

```
1: #!/bin/bash
2: # shellcheck source=kzcommon.sh
3: # -----
4: # Scripts archiveren.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #
8: # Auteursrecht (c) 2013-2021 Karel Zimmer.
9: # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>.
10: #
11: # RelNum=17.03.00
12: # RelDat=2021-01-18
13: # -----
14: #
15: # -----
16: # Global constants
17: # -----
18: source "$(dirname "$0")"/kzcommon.sh
19: readonly ARCHIVE_FROM=$HOME/uploads/karelzimmer.nl/httpdocs/data/linux
20: readonly ARCHIVE_TO=$HOME/uploads/karelzimmer.nl/httpdocs/data/linux-archieff
21: readonly CHANGELOG=$HOME/kzscripts/CHANGELOG
22: readonly CHANGELOG_GZ=$HOME/kzscripts/CHANGELOG.gz
23: readonly DELETE_FROM_1=$HOME/kzscripts
24: readonly DELETE_FROM_2=$HOME/kzscripts.deb
25: readonly DELETE_FROM_3=$HOME/Documenten/Checklists
26: #
27: # Bij aanpassingen ook .completion aanpassen!
28: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
29: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
30: readonly USAGE="Gebruik: $PROGRAMME $OPTIONS_USAGE_COMMON
31:                 [SELECTIE...]"
32: readonly HELP="Gebruik: $PROGRAMME [OPTIE...] [--] [SELECTIE...]"
33: #
34: Scripts archiveren.
35: #
36: Opties:
37: $OPTIONS_HELP_COMMON
38: #
39: Argumenten:
40:   SELECTIE      zoek naar bestanden overeenkomend met opgegeven SELECTIES"
41: #
42: # -----
43: # Global variables
44: # -----
45: declare ARCHIVED=false
46: declare ARGUMENT_SELECTION=false
47: declare -a SELECTION_ARGUMENT=''
48: declare DELETED=false
49: declare NO_FILES_FOUND=true
50: #
51: # -----
52: # Functions
53: # -----
54: check_input() {
55:     local -i getopt_rc=0
56:     local -i search4_arg_num=0
57:     local parsed=''
58: #
59:     parsed=$(
60:         getopt --alternative          \
61:                --options              "$OPTIONS_SHORT"          \
62:                --longoptions         "$OPTIONS_LONG"           \
63:                --name                 "$PROGRAMME"              \
64:                -- "$@"
65:     ) || getopt_rc=$?
66:     if [[ $getopt_rc -ne 0 ]]; then
67:         printf '%s\n' "$USAGELINE" >&2
68:         exit $ERROR
```

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

```
137:         if [[ $dirname = "$ARCHIVE_FROM"* ]]; then
138:             process_archive
139:         else
140:             process_remove
141:         fi
142:     done <<(
143:         find      "$ARCHIVE_FROM"      \
144:                 "$DELETE_FROM_1"     \
145:                 "$DELETE_FROM_2"     \
146:                 "$DELETE_FROM_3"     \
147:                 -iname                 \
148:                 '*"$search4"*'        \
149:                 -type f                \
150:                 -print                  |
151:                 sort
152:     )
153: done
154:
155: if $NO_FILES_FOUND; then
156:     warning "Geen bestanden gevonden met '${SELECTION_ARGUMENT[*]}' in de \
157: naam."
158:     exit $WARNING
159: fi
160: }
161:
162: process_archive() {
163:     printf "\n${BOLD}%s\n%s${NORMAL}\n" "$basename" "$dirname"
164:     while true; do
165:         read -rp 'archiveren/Overslaan/stoppen? [a/O/s]: ' < /dev/tty
166:         case $REPLY in
167:             a*|A*)
168:                 archive_file
169:                 break
170:             ;;
171:             o*|O*|'')
172:                 printf '%s\n' 'Overgeslagen.'
173:                 break
174:             ;;
175:             s*|S*)
176:                 printf '%s\n' 'Gestopt.'
177:                 exit $SUCCESS
178:             ;;
179:             *)
180:                 printf '%s' "${UP_ONE_LINE}${ERASE_LINE}"
181:                 continue
182:             ;;
183:         esac
184:     done
185: }
186:
187: archive_file() {
188:     local olddir=""
189:     local newdir=""
190:
191:     olddir=$dirname
192:     newdir=${olddir//linux/linux-archief}
193:     mv "$olddir/$basename" "$newdir" |& $LOGCMD
194:     printf '%s\n' 'Gearchiveerd.'
195:     warning 'Pas website aan!'
196:     ARCHIVED=true
197: }
198:
199: process_remove() {
200:     printf "\n${BOLD}%s\n%s${NORMAL}\n" "$basename" "$dirname"
201:     while true; do
202:         read -rp 'verwijderen/Overslaan/stoppen? [v/O/s]: ' < /dev/tty
203:         case $REPLY in
204:             v*|V*)
```

```
205:         remove_file
206:         break
207:         ;;
208:     o*|O*|'')
209:         printf '%s\n' 'Overgeslagen.'
210:         break
211:         ;;
212:     s*|S*)
213:         printf '%s\n' 'Gestopt.'
214:         exit $SUCCESS
215:         ;;
216:     *)
217:         printf '%b' "${UP_ONE_LINE}${ERASE_LINE}"
218:         continue
219:         ;;
220:     esac
221: done
222: }
223:
224: remove_file() {
225:     rm "$file" |& $LOGCMD
226:     if [[ $dirname = "$DELETE_FROM_1" ]]; then
227:         gunzip "$CHANGELOG_GZ" |& $LOGCMD
228:         sed --in-place \
229:             --expression="/^$basename /d" \
230:             "$CHANGELOG"
231:         gzip --best \
232:             --force \
233:             "$CHANGELOG" |& $LOGCMD
234:     fi
235:     printf '%s\n' 'Verwijderd.'
236:     DELETED=true
237: }
238:
239: term_script() {
240:     local exit_msg=''
241:
242:     if $DELETED; then
243:         exit_msg='\nEr zijn bestanden verwijderd.'
244:     else
245:         exit_msg='\nEr zijn geen bestanden verwijderd.'
246:     fi
247:     if $ARCHIVED; then
248:         exit_msg="$exit_msg\n\nEr zijn bestanden gearchieveerd."
249:     else
250:         exit_msg="$exit_msg\n\nEr zijn geen bestanden gearchieveerd."
251:     fi
252:     info "$exit_msg"
253:     exit $SUCCESS
254: }
255:
256: # -----
257: # Main line
258: # -----
259: main() {
260:     init_script
261:     check_input "$@"
262:     process_input
263:     term_script
264: }
265:
266: main "$@"
267:
268: # EOF
```