Le's Zone

Build Your Custom OpenWrt Image

2021/05/10 | 访问量：2687 次 OpenWrt

This guides to build your OpenWrt Image integrated with custom application step by step.

Setup OpenWrt

Pre-requisites

Debian / Ubuntu

```
sudo apt update
sudo apt install build-essential ccache ecj fastjar file g++ gawk \
gettext git java-propose-classpath libelf-dev libncurses5-dev \
libncursesw5-dev libssl-dev python python2.7-dev python3 unzip wget \
python3-distutils python3-setuptools rsync subversion swig time \
xsltproc zlib1g-dev
```

CentOS / Fedora

```
sudo dnf --skip-broken install bash-completion bzip2 gcc gcc-c++ git \
make ncurses-devel patch perl-Data-Dumper perl-Thread-Queue python2 \
python3 rsync tar unzip wget perl-base perl-File-Compare \
perl-File-Copy perl-FindBin diffutils which
```

For other Linux distributions, you can take this for reference.

Build OpenWrt form source code

You need leave the `sudo` privileged mode to get better cross-compile by `ctrl + d` or `exit`.

1. Get the `OpenWrt` source code

   ```
   git clone https://git.openwrt.org/openwrt/openwrt.git
   cd openwrt
   ```

2. Build from main branch or stable release branch

   ```
   # To build from a specific version
   git branch
   git checkout <branch name>
   # Or just skip to 3 to build from main branch
   ```

3. Update and install feeds

   ```
   ./scripts/feeds update -a
   ./scripts/feeds install -a
   ```

4. Configure

   ```
   make menuconfig
   ```

   E.g. for target “TL-WR841N v11” Wi-Fi router do following:

   - “Target System” ⇒ “Select” ⇒ “Atheros AR7xxx/AR9xxx” ⇒ “Select”
   - “Subtarget” ⇒ “Select” ⇒ “Devices with small flash” ⇒ “Select”
   - “Target Profile” ⇒ “Select” ⇒ “TP-LINK TL-WR841N/ND v11” ⇒ “Select”

   To exit `OpenWrt` configuration and save target with options settings do following:

   - “Exit” ⇒ “Yes”

5. Tool Chain

   ```
   make toolchain/install
   ```

   The target-independent tools and the toolchain are deployed to the `staging_dir/host/` and `staging_dir/toolchain/` directories. This applies to the executables built in the above section as well the pre-built executables available in the SDK. We can adjust the `PATH` variable:

   ```
   export PATH=/home/icn/openwrt/staging_dir/host/bin:$PATH
   ```

6. Build image

   ```
   make
   # For faster compiling, use `make -j N`, where N is the number of CPU cores +
   ```

   And the target image will be generated into `./bin/target/...`

Create your own application

Locate your application

Suppose we create a project named `hellosdewan`.

```
mkdir hellosdewan
cd hellosdewan
pwd
# /home/icn/hellosdewan
```

We create the sole source code file named `hellosdewan.c`.

```
touch hellosdewan.c
```

And input the following content:

```
#include <stdio.h>

int main(void)
{
```

```
    printf("\nHello, sdewan!\n\n");
    return 0;
}
```

Compile and test

Our example application is a `c` application.

```
gcc -c -o hellosdewan.o hellosdewan.c -Wall
gcc -o hellosdewan hellosdewan.o
```

And then, run the `hellosdewan` binary

```
./hellosdewan
# Hello, sdewan!
```

Generate the application package

Create package and configure it

Firstly create a directory to store your own application, like the `openwrt` ,

```
mkdir -p myapps/test/hellosdewan
cd myapps/test/hellosdewan
pwd
# /home/icn/myapps/test/hellosdewan
```

Then we need deliver a package manifest file for `hellosdwan` application, which is responsible for describing the package, what it does and provide instructions on where to obtain the source code, how to build it and which should be contained in the final installable package.

