2nd exercise

NXSOL-OJT-2024

Exported on 2024-01-24 16:15:36

Table of Contents

1 1: 3

1.1 Set up the environment 3

1.1.1 Download base poky 3

1.1.2 Set the build environment 3

1.2 Create a custom layer 3

1.2.1 Create a layer 3

1.2.2 Add the layer 3

1.3 Create a custom image recipe 4

1.3.1 Create an "images" directory in the layer created above 4

1.3.2 Create [name]-image.bb in this directory and write the image recipe in it 4

1.4 Create a hello world recipe 4

1.4.1 Create hello world source 4

1.4.2 Create a hello world recipe 4

1.5 Add the hello world recipe to the core image recipe 5

1.6 Build and run the image 6

2 2: 7

2.1 런타임 패키지 변수에 대해 설명하시오: 7

# 1:

## Set up the environment

### Download base poky

|  |
| --- |
| git clone -b kirkstone git://git.yoctoproject.org/poky.git |

### Set the build environment

|  |
| --- |
| source poky/oe-init-build-env ; cd .. |

## Create a custom layer

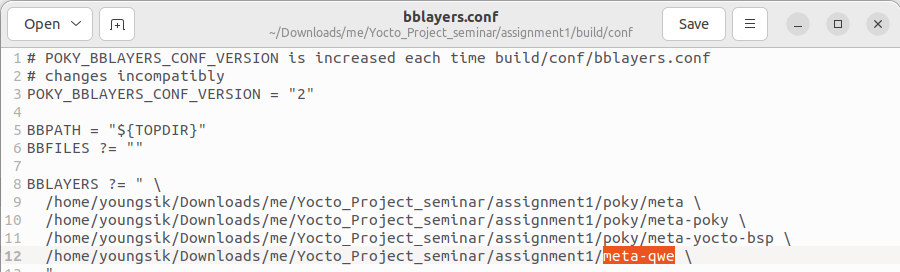
### Create a layer

meta-[name] (e.g. meta-qwe)

|  |
| --- |
| bitbake-layers create-layer meta-[name] |

### Add the layer

Open build/conf/bblayers.conf  and add [root]/meta-[name]



## Create a custom image recipe

### Create an "images" directory in the layer created above

|  |
| --- |
| mkdir -p [root]/meta-[name]/recipes-core/images |

### Create [name]-image.bb in this directory and write the image recipe in it

|  |
| --- |
| DESCRIPTION = "A core image for QWE"  LICENSE = "MIT"    # Core files **for** basic console boot  IMAGE\_INSTALL = "packagegroup-core-boot"    # Add our desired extra files  IMAGE\_INSTALL += "psplash dropbear"    inherit core-image    IMAGE\_ROOTFS\_SIZE ?= "8192" |

## Create a hello world recipe

### Create hello world source

|  |
| --- |
| mkdir -p [root]/meta-[name]/recipes-core/hello/files  touch [root]/meta-[name]/recipes-core/hello/files/hello.c |

|  |
| --- |
| #include <stdio.h>    int main(int argc, char \*\*argv) {  printf("Hello World\n");  **return** 0;  } |

Code Block 1 hello.c

### Create a hello world recipe

|  |
| --- |
| touch [root]/meta-[name]/recipes-core/hello/hello\_1.0.bb |

|  |
| --- |
| DESCRIPTION = "Hello World example"  LICENSE = "MIT"    LIC\_FILES\_CHKSUM = "file://${COREBASE}/meta/COPYING.MIT;md5=3da9cfbcb788c80a0384361b4de20420"    S = "${WORKDIR}"    SRC\_URI = "file://hello.c"    do\_compile() {  ${CC} ${CFLAGS} ${LDFLAGS} hello.c -o hello  }    do\_install() {  install -d -m 0755 ${D}/${bindir}  install -m 0755 hello ${D}/${bindir}/hello  } |

