安装deb文件

Dpkg -i xxx.deb文件

在yocto中添加deb文件

**1. 创建自定义的 Layer**

首先，您需要创建一个新的目录来存放您的 layer。我们命名为 meta-custom。

mkdir meta-custom

进入这个新创建的目录，并创建必要的目录结构：

cd meta-custom

mkdir -p recipes-custom/mydebpackage

在 meta-custom 目录下，创建一个名为 conf/layer.conf 的配置文件，这是每个 Yocto layer 必须有的文件。

touch conf/layer.conf

打开 conf/layer.conf 并添加以下内容：

代码段

# We have a conf and classes directory, add to BBPATH

BBPATH .= ":${LAYERDIR}"

# We have recipes-\* directories, add to BBFILES

BBFILES += "${LAYERDIR}/recipes-\*/\*.bb \

${LAYERDIR}/recipes-\*/\*.bbappend"

BBFILE\_COLLECTIONS += "custom"

BBFILE\_PATTERN\_custom = "^${LAYERDIR}/"

BBFILE\_PRIORITY\_custom = "5"

这个文件告诉 Yocto 的构建系统您的 layer 的基本信息和包含的 recipes 文件。

**2. 添加 xx.deb 包到 Layer**

将您的 xx.deb 包文件放入 meta-custom/recipes-custom/mydebpackage/ 目录下。

# 假设您的deb包在 /path/to/your/deb/xx.deb

cp /path/to/your/deb/xx.deb meta-custom/recipes-custom/mydebpackage/

**3. 创建 Recipe 文件**

在同一目录下，创建一个 .bb 文件来告诉 Yocto 如何处理这个 deb 包。文件名可以随意，但最好能反映包名，例如 mydebpackage\_1.0.bb。

touch meta-custom/recipes-custom/mydebpackage/mydebpackage\_1.0.bb

打开 mydebpackage\_1.0.bb 文件并添加以下内容：

代码段

DESCRIPTION = "A custom recipe to install xx.deb"

LICENSE = "CLOSED"

LIC\_FILES\_CHKSUM = ""

PR = "r0"

SRC\_URI = "file://xx.deb"

S = "${WORKDIR}"

# The do\_install function is used to install the debian package.

# We use dpkg to extract the contents and then copy them to the image root.

do\_install() {

# Extract the debian package contents

dpkg-deb -R ${WORKDIR}/xx.deb ${D}

}

# Explicitly state that this package should be a part of the image

# so that its contents are included in the rootfs.

PACKAGES = "${PN}"

FILES\_${PN} = "\*"

# This makes sure the recipe is marked for installation when building an image.

INHIBIT\_PACKAGE\_STRIP = "1"

**关键点解释：**

* **SRC\_URI = "file://xx.deb"**: 告诉 Yocto 这个包的源文件就在本地的 recipes 目录中。
* **do\_install()**: 这是最重要的一部分。我们使用 dpkg-deb -R 命令来解压 xx.deb 包，并将其内容安装到 ${D} 目录下。${D} 是 Yocto 构建系统为每个 recipe 提供的临时安装目录，最后这些文件会从这里被复制到最终的镜像中。

**4. 将 Layer 添加到构建中**

现在，您需要让 Yocto 的构建系统知道您创建的 meta-custom layer。

在您的构建目录 (poky 或其他名称) 中，找到 conf/bblayers.conf 文件。打开它，并在 BBLAYERS 变量中添加您的 meta-custom layer 的路径。

代码段

# POKY\_BBLAYERS\_CONF\_VERSION is increased each time build/conf/bblayers.conf

# changes in incompatible ways

POKY\_BBLAYERS\_CONF\_VERSION = "2"

BBPATH = "${TOPDIR}"

BBFILES ?= ""

BBLAYERS ?= " \

/path/to/poky/meta \

/path/to/poky/meta-poky \

/path/to/poky/meta-yocto-bsp \

/path/to/your/meta-custom \

"

**请务必将 /path/to/your/meta-custom 替换为您的 meta-custom layer 的实际路径。**

**5. 将包添加到镜像中**

最后，您需要将您的 mydebpackage 添加到您正在构建的镜像中。

在您的镜像 recipe (例如 core-image-full-cmdline.bb) 中，添加您的包名到 IMAGE\_INSTALL 变量中。

代码段

IMAGE\_INSTALL += "mydebpackage"

如果您不想修改现有的镜像 recipe，可以创建一个 .bbappend 文件来追加内容。例如，创建一个名为 core-image-full-cmdline.bbappend 的文件在您的 meta-custom layer 中，并添加以下内容：

代码段

IMAGE\_INSTALL += "mydebpackage"

**6. 构建镜像**

现在，您可以正常地构建您的镜像了。Yocto 会自动识别您的自定义 layer 和 recipe，并将 xx.deb 包的内容安装到最终的镜像中。

Bash

bitbake core-image-full-cmdline

完成这些步骤后，您就可以成功地将您的 xx.deb 包集成到 Yocto 构建的镜像中了。

这将显示如下的目录结构：

meta-custom/

├── conf

│ └── layer.conf

└── recipes-custom

└── mydebpackage

├── mydebpackage\_1.0.bb

└── xx.deb

**结构说明**

* **meta-custom/**：这是你自定义的 layer 的根目录。
* **conf/**：这个目录包含了 layer 的配置信息。
  + **layer.conf**：这是 Yocto 识别并加载此 layer 的关键配置文件，它定义了 layer 的优先级和包含的 recipe 文件模式。
* **recipes-custom/**：这是一个用于存放自定义 recipes 的目录。你可以根据功能或类型创建子目录，例如这里的 mydebpackage。
  + **mydebpackage/**：这个子目录包含了所有与安装 xx.deb 包相关的源文件和 recipe。
    - **mydebpackage\_1.0.bb**：这是 recipe 文件，它告诉 Yocto 如何处理和安装 xx.deb 包。
    - **xx.deb**：这是你想要安装的 Debian 包的源文件。