In current directory, create a `Makefile` :

```
include $(TOPDIR)/rules.mk

# Name, version and release number
# The name and version of your package are used to define the variable to point t
PKG_NAME:=hellosdewan
PKG_VERSION:=1.0
PKG_RELEASE:=1

# Source settings (i.e. where to find the source codes)
# This is a custom variable, used below
SOURCE_DIR:=/home/icn/myapps/test/hellosdewan

include $(INCLUDE_DIR)/package.mk

# Package definition; instructs on how and where our package will appear in the o
define Package/hellosdewan
        SECTION:=test
        CATEGORY:=MyApps
        TITLE:=Hello, sdewan!
endef

# Package description; a more verbose description on what our package does
define Package/hellosdewan/description
        A simple "Hello, sdewan!" -application.
endef

# Package preparation instructions; create the build directory and copy the sourc
# The last command is necessary to ensure our preparation instructions remain com
define Build/Prepare
        mkdir -p $(PKG_BUILD_DIR)
        cp $(SOURCE_DIR)/* $(PKG_BUILD_DIR)
        $(Build/Patch)
endef

# Package build instructions; invoke the target-specific compiler to first compil
define Build/Compile
        $(TARGET_CC) $(TARGET_CFLAGS) -o $(PKG_BUILD_DIR)/hellosdewan.o -c $(PKG_
        $(TARGET_CC) $(TARGET_LDFLAGS) -o $(PKG_BUILD_DIR)/$1 $(PKG_BUILD_DIR)/he
endef

# Package install instructions; create a directory inside the package to hold our
define Package/hellosdewan/install
        $(INSTALL_DIR) $(1)/usr/bin
        $(INSTALL_BIN) $(PKG_BUILD_DIR)/hellosdewan $(1)/usr/bin
endef

# This command is always the last, it uses the definitions and variables we give
$(eval $(call BuildPackage,hellosdewan))
```

Integrate the package to build system

`OpenWrt` build system uses a file named `feeds.conf.default` which indicates the package feeds that will be made available during the firmware configuration stage.

```
pwd
# /home/icn/openwrt
cat feed.conf.default
`
src-git packages https://git.openwrt.org/feed/packages.git
```

```
src-git luci https://git.openwrt.org/project/luci.git
src-git routing https://git.openwrt.org/feed/routing.git
src-git telephony https://git.openwrt.org/feed/telephony.git
#src-git video https://github.com/openwrt/video.git
#src-git targets https://github.com/openwrt/targets.git
#src-git management https://github.com/openwrt-management/packages.git
#src-git oldpackages http://git.openwrt.org/packages.git
#src-link custom /usr/src/openwrt/custom-feed
`
```

Modify the file to link to `myapps` package feed with `vim` or other editor in `feeds.conf.default` :

```
src-link myapps /home/icn/myapps
```

Updating and installing feeds

The new feed is defined, so update the build system

```
pwd
# /home/icn/openwrt
./scripts/feeds update myapps
./scripts/feeds install -a -p myapps
```

Build image and test

Build image with custom application

Re-configure your `openwrt` build system

```
pwd
# /home/icn/openwrt
make menuconfig
```

In the menu list, we could find the category named `MyApps` and enter into this category and choose the application we deployed named `hellosdewan` here. Save the configuration and exit.

The changes will be stored to `.config` , then start to build the image

```
make -j 70
```

```
# To locate the error, you can run `make -j1 V=sc`
```

Import image and test

After finishing the image build, we can find the `rootfs` of the image in the `./bin` directory

```
pwd
# /home/icn/openwrt
# Dive into the target to find the `rootfs` image
cd ./bin/targets/x86/64
ls openwrt-x86-64-generic-rootfs.tar.gz
```

Import the image to docker as a docker container

```
sudo docker import openwrt-x86-64-generic-rootfs.tar.gz myapps:hellosdewan
```

Then the image is imported into docker as `myapps` and tag is `hellosdewan`. We can run the docker image and test the application `hellosdewan`

```
sudo docker run -ti myapps:hellosdewan hellosdewan
# Hello, sdewan!
```

