napi-rs-opencv交叉编译ohos动态库全流程

- 参考：rust-cross
- 环境：wsl2 + Ubuntu20.04
- 目标：Rust项目编译OpenCV NAPI动态库，目标平台为ohos3.0LTS（aarch64）

1- Rust配置

1.1 安装Rust

Windows 的 Linux 子系统（WSL）下：

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

1.2 更换cargo版本为nightly

```
rustup default nightly
```

```
$ cargo version
cargo 1.60.0-nightly (95bb3c92b 2022-01-18)
```

2- NAPI-RS配置

> **NAPI-RS** is a framework for building pre-compiled **Node.js** addons in **Rust**.
>
> 参考文档：napi-rs-Quick-Start.md

- 参考：Home – NAPI-RS

3- 配置aarch64交叉编译工具链

3.1 拷贝交叉编译所需工具clang和musl (ohos3.0) 到~/env/目录下

> 百度网盘：https://pan.baidu.com/s/17R8pwdSqGYXnWbNitFKmxQ
> 提取码：nuc9

3.2 加入可执行权限

```
~/env/clang/ohos/linux-x86_64/llvm/bin$ sudo chmod +x *
```

```
lynn@DESKTOP-M96JUD3:~/env/clang/ohos/linux-x86_64/llvm/bin$ ls -la
total 723436
drwxr-xr-x 2 lynn lynn     4096 May  6 10:41 .
drwxr-xr-x 7 lynn lynn     4096 May  6 10:42 ..
-rwxr-xr-x 1 lynn lynn 77400400 May  6 10:40 clang
-rwxr-xr-x 1 lynn lynn 77400400 May  6 10:40 clang++
-rwxr-xr-x 1 lynn lynn 77400400 May  6 10:40 clang-10
-rwxr-xr-x 1 lynn lynn 52447872 May  6 10:40 clang-check
-rwxr-xr-x 1 lynn lynn 77400400 May  6 10:40 clang-cl
......
```

4- CMake交叉编译OpenCV(含contrib)

- 参考：OpenCV: Installation in Linux

> 编译完成：aarch64平台下OpenCV4 (含opencv_contrib)
>
> 位置：/home/hit/wxc/OpenCV/out（实验室服务器）

5- 配置OpenCV napi-rs binding

- 参考：twistedfall/opencv-rust: Rust bindings for OpenCV 3 & 4 (github.com)

5.1 拷贝lib/和include/目录到编译主机`opencv_cross_out`目录下

> opencv_cross_out目录为：/home/lynn/opencv_crosscompile/out_contrib

> 将lib.tar拷贝到指定位置，然后`tar -xvf lib.tar`解压
>
> 这样动态库是包含软链接的

5.2 添加环境变量

```
vim ~/.bashrc
```

> 注释：
>
> ```
> #export OpenCV_DIR=/usr/local/lib/cmake/opencv4
> #export LD_LIBRARY_PATH=/usr/local/lib
> ```

append：

```
export OpenCV_DIR=/home/lynn/opencv_crosscompile/out_contrib/lib/cmake/opencv4
export LD_LIBRARY_PATH=/home/lynn/opencv_crosscompile/out_contrib/lib
```

> Getting the OpenCV library (linux):
>
> build OpenCV manually and set up the following environment variables prior to building the project with `opencv` crate:
>
> - `PKG_CONFIG_PATH` for the location of `*.pc` files or `OpenCV_DIR` **for the location of** `*.cmake` **files**
> - `LD_LIBRARY_PATH` **for where to look for the installed** `*.so` **files during runtime**

使环境变量生效：

```
source ~/.bashrc
```

5.3 安装Clang

> Clang (part of LLVM, needed for automatic binding generation)

Additionally, please make sure to install `clang` package or its derivative that contains `libclang.so` and `clang` binary.

