Building a Yocto Image

Advanced Embedded Linux Development

with **Dan Walkes**

University of Colorado **Boulder**

Learning objectives:

Yocto build by example

Overview of yocto build helper scripts

Building your Custom Image

- Run ./build.sh
  - Initializes your poky submodule
  - source poky/oe-init-build-env (note: creates/changes directory to the build dir!)
  - Checks whether your build/conf/local.conf file already contains your MACHINE definition, if not adds it
  - Checks whether your meta-aesd layer is already included, if not adds it
  - Runs bitbake core-image-aesd
  - All commands after source command are run from the build dir!

```
git submodule init
git submodule sync
git submodule update
```

```
source poky/oe-init-build-env
```

```
CONFLINE="MACHINE = \"qemuarm64\""

cat conf/local.conf | grep "${CONFLINE}" > /dev/null
local_conf_info=$?
```

```
if [ $local_conf_info -ne 0 ];then
        echo "Append ${CONFLINE} in the local.conf file"
        echo ${CONFLINE} >> conf/local.conf
```

```
bitbake-layers show-layers | grep "meta-aesd" > /dev/null
layer_info=$?
```

```
if [ $layer_info -ne 0 ];then
        echo "Adding meta-aesd layer"
        bitbake-layers add-layer ../meta-aesd
```

```
bitbake core-image-aesd
```

https://github.com/cu-ecen-5013/yocto-assignments-base/blob/master/build.sh

Running your custom image

- source poky/oe-init-build-env
- runqemu nographic
  - Uses environment info to find MACHINE, and location of image.
  - Use runqemu --help to find additional argument options
- Default login is root with no password
    - Modified for you to set to “root” to match buildroot setup

runqemu.sh

```
You can also run generated qemu images with a command like 'runqemu qemux86'
runqemu - INFO - Running bitbake -e...
runqemu - INFO - Continuing with the following parameters:

KERNEL: [/home/aesdbuildbot/yocto-complete-private/build/tmp/deploy/images/qemuarm64/Image--5.0.19+git0+31de88e51d_00638cdd8f-r0-qemuarm64-20200522201347.bin]
MACHINE: [qemuarm64]
FSTYPE: [ext4]
ROOTFS: [/home/aesdbuildbot/yocto-complete-private/build/tmp/deploy/images/qemuarm64/core-image-aesd-qemuarm64-20200522201347.rootfs.ext4]
CONFFILE: [/home/aesdbuildbot/yocto-complete-private/build/tmp/deploy/images/qemuarm64/core-image-aesd-qemuarm64.qemuboot.conf]

runqemu - INFO - Port forward: hostfwd=tcp::10022-:22 hostfwd=tcp::9000-:9000
```

- Sets up slirp network interface for you, does not require root access (as opposed to tap interface default)
  - Matches what we used for buildroot assignments
- All target ports will now be available at 10.0.2.15 (localhost)
  - Use sockettest.sh to reach the target qemu

https://git.yoctoproject.org/poky/tree/scripts/runqemu.README?id=08127d444e05d8d33ce39abdce677655bd4766ea#n49
https://wiki.qemu.org/Documentation/Networking#User_Networking_.28SLIRP.29

Adding your recipe (already done for you)

- Easiest option - use devtool
- Source the build script from the poky folder
  - source poky/oe-init-build-env
- Add the recipe using devtool
  - devtool add aesd-assignments <path to your aesd assignments repo>
    - Generates a recipe based on content of git repo in build/workspace/recipes/recipes-aesd-assignments/aesd-assignments_git.bb

Adding your recipe (already done for you)

- devtool build aesd-assignments
  - Builds the source, verifies it builds successfully
- devtool finish aesd-assignments ../../meta-aesd/
  - Places the .bb file in your yocto recipes tree within your layer.

Modifying your Makefile

- Yocto expects to completely override your makefile variables
  - CC
  - LDFLAGS
  - CFLAGS
- Set these with ?= to allow overrides
- Use CC, CFLAGS, INCLUDES, LDFLAGS IN YOUR COMPILE STEP

```
CC ?= $(CROSS_COMPILE)gcc
CFLAGS ?= -g -Wall -Werror
TARGET ?= aesdsocket
LDFLAGS ?= -lpthread -lrt
```

```
$(CC) $(CFLAGS)  $^ -o $@  $(INCLUDES) $(LDFLAGS)
```

Implementing/Finishing your Recipe

- Implement TODOs in aesd_assignments_git.bb
  - Setup your assignments repo
  - Important to use ssh-agent with yocto, builds will fail if you need interactive login for your SSH password.
- Setup your commit hash in SRCREV

https://github.com/cu-ecen-5013/yocto-assignments-base/blob/master/meta-aesd/recipes-aesd-assignments/aesd-assignments/aesd-assignments_git.bb

Implementing/Finishing your Recipe

- Include files you will be adding to the rootfs in FILES:${PN}
- Add LDFLAGS from your makefile into TARGET_LDFLAGS
- Customize do_install() step to install your files