Note you can build the `opkg` package solely

```
pwd
# /home/icn/openwrt
make package/hellosdewan/compile
# The `hellosdewan_1.0-1_x86_64.ipk` will be on `./bin/packages/x86_64/myapps`
# And you can use `opkg` to directly install it
opkg install <path-to-package>/hellosdewan_1.0-1_x86_64.ipk
# Or you can remove the installation
opkg remove hellosdewan
```

Patch your application

Prepare

Check the tool for patch is ready

```
pwd
# /home/icn/openwrt
quilt --version
# the path of tools like quilt in under `./staging_dir/host/bin/`
```

Prepare the source code and navigate into the build directory.

```
make package/hellosdewan/{clean,prepare} QUILT=1
cd build_dir/target-x86_64_musl/hellosdewan-1.0
quilt push -a
```

Create a simple patch

```
quilt new 000-add.patch
# If editing existing files
quilt edit hellosdewan.c
# If adding new fiels
quilt add function.c
quilt add function.h
touch function.c
touch function.h
quilt edit function.c
quilt edit function.h
```

`function.c`

```
int add(int a, int b)
{
    return a + b;
}
```

`function.h`

```
int add(int, int);
```

And we can review the change and accept changes as content of patch

```
quilt diff
quilt refresh
```

Include the patch

Migrate the patch data into package

```
pwd
# /home/icn/openwrt
make package/hellosdewan/update
```

We now can review the package feed folder and see the changes

```
ls -la /home/icn/myapps/test/hellosdewan
ls -la /home/icn/hellosdewan
```

`OpenWrt` build system have migrated the patch to the package manifest folder.

Then build and check the new files and changes are present in the build directory

```
pwd
# /home/icn/openwrt
make package/hellosdewan/{clean,prepare}
ls -la build_dir/target-x86_64_musl/hellosdewan-1.0
```

8 comments

Anonymous

Leave a comment

ⓘ Markdown is supported

Login with GitHub

Preview

penaivanalejandro commented 5 months ago

Excellent..!!! Thanks

thinhvu21 commented 4 months ago

Very nice notes, thanks.

After I execute the following commands, I got compile errors when trying to build just a hellosdewan ipk application package alone. I think it could not find a package manifest and other files? Do you know how what is missing?

> make package/hellosdewan/compile

leyao-daily commented 3 months ago

> @thinhvu21
> Very nice notes, thanks.
>
> After I execute the following commands, I got compile errors when trying to build just a hellosdewan ipk application package alone. I think it could not find a package manifest and other files? Do you know how what is missing?
>
> > make package/hellosdewan/compile

Hi, do you config the source dir of the source code correctly. And we should replace `space` with `tab` of the makefile in this webpage either. Do you have any logs for your errors.

thinhvu21 commented 3 months ago

> @leyao-daily
>
> > @thinhvu21
> > Very nice notes, thanks.
> >
> > After I execute the following commands, I got compile errors when trying to build just a hellosdewan ipk application package alone. I think it could not find a package manifest and other files? Do you know how what is missing?
> >
> > > make package/hellosdewan/compile
>
> Hi, do you config the source dir of the source code correctly. And we should replace `space` with `tab` of the makefile in this webpage either. Do you have any logs for your errors.

You are right! It works after I replace "8 spaces" with a "tab"...Thanks.

Do you have a document or a web link to the Docker tool that you are using? I ran the Docker commands you wrote in the above notes but it did not work. I might have missing something. Here is the log after I ran the Docker commands (Note: my rootfs file name is different from yours because of the different in Target):

> sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz openwrt-ath79-generic-8dev_carambola2-rootfs

> ```
> sha256:6d1d6d3266aa2b6f784aa2ab56775867e6556eb9166cdfee1762b47245fc160a
> ```

