

# 數字媒體軟件與系統開發

干皓丞，2101212850, 信息工程學院

2022 年 3 月 6 日

## 1 作業目標與章節摘要

下載 GPAC，理解並描述 random access 過程。

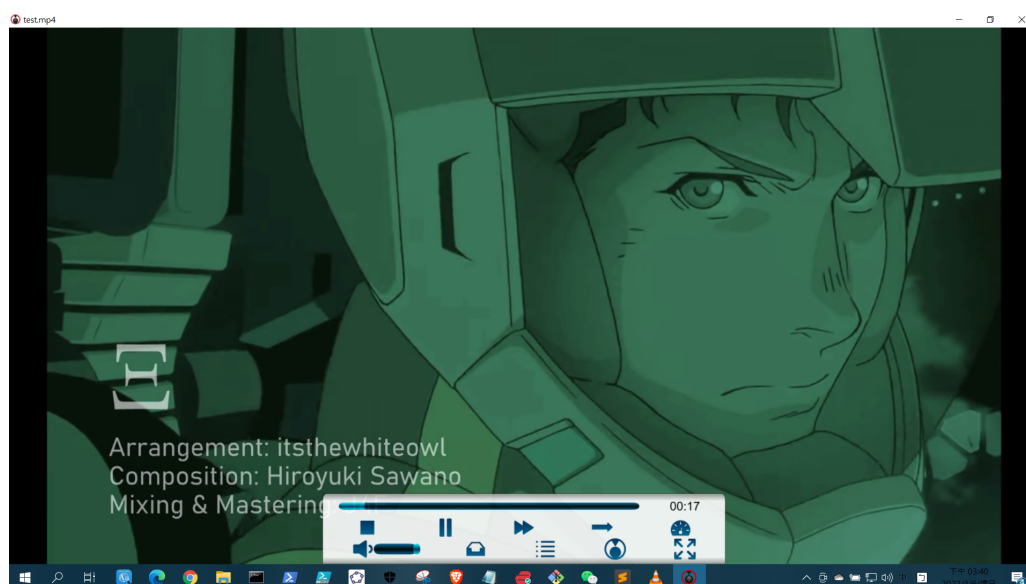


Fig. 1. 使用安裝好的 GPAC 的 Osmo4 播放

## 2 文章與作業狀況

作業可以從 GitHub 下的 [kancheng/kan-cs-report-in-2022](#) 專案找到，作業程式碼與文件目錄為 [kan-cs-report-in-2022/DMSASD/gpac-random-access](#)。實際執行的環境與實驗設備為 Google 的 Colab、MacBook Pro (Retina, 15-inch, Mid 2014)、Acer Aspire R7 與 HP Victus (Nvidia GeForce RTX 3060)。

## 3 作業內容概述

此作業分二大部分，第一部分說明 GPAC 使用與理解，第二部分則描述描述 Random Access 過程。

1. GPAC 使用與理解
2. 描述 Random Access 過程

## 4 GPAC 使用與理解

GPAC 是一個 LGPL v2.1 且在大多數情況下也可以在商業許可下使用的開源多媒體框架，其專案提供了使用者在處理、檢查、打包、流式傳輸、播放和與媒體內容交互的工具。此類內容可以為音頻、影像、字幕、元數

據、可縮放圖形、加密媒體、2D/3D 圖形和 ECMAScript 等任意組合。GPAC 以其廣泛的 MP4/ISOBMFF 功能而聞名，深受影像愛好者、學術研究人員、標準化機構和專業廣播公司的歡迎。

