



链滴

关于使用 ExifTool 批量修复照片、视频的媒体创建时间，使其在相册正确排序，学习参考代码

作者: [HaujetZhao](#)

原文链接: <https://ld246.com/article/1634824702145>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

如果一个照片、视频的元数据中没有媒体创建时间的记录，那在相册中时，相册会默认以文件的创建时间、修改时间排序。

从网上下载来的照片、手机的截图、录屏视频，通常是没有媒体创建时间的。

假如你创建了一个此图片、视频的副本、修改版、压缩版，新文件的创建时间就会很新，本来一个拍日期很久远的照片，在相册中就会排到很靠前。

因此，有必要为视频、图片添加上正确的创建时间。

众所周知，视频、照片、截图、录屏的自动命名，都会用到创建时的时间戳，所以我们就以两个例子，对媒体文件的媒体创建时间元数据进行纠正：

- 如果文件名中有正确的时间戳，就以文件名为线索纠正，写入到元数据
- 如果文件名中没有正确的时间戳，就以文件的创建时间为线索，写入到元数据

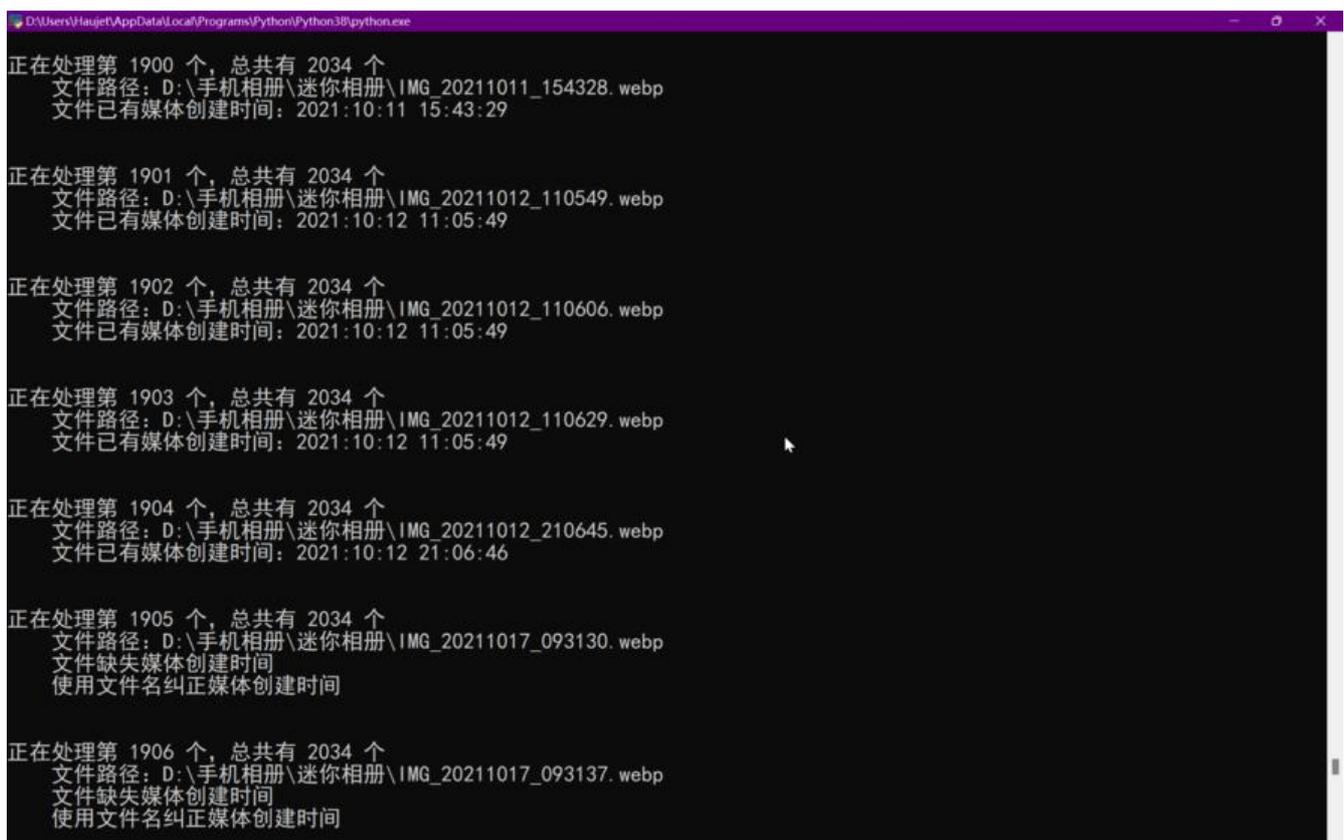
ExifTool 可以对多种格式（包括视频、图片）读写元数据，用它来修复再合适不过了。