> ```
> sudo docker images
> REPOSITORY TAG IMAGE ID CREATED SIZE
> openwrt-ath79-generic-8dev_carambola2-rootfs latest 6d1d6d3266aa 13 seconds ago 9.46MB
> myapps hellosdewan f025f9d31e12 3 days ago 9.46MB
> ae13cb54c0a5 3 days ago 9.46MB
> ff20c0ac9923 3 days ago 9.46MB
> 477072121e0d 3 days ago 9.46MB
> myhelloworldcxx_sdk helloworldcxx 8e76709de225 5 days ago 1.01GB
> ```

> ```
> sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz myapps:hellosdewan
> sha256:6dda9483a18e83d2b13c29d21b5754d7800984d997b7ff3946219e5b71a64a7b
> ```

> ```
> sudo docker run -ti myapps:hellosdewan hellosdewan
> docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "hellosdewan": executable file not found in $PATH: unknown.
> ERRO[0000] error waiting for container: context canceled
> ```

leyao-daily commented 3 months ago

> @thinhvu21
>
>> @leyao-daily
>>
>>> @thinhvu21
>>> Very nice notes, thanks.
>>>
>>> After I execute the following commands, I got compile errors when trying to build just a hellosdewan ipk application package alone. I think it could not find a package manifest and other files? Do you know how what is missing?
>>>
>>>> make package/hellosdewan/compile
>>
>> Hi, do you config the source dir of the source code correctly. And we should replace `space` with `tab` of the makefile in this webpage either. Do you have any logs for your errors.
>
> You are right! It works after I replace "8 spaces" with a "tab"...Thanks.

> Do you have a document or a web link to the Docker tool that you are using? I ran the Docker commands you wrote in the above notes but it did not work. I might have missing something. Here is the log after I ran the Docker commands (Note: my rootfs file name is different from yours because of the different in Target):
>
> > sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz openwrt-ath79-generic-8dev_carambola2-rootfs
> > sha256:6d1d6d3266aa2b6f784aa2ab56775867e6556eb9166cdfee1762b47245fc160a
>
> > sudo docker images
> > REPOSITORY TAG IMAGE ID CREATED SIZE
> > openwrt-ath79-generic-8dev_carambola2-rootfs latest 6d1d6d3266aa 13 seconds ago 9.46MB
> > myapps hellosdewan f025f9d31e12 3 days ago 9.46MB
> > ae13cb54c0a5 3 days ago 9.46MB
> > ff20c0ac9923 3 days ago 9.46MB
> > 477072121e0d 3 days ago 9.46MB
> > myhelloworldcxx_sdk helloworldcxx 8e76709de225 5 days ago 1.01GB
>
> > sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz myapps:hellosdewan
> > sha256:6dda9483a18e83d2b13c29d21b5754d7800984d997b7ff3946219e5b71a64a7b
>
> > sudo docker run -ti -ti myapps:hellosdewan hellosdewan
> > docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "hellosdewan": executable file not found in $PATH: unknown.
> > ERRO[0000] error waiting for container: context canceled

My host kernel is 5.4.0-88-generic Ubuntu 20.04.3 and docker version is 20.10.9
I think to run it correctly, you can start with `docker run -ti myapps:hellosdewan bash` or `docker run -ti myapps:hellosdewan sh`,, make sure the command you exec in docker exactly exist.

thinhvu21 commented 3 months ago

Docker still gives errors when I run myapps:hellosdewan. If the commands I entered are correct, then there is a problem with Docker to run an openWrt image as a container. I captured a log below:

> sudo docker version

Client: Docker Engine - Community
Version: 20.10.21
API version: 1.41
Go version: go1.18.7
Git commit: baeda1f
Built: Tue Oct 25 18:02:21 2022
OS/Arch: linux/amd64
Context: default
Experimental: true

> uname -a

Linux Dell 5.15.0-52-generic #58~20.04.1-Ubuntu SMP Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux

> sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz myapps:hellosdewan

> sudo docker images

REPOSITORY TAG IMAGE ID CREATED SIZE
myapps hellosdewan 1005e3ba0150 13 minutes ago 9.46MB

> ls /usr/bin/bash
> /usr/bin/bash

sudo docker run -ti myapps:hellosdewan /usr/bin/bash
docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/bash": stat /usr/bin/bash: no such file or directory: unknown.
ERRO[0000] error waiting for container: context canceled

