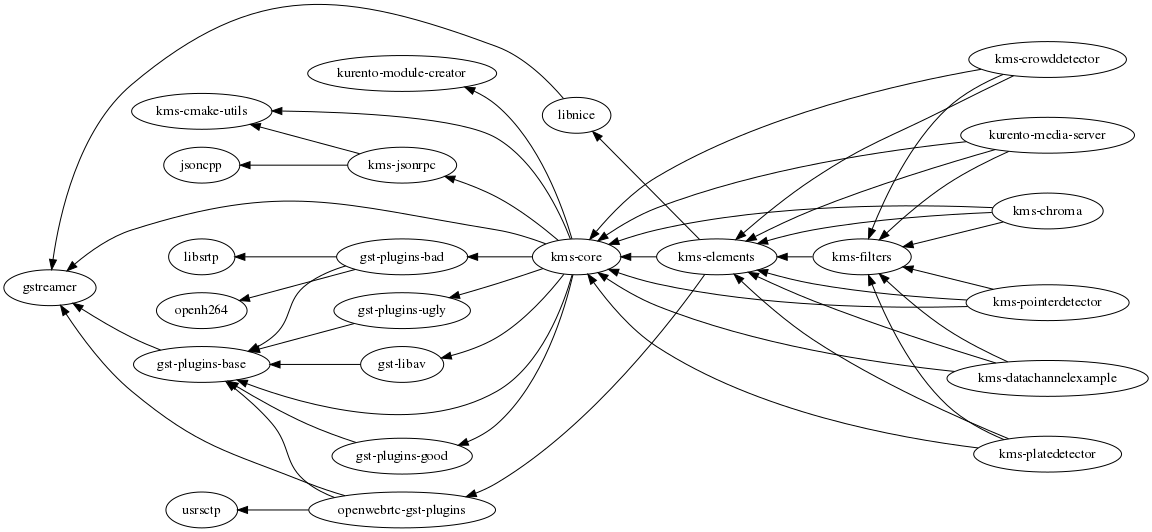
**KMS搭建**

**简介**

KMS使用的工具和技术的概述：

* 使用C和C++语言编写
* CMake是首选的构建工具，用于构建所有模块
* 官方支持的平台是Ubuntu 16.04和Ubuntu18.04
* GStreamer多媒体框架是KMS核心
* 其它库：boost，jsoncpp，libnice等

构成KMS的所有repos依赖关系：



主要的repos：

* kurento-module-creator

代码生成工具，用于为插件生成代码支架，包括KMS代码和Kurento客户代码。主要是Java代码。

* kms-cmake-utils

用于使用Cmake构建KMS的应用程序

* kms-jsonrpc

Kurento协议基于JsonRpc，并使用了此库中包含的JsonRpc库，C++代码

* kms-core

包含核心Gstreamer代码，是其它库所需的基础库

* kms-elements

包含提供管道功能的主要元素，例如WebRTC、RTP、Player、Recorder等

* kms-filters

包含KMS中基本的视频过滤器

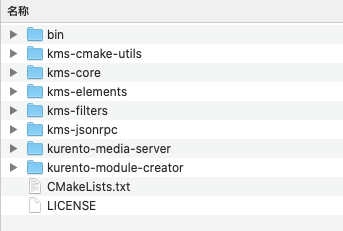
* kurento-media-server

是KMS的主入口，提供服务器可执行的代码main()函数

**源码编译**

**编译环境**

KMS是以Ubuntu系统为目标的C/C++项目，其依赖项管理和分发基于Ubuntu软件包系统。

****

**安装所需工具**

sudo apt-get update && sudo apt-get install --no-install-recommends --yes **\**

build-essential **\**

ca-certificates **\**

cmake **\**

gnupg

**添加Kurento存储库**

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5AFA7A83

DISTRO="xenial" # KMS for Ubuntu 16.04 (Xenial)