前往 GPAC (<https://gpac.wp.imt.fr/>) 下載，安裝後即可以使用。開啟終端機查看版本。

1. GPAC 的官方開發文件: <https://doxygen.gpac.io/>
2. GPAC 的 Wiki: <https://github.com/gpac/gpac/wiki/Howtos>

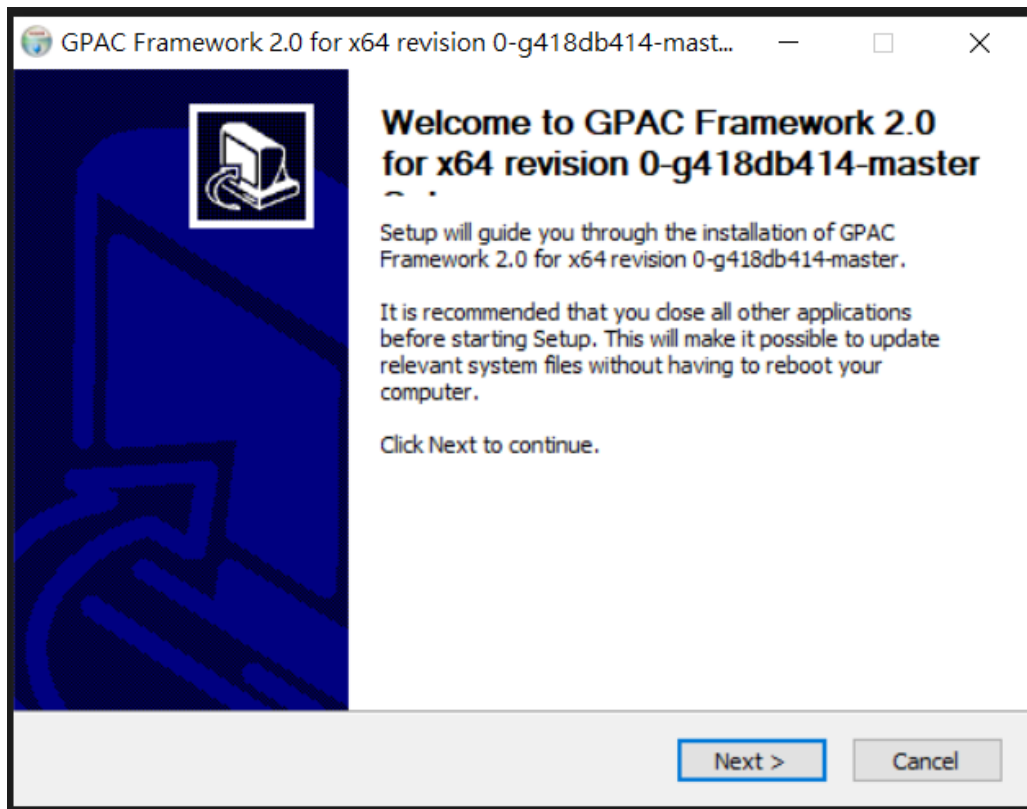


Fig. 2. 在此使用 GPAC 的 Windows 版本安裝

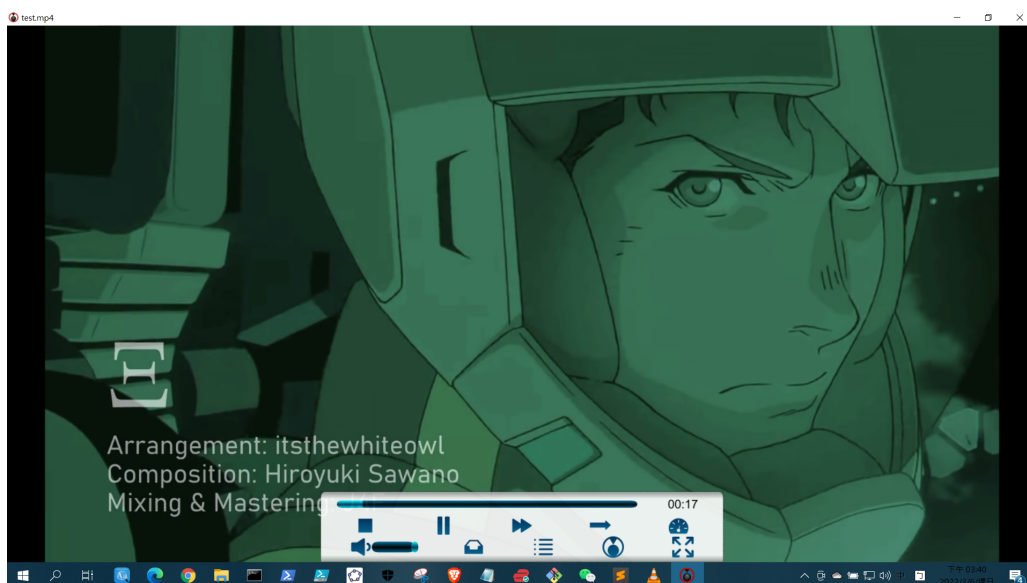
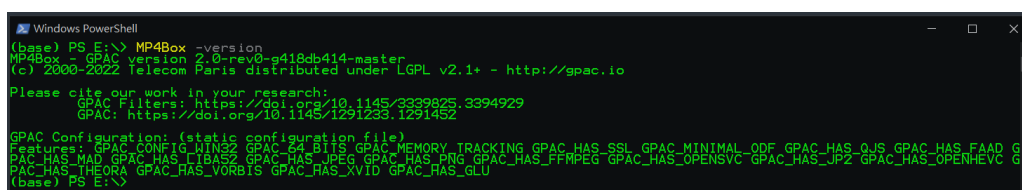


