

procesa descargas

objetivo

crear un método para gestionar la descarga de los ficheros **.torrent**, desde el fichero .torrent hasta el fichero final, automatizando en todo lo posible el proceso

para ello nos apoyamos en una serie de herramientas:

1. lector de RSS
2. sites de películas/series/etc que usen este sistema (RSS)
3. un servidor externo corriendo rTorrent
4. una raspberry ejecutando tareas de sincronización de directorios

método

1. esta parte indica la URL del site (conseguida a través del lector RSS) donde encontraremos el fichero .torrent (no el .torrent directamente) para descargarlo y proceder a su subida en el servidor externo con rTorrent, donde se procederá a su descarga
2. la parte de sincronización entre el servidor y la raspberry (mediante rsync)
3. el proceso de los ficheros descargados ya en local (desde la raspberry) y su «normalización» (incluso extracción si están comprimidos)

1

[newpct+down+scp+pass+kde.sh](#)

```
#!/bin/bash

# versiones
# PTE cambiar variable URL_TMP por URL2 directamente
#
# 5.8 informar solo en casode error del log de intento de conexión, si no,
# dar OK en popup emergente
# 5.7 modificación para nuevas URLs de tumejortorrent.com
# han modificado la ubicación de la URL (ahora está en una función llamada
# OpenTorrent), pero la detección sigue funcionando igual
# OK en sentencias IF, hacer que reconozca el patrón
# http://*<loquesea>/<loquesea>/*, las pruebas realizadas no reconocen el primer
# * > ^http://.*<loquesea>/<loquesea>/*
# 5.6 añadido soporte para las URLs de torrentrapid.com
# torrentrapid.com = torrentlocura.com = tumejortorrent.com ==>
# tumejorjuego.com
# 5.5 añadido soporte para las URLs de torrentlocura.com
# 5.4.1 realizadas modificaciones para reconocer contraseñas del site
# "torrentlocura.com".
# > No funciona pq la gente de "tumejortorrent.com" 'sablean' a los de
# "torrentlocura.com", de manera que en el primer site no aparece la contraseña
# de descarga.
```

```
# 5.4 realizadas modificaciones para aceptar la URL a través de parámetro
# además de por el diálogo
#
# 5.3 cambio de ficheros temporales a variables en memoria para procesamiento
# de URL
# 5.2 modificación para nuevas URL de tumejortorrent
# 5.1 añadida opción de ACTIVA_GUI, para desactivar si necesario las salidas
# por KDE
# 5.0 integración con KDE
# las contraseñas se guardan en fichero dentro del DIR_PASS
# 4.2 recuperación de contraseña
# versiones anteriores sin comentarios

# retales, mate, 03/10/2016
#
https://techbase.kde.org/Development/Tutorials/Shell\_Scripting\_with\_KDE\_Dialogs
#
http://www.linux-magazine.com/var/linux\_magazin/storage/images/linux-magazine.com/issues/2009/99/shell-click/table2.png/428127-1-eng-US/table2.png\_reference.png
# https://mostlylinux.wordpress.com/bashscripting/kdialog/
# https://forum.kde.org/viewtopic.php?f=66&t=135037 (ejemplo con fichero de
# apoyo para la generación de la ventana en KDE)

# pruebas de conversación entre KDE y BASH
# kdialog --password "Please enter the server access code:" 2>/dev/null

#
# FUNCIONES
#
function aleatorio {

    if [ "$NOMBRE" == "" ]; then
        NOMBRE=$(apg -a 0 -n 1 -q -MC -m 8);
    fi
}

ACTIVA_GUI=1 # puesto a 0 no muestra salidas información por GUI
escritorio
SALIDA_BUCLE=0 # bucle general (salida en modo CLI)
CODIGO_SALIDA=0 # establecido a 0 para salida OK, a 1 para salida KO
PARAMETROS=$# # recogida de parámetros de entrada
MOSTRAR_LOG=0 # fuerza mostrar en GUI el log de acciones del script aunque
ACTIVA_GUI=0
CODIGOS_ERROR=0 # recoge los códigos de retorno de WGET y SCP para
determinar si se ha realizado con éxito la transferencia,
# y evitar que muestre el LOG si no es necesario

DIR_TORRENT="/media/mate/WD2-3TB-EXT4/__VIDEO/4.torrent/"
SSH_KEY="/home/mate/.ssh/user@seedbox.torresegea.es"
REMOTE_LOGIN="user@seedbox.torresegea.es"
REMOTE_DIR_TORRENT="/home/user/torrent"
DIR_PASS="/media/mate/WD2-3TB-EXT4/__VIDEO/0.PROCESAR/"

APP_NAME="Torrent Downloader v5.8"
```

```
export DISPLAY=:0.0

while [ $SALIDA_BUCLE -eq 0 ] ; do

    if [[ $PARAMETROS -ge 1 && ! -z "$1" ]]; then

        URL=$1
        SALIDA_BUCLE=1
        ACTIVA_GUI=0
        MOSTRAR_LOG=1

    else

        if [ $ACTIVA_GUI -eq 1 ]; then
            sleep .3
            URL=$(kdialog --title "$APP_NAME" --inputbox "Introduce la URL"
2>/dev/null)
        else
            read -p "$APP_NAME (vacío acaba)> " URL
        fi

        if [ $? != 0 ]; then
            echo "saliendo...."
            exit
        fi

    fi

    if [ "$URL" != "" ]; then
        # extrae el nombre del futuro archivo .torrent
        NOMBRE=$(echo ${URL//\\/_} | cut -d "_" -f 5-8)
