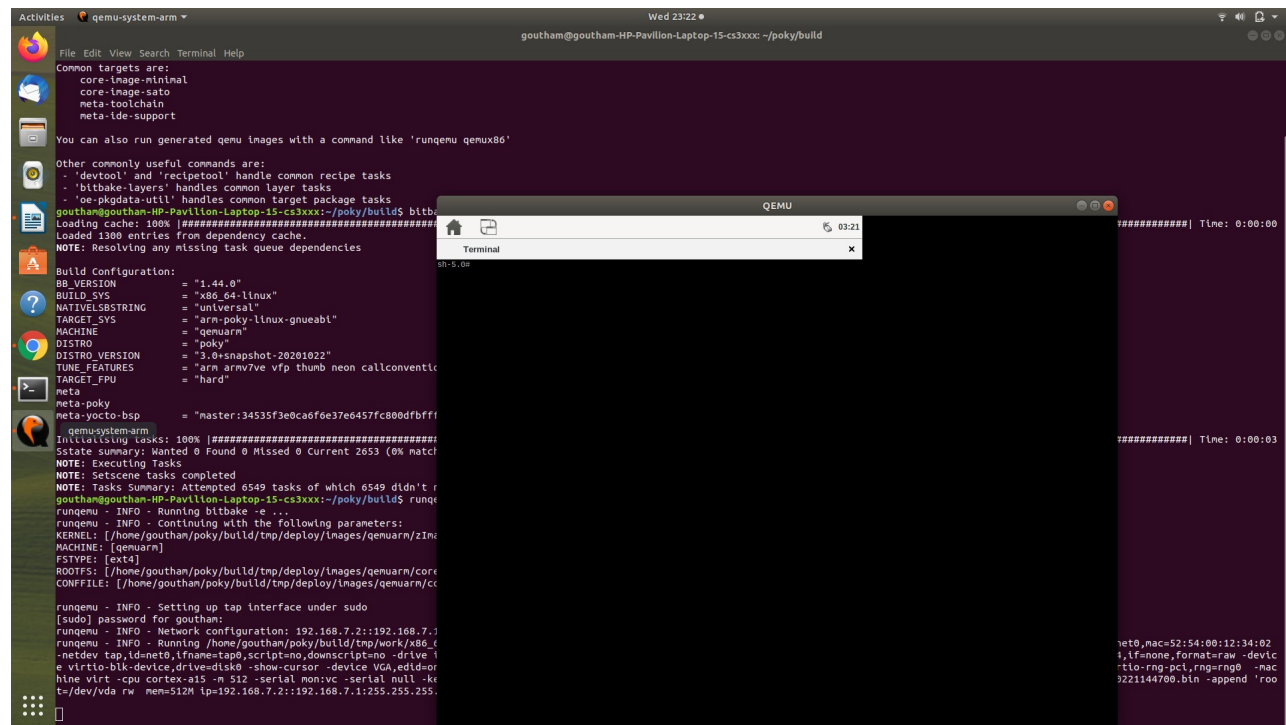
Applications in QEMU

The screenshot displays a QEMU virtual machine environment. On the left, a terminal window shows the output of a build process. The output includes common targets (core-image-minimal, core-image-sato, meta-toolchain, meta-ide-support), useful commands (devtool, recipetool, bitbake-layers, oe-pkgdata-util), and build configuration details (BB_VERSION, BUILD_SYS, NATIVELSBSTRING, TARGET_SYS, MACHINE, DISTRO, DISTRO_VERSION, TUNE_FEATURES, TARGET_FPU, meta, meta-poky, meta-yocto-bsp). It also shows the progress of task initialization and execution, including a state summary and a list of tasks attempted. The build process is running bitbake and setting up a tap interface under sudo.

