

The background of the entire page is a dark, high-contrast image of musical notation. It features multiple staves with various notes, rests, and clefs, all rendered in a light gray or white color against a black background. The notation is slightly blurred and has a grainy texture, giving it an artistic, almost ethereal feel. The staves are arranged in a slightly overlapping, diagonal pattern across the page.

# Ordea música

Documentación

Por Delio Docampo Cordeiro

Versión: 20151215



# Índice

Ordea música Documentación.....	1
Objetivo.....	6
Uso.....	6
Exemplo.....	6
Algoritmo.....	7
sort_int_numbers().....	7
older_mp3().....	8
copy_mp3_files().....	9
mp3_organize().....	10
music_sort().....	10



## Obxectivo

O obxectivo deste script é procesar un directorio que conteña unha discografía dun grupo, procesar as súas cancións para copialas noutro directorio cun nome que as orde cronoloxicamente por número de álbum e logo por número de canción dentro de ese álbum.

As cancións quedarán co seguinte formato:

<número\_arquivo>-<número\_pista\_do\_álbum>-<título\_canción>

## Uso

Se executa o script engadido como parámetro o directorio raíz da discografía:

```
./<nome_script>.sh <directorio_raíz_discografía>
```

Neste caso procesará as cancións do directorio pasado e subdirectorios deste, e a saída será no mesmo directorio no que se executa o script, dentro dun directorio chamado “OUT”.

De maneira opcional se poden engadir como parámetros o directorio raíz e o directorio de destino:

```
./<nome_script>.sh <directorio_raíz_discografía> <directorio_destino>
```

Procesará as cancións do directorio raíz e os seus subdirectorios, e a saída irá a parar o directorio destino.

## Exemplo

Temos o seguinte directorio dunha discografía “Banda\_piruletas” e o destino sería o de por defecto:

Orixen:

```
/Banda_piruletas
../Amorodos doces ( 2012 )
../pista1.mp3
../pista2.mp3
../pista3.mp3
../portada.jpg

../Z peta z ( maqueta 1999 )
../azucres e porcallada.mp3

../Coyacs pelados ( 2003 )
../CD1
../recheo a chicle baby.mp3
../sabor a caramelo.mp3
../etc..

../CD2
../Melóns agrios
../directo finisterre.mp3
../entrevista na radio.mp3
../etc..
```

Destino:

```
01-01-azucres e porcallada.mp3
02-01-recheo a chicle baby.mp3
02-02-sabor a caramelo.mp3
02-03-directo finisterre.mp3
02-04-entrevista na radio.mp3
03-01-pista1.mp3
03-02-pista2.mp3
03-03-pista3.mp3
```

# Algoritmo

Se non hai parámetro 1, establécese a raíz no directorio actual, se informa o usuario coa seguinte cadea: “Non se definiu o primer parámetro, se usará o actual directorio como raíz:”

Se hai parámetro 1, ROOT collará o valor do parámetro.

Se non hai parámetro 2:

Se define:

DATE\_TITLE:  
\_<ano><mes><día>\_<hora>:<minuto>:<segundo>

OUTPUT\_TITLE:  
OUTPUT<valor\_DATE\_TITLE>

DESTINY\_DIR:  
<directorio\_actual>/<valor\_OUTPUT\_TITLE>

Se crea un directorio e subdirectorios co valor de DESTINY\_DIR.

Se informa o usuario coa seguinte cadea:

*Non se definiu o segundo parámetro, se usará ./<valor\_OUTPUT\_TITLE> como directorio de saída:*  
<valor\_DESTINY\_DIR>

Se hai parámetro 2:

DESTINY\_DIR:  
*É igual ao parámetro 2.*

Se fai o directorio destino e subdirectorios definidos en DESTINY\_DIR

## sort\_int\_numbers()

---

Iniciar SORT\_NUMBERS, NUMBER\_TO\_WORK, NUMBERS

Iniciar NUMBERS co contido da matriz do parámetro 1

Iniciar NUMBERS\_TO\_WORK con <valores matriz NUMBERS>

Mentres

Iniciar C co valor 0;