DISTRO="bionic" # KMS for Ubuntu 18.04 (Bionic)

sudo tee "/etc/apt/sources.list.d/kurento.list" >/dev/null <<EOF

# Kurento Media Server - Nightly packages

deb [arch=amd64] http://ubuntu.openvidu.io/dev $DISTRO kms6

EOF

sudo apt-get update

**安装编译依赖**

sudo apt-get update && sudo apt-get install --no-install-recommends --yes **\**

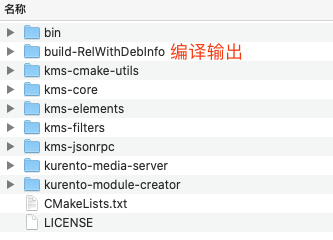
kurento-media-server-dev

**编译**

进入到kurento-omni-build目录

export MAKEFLAGS="-j$(nproc)"

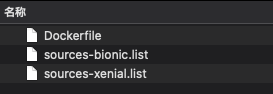
./bin/kms-build-run.sh –build-only --release



**构建Docker镜像**

构建KMS镜像耗时长，大多集中在运行环境的构建，并且运行环境一般情况不变，故把运行环境与编译文件分开构建。

1. **构建编译环境kms-environment**



* 1. 定义构建参数：UBUNTU\_VERSION

ARG UBUNTU\_VERSION="xenial"

* 1. 指定基础镜像为ubuntu:${UbuntuVersion}，”xenial”or “bionic”

FROM ubuntu:${UBUNTU\_VERSION}

* 1. 安装工具gnupg、curl

RUN apt-get update && apt-get install --yes gnupg curl

* 1. 添加Kurento仓库

RUN echo "deb [arch=amd64] http://ubuntu.openvidu.io/6.13.0 ${UBUNTU\_VERSION} kms6" > /etc/apt/sources.list.d/kurento.list

apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5AFA7A83

* 1. 安装依赖环境

RUN apt-get update && apt-get install --yes kurento-media-server-dev

构建命令

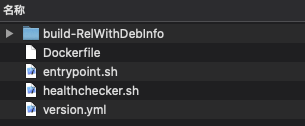
docker build [Args...] --tag huayu/kms-environment:[xenial or bionic] .

构建参数

--build-arg UBUNTU\_VERSION=<UbuntuVersion>

"xenial", "bionic"，可选. 默认: "xenial".

1. **构建编译文件kms**



* 1. 定义构建参数：UBUNTU\_VERSION

ARG UBUNTU\_VERSION="xenial"

* 1. 指定基础镜像为ubuntu:${UbuntuVersion}，”xenial”or “bionic”

FROM huayu/kms-environment:${UBUNTU\_VERSION}

* 1. 复制文件

COPY ./build-RelWithDebInfo /kms/

COPY ./entrypoint.sh /kms/entrypoint.sh

COPY ./healthchecker.sh /kms/healthchecker.sh

* 1. 修改权限

RUN chmod -R 777 /kms

* 1. 配置KMS环境

ENV GST\_DEBUG="3,Kurento\*:4,kms\*:4,sdp\*:4,webrtc\*:4,\*rtpendpoint:4,rtp\*handler:4,rtpsynchronizer:4,agnosticbin:4" \

GST\_DEBUG\_NO\_COLOR=1 \

KMS\_MTU="" \

KMS\_EXTERNAL\_ADDRESS="" \

KMS\_NETWORK\_INTERFACES="" \

KMS\_STUN\_IP="" \

KMS\_STUN\_PORT="" \

KMS\_TURN\_URL=""

* 1. 暴露端口

EXPOSE 8888

* 1. 配置健康检查脚本

HEALTHCHECK --start-period=15s --interval=30s --timeout=3s --retries=1 CMD /kms/healthchecker.sh

* 1. 配置入口脚本

ENTRYPOINT ["/kms/entrypoint.sh"]

构建命令

docker build [Args...] --tag huayu/kms:latest .

构建参数

--build-arg UBUNTU\_VERSION=<UbuntuVersion>

"xenial", "bionic"，可选. 默认: "xenial".

运行命令

docker run -d --name kms -p 8888:8888 \

-v /tmp:/tmp \

-e KMS\_STUN\_IP="10.10.10.10" \

-e KMS\_STUN\_PORT="3478" \

-e KMS\_TURN\_URL="kurento:kurento@10.10.10.10:3478" \

kurento/kurento-media-server:latest