Start Script Implementation in Yocto

- Yocto uses a similar but slightly different startup
  - System V vs Busybox init
- Your start script can be installed using the update-rc.d class

```
root@qemuarm64:~# ls -la /etc/rc5.d/
drwxr-xr-x    2 root     root          1024 Oct 10 20:01 .
drwxr-xr-x   24 root     root          3072 Oct 10 20:14 ..
lrwxrwxrwx    1 root     root            20 Oct 10 20:01 S01networking -> ../init.d/networking
lrwxrwxrwx    1 root     root            16 Oct 10 20:01 S02dbus-1 -> ../init.d/dbus-1
lrwxrwxrwx    1 root     root            17 Oct 10 20:01 S12rpcbind -> ../init.d/rpcbind
lrwxrwxrwx    1 root     root            21 Sep 10 04:31 S15mountnfs.sh -> ../init.d/mountnfs.sh
lrwxrwxrwx    1 root     root            19 Oct 10 20:01 S19nfscommon -> ../init.d/nfscommon
lrwxrwxrwx    1 root     root            31 Oct 10 20:01 S20aesdsocket-start-stop -> ../init.d/aesdsocket-start-stop
lrwxrwxrwx    1 root     root            19 Oct 10 20:01 S20bluetooth -> ../init.d/bluetooth
lrwxrwxrwx    1 root     root            16 Oct 10 20:01 S20distcc -> ../init.d/distcc
lrwxrwxrwx    1 root     root            20 Oct 10 20:01 S20hwclock.sh -> ../init.d/hwclock.sh
lrwxrwxrwx    1 root     root            19 Oct 10 20:01 S20nfsserver -> ../init.d/nfsserver
lrwxrwxrwx    1 root     root            16 Oct 10 20:01 S20syslog -> ../init.d/syslog
lrwxrwxrwx    1 root     root            22 Oct 10 20:01 S21avahi-daemon -> ../init.d/avahi-daemon
lrwxrwxrwx    1 root     root            22 Sep 10 04:31 S99rmnologin.sh -> ../init.d/rmnologin.sh
lrwxrwxrwx    1 root     root            23 Sep 10 05:40 S99stop-bootlogd -> ../init.d/stop-bootlogd
root@qemuarm64:~#
```

http://www.embeddedlinux.org.cn/OEManual/update-rc-d_class.html

Start Script Implementation in Yocto

- Add these lines to your recipe:
- inherit update-rc.d
  - References class which handles install scripts
- INITSCRIPT_PACKAGES = “${PN}”
  - flag your package as one which uses init scripts
- INITSCRIPT_NAME:${PN} = “your-start-script-name”
- Then in do_install()
  - install -d ${D}${sysconfdir}/init.d
  - install -m ${S}/<your-start-script-name> ${D}${sysconfdir}/init.d

http://www.embeddedlinux.org.cn/OEManual/update-rc-d_class.html - Uses pre-kirkstone references for overrides, replace _ with : before ${PN}

Adding packages to the image

- Add an IMAGE_INSTALL += to your core-image-aesd.bb file referencing your aesd-assignments recipe
  - You can specify other recipes on this line, space separated, to customize your image

https://github.com/cu-ecen-5013/yocto-assignments-base/blob/master/meta-aesd/recipes-aesd-assignments/images/core-image-aesd.bb

Yocto Project Example

- See branch https://github.com/cu-ecen-aeld/yocto-hello-world/tree/ecen5013-hello-world for working example.
- See pull request https://github.com/cu-ecen-aeld/yocto-hello-world/pull/1 for changes to support adding hello world example package to the base project

Devtool Modify and Iterative Changes

- Equivalent of FOO_OVERRIDE_SRCDIR in buildroot to build based on modified source of a specific component.
  - source poky/oe-init-build-env
  - devtool modify <recipename> (ie aesd-assignments)
    - Will output a workspace location
    - cd to this location and edit source code there

https://docs.yoctoproject.org/sdk-manual/extensible.html#use-devtool-modify-to-modify-the-source-of-an-existing-component

Devtool Modify and Iterative Changes

```
dwalkes@AESD-001:~/yocto-complete-private/build$ devtool modify aesd-assignments
INFO: Source tree extracted to /home/dwalkes/yocto-complete-private/build/workspace/sources/aesd-assignments
INFO: Using source tree as build directory since that would be the default for this recipe
INFO: Recipe aesd-assignments now set up to build from /home/dwalkes/yocto-complete-private/build/workspace/sources/aesd-assignments
```

- Next bitbake run (or build.sh) will use source there instead of at your git repository/commit hash
- git add/ commit/git push to push to your repo (switch to master branch)
- Update commit hash in your .bb file
- devtool reset <recipename> to go back to the settings in your .bb file

https://docs.yoctoproject.org/sdk-manual/extensible.html#use-devtool-modify-to-modify-the-source-of-an-existing-component