将ffmpeg编译为dll供外部调用

将ffmpeg 编译为dll供外部调用

创建qt应用程序，设置项目为lib

```
// ffmpeg-dll.pro
#TEMPLATE = app
#CONFIG += console
TEMPLATE = lib
```

增加dll函数导出

```
// main.c
__declspec(dllexport) int __cdecl ffmain(int argc, char const *argv[]){
    return main(argc,argv);
}
```

处理命令行参数

既ffmpeg的传入参数并不是以传入的为准，而是内部使用操作系统API又进行了替换，如果在某些系统出现兼容性问题，需要再次修改源代码

```
// cmdutils.c
// split_commandline
// 注释掉 785 行
// prepare_app_arguments(&argc, &argv);
```

使用python 调用

```
import ctypes as ctypes
import time as time



def ffmpeg():
    dll = ctypes.windll.LoadLibrary(
        "D:/cpp-sdk/FFmpeg-n7.0/ffmpeg-exe-dll/Release/release/ffmpeg-dll.dll"
    )
```

```
    ffmpeg_main = dll.ffmain

    ffmpeg_main.argtypes = [ctypes.c_int, ctypes.POINTER(ctypes.c_char_p)]
    ffmpeg_main.restype = ctypes.c_int
    argv = (ctypes.c_char_p * 10)()
    argv[0] = bytes("ffmpeg", "utf-8")
    argv[1] = bytes("-f", "utf-8")
    argv[2] = bytes("gdigrab", "utf-8")
    argv[3] = bytes("-i", "utf-8")
    argv[4] = bytes("desktop", "utf-8")
    argv[5] = bytes("-vcodec", "utf-8")
    argv[6] = bytes("libx264", "utf-8")
    argv[7] = bytes("-f", "utf-8")
    argv[8] = bytes("flv", "utf-8")
    argv[9] = bytes("rtmp://srs-push.qq829.cn:31935/live/557254322", "utf-8")
    ret = ffmpeg_main(10, argv)
    ctypes.windll.kernel32.FreeLibrary.argtypes = [ctypes.c_void_p]
    ctypes.windll.kernel32.FreeLibrary.restype = ctypes.c_int
    ctypes.windll.kernel32.FreeLibrary(dll._handle)
    return ret


while ffmpeg() < 0:
    print("ffmpeg error")
    time.sleep(5)
```

□□□ #2
□ □□□ □□□ 1 □□ 2024 18:01:19
□ □□□ □□□ 18 □□ 2025 03:45:38