Valor de C sexa menor que número elementos matriz SORT\_NUMBERS;

Incrementar C;

Mentres

Iniciar C2 co valor 0;

<valor C2> menor ca <número elementos matriz NUMBERS>;

Incrementar C2;

Se <valor de SORT\_NUMBERS[valor C]> igual a <valor NUMBERS\_TO\_WORK[valor C2]>

Engadir a matriz INDEX <valor de C2>

Definir valor de matriz NUMBERS\_TO\_WORK[<valor C2>] co valor “-1”

Sair do bucle mentres

Fin Se

Fin Mentres

Fin Mentres

## older\_mp3()

---

Iniciar PROCESS\_DIR igual a <valor parámetro 1>

Iniciar LIST\_SONGS

Iniciar TEMP\_DATE

Iniciar OLD\_DATE

Iniciar LIST\_SONGS igual a resultado de `find <valor PROCESS\_DIR> -type f -iname \*.mp3`

Mientras SONG en <contenido matriz LIST\_SONGS>

    Iniciar TEMP\_DATE con resultado de `mediainfo <valor SONG> 2> /dev/null | grep date | grep [0-9]{4} -Po | uniq`

    Se <valor OLD\_DATE> é null

        Se <valor TEMP\_DATE> é null

            Iniciar TEMP\_DATE co resultado de `echo <valor de PROCESS\_DIR> | grep [0-9]{4} -Po | uniq`

        Fin Se

        Iniciar OLD\_DATE con <valor TEMP\_DATE>

    Fin Se

Se <valor TEMP\_DATE> non é null

    Se <valor TEMP\_DATE> menor ca <valor OLD\_DATE>

        Iniciar OLD\_DATE con <valor TEMP\_DATE>

    Fin Se

Fin Se

Fin Mientras

Se <valor OLD\_DATE> non é NULL

    Amosar: <valor de OLD\_DATE> | grep [0-9]{4} -Po | uniq

Se é NULL

    Amosar cadea NULL

Fin Se

---

## copy\_mp3\_files()

---

Inicializar NDIR co contido da matriz DIR\_PATH\_LOCAL

```

Mentres C_DIR = 0 ; valor de C_DIR menor ca valor NDIRS , incrementar C_DIR
eliminar Z , DESTINY_FILES, FILE_PATH_OLD DIR_NUM_RAW
inicializar DIR_NUM_RAW = ( <valor_d_dir> + 1 )
Mentres
  ZC_DIR = 0;
  <valor de ZC_DIR> menor ca ( <número elementos matriz NDIRS> - <número elementos DIR_NUM_RAW>);
  incrementar ZC_DIR;
  inicializar Z igual a "<valor Z>0"
fin Mentres
Se <número de elementos da matriz NDIRS> menos <número de elementos da matriz DIR_NUM_RAW> é igual a 0 e o
valor de DIR_NUM_RAW é menor ca 10
  inicializar Z igual a 0

inicializar DIR_NUM igual a "<valor Z>+<valor DIR_NUM_RAW>"

Se <valor C_DIR> menor ca <valor NDIRS>
  Inicializar DIR igual a <valor DIR_PATH[<valor C_DIR>]>
  Inicializar LIST_FILES_RAW = `ls "<valor DIR>"/*.{mM}[pP]3`
  eliminar LIST_FILES

  Mentres FILE en <valores matriz de LIST_FILES_RAW>
    Engadir a matriz FILE_PATH_OLD valor de FILE
    Inicializar FILE_TITLE con `basename <valor FILE>`
    Definir FILE_TITLE con `echo <valor FILE_TITLE> | awk 'BEGIN { FS = "[0-9._-]*" } { print $2 }`
    Se <valor FILE_TITLE> é igual a NULL
      Inicializar FILE_TITLE con `basename <valor FILE>`
    Engadir a LIST_FILES <valor FILE_TITLE>
  Fin Mentres
  eliminar FILE , CF , ZF
  Iniciar NFILES co contido da matriz LIST_FILES
  Mentres
    CF = 0;
    CF menor que número elementos de LIST_FILES;
    Incrementar CF;
    Iniciar FILE_TITLE con <valor matriz LIST_FILES[<valor CF>]>
    eliminar Z

    Mentres
      Iniciar ZC a 0;
      <valor ZC> mentres sexa menor a (<número elemento matriz NFILES>-<número elementos matriz CF>);
      Incrementar ZC;
      Iniciar Z con <valor Z>0
    Fin Mentres

    Iniciar TEMP_FILE_NUM con <valor Z>(valor CF + 1)
    Se <número elementos matriz TEMP_FILE_NUM> maior <número elementos matriz NFILES>
      Inicializar TEMP_FILE_NUM con (<valor CF>+1)
      Inicializar FILE_NUM con <valor TEMP_FILE_NUM>
      Inicializar NEW_FILE_TITLE con (<valor DIR_NUM>-<valor FILE_NUM>-<valor FILE_TITLE>)
      Engadir a matriz DESTINY_FILES ("<valor DESTINY_DIR>/<valor NEW_FILE_TITLE>")
    Fin Mentres

  Eliminar FILE_TITLE, CF
  Mentres
    CF igual a 0;
    <valor CF> menor ca <número elementos FILE_PATH_OLD>;
    Incrementar CF;
    cp -v "<valor de FILE_PATH_OLD[<valor CF>]>" "<valor DESTINY_FILES[<valor CF>]>"
  Fin Mentres
Fin Mentres