Fig. 3. 完成

1. 查看 mp4box 版本

```
1 mp4box -version
```



```
Windows PowerShell
(base) PS E:\> MP4Box -version
MP4Box - GPAC version 2.0-rev0-g418db414-master
(c) 2000-2022 Telecom Paris distributed under LGPL v2.1+ - http://gpac.io

Please cite our work in your research:
GPAC Filters: https://doi.org/10.1145/3339825.3394929
GPAC: https://doi.org/10.1145/1291233.1291452

GPAC Configuration: (static configuration file)
Features: GPAC_CONFIG_WIN32 GPAC_24_BITS GPAC_MEMORY_TRACKING GPAC_HAS_SSL GPAC_MINIMAL_ODF GPAC_HAS_QJS GPAC_HAS_FAAD G
PAC_HAS_MAD GPAC_HAS_LIBAS2 GPAC_HAS_JPEG GPAC_HAS_PNG GPAC_HAS_FFMPEG GPAC_HAS_OPENSVC GPAC_HAS_JP2 GPAC_HAS_OPENHEVC G
PAC_HAS_THEORA GPAC_HAS_VORBIS GPAC_HAS_XVID GPAC_HAS_GLU
(base) PS E:\>
```

Fig. 4. 查看 GPAC 的 mp4box 版本

## 2. 查看 mp4box 操作指令

```
1 # 1
2 mp4box -h
3 # 查看 mp4box 中的所有幫助信息
4
5 # 2
6 mp4box -h general
7 # 查看 mp4box 中的通用幫助信息
8
9 # 3
10 mp4box -info test.mp4
11 # 查看 test.mp4 文件是否有問題
12
13 # 4
14 mp4box -add test.mp4 test-new.mp4
15 # 修復 test.mp4 文件格式不標準的問題，並把新文件保存在 test-new.mp4 中
16
17 # 5
18 mp4box -inter 10000 test-new.mp4
19 # 解決開始播放 test-new.mp4 卡一下的問題，為 HTTP 下載快速播放有效，10000ms
20
21 # 6
22 mp4box -add file.avi new_file.mp4
23 # 把 avi 文件轉換為 mp4 文件
24
25 # 7
26 mp4box -hint file.mp4
27 # 為 RTP 準備，此指令將為文件創建 RTP 提示跟蹤信息。
28 # 這使得經典的流媒體服務器像 darwinstreamingserver 或 QuickTime 的流媒體服務器通
    過 RTSP/RTP 傳輸文件
29
30 # 8
31 mp4box -cat test1.mp4 -cat test2.mp4 -new test.mp4
32 # 把 test1.mp4 和 test2.mp4 合併到一個新的文件 test.mp4 中，要求編碼參數一致
33
34 # 9
35 mp4box -force-cat test1.mp4 -force-cat test2.mp4 -new test.mp4
```

```

36 # 把 test1.mp4 和 test2.mp4 強制合併到一個新的文件 test.mp4 中，有可能不能播放
37
38 # 10
39 mp4box -add video1.264 -cat video2.264 -cat video3.264 -add audio1.aac -cat
    audio2.aac -cat audio3.aac -new muxed.mp4 -fps 24
40 # 合併多段音視頻並保持同步
41
42 # 11
43 mp4box -split *time_sec* test.mp4
44 # 切取 test.mp4 中的前面 time_sec 秒的視頻文件
45
46 # 12
47 mp4box -split-size *size* test.mp4
48 # 切取前面大小為 size KB的視頻文件
49
50 # 13
51 mp4box -split-chunk *S:E* test.mp4
52 # 切取起始為 S 秒，結束為 E 秒的視頻文件
53
54 # 14
55 mp4box -add 1.mp4#video -add 2.mp4#audio -new test.mp4
56 # test.mp4 由 1.mp4 中的視頻與 2.mp4 中的音頻合併生成