        aleatorio # si no, nos inventamos uno

        wget -O /tmp/url.tmp $URL > /tmp/text.tmp 2>&1

        # causística para tumejortorrent.com
        if [[ "$URL" =~ ^http://.*tumejortorrent.com* ]]; then

            COUNTER_LOOP=3
            while [ $COUNTER_LOOP -ge 0 ]; do
                case $COUNTER_LOOP in
                    0) # v1 URL
                        #URL_TMP=$(grep -o
'http://tumejorjuego.com/redirect/index.php?link=descargar-torrent/.*/'
/tmp/url.tmp | tee /tmp/url2.tmp)
                        URL_TMP=$(grep -o
'http://tumejorjuego.com/redirect/index.php?link=descargar-torrent/.*/'
/tmp/url.tmp)
                        [ -n "$URL_TMP" ] && URL_TMP=${URL_TMP::-1} #
eliminamos último carácter ("), si hay algo que quitar
                        ;;
                    1) # v2 URL
                        #URL_TMP=$(grep -o
```

```
'http://tumejorjuego.com/download/index.php?link=descargar-torrent/*.html'
/tmp/url.tmp | tee /tmp/url2.tmp)
    URL_TMP=$(grep -o
'http://tumejorjuego.com/download/index.php?link=descargar-torrent/*.html'
/tmp/url.tmp)

;;
2)    # v3 URL
    #URL_TMP=$(grep -o
'http://www.tumejortorrent.com/descargar-torrent./.*'" /tmp/url.tmp | tee
/tmp/url2.tmp)
    URL_TMP=$(grep -o
'http://www.tumejortorrent.com/descargar-torrent./.*'" /tmp/url.tmp)
    [ -n "$URL_TMP" ] && URL_TMP=${URL_TMP::-1}    #
eliminamos último carácter ("), si hay algo que quitar
;;
3)    # v4 URL
    URL_TMP=$(grep -o
'http://tumejorjuego.com/download/tumejortorrent.php?link=descargar-torrent./.*'
/' /tmp/url.tmp)
;;
esac

[ -z "$URL_TMP" ] && COUNTER_LOOP=$((COUNTER_LOOP-1)) ||
COUNTER_LOOP=-1

done

fi

# causística para newpct.com
if [[ "$URL" =~ ^http://www.newpct.com* ]]; then

    #URL_TMP=$(grep -o
'http://tumejorserie.com/descargar/index.php?link=torrents/.....torrent'
/tmp/url.tmp | tee /tmp/url2.tmp)
    URL_TMP=$(grep -o
'http://tumejorserie.com/descargar/index.php?link=torrents/.....torrent'
/tmp/url.tmp)

fi

# causística para torrentlocura.com
if [[ "$URL" =~ ^http://torrentlocura.com* ]]; then

    URL_TMP=$(grep -o
'http://tumejorjuego.com/redirect/torrentlocura.php?link=descargar-torrent./.*'
'" /tmp/url.tmp)
    [ -n "$URL_TMP" ] && URL_TMP=${URL_TMP::-1}    # eliminamos
último carácter ("), si hay algo que quitar
fi

# causística para torrentrapid.com
if [[ "$URL" =~ ^http://torrentrapid.com* ]]; then

    URL_TMP=$(grep -o
'http://tumejorjuego.com/redirect/torrentrapid.php?link=descargar-torrent./.*'
```

```
' /tmp/url.tmp)
    [ -n "$URL_TMP" ] && URL_TMP=${URL_TMP::-1}      # eliminamos
último carácter ("), si hay algo que quitar
fi

# procesa url
#URL2=$(cat /tmp/url2.tmp)
URL2=$URL_TMP

# descarga el torrent
wget $URL2 -O $DIR_TORRENT$NOMBRE.torrent >> /tmp/text.tmp 2>&1
CODIGOS_ERROR_TEMP=$?
CODIGOS_ERROR=$((CODIGOS_ERROR+CODIGOS_ERROR_TEMP))

# subir el torrent al servidor de descargas
scp -i $SSH_KEY $DIR_TORRENT$NOMBRE.torrent
$REMOTE_LOGIN:$REMOTE_DIR_TORRENT >> /tmp/text.tmp 2>&1
CODIGOS_ERROR_TEMP=$?
CODIGOS_ERROR=$((CODIGOS_ERROR+CODIGOS_ERROR_TEMP))

# mira si tiene contraseña
if [[ "$URL" =~ ^http://*tunejortorrent.com* ]] || [[ "$URL" =~
^http://torrentlocura.com* ]] || [[ "$URL" =~ ^http://torrentrapid.com* ]] ;
then
    grep -o '<input type="text" id="txt_password" name="txt_password"
value=".*"' /tmp/url.tmp > /tmp/pass.tmp
    PASS=$(cat /tmp/pass.tmp | tr -d "\"" | awk '{print $5}' | tr -s
'=' ' ' | awk '{print $2}')

    if [[ "$PASS" == "" && $CODIGOS_ERROR -eq 0 ]]; then
        echo "Descarga sin contraseña"
        if [ $ACTIVA_GUI -eq 1 ]; then
            kdialog --title "$APP_NAME" --passivepopup "Descarga sin
contraseña" 5
        fi
    fi
    if [[ "$PASS" != "" && $CODIGOS_ERROR -eq 0 ]]; then
        echo "DESCARGA CON CONTRASEÑA, ANÓTALA: $PASS"
        if [ $ACTIVA_GUI -eq 1 ]; then
            kdialog --title "$APP_NAME" --passivepopup "Esta descarga
contenía contraseña: $PASS" 10
        fi
        touch "$DIR_PASS$NOMBRE = $PASS"
    fi
fi

unset NOMBRE
unset URL2
unset PASS
unset CODIGOS_ERROR_TEMP

if [[ $ACTIVA_GUI -eq 1 || $MOSTRAR_LOG -eq 1 ]]; then
    if [[ $CODIGOS_ERROR -ne 0 || $MOSTRAR_LOG -eq 1 ]]; then
        kdialog --error "error subiendo .torrent"
```

```
        kdialog --title "$APP_NAME" --textbox /tmp/text.tmp 1024 636 &
    else
        kdialog --title "$APP_NAME" --passivepopup ".torrent subido
correctamente" 5
    fi
fi

CODIGOS_ERROR=0

else
    CODIGO_SALIDA=1
    if [ $ACTIVA_GUI -eq 1 ]; then
        kdialog --title "$APP_NAME" --sorry "la URL está vacía"
2>/dev/null
    fi
fi

done

exit $CODIGO_SALIDA
```