```

---



## mp3\_\_organize()

---

Inicializar localmente DIR\_PATH\_LOCAL, DIR\_YEAR\_LOCAL, YEAR, ORDER

Inicializar DIR\_YEAR\_LOCAL co contido do array DIR\_YEAR

Inicializar DIR\_PATH\_LOCAL co contido do array DIR\_PATH

Inicializar ORDER co valor “chronologically”

mentres YEAR en DIR\_YEAR\_LOCAL facer:

    Se, <valor\_year> é igual a “NULL”:  
        definir ORDER coa cadea “alphabetical”

Se non está definido o valor de ORDER:

    definir ORDER coa cadea “alphabetical”

*# Nos aseguramos de que ORDER en caso de que falte un dos valores de YEAR, sexa por orde alfabético para evitar fallos.*

Se ORDER non é igual a “NULL”:

    Se ORDER é igual a “chronologically”:  
        Chamar a sort\_int\_numbers() co parámetro DIR\_YEAR\_LOCAL  
        Inializar COUNT = 0  
        mentres C\_YEAR en INDEX:  
            definir DIR\_YEAR[<valor COUNT>] con <valor DIR\_YEAR\_LOCAL[<valor  
            C\_YEAR>]  
            definir DIR\_PATH[<valor COUNT>] con <valor DIR\_PATH\_LOCAL[<valor  
            C\_YEAR>]  
            incrementar COUNT

    chamar a copy\_mp3\_files

Inicializar DIRECTORIES con:

    Resultado de buscar con “find” tódolos arquivos .mp3 que estén aloxados baixo a raíz de <ROOT> , filtrando o seu resultado con “uniq” para evitar duplicidades, e logo filtrando a saída con “sort”

borrar COUNT

Chamar copy\_mp3\_files()

---

## music\_sort()

---

Inicializar COUNT = 0

Inicializar DIRECTORIES con:

    Resultado de buscar con “find” tódolos arquivos .mp3 que estén aloxados baixo a raíz de <ROOT> , filtrando o seu resultado con “uniq” para evitar duplicidades, e logo filtrando a saída con “sort”

mentres DIR en DIRECTORIES facer:

    Inicializar DIR\_PATH[<valor\_count>] = <valor DIR>  
    Inicializar DIR\_YEAR[<valor\_count>] = (resultado función older\_mp3 <valor DIR> >)  
    Incrementar COUNT

borrar COUNT

Chamar mp3\_\_organize()

---

chamar a music\_sort