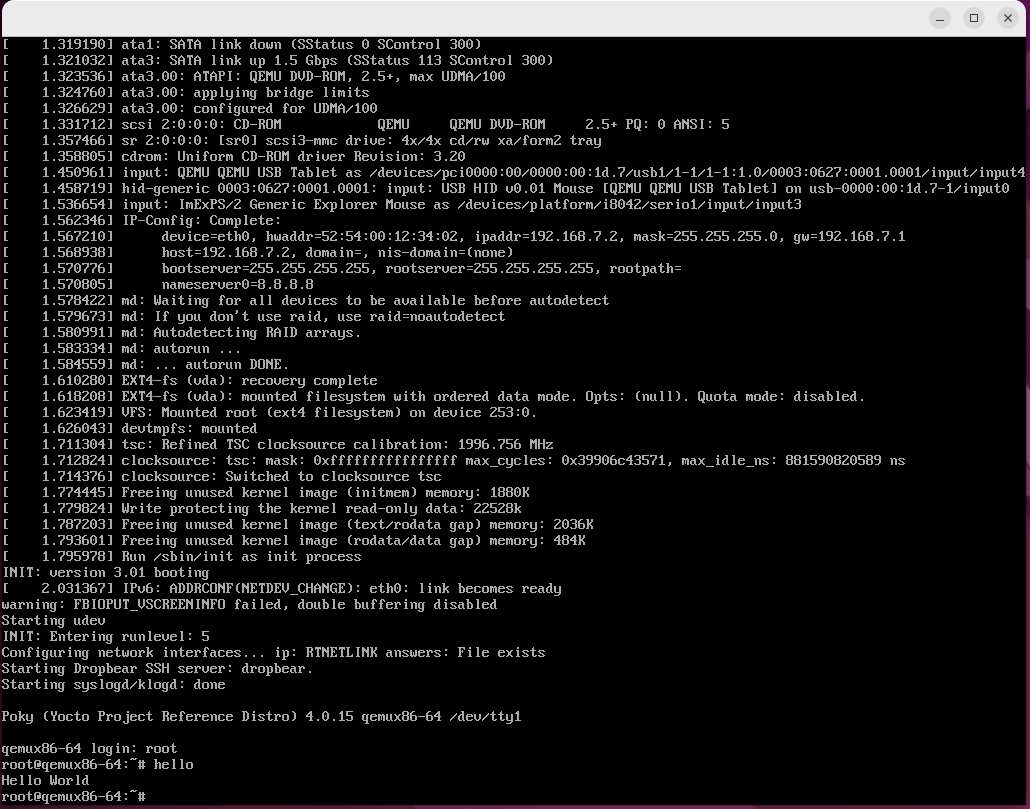
## Add the hello world recipe to the core image recipe

[root]/meta-[name]/recipes-core/images/[name]-image.bb



## Build and run the image

|  |
| --- |
| bitbake -k qwe-image  runqemu qwe-image |



# 2:

## 런타임 패키지 변수에 대해 설명하시오:

* RRECOMMENDS:
  + **Description**: This variable lists packages that are not essential but are recommended to be installed along with the package being defined. If these recommended packages are available, they will be installed by default.
  + **Usage**: It's used when a package can provide additional functionality if another package is present, but can still function without it.
* RSUGGESTS:
  + **Description**: Similar to RRECOMMENDS, RSUGGESTS lists packages that are suggested to be used with the package being defined, but are even less critical. These suggestions are typically not installed by default and are just informational.
  + **Usage**: It's useful for indicating optional packages that could enhance the functionality or user experience but are not directly tied to the package's operation.
* RPROVIDES:
  + **Description**: This variable is used to specify that the package being defined provides the features or functionality of another package (or multiple packages). It's a way of saying, "Installing this package is as good as installing the listed ones."
  + **Usage**: Commonly used in situations where multiple packages can fulfill the same dependency, or when creating a virtual/meta-package that represents a group of packages.
* RCONFLICTS:
  + **Description**: RCONFLICTS is used to declare that the package being defined cannot be installed at the same time as the listed package(s). It's a way to prevent incompatible packages from being installed together.
  + **Usage**: Important for ensuring that packages that would cause problems if installed together are kept separate. For example, two packages that try to install the same file.
* RREPLACES:
  + **Description**: This variable is used to indicate that the package being defined should replace the listed packages. It is often used in conjunction with RCONFLICTS to specify that this package not only conflicts with but also supersedes another package.
  + **Usage**: Commonly used during upgrades or when one package is meant to entirely substitute another, effectively making the old package obsolete.