2

3

procesar.sh

```
#!/bin/bash -x

#
# VARIABLES GLOBALES
#
DIR_ORIGEN="/media/mate/WD2-3TB-EXT4/__VIDEO/0.PROCESAR"
DIR_DESTINO="/media/mate/WD2-3TB-EXT4/__VIDEO/1.VISUALIZAR"
FICHERO_IGNORE=".ignore"
UNRAR="/usr/bin/unrar"
ESRAR=""

#
# FUNCIONES
#
function mirarCarpeta() {
    CARPETA=$1
    if [ ! -e "$CARPETA/$FICHERO_IGNORE" ]; then
        mirarRAREnCarpeta "$CARPETA"
        mirarFicherosCarpeta "$CARPETA"
    # else
    #     echo "Carpeta Ignorada: ${CARPETA##*/}"
    #     echo "+++"
    fi
}

}
```

```

function mirarRAREnCarpeta() {
# prueba a descomprimir un RAR (por si no lleva contraseña)
  CARPETA=$1
  NOMBRE_CARPETA=${CARPETA##*/}

  for fichero in "$CARPETA"/* ; do
    NOMBRE_FICHERO=${fichero##*/}
    EXTENSION=${NOMBRE_FICHERO##*.*}

    if [ "$EXTENSION" == "rar" ]; then

      ESRAR="1"

      LLEVA_PASS=$(($UNRAR l "$fichero" | grep \*))
      # mirar la lista de archivos, si aparece un * es que está
      encriptado

      if [ "$LLEVA_PASS" == "" ]; then
        # no lleva contraseña

        $UNRAR e "$fichero" "$CARPETA"
        if [ "$?" == "0" ]; then
          rm "$fichero"
          rm "$CARPETA"/*.url
          rm "$CARPETA"/*.txt
        fi
      else
        touch "$CARPETA"/LLEVA_PASS!
      fi
    fi
  done
}

function mirarFicherosCarpeta {
  CONTADOR=0 # si una carpeta contiene más de un fichero de video,
añadirle un numerador para evitar perder el fichero en el MV
  CARPETA=$1
  NOMBRE_CARPETA=${CARPETA##*/}

# echo "$NOMBRE_CARPETA"

# se podría usar for fichero in "$CARPETA"/.mkv "$CARPETA"/.avi ; do ???
# habría que mirar de ordenarlos antes de renombrarlos para que conserven el
orden
# o igual dejarlos con el nombre original para hacerlo a mano

  for fichero in "$CARPETA"/* ; do
    NOMBRE_FICHERO=${fichero##*/}
    EXTENSION=${NOMBRE_FICHERO##*.*}
    if [ "$EXTENSION" == "mkv" -o "$EXTENSION" == "avi" ]; then
      ORIGEN="$fichero"
      if [ $CONTADOR -eq 0 ]; then
        DESTINO="$DIR_DESTINO/$NOMBRE_CARPETA.$EXTENSION"
      else
        DESTINO="$DIR_DESTINO/$NOMBRE_CARPETA.$CONTADOR.$EXTENSION"
      fi
    fi
  done
}

