

```

#!/bin/bash
. digit03ltp.sh
#-----
create_sip()
{
    local sip=$SIP_PATH/$(encode_sip $candidate)

    #create symlink if not already exists
    if ! test -L "$sip"
    then
        ln -fs "$candidate" "$sip"
        echo "$(date +%b %e %H:%M') [find candidate] $sip from ${candidate##*/}"
    fi
}
export -f create_sip
#-----
treat_candidate()
{
    local candidate=$1

    if ! test -d "$candidate"
    then
        echo "[error] candidate '$candidate' does not exist anymore!" >&2
    elif ! test -f "$candidate.xml"
    then
        #new sip
        create_sip
    elif test "$candidate.xml" -ot "$candidate/"
    then
        #modified sip
        rm -v $candidate.xml
        create_sip
    fi
}
export -f treat_candidate
#-----
treat_dataset()
{
    local dataset ndays
    IFS=';' read dataset ndays <<< $@
    local opt=""
    find ${dataset%/*} -mindepth 1 -maxdepth 1 -name "${dataset##*/}" $opt -type d \
        | parallel --will-cite -rj1 treat_candidate
}
export -f treat_dataset
#-----
treat_new_zips()
{
    cd $MAIN_PATH
    grep ^/ $DATASET_FILE \
        | parallel --will-cite -rj1 treat_dataset \
        | nl
}
export -f treat_new_zips
#-----
treat_candidate_zip()
{
    echo "in treat_candidate_zip"
    #zip directories of NAS
    local candidate=$1
    if ! test -d "$candidate"
    then
        echo "[error] candidate '$candidate' does not exist anymore!" >&2
    else
        echo "$(date +%b %e %H:%M') zipping $candidate"

        #zip variables
    fi
}

```

```

oldstr=" "
newstr=" "
candidate2=$(echo $candidate | sed "s/$oldstr/$newstr/g")

strtoadd=""
strtoremove=$PATH_TO_DATA_IN_MNT
prezip="${candidate2/$strtoremove/$strtoadd}.zip"
#print "prezip-----$prezip"
elems=$(echo $prezip | tr "/" "\n")
#IFS='/' read -r -a array <<< "$prezip"

collection=""
counter=0
sizearray=${#elems[@]}
local prezip_path_to_zip=""
local prezip_zip_only=""

for elem in $elems
do
    prezip_zip_only="${elem}"
    ((counter++))
    prezip_path_to_zip+="/${elem}"
done
prezip_path_to_zip="${prezip_path_to_zip:1}"

strtoadd=""
strtoremove=$prezip_zip_only
#export prezip_zip_only
prezip_path_to_zip="${prezip_path_to_zip/$strtoremove/$strtoadd}"

date1="$(date +%d-%m-%Y %H:%M)"
strtoadd="-"
strtoremove=""
date2="${date1}/${strtoremove/$strtoadd}"
strtoadd="h"
strtoremove=":"
date="${date2/$strtoremove/$strtoadd}"

strtoadd="-"
strtoremove="/"
method1="${prezip_path_to_zip}/${strtoremove/$strtoadd}"
strtoadd=""
strtoremove="mnt-NAS-"
method="${method1/$strtoremove/$strtoadd}"
method="${method::-1}"
#echo "method1=$method"
text=$prezip_path_to_zip
search="."
prefix=${text%%$search*}
pospoint=${#prefix}
pospoint1=$((pospoint-4))
method=${prezip_path_to_zip:pospoint1:4}
#echo "method2=$method"
strtoadd=""
strtoremove=".zip"
zipname="${prezip_zip_only/$strtoremove/$strtoadd}-$date-$method.zip"
#zipname="${prezip_zip_only/$strtoremove/$strtoadd}-$date.zip"

#print "zipname=====zipname"
#print "prezip_path_to_zip=====prezip_path_to_zip"
path_to_create="$ORIGINALDATA_PATH/${prezip_zip_only/$strtoremove/
$strtoadd}"

if [ ! -d "$path_to_create/" ]; then
    mkdir -pv $path_to_create

```

```

        cd $candidate2
        ls -lR > $path_to_create/content.txt
        #echo "-----$path_to_create/$zipname.tmp"
        #create zip and check zip integrity
        #print "/usr/bin/time -f '%e' -o $UPLOAD_PATH/
$zipname.prezip.create.time zip -rv0 '$path_to_create/$zipname.tmp' $candidate/
$path_to_create/content.txt &>$UPLOAD_PATH/$zipname.prezip.create.log"
        if /usr/bin/time -f "%e" -o $UPLOAD_PATH/
$zipname.prezip.create.time zip -rv0 "$path_to_create/$zipname.tmp" "$candidate/"
$path_to_create/content.txt &>$UPLOAD_PATH/$zipname.prezip.create.log
        then
            local size=$(stat -c "%s" "$path_to_create/$zipname.tmp")
            local duration=$(<$UPLOAD_PATH/
$zipname.prezip.create.time)
            mv -f $path_to_create/$zipname.tmp $path_to_create/
$zipname

            rm $path_to_create/content.txt
            echo "$(date +%b %e %H:%M') [create prezip] ... $
(get_speed $size $duration)"
            treat_new_zips
        else
            echo "[error] zip '${candidate/$strtoremove/$strtoadd}/
$prezip' failed [ $? ]; see log '$prezip.prezip.create.log'." >&2
        fi
    fi
    treat_new_zips
}
export -f treat_candidate_zip
#-----
treat_dataset_zip()
{
    local dataset ndays
    IFS=';' read dataset ndays <<< @$@
    local opt=""

    #find /media/jmherpers/Elements/ltp_storage/PHOTOGRAPHY/MACROSCOPY/STACKING/
PH31.Macro_Stacking -mindepth 1 -maxdepth 1 -name "BE_RBINS-PAL-a 0086*" $opt -type d \
    #| parallel --will-cite -rj1 treat_candidate_zip

    find ${dataset%/*} -mindepth 1 -maxdepth 1 -name "${dataset##*/}" $opt -type d \
    | parallel --will-cite -rj1 treat_candidate_zip
    #find /media/jmherpers/BE-RBINS-BACKUP-006/COMPUTED_TOMOGRAPHY/MICROCT/MERCO-
XRE_TESCAN_UNITOM/CT03.TOMOGRAPHY -mindepth 1 -maxdepth 1 -name "BE-RBINS-INV-TYPES-
MT-34*" $opt # -type d \

}
export -f treat_dataset_zip
#-----
#main
echo "$(date +%b %e %H:%M') Start treatment"

#rm -R $ORIGINALDATA_PATH/*
wait
rm -R $SIP_PATH/*
wait
#rm -R $XML_PATH/*
#wait
rm -R $UPLOAD_PATH/*
wait
#rm -R $INGESTED_PATH/*
#wait
rm -R $DOWNLOAD_PATH/*
wait
#rm -R $CSVPLONE_PATH/*

```

```
#wait
rm -R $OUT_PYTHON_PATH/*
wait

create_lock

grep ^/ $DATASET_PREZIP_FILE \
| parallel --will-cite -rj1 treat_dataset_zip \
| nl

#treat_new_zips

#next treatment: create packet (couple xml+zip)

./create_packets.sh &
#nohup ./create_packets.sh &
exit 0
#test without parallel
#while IFS=';' read dataset ndays
#do
#   opt=${ndays:+"-mtime +$ndays"}
#   while read candidate
#   do
#       ...
#   done <<(find ${dataset%/*} -mindepth 1 -maxdepth 1 -name "${dataset##*/}" $opt -type
d)
#done <<(grep ^/ $DATASET_FILE)
```