- Debian, Ubuntu: `clang` and `libclang-dev`

```
 sudo apt-get update -y
 sudo apt-get install -y libclang-dev
 sudo apt install clang
```

验证安装成功:

```
lynn@DESKTOP-M96JUD3:~$ clang --version
clang version 10.0.0-4ubuntu1
Target: x86_64-pc-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
lynn@DESKTOP-M96JUD3:~$ clang++ --version
clang version 10.0.0-4ubuntu1
Target: x86_64-pc-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
```

5.4 安装ninja

6- 交叉编译

Rust项目中配置.cargo/config.toml

```
[build]
target = "aarch64-unknown-linux-musl"

[env]
CXX = "/home/lynn/env/clang/ohos/linux-x86_64/llvm/bin/clang++"
CC = "/home/lynn/env/clang/ohos/linux-x86_64/llvm/bin/clang"
CFLAGS = "--sysroot=/home/lynn/env/musl --target=aarch64-linux-ohosmusl"
CXXFLAGS = "--sysroot=/home/lynn/env/musl --target=aarch64-linux-ohosmusl"
CXXSTDLIB = "c++"

[target.aarch64-unknown-linux-musl]
linker = "/home/lynn/env/clang/ohos/linux-x86_64/llvm/bin/ld.lld"

rustflags = [
    "-Ctarget-feature=-crt-static",
    "-Clink-self-contained=off",
    "-L/home/lynn/env/musl/usr/lib/aarch64-linux-ohosmusl",
    "-L/home/lynn/env/clang/ohos/linux-x86_64/llvm/lib/aarch64-linux-ohosmusl",
    "-L/home/lynn/env/clang/ohos/linux-x86_64/llvm/lib/aarch64-linux-
ohosmusl/c++",
    "-L/home/lynn/env/clang/ohos/linux-x86_64/llvm/lib"
]

# "-Cembed-bitcode=yes",
#    "-Clink-arg=--gc-sections",
#    "-Clto",

[unstable]
build-std = ["core", "compiler_builtins", "alloc", "std", "panic_unwind"]
build-std-features = ["llvm-libunwind"]
```

```
cargo check
```

```
cargo build --release
```

生成的动态库在//target/aarch64-unknown-linux-musl/release/下

```
lynn@DESKTOP-KIMSKHG:~/ohos-opencv/rust-opencv/target/aarch64-unknown-linux-
musl/release$ file libjsopencv.so
libjsopencv.so: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV),
dynamically linked, not stripped
```

troubleshooting

1- 编译遇到

```
ld.lld: error: unable to find library -lgcc_s
```

改变了基本库（如libgcc），因此要换成不稳定的nightly版本

```
rustup default nightly
```

2-权限问题

```
lynn@DESKTOP-M96JUD3:~/openharmony/marker-detection$ cargo check
error occurred: Command "/home/lynn/env/clang/ohos/linux-
x86_64/llvm/bin/clang++" "-O0" "-ffunction-sections" "-fdata-sections"        "-
fPIC" "-g" "-fno-omit-frame-pointer" "--target=aarch64-unknown-linux-musl" "--
sysroot=/home/lynn/env/musl" "--target=aarch64-l       inux-ohosmusl" "-I"
"/home/lynn/.cargo/registry/src/github.com-1ecc6299db9ec823/opencv-
0.63.1/src_cpp" "-I" "/home/lynn/openharmo       ny/marker-
detection/target/aarch64-unknown-linux-musl/debug/build/opencv-
247faedf2a86603d/out" "-I" "." "-I" "/home/lynn/opencv_c
rosscompile/out_contrib/include/opencv4" "-std=c++11" "-o"
"/home/lynn/openharmony/marker-detection/target/aarch64-unknown-linux-
musl/debug/build/opencv-247faedf2a86603d/out/alphamat.o" "-c"
"/home/lynn/openharmony/marker-detection/target/aarch64-unknown-lin       ux-
musl/debug/build/opencv-247faedf2a86603d/out/alphamat.cpp" with args "clang++"
failed to start: Os { code: 13, kind: Permissio       nDenied, message:
"Permission denied" }
```

```
~/env/clang/ohos/linux-x86_64/llvm/bin$ sudo chmod +x *
```

3-OpenCV-binding配置问题

参考：unresolved imports `opencv::highgui`, `opencv::videoio` · Issue #230 · twistedfall/opencv-rust (github.com)

Specifying the OpenCV *build* directory as *install* directory falsely:

```
OpenCV_DIR = Some("/home/rprata/Projects/opencv/build/")
```

this is currently not supported. Install OpenCV after building and then specify the directory where it was installed to.