```

## 5 描述 Random Access 過程

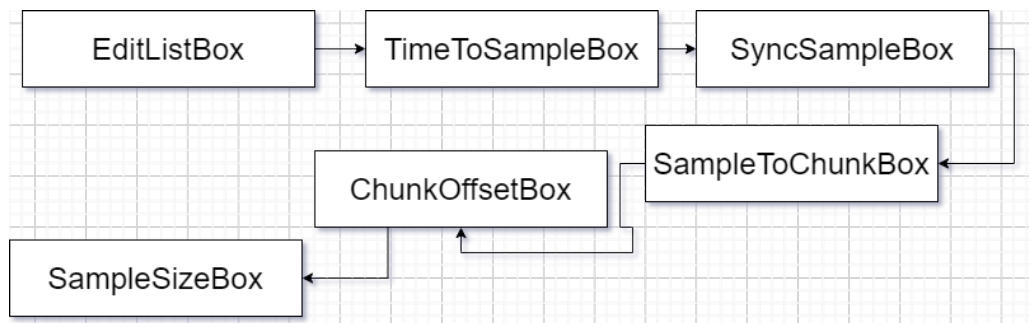


Fig. 5. 描述 Random Access 過程圖

If you want to seek a given track to a time  $T$ ,

假如想要最一個文件進行隨機訪問，而該訪問進行為  $T$  時刻 EX: 第三分五十秒。

1. If the track contains an edit list, determine which edit contains the time  $T$  by iterating over the edits.  $T_{movie} = T_{start} + T'$

如果軌道包含編輯列表，則通過迭代編輯確定哪個編輯包含時間  $T$ 。 $T_{movie} = T_{start} + T'$

先找當中有沒有 edit list，當中 edit list 實際上存著一段段的樣本信息，這裡面會有一個關鍵字段的有開始時刻 ( $T_{start}$ ) 的連續樣本。這個過程就是將  $T$  換算成  $T_{start} + T'$ ，而這個  $T'$  就是換算出來的新的時間。

2. Convert to media time scale  $T_{media} = T_{start}' + T''$

然後再將原先的過程轉換為媒體時間尺度  $T_{media} = T_{start}' + T''$

3. Use time-to-sample box to find the first sample prior to the given time

使用 time-to-sample box 找到給定時間之前的第一個樣本，也就是對應的樣本編號。

4. Consult the sync sample table to seek to which sample is closest to, but prior to, the sample found above 因為找到的樣本不一定是可以解的，為了保險會先查閱同步樣本表以尋找最接近但先於上面所找到的樣本的樣本編號。

5. Use the sample-to-chunk table to determine in which chunk this sample is located.

使用 sample-to-chunk 來確定該樣本位於哪個 chunk 中。

6. use the chunk offset box to figure out where that chunk begins

找到後根據使用 the chunk offset box 來確定該 chunk 開始的物理存儲位置。

7. Starting from this offset, you can use the information contained in the sample-to-chunk box and the sample size box to figure out where within this chunk the sample in question is located.

從這個偏移量開始，您可以使用包含在 sample-to-chunk box 和 the sample size box 中的信息來確定有問題的樣本在這個 chunk 中的位置。