> ls /usr/bin/sh
> /usr/bin/sh

> sudo docker run -ti myapps:hellosdewan /usr/bin/sh
> docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/sh": stat /usr/bin/sh: no such file or directory: unknown.
> ERRO[0000] error waiting for container: context canceled

leyao-daily commented 3 months ago

> @thinhvu21
> Docker still gives errors when I run myapps:hellosdewan. If the commands I entered are correct, then there is a problem with Docker to run an openWrt image as a container. I captured a log below:
>
> > sudo docker version
>
> ```
> Client: Docker Engine - Community
> Version: 20.10.21
> API version: 1.41
> Go version: go1.18.7
> Git commit: baeda1f
> Built: Tue Oct 25 18:02:21 2022
> OS/Arch: linux/amd64
> Context: default
> Experimental: true
> ```
>
> > uname -a
>
> ```
> Linux Dell 5.15.0-52-generic #58~20.04.1-Ubuntu SMP Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
> ```
>
> > sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz myapps:hellosdewan
>
> > sudo docker images
>
> ```
> REPOSITORY TAG IMAGE ID CREATED SIZE
> myapps hellosdewan 1005e3ba0150 13 minutes ago 9.46MB
> ```
>
> > ls /usr/bin/bash
> > /usr/bin/bash
>
> ```
> sudo docker run -ti myapps:hellosdewan /usr/bin/bash
> docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/bash": stat /usr/bin/bash: no such file or directory: unknown.
> ERRO[0000] error waiting for container: context canceled
> ```
>
> > ls /usr/bin/sh
> > /usr/bin/sh
>
> > sudo docker run -ti myapps:hellosdewan /usr/bin/sh
> > docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/sh": stat /usr/bin/sh: no such file or directory:

> > unknown.
> > ERRO[0000] error waiting for container: context canceled

It seems that your image can not work, whar your target platform and arch? Is `sudo docker run -ti myapps:hellosdewan ls` works?

thinhvu21 commented 3 months ago

> @leyao-daily
>
> > @thinhvu21
> > Docker still gives errors when I run myapps:hellosdewan. If the commands I entered are correct, then there is a problem with Docker to run an openWrt image as a container. I captured a log below:
> >
> > > sudo docker version
> >
> > Client: Docker Engine - Community
> > Version: 20.10.21
> > API version: 1.41
> > Go version: go1.18.7
> > Git commit: baeda1f
> > Built: Tue Oct 25 18:02:21 2022
> > OS/Arch: linux/amd64
> > Context: default
> > Experimental: true
> >
> > > uname -a
> >
> > Linux Dell 5.15.0-52-generic #58~20.04.1-Ubuntu SMP Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
> >
> > > sudo docker import openwrt-ath79-generic-8dev_carambola2-rootfs.tar.gz myapps:hellosdewan
> >
> > > sudo docker images
> >
> > REPOSITORY TAG IMAGE ID CREATED SIZE
> > myapps hellosdewan 1005e3ba0150 13 minutes ago 9.46MB
> >
> > > ls /usr/bin/bash
> > > /usr/bin/bash
> >
> > sudo docker run -ti myapps:hellosdewan /usr/bin/bash
> > docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/bash": stat /usr/bin/bash: no such file or

> > directory: unknown.
> > ERRO[0000] error waiting for container: context canceled
> >
> > > ls /usr/bin/sh
> > > /usr/bin/sh
> >
> > > sudo docker run -ti myapps:hellosdewan /usr/bin/sh
> > > docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "/usr/bin/sh": stat /usr/bin/sh: no such file or directory: unknown.
> > > ERRO[0000] error waiting for container: context canceled
>
> It seems that your image can not work, whar your target platform and arch? Is `sudo docker run -ti myapps:hellosdewan ls` works?

It works after I install Docker Desktop on Linux. Thank you so much for your helps and the documents...

Search

Search

本站总访问量：8085次