

```
#!/bin/bash
#
# Visualization of occupied space by directories and specifications

#####
# Command error message
# Outputs:
#   Output to STDERR
#####
usage() { echo "Usage: $0 [-d <date>] [-s <size>] [-l <limit>] [-r] [-a] [-n <regex>] [<directory>]" 1>&2; exit 1; }

#####
# Argument error message
# Outputs:
#   Output to STDERR
#####
argError() { echo "ERROR: \"$1\" arg is invalid." 1>&2; exit 1; }

#####
# Invalid directory error
# Outputs:
#   Output to STDERR
#####
invalidDirectory() { echo "ERROR: \"$1\" directory is invalid." 1>&2; exit 1; }

#####
# Defaults
#####
directories="."
sort_option="-k1,1nr" # default sort
size=0
name_exp=".*"
date_ref=$(LC_TIME=en_US.utf8 date "+%Y-%m-%d %H:%M:%S")

header="$*"
#####

# getopt to parse command-line options
while getopt "d:s:l:ran::" opt 2>/dev/null; do
  case "$opt" in
    d) # -d " " - print the current date
      [[ -z "$OPTARG" ]] && argError "-d"
      date_ref=$(LC_TIME=en_US.utf8 date -d "$OPTARG" "+%Y-%m-%d %H:%M:%S" 2>/dev/nu
11)
      [[ $? -ne 0 ]] && argError "-d"
      ;;
    s)
      [[ -z "$OPTARG" ]] && argError "-s"
      { [[ $OPTARG =~ ^[0-9]+$ ]] && [[ "$OPTARG" -ge 0 ]] ;} || argError "-l"
      size="$OPTARG"
      ;;
    l)
      [[ -z "$OPTARG" ]] && argError "-l"
      { [[ "$OPTARG" =~ ^[0-9]+$ ]] && [[ "$OPTARG" -gt 0 ]] ;} || argError "-l"
      limit_lines="$OPTARG"
      ;;
    r)
      if [[ "$sort_option" = "-k1,1nr" ]]; then
        sort_option="-k1,1n -k1,1r -k2,2r"
      else
        sort_option="-k2,2 -k1,1n -k1,1r" # -a -r
      fi
      ;;
    a)
      if [[ "$sort_option" = "-k1,1nr" ]]; then
        sort_option="-k2,2"
      else

```

```

        sort_option="-k1,1n -k1,1r -k2,2" # -r -a
    fi
    ;;
n)
    name_exp="$OPTARG"
    ;;
*) usage ;;
esac
done

# removes options and their associated values,
# allows your script to access and process the non-option arguments
shift $((OPTIND-1))

# LC_ALL=EN_us.utf8 for uniform date

if [[ $# -ge 1 ]]
then
    directories=()
    for dir in "$@"; do
        [[ -d "$dir" ]] || invalidDirectory "$dir" # check: valid directory
        [[ "$dir" = "/" ]] || [[ "$dir" = "//" ]] || dir=$(echo "$dir" | sed 's:/*$::')
        directories+=("$dir")
    done
fi

# all folders in directory
folders=$(find "${directories[@]}" -type d 2>/dev/null | sort -u)

if [[ -z "$limit_lines" ]]; then
    limit_lines=$(echo "$folders" | wc -l)
fi

# header
echo "SIZE NAME $(LC_TIME=en_US.utf8 date "+%Y%m%d") $header"

while IFS= read -r f; do

    if [[ -x "$f" ]] && [[ -r "$f" ]]; then
        bytes=$(find "$f" -not -newermt "$date_ref" -type f \( -regex "$name_exp" -o -name "$name_exp" \) -not -size "$size"c -exec du -bc {} + 2>/dev/null | tail -n 1 | awk '{print $1}')
        [[ -z "$bytes" ]] && bytes=0
        echo "$bytes" "$f"
    else
        echo "NA" "$f"
    fi
done <<< "$folders" | sort $sort_option | head -n "$limit_lines"

```