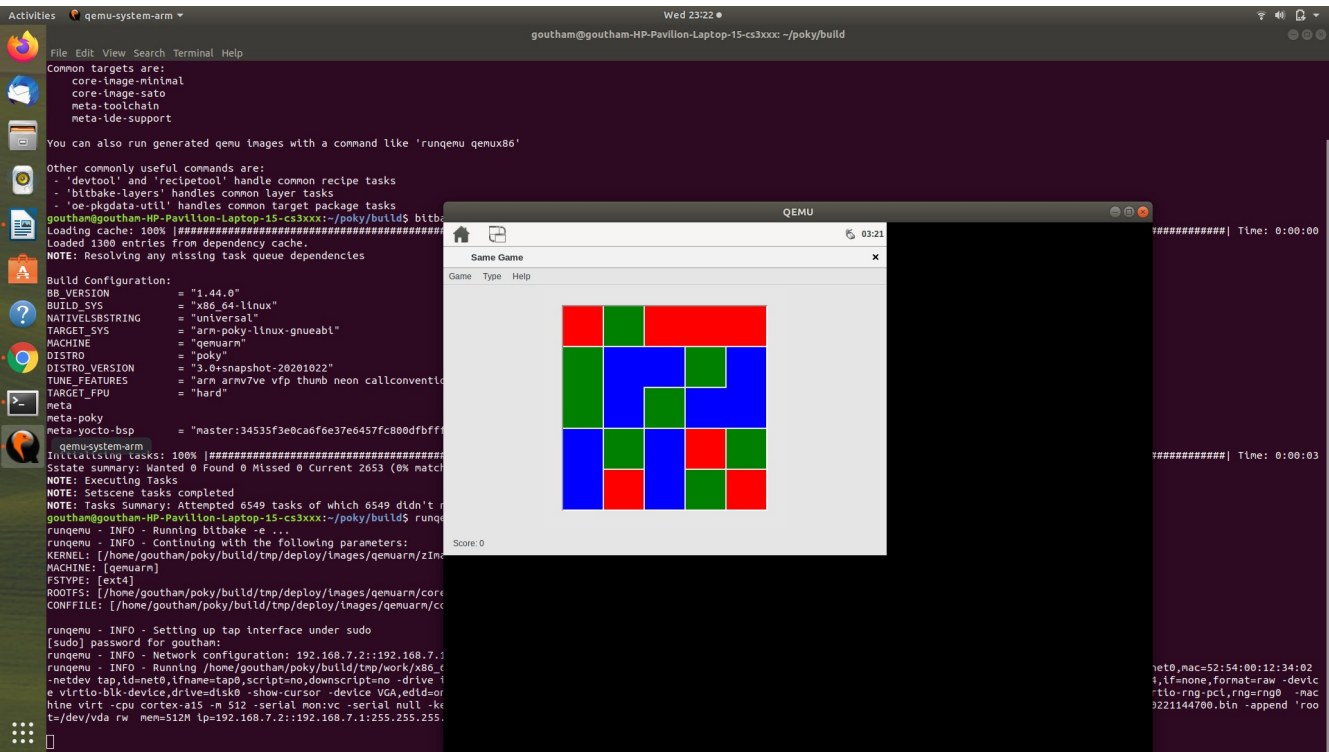
On the right, an applications menu is visible, listing various applications such as Calibrate Touchs..., File Manager PC..., L3afpad, Media Player, Shutdown, and Terminal. The menu is titled 'Applications' and includes a search bar and a list of icons representing the applications.

The bottom right corner of the terminal window shows the output of the 'runqemu' command, including the kernel version (3.0+snapshot-20201022), the target architecture (armv7ve), and the target platform (hard). It also shows the output of the 'runqemu' command, including the kernel version (3.0+snapshot-20201022), the target architecture (armv7ve), and the target platform (hard).

Terminal in QEMU:



Games In QEMU:



CONCLUSION:

Building the Yocto project has been done. We have build our “core-image-sato” operating systems and after that it ran on an emulated machine using qemuarm. By doing the Yocto project we have build an reference embedded operating system called Poky. In QEMU we have different applications build, it is a qucik EMUlator along with Yocto-project.