```

```
        fi
#       echo -e "Origen: $ORIGEN\n"
#       echo -e "Destino: $DESTINO\n"
#       mv "$ORIGEN" "$DESTINO"
#       borrarCarpetaVacía "$CARPETA"
#       CONTADOR=$((CONTADOR + 1))
    fi

    if [ "$ESRAR" == "0" ]; then
        rm "$CARPETA"/*.txt
    fi
#   echo "contador: $CONTADOR"
done

# echo "+++"
}

#
# MAIN
#
for carpeta in $DIR_ORIGEN/*
do
    if [ -d "$carpeta" ]; then
        ESRAR="0"
        mirarCarpeta "$carpeta"
    # else
    #   echo "no directorio: $carpeta"
    fi
done

# elimina las carpetas vacías
find "$DIR_ORIGEN" -mindepth 1 -type d -empty -delete
```

OLD

procesa.sh

```
#!/bin/bash

#
# VARIABLES GLOBALES
#
DIR_ORIGEN="/media/WD1-3TB-EXT4/__VIDEO/0.REVISAR"
DIR_DESTINO="/media/WD1-3TB-EXT4/__VIDEO/1.DOWNLOADED"
FICHERO_IGNORE=".ignore"

#
# FUNCIONES
#
```

```
function mirarCarpeta() {
    CARPETA=$1
    if [ ! -e "$CARPETA/$FICHERO_IGNORE" ]; then
        mirarFicherosCarpeta "$CARPETA"
    else
        echo "Carpeta Ignorada: ${CARPETA##*/}"
        echo "+++"
    fi
}

function mirarFicherosCarpeta {
    CONTADOR=0      # si una carpeta contiene más de un fichero de video,
    añadirle un numerador para evitar perder el fichero en el MV
    CARPETA=$1
    NOMBRE_CARPETA=${CARPETA##*/}

    echo "$NOMBRE_CARPETA"

    # se podría usar for fichero in "$CARPETA"/.mkv "$CARPETA"/.avi ; do ???
    for fichero in "$CARPETA"/* ; do
        NOMBRE_FICHERO=${fichero##*/}
        EXTENSION=${NOMBRE_FICHERO##*.*}
        if [ "$EXTENSION" == "mkv" -o "$EXTENSION" == "avi" ]; then
            ORIGEN="$fichero"
            if [ $CONTADOR -eq 0 ]; then
                DESTINO="$DIR_DESTINO/$NOMBRE_CARPETA.$EXTENSION"
            else
                DESTINO="$DIR_DESTINO/$NOMBRE_CARPETA.$CONTADOR.$EXTENSION"
            fi
            echo -e "Origen: $ORIGEN\n"
            echo -e "Destino: $DESTINO\n"
            mv "$ORIGEN" "$DESTINO"
        fi
        # borrarCarpetaVacía "$CARPETA"
        CONTADOR=$((CONTADOR + 1))
        echo "contador: $CONTADOR"
    done

    echo "+++"
}

function borrarCarpetaVacía {
    # esta función no se ejecuta
    # no he encontrado una manera de comprobar si un directorio está vacío
    # se utiliza un FIND al final del bucle de carpetas para eliminar las vacías

    CARPETA=$1

    if [ -s "$CARPETA" ]; then
        echo "existe y no está vacío"
    else
        echo "no existe o está vacío?"
    fi
}
```

```
#  
# MAIN  
#  
for carpeta in $DIR_ORIGEN/*  
do  
    if [ -d "$carpeta" ]; then  
        mirarCarpeta "$carpeta"  
    else  
        echo "no directorio: $carpeta"  
    fi  
done  
  
# elimina las carpetas vacías  
find "$DIR_ORIGEN" -mindepth 1 -type d -empty -delete
```

From:
<https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link:
<https://miguelangel.torresegea.es/wiki/misproyectos:procesadescargas?rev=1512692792>

Last update: **07/12/2017 16:26**

