
c++ media ...

发布 0.1

feiyu

2022 年 03 月 20 日

Contents

1	c++	1
1.1	makefile and autotools	1
1.2	gdb	1
2	media	3
2.1	ffmpeg structure	4
2.2	gcc	4
2.3	log system	4
2.4	file delete and rename (api : http://www.ffmpeg.org/doxygen/4.1/avio_8h.html)	4
2.5	list dir	4
2.6	read dic api	5
2.7	Some basic struct for ffmpeg:	5
2.8	Basic Steps	5
2.9	get meta data of audio and video	5
2.10	get audio info	6
2.11	get video info	6
3	python	7
4	java	9

c++ learning space

1.1 makefile and autotools

c++ project 的编译方式大致有以下几种: Makefile, autotools, cmake. 这里只是关于 makefile 以及 autotools 相关入门知识整理, 对于 cmake 因为还没用到暂且不表.

参考资料

1. [跟我一起写 Makefile](#)
2. [tutorial for autotools](#)

1.2 gdb

commands:

1. gcc -g test.c -o test.out
2. gdb ./test.out
3. b(breakpoint) home/xxx/test.c:13
4. r(for run, stop before line 13)
5. n(for next, one step but not step into)

6. Enter(repeat last command)
7. s(for step into)
8. k(kill, kill debugging)
9. info b(show breakpoints)
10. d 1(delete breakpoint 1)
11. c(continue to next breakpoint)
12. bt(back trace, show function call trace)
13. watch(watch point for variable, e.g watch i)
14. info r(show all register)
15. info variables
16. p (for print, show variable value, e.g p i)
17. layout src(ctrl x + a)

CHAPTER 2

media

三角函数

$$x[n] = \text{Amplitude} \cos(\text{frequency} * n + \text{phase})$$

对于连续信号来说, frequency 和 phase always 可以等价转换的, 但是对于离散信号是不成立的, 因为在离散信号中 n always is integer, 并非总是能找到对应的点.

对于离散信号来说一定可以找到另一个 frequency 不同却相同的信号 (例如 frequency - 2π), 但是对于连续信号不成立.

离散信号的周期并非一定是 $2\pi/\text{frequency}$, 因为要取整, 且离散信号并非一定有周期, 当 frequency 与 π 无法约去的时候, $2\pi/\text{frequency}$ 是一个无理数

复指数信号可以通过欧拉公司与三角函数信号进行转换, 所以也是基础信号之一

in discrete time case, 可以通过求和来进行脉冲和阶跃的转换. in continues time case, impulse is the derivative of the step. 单位脉冲函数是阶跃函数的导数.

因果系统 Causality : Output at any time depends only on input prior or equal to that time. 或 System can't anticipate "future" inputs.

Time invariant : “ if $x(t) \rightarrow y(t)$ then $x(t-t_0) \rightarrow y(t-t_0)$

same for $x[t]$ “

linear: ` if  $x(t) \rightarrow y(t)$  then  $x(kt) \rightarrow k*y(t)$  `

ffmpeg tutorial

2.1 ffmpeg structure

1. libavcodec
2. libavformat
3. libavutil
4. libavfilter
5. libavdevice
6. libswresample
7. libswscale

2.2 gcc

compile option :: `g++ -I /usr/local/ffmpeg/include delete_file.cpp -L /usr/local/ffmpeg/lib -lavformat -o delete_file.o`
`g++ -I /usr/local/ffmpeg/include ffmpeg_list_dir.cpp -L /usr/local/ffmpeg/lib -lavformat -lavutil -o ffmpeg_list_dir.o`

2.3 log system

1. should include `<libavutil/log.h>`
2. api:

log :: `av_log_set_level()` `av_log()`

2.4 file delete and rename (api : http://www.ffmpeg.org/doxygen/4.1/avio_8h.htm)

1. access to file system should include `<libavformat/avio.h>`

file option :: `avpriv_io_delete()` `avpriv_io_move()`

2.5 list dir

list dir :: `avio_open_dir()` `avio_open_dir()` `avio_open_dir()`

2.6 read dic api

read dic :: AVIODirContext : context AVIODirEntry : entry

2.7 Some basic struct for ffmpeg:

Basically audio and video are both containers. Each container has many stream that do not cross over with each other. each stream is made by several packet and there is one or more frame in each packet.

- AVFormatContext : show which container is deal with
- AVStream : show which stream is deal with
- AVPacket

2.8 Basic Steps

demux(解复用) -> get streams -> read packets -> release resources

2.9 get meta data of audio and video

apis ::

- av_register_all()
- avformat_open_input()/ avformat_close_input(); output : AVFormatContext
- av_dump_format(); output : meta data

e.g:

compile : g++ -I /usr/local/ffmpeg/include mediainfo.cpp -L /usr/local/ffmpeg/lib -lavutil -lavformat -o mediainfo.o

output:: Input #0, mov,mp4,m4a,3gp,3g2,mj2, from ‘./1.mp4’ : Metadata:

major_brand : isom minor_version : 512 compatible_brands: isomiso2avc1mp41 encoder :
Lavf59.10.100

Duration: 00:00:08.24, bitrate: N/A Stream #0:0[0x1](und): Video: h264 (avc1 / 0x31637661), none(tv, bt709),
1920x1080, 3618 kb/s, SAR 1:1 DAR 16:9, 30 fps, 30 tbr, 15360 tbn (default)

Metadata: handler_name : ISO Media file produced by Google Inc. vendor_id : [0][0][0][0]

Stream #0:1[0x2](und): Audio: aac (mp4a / 0x6134706D), 44100 Hz, 2 channels, 127 kb/s (default)

Metadata: handler_name : ISO Media file produced by Google Inc. vendor_id : [0][0][0][0]

2.10 get audio info

api: `av_init_packet()` `av_find_best_stream()` `av_read_frame()` / `av_packet_unref()`

2.11 get video info

CHAPTER 3

python

CHAPTER 4

java

Hibernate Validator 7.0.4.Final

Jakarta Bean Validation API: jakarta.validation Hibernate Validator core: org.hibernate.validator

annotations are used to declare the constraints, e.g : @NotNull, @Size, annotations should import jakarta.validation.constraints.*

ref:

https://docs.jboss.org/hibernate/stable/validator/reference/en-US/html_single/#validator-gettingstarted