尋找專案中有關 Random Access 的部分，利用 find 和 grep 指令，尋找 GPAC 與 Random Access 過程有關的檔案。

```
1 find . -name "*.h" | xargs grep "random access" *.*
```

```
USER@Aspire-R7 MINGW64 /d/USERDATA/Downloads/gpac-master/gpac-master
$ find . -name "*.h" | xargs grep "random access" *.*
grep: ./: Is a directory
grep: ./github: Is a directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/archive.bat: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/build_installer.bat: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/gpac.inf: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/readme.txt: No such file or directory
grep: ./applications/generators/SVG/Tiny-1.2-NG: Is a directory
./applications/mp4box/fileimport.c: /*make random access for storage*/
./applications/mp4box/main.c: "The input file must have enough random access points in order to
./applications/mp4box/main.c: MP4BOX_ARG("rap", "ensure that segments begin with random access points,
FUN | ARG_EMPTY),
./applications/mp4box/main.c: MP4BOX_ARG("frag-rap", "ensure that all fragments begin with random access
./applications/mp4box/main.c: "MP4Box can encode BIFS or LAsER streams and insert random access
urn in the scene at the same time. In MPEG-4 terminology, this is called the __scene carousel__."
./applications/mp4box/main.c: MP4BOX_ARG("rap", "signal random access points in RTP packets (MPEG-4 Sys
grep: ./build/xcode/gpac.xcodeproj: Is a directory
grep: ./build/xcode/gpac4ios.xcodeproj: Is a directory
./extra_lib/include/zlib/zlib.h: random access is desired. Using Z_FULL_FLUSH too often can seriously de
./include/gpac/constants.h: /*!M: is-RAP: DIMS unit is a random access point*/
./include/gpac/filters.h: - Packets have default attributes such as timestamps, size, random access
./include/gpac/html5_media.h: Bool needs_rap; /* MSE need random access point flag
./include/gpac/ietf.h: /*random access point flag present*/
./include/gpac/internal/swf_dev.h: 1- all BIFS AUs in sprites are random access
./include/gpac/isomedia.h:/*! enables mfra (movie fragment random access computing) when writing movie fr
./include/gpac/isomedia.h:\param IsRandomAccessPoint set to GF_TRUE if you want to indicate that this is
./include/gpac/isomedia.h:The resulting sample will always be marked as random access
./include/gpac/media_tools.h:\param is_rap set to GF_TRUE to signal a random access point
./include/gpac/mpegts.h: /*! random access indicator*/
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/scene_manager.h: /*AU is RAP - random access indication - may be overridden by encoder*/
./include/gpac/scene_manager.h:stream, this AU being a random access for the stream
grep: ./packagers/osx/GPAC_app: Is a directory
```

Fig. 6. 尋找 GPAC 與 Random Access 過程有關的檔案


```
1 ./ applications /mp4box/ fileimport .c
2 ./ applications /mp4box/ main .c
3 ./ extra_lib /include/ zlib/ zlib .h
4 ./ include/gpac/ constants .h
5 ./ include/gpac/ filters .h:
6 ./ include/gpac/ html5_media .h:
7 ./ include/gpac/ ietf .h:
```

```

8 ./include/gpac/internal/swf_dev.h:
9 ./include/gpac/isomedia.h:
10 ./include/gpac/isomedia.h:
11 ./include/gpac/media_tools.h:
12 ./include/gpac/mpegts.h:
13 ./include/gpac/rtp_streamer.h:
14 ./include/gpac/scene_manager.h:
15 ./share/doc/man/gpac-filters.1:mfra (bool, default: false):
16 ./share/doc/man/mp4box.1:
17 ./src/filters/mux_isom.c:
18 ./src/media_tools/html5_mse.c:
19 ./src/media_tools/m2ts_mux.c:
20 ./src/scene_manager/text_to_bifs.c:

```

同時可以從 GPAC 的官方文件看到 Random Access 的部分。



**libgpac**  
Documentation of the core library of GPAC

Main Page Modules Data Structures Files

GF\_M2TS\_TimelineCodebook  
GF\_M2TS\_Program  
GF\_M2TS\_ES  
GF\_M2TS\_PESHeader  
GF\_M2TS\_SECTION\_ES  
tag\_m2ts\_dvb\_sub  
tag\_m2ts\_dvb\_teletext  
tag\_m2ts\_metadata\_descrip  
GF\_M2TS\_Header  
GF\_M2TS\_AdaptationField  
discontinuity\_indicator  
random\_access\_indicator  
priority\_indicator  
PCR\_flag  
PCR\_base

GF\_M2TS\_AdaptationField

struct GF\_M2TS\_AdaptationField

MPEG-2 TS packet adaptation field

Data Fields		
u32	discontinuity_indicator	discontinuity indicator (for timeline splicing)
u32	random_access_indicator	random access indicator
u32	priority_indicator	priority indicator
u32	PCR_flag	PCR present flag
u64	PCR_base	PCR base value

Fig. 7. 文件 GPAC 與 Random Access