考虑到可能有许多文件需要批处理，还要判断原文件是否有正确的时间元数据，因此一般几行命令行法满足需求，因此我写了个 Python 脚本，可以

- 对文件、文件夹自动判断，如果是文件夹，就进行递归操作
- 判断有没有正确的时间元数据
- 根据上述两条线索修复时间元数据
- 利用 ExifTool 的 `-stay_open` 参数，在一个实例中操作所有图片，而不是每一个图片都要打开一个 ExifTool 实例，因此操作速度飞起！

另外也学习到，原来文件名中加入时间戳这么重要！

实现效果：



```
D:\Users\Hajet\AppData\Local\Programs\Python\Python38\python.exe
正在处理第 1900 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211011_154328.webp
文件已有媒体创建时间：2021:10:11 15:43:29

正在处理第 1901 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211012_110549.webp
文件已有媒体创建时间：2021:10:12 11:05:49

正在处理第 1902 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211012_110606.webp
文件已有媒体创建时间：2021:10:12 11:05:49

正在处理第 1903 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211012_110629.webp
文件已有媒体创建时间：2021:10:12 11:05:49

正在处理第 1904 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211012_210645.webp
文件已有媒体创建时间：2021:10:12 21:06:46

正在处理第 1905 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211017_093130.webp
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 1906 个，总共有 2034 个
文件路径：D:\手机相册\迷你相册\IMG_20211017_093137.webp
文件缺失媒体创建时间
使用文件名纠正媒体创建时间
```

原文链接：[关于使用 ExifTool 批量修复照片、视频的媒体创建时间，使其在相册正确排序，学习参考代码](#)

```
D:\Users\Hajiet\AppData\Local\Programs\Python\Python38\python.exe
文件路径: D:\手机相册\迷你相册\VID_20210701_155817~2. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 2030 个, 总共有 2034 个
文件路径: D:\手机相册\迷你相册\VID_20210710_175621. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 2031 个, 总共有 2034 个
文件路径: D:\手机相册\迷你相册\VID_20210929_164151_HSR_120. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 2032 个, 总共有 2034 个
文件路径: D:\手机相册\迷你相册\VID_20210929_164159_HSR_120. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 2033 个, 总共有 2034 个
文件路径: D:\手机相册\迷你相册\VID_20211016_134312. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

正在处理第 2034 个, 总共有 2034 个
文件路径: D:\手机相册\迷你相册\VID_20211017_174815. mp4
文件缺失媒体创建时间
使用文件名纠正媒体创建时间

处理结束, 回车结束
```

代码链接: [拖入处理文件/修复照片视频创建时间.py · 淳帅二代/我的 Python 脚本](#)

代码:

```
'''
```

脚本作用:

使用 exiftool 根据文件名、新建时间, 纠正文件元数据中的媒体创建时间。
使用前要先确保 exiftool 可执行文件路径已添加到环境变量

使用方法:

python 本脚本名字.py 文件或文件夹路径 其它文件或文件夹路径

(如果是文件夹路径, 会递归修复。如果检测到有媒体创建时间, 则不会改动原文件)

```
'''
```

```
import sys, time, os
from os import path
import subprocess, shlex, re
import locale
from pprint import pprint
from datetime import datetime
```

```
# 合法图片后缀, 一行一个
图片后缀 = '''
jpg
```

```

.jpeg
.png
.webp
.heic
"".strip()

# 合法视频后缀，一行一个
视频后缀 = ''
.mp4
.webm
.flv
.mkv
"".strip()

def 处理文件(process, 文件路径):

    def 开始执行():
        process.stdin.write(f'-execute\n')
        process.stdin.flush()
        fd = process.stdout.fileno()
        output = b''
        while not output.decode(locale.getpreferredencoding()).strip().endswith(r'{ready}'):
            output += os.read(fd, 4096)
        output = output.decode().strip()[:-len(r'{ready}')].strip()
        return output

    # 时区用于调整视频 UTC 时间
    时区 = time.strftime("%z", time.gmtime())
    时区小时, 时区分钟 = 时区[1:3], 时区[3:5]

    # 先看下有没有媒体创建时间
    process.stdin.write(f'-s\n')
    process.stdin.write(f'-s\n')
    process.stdin.write(f'-s\n')
    process.stdin.write(f'-createdate\n')
    process.stdin.write(f'{文件路径}\n')
    output = 开始执行()
    # 对于图片文件，只要有 createdate，就是有
    # 对于视频文件，如果没有 createdate，读出来的就全是 0000:00:00 00:00:00
    # 这时，如果都是 0，也应当纠正下时间
    output_num = "".join(re.findall(r'\d+', output))

    if output and int(output_num):
        print(f' 文件已有媒体创建时间: {output}')
    else:
        print(f' 文件缺失媒体创建时间')
        文件名 = path.basename(文件路径)
        文件名时间 = 从文件名得到时间(文件名)
        if 文件名时间:
            print(f' 使用文件名纠正媒体创建时间')
            process.stdin.write(f'-alldates={文件名时间}\n')
        else:
            print(f' 使用文件创建时间纠正媒体创建时间')

```

```

    process.stdin.write(f'-alldates<FileCreateDate\n')

process.stdin.write(f'-overwrite_original\n')
process.stdin.write(f'{文件路径}\n')
开始执行()

if path.splitext(文件路径)[1] in 视频后缀.splitlines():
    print(f' 对视频文件元数据纠正 UTC 偏差: {时区小时}:{时区分钟}:00')
    process.stdin.write(f'-alldates-={时区小时}:{时区分钟}:00\n')
    process.stdin.write(f'-overwrite_original\n')
    process.stdin.write(f'{文件路径}\n')
    开始执行()
print('\n')

def 获得文件夹内所有媒体文件(文件夹路径):
    文件列表 = []
    for root, dirs, files in os.walk(文件夹路径):
        for file in files:
            文件路径 = path.join(root, file)
            if path.splitext(文件路径)[1].lower() in 图片后缀.splitlines() + 视频后缀.splitlines():
                文件列表.append(文件路径)
    return 文件列表

def 从文件名得到时间(文件名):
    全数字 = ''.join(re.findall(r'\d+', 文件名))
    当前时间 = datetime.now()

    try:
        # 类似 IMG_20170606_192311.jpg
        数字转时间 = datetime.fromisoformat(
            f'{全数字[0:4]}-{全数字[4:6]}-{全数字[6:8]} {全数字[8:10]}:{全数字[10:12]}:{全数字[12:14]}'
        )
        if 数字转时间 < 当前时间:
            return 数字转时间
    except:
        ...

    try:
        # 类似 mmexport1111467816512.jpg
        数字转时间 = datetime.fromtimestamp(
            int(全数字[0:13]) / 1000
        )
        if 数字转时间 < 当前时间:
            return 数字转时间
    except:
        ...

    try:
        # 类似 121301190818c8f0c7010755.mp4
        数字转时间 = datetime.fromisoformat(
            f'20{全数字[10:12]}-{全数字[8:10]}-{全数字[6:8]} {全数字[4:6]}:{全数字[2:4]}:{全数字[0:2]}').t
mestamp()
        if 数字转时间 < 当前时间:

```

```

        return 数字转时间
    except:
        ...

    return False

def main():
    process = subprocess.Popen(
        shlex.split(
            f'exiftool -stay_open True -@ -'
        ),
        stdin=subprocess.PIPE, stdout=subprocess.PIPE, stderr=subprocess.DEVNULL,
        encoding=locale.getpreferredencoding()
    )

    总文件列表 = []
    for 路径 in sys.argv[1:]:
        if path.isfile(路径):
            if path.splitext(路径)[1].lower() in 图片后缀.splitlines() + 视频后缀.splitlines():
                总文件列表.append(路径)
            elif path.isdir(路径):
                总文件列表 += 获得文件夹内所有媒体文件(路径)

    for index, 文件 in enumerate(总文件列表):
        print(f'正在处理第 {index + 1} 个， 总共有 {len(总文件列表)} 个')
        print(f' 文件路径: {文件}')
        处理文件(process, 文件)

    process.communicate('-stay_open\nFalse\n')
    input('处理结束， 回车结束\n')

if __name__ == '__main__':
    main()

```