Tutorial para crear las receta para cocinar imagen personalizada usando yocto

Descripcion del sistema Host:

- Se utilizo una maquina virtual con 12 nucleos y 12GB de memoria ram
- Sistema operativo:

```
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.5 LTS
Release:        22.04
Codename:       jammy
```

Aplicacion seleccionada:

Deteccion de vehiculos, personas y sus caracteristicas

Descripcion del problema a resolver:

En los talleres mecanicos es de gran importancia registrar un vehiculo al momento de que este ingrese al taller para algun tipo de mantenimiento ya que es necesario tomar datos como lo son la matricula y ademas tomar fotografias del mismo. Por lo cual, la idea es basandose en este sistema de deteccion de vehiculos y sus caracteristicas, crear un sistema que automaticamente obtenga la placa del vehiculo y guarde fotografias del mismo en una base de datos.

Dependencias requeridas por la aplicacion:

- python 3
- opencv habilitado para usar gstreamer
- gstreamer
- dlstreamer
- openvino

Recetas:

- Clonamos el repo de poky y usamos la version deseada con el siguiente comando:

```
git clone https://git.yoctoproject.org/git/poky && git checkout -t
origin/scarthgap -b my-scarthgap
```

- Clonamos los repos que contienen las capas dentro del repo de Poky:

```
git clone https://git.yoctoproject.org/meta-intel
```

```
git clone https://git.openembedded.org/meta-openembedded
```

`git clone https://github.com/kraj/meta-clang.git`

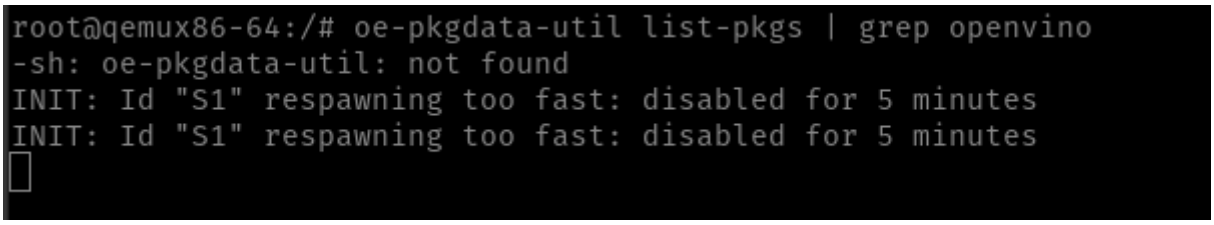
- Ahora dentro de cada uno de los repositorios clonados hay que cambiarlos a la version de yocto que se est'a usando usamos el commando:

`git checkout -t origin/scarthgap -b my-scarthgap`

- Añadimos soporte para Gstreamer y networking agregando las siguientes capas asumiendo que estamos dentro de `build`:

```
bitbake-layers add-layer ../meta-openembedded/meta-multimedia
bitbake-layers add-layer ../meta-openembedded/meta-networking
bitbake-layers add-layer ../meta-intel
bitbake-layers add-layer ../meta-openembedded/meta-oe
bitbake-layers add-layer ../meta-openembedded/meta-python
bitbake-layers add-layer ../meta-clang
```

- La imagen corrio bien pero no tiene los paquetes:



- Ese error no tiene sentido porque el comando se corre en el host machine se puede observar que corre con exito

- Se agrego la capa meta-multimedia para el soporte de gstreamer pero olvide modificar el local.conf agregar lo siguiente:

Añadiendo soporte para GStreamer

- Error debido al requerimiento de licencia

- Correccion añadimos lo siguiente a local.conf `LICENSE_FLAGS_ACCEPTED += "commercial"`
- Correccion eliminar lo anterior para gstreamer y añadir lo siguiente:

GStreamer packages

```
IMAGE_INSTALL:append = " \
    gstreamer1.0 \
    gstreamer1.0-plugins-base \
    gstreamer1.0-plugins-good \
    gstreamer1.0-plugins-bad \
    gstreamer1.0-libav \
    gstreamer1.0-python \
    gstreamer1.0-rtsp-server \
    gstreamer1.0-meta-base \
"
```

Agregando la capa para la aplicacion

- Clonamos el repositorio meta-myapp:

`https://github.com/xavier2200/meta-myapp`

- Ejecutamos:

`bitbake-layers add-layer ../meta-myapp`

- Agregamos el soporte necesario dentro de local.conf

`person-vehicle-detection`

Agregamos dependencias para el programa, utilidades extra para comunicaciones y nuestra aplicacion:

```
IMAGE_INSTALL:append = openssh \
                    openssh-sshd\
                    openssh-sftp \
                    openssh-sftp-server \
                    python3-core \
```

```
                    python3-modules \
                    opencv \
                    dhcpcd \
                    xauth \
                    person-vehicle-detection"
```

Con lo siguiente habilitamos el redireccionamiento del servidor X para las ventanas emergentes

```
PACKAGECONFIG:append:pn-openssh = " x11"
```

Que pasa si se borra la carpeta donde se crea la imagen?

En este caso es necesario borrar la carpeta `/tmp` y volver a cocinar para evitar conflictos.

Ahora si, a cocinar:

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

Referencias:

[1] https://docs.openvino.ai/2024/get-started/install-openvino/install-openvino-yocto.html

[2] https://github.com/yoctoproject/poky

[3] https://gstreamer.freedesktop.org/

[4] https://dlstreamer.github.io/dev_guide/advanced_install/advanced_install_guide_index.html

[5] https://github.com/xavier2200/meta-myapp