

Tutorial: Yocto Project

1. Proyecto yocto

El Proyecto Yocto es una colaboración de código abierto, enfocada a desarrolladores de Linux embebido, que permite crear imágenes de Linux personalizables para diferentes plataformas de hardware.

El Proyecto Yocto, a través del sistema de construcción OpenEmbedded, proporciona un entorno de desarrollo de código abierto dirigido a las arquitecturas ARM, MIPS, PowerPC y x86, para una variedad de plataformas, incluyendo beaglebone, x86-64, emuladores, entre otras. Se pueden utilizar, además, los componentes de Yocto para diseñar, desarrollar, construir, depurar, simular y verificar las distintas aplicaciones de software usando Linux, Sistema X Window, GNOME orientado a móviles y frameworks basados en Qt.

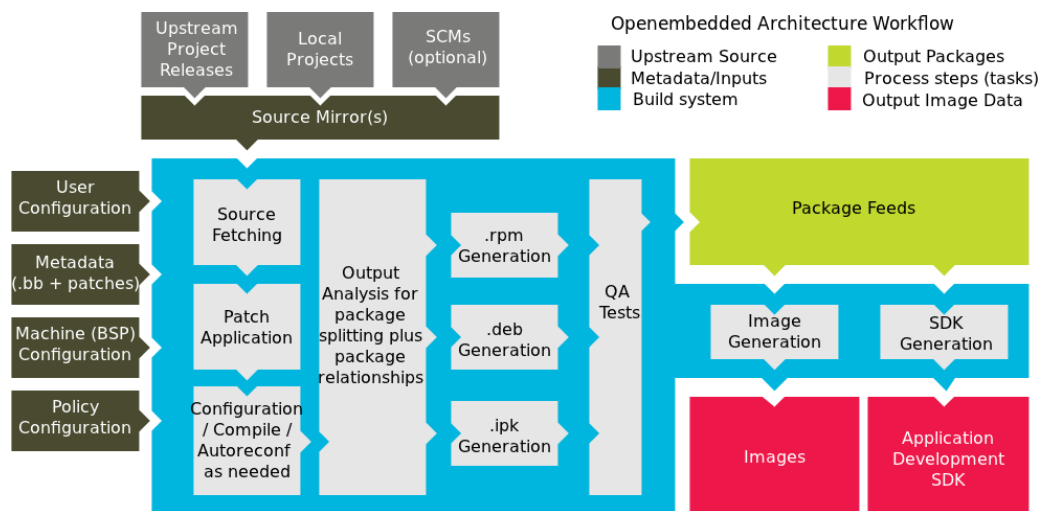


Figura 1: Entorno de desarrollo del Proyecto Yocto

Algunas de las principales características del Proyecto Yocto son:

- Provee un kernel de Linux reciente, junto con un set de comandos y bibliotecas de sistema adecuadas para el entorno embebido.

- Ofrece disponibilidad de componentes de sistema como X11, GTK+, QT, Clutter y SDL (entre otros) lo que permite una experiencia rica en dispositivos con métodos de visualización. Para dispositivos sin esta característica, estos componentes no necesitan ser instalados.
- Crea un núcleo enfocado y estable, compatible con OpenEmbedded, con el cual se puede desarrollar y construir fácil y confiablemente.
- Soporta un amplio rango de hardware y dispositivos de emulación a través de QEMU.

2. Instalando Yocto

Antes de comenzar con la instalación de Yocto, se debe asegurar que se cuenta con el sistema y los paquetes necesarios para su ejecución.

- El sistema anfitrión (host) debe tener una versión soportada de Linux (Fedora, openSUSE, CentOS, Debian, y Ubuntu).
- Suficiente espacio de memoria en disco para la creación de las imágenes (hasta 50GB dependiendo de la imagen).
- Paquetes básicos: Git 1.7.5 o mayor, tar 1.24 o mayor, Python 2.7.3 o mayor excluyendo Python 3.x, el cual no es soportado.

Paquetes requeridos

Se debe instalar, además una lista de paquetes esenciales, gráficos, de soporte y documentación y extras, dependiendo de la distribución de Linux utilizada, los comandos para su instalación son:

En **Ubuntu**:

```
sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib
build-essential chrpath socat libsdl1.2-dev xterm
```

Release Yocto

La última versión de Yocto se puede obtener de: <https://www.yoctoproject.org/software-overview/downloads/> o mediante comando (versión 19.0.1 únicamente) :

```
wget http://downloads.yoctoproject.org/releases/yocto/yocto-2.5.1/poky-sumo-19.0.1.tar.bz2
```

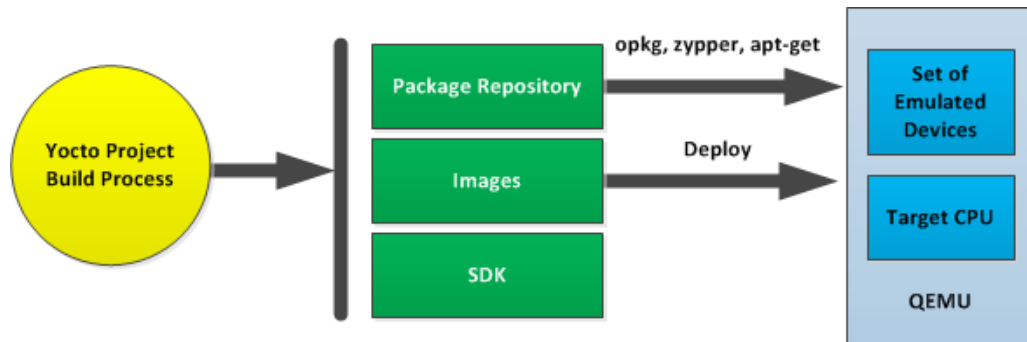


Figura 2: Proceso de construcción de imagen para simulador QEMU

3. Creación de imagen para arquitectura simulada

Una vez que se tiene todos los paquetes instalados junto con Yocto, se puede proceder a realizar la primera imagen y emularla por medio de QEMU.

Para realizar esto, se debe seguir los siguientes comandos:

1. Extraer el archivo comprimido del proyecto yocto e ingresar a directorio:

```
tar -xvf poky-pyro-19.0.1.tar.bz2
cd poky-pyro-19.0.1
```

2. Crear directorio para imagen:

```
mkdir emuARM
```

3. Crear directorio de descargas general (opcional, pero MUY recomendado)

```
mkdir downloads
```

4. Ejecutar script para configuración de ambiente necesario para variables y comandos del proyecto Yocto:

```
source oe-init-build-env emuARM
```

5. Editar archivo de configuración local: (conf/local.conf):

- Agregar al final: (las rutas deben cambiarse según cada usuario y máquina)

```
DL_DIR ?= "/home/usuario/poky-pyro-19.0.1/downloads"
INHERIT += "rm_work"
MACHINE ?= "qemuarm"
```

6. Iniciar construcción de imagen. Puede tardar de 2-6h, dependiendo conexión a internet y capacidad de procesamiento de máquina host:

```
bitbake core-image-minimal
```

7. Al finalizar, podrá ejecutar el simulador con la imagen recién creada, ejecutando:

```
runqemu qemuarm
```

4. Evaluación

Entregable

- Documento con toma de pantalla de la simulación, con la salida del comando `uname -a`. Entrega en tecDigital.