

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # USB-sticks maken.
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-mkusb
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2014
13:
14: VERSION_NUMBER=18.06.10
15: VERSION_DATE=2021-10-04
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=true
23: readonly OPTIONS_SHORT=${OPTIONS_SHORT_COMMON}s:t:