

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Scripts archiveren.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #####
8: PROGRAM_PATH=$(realpath "$(dirname "$0")")
9: source "$PROGRAM_PATH"/kz-common.sh
10: PROGRAM_NAME=kz-archive
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2013
13:
14: VERSION_NUMBER=18.03.00
15: VERSION_DATE=2021-10-08
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $DISPLAY_NAME $OPTIONS_USAGE_COMMON
26: [SELECTIE...]"
27: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...] [SELECTIE...]"
28:
29: Scripts archiveren.
30:
31: Opties:
32: $OPTIONS_HELP_COMMON
33:
34: Argumenten:
35: SELECTIE zoek naar bestanden overeenkomend met opgegeven SELECTIES"
36:
37: readonly ARCHIVE_FROM=$HOME/kz-uploads/dist/data/linux
38: readonly ARCHIVE_TO=$HOME/kz-uploads/app/data/linux-archieff
39: readonly DELETE_FROM_1=$HOME/kz-scripts
40: readonly DELETE_FROM_2=$HOME/kz-docs
41: readonly DELETE_FROM_3=$HOME/kz-deb
42:
43:
44: #####
45: # Global variables
46: #####
47:
48: declare ARCHIVED=false
49: declare ARGUMENT_SELECTION=false
50: declare -a SELECTION_ARGUMENT=()
51: declare DELETED=false
52:
53:
54: #####
55: # Functions
56: #####
57:
58: check_input() {
59:     local -i getopt_rc=0
60:     local -i search4_arg_num=0
61:     local dir=''
62:     local parsed=''
63:
64:     parsed=$(
65:         getopt --alternative \
66:             --options "$OPTIONS_SHORT" \
67:             --longoptions "$OPTIONS_LONG" \
68:             --name "$DISPLAY_NAME" \
```

```
69:         -- "$@"
70:     ) || getopt_rc=$?
71:     if [[ $getopt_rc -ne $SUCCESS ]]; then
72:         printf '%s\n' "$USAGELINE" >&2
73:         exit $ERROR
74:     fi
75:     eval set -- "$parsed"
76:     process_common_options "$@"
77:
78:     while true; do
79:         case $1 in
80:             --)
81:                 shift
82:                 break
83:                 ;;
84:             *)
85:                 shift
86:                 ;;
87:         esac
88:     done
89:
90:     while [[ "$*" ]]; do
91:         ARGUMENT_SELECTION=true
92:         SELECTION_ARGUMENT[$search4_arg_num]=$ (basename "$1")
93:         ((+search4_arg_num))
94:         shift
95:     done
96:
97:     # Een non-gui script gestart met optie gui.
98:     if $OPTION_GUI; then
99:         OPTION_GUI=false
100:        DESKTOP_TERMINAL=false
101:    fi
102:
103:    check_user
104:
105:    for dir in "$ARCHIVE_FROM" \
106:              "$ARCHIVE_TO" \
107:              "$DELETE_FROM_1" \
108:              "$DELETE_FROM_2" \
109:              "$DELETE_FROM_3"; do
110:        if ! [[ -d $dir ]]; then
111:            error "$DISPLAY_NAME: map '$dir' bestaat niet"
112:            exit $ERROR
113:        fi
114:    done
115:
116:    request_input
117: }
118:
119:
120: request_input() {
121:     if $ARGUMENT_SELECTION; then
122:         return $SUCCESS
123:     fi
124:     read -rp 'Wat archiveren/verwijderen? [leeg=alles]: '
125:     SELECTION_ARGUMENT[0]=$REPLY
126: }
127:
128:
129: process_input() {
130:     local basename=''
131:     local dirname=''
132:     local files_found=false
133:     local search4=''
134:
135:     for search4 in "${SELECTION_ARGUMENT[@]}; do
136:         while read -r file; do
```

```
137:
138:     files_found=true
139:
140:     dirname=${dirname "$file"}
141:     basename=${basename "$file"}
142:
143:     if [[ $dirname = $ARCHIVE_FROM* ]]; then
144:         process_archive
145:     else
146:         process_delete
147:     fi
148: done <<(
149:     find "$ARCHIVE_FROM" \
150:         "$DELETE_FROM_1" \
151:         "$DELETE_FROM_2" \
152:         "$DELETE_FROM_3" \
153:         -iname \
154:         '*"$search4"*' \
155:         -not -path '*/.git/*' \
156:         -type f \
157:         -print |
158:         sort
159: )
160: if ! $files_found; then
161:     warning "\nGeen bestanden gevonden met '$search4' in de naam."
162: fi
163: files_found=false
164: done
165:
166: }
167:
168:
169: process_archive() {
170:     printf "\n${BOLD}%s\n${NORMAL}\n" "$basename" "$dirname"
171:     while true; do
172:         # < /dev/tty want FD 1 al in gebruik door for-while constructie.
173:         read -rp 'archiveren/Overslaan/stoppen? [a/O/s]: ' < /dev/tty
174:         case $REPLY in
175:             a*|A*)
176:                 archive_file
177:                 break
178:                 ;;
179:             o*|O*|'')
180:                 printf '%s\n' 'Overgeslagen.'
181:                 break
182:                 ;;
183:             s*|S*)
184:                 exit $SUCCESS
185:                 ;;
186:             *)
187:                 continue
188:                 ;;
189:         esac
190:     done
191: }
192:
193:
194: archive_file() {
195:     local olddir=''
196:     local newdir=''
197:
198:     olddir=$dirname
199:     newdir=${olddir//dist/app}
200:     newdir=${newdir//linux/linux-archieff}
201:     mv "$olddir/$basename" "$newdir" |& $LOGCMD
202:     printf '%s\n' 'Gearchiveerd.'
203:     ARCHIVED=true
204: }
```

```
205:
206:
207: process_delete() {
208:     printf "\n${BOLD}%s\n%s${NORMAL}\n" "$basename" "$dirname"
209:     while true; do
210:         # < /dev/tty want FD 1 al in gebruik door for-while constructie.
211:         read -rp 'verwijderen/Overslaan/stoppen? [v/O/s]: ' < /dev/tty
212:         case $REPLY in
213:             v*|V*)
214:                 delete_file
215:                 break
216:                 ;;
217:             o*|O*|'')
218:                 printf '%s\n' 'Overgeslagen.'
219:                 break
220:                 ;;
221:             s*|S*)
222:                 exit $SUCCESS
223:                 ;;
224:             *)
225:                 continue
226:                 ;;
227:         esac
228:     done
229: }
230:
231:
232: delete_file() {
233:     rm "$file" |& $LOGCMD
234:     printf '%s\n' 'Verwijderd.'
235:     DELETED=true
236: }
237:
238:
239: term_script() {
240:     if $DELETED; then
241:         TEXT='\nEr zijn bestanden verwijderd.'
242:     else
243:         TEXT='\nEr zijn geen bestanden verwijderd.'
244:     fi
245:     if $ARCHIVED; then
246:         TEXT="$TEXT\nEr zijn bestanden gearchiveerd."
247:     else
248:         TEXT="$TEXT\nEr zijn geen bestanden gearchiveerd."
249:     fi
250:     info "$TEXT"
251:     if $DELETED || $ARCHIVED; then
252:         warning 'Pas website aan en commit de wijzigingen!'
253:     fi
254:     exit $SUCCESS
255: }
256:
257:
258: #####
259: # Main line
260: #####
261:
262: main() {
263:     init_script "$@"
264:     check_input "$@"
265:     process_input
266:     term_script
267: }
268:
269:
270: main "$@"
271:
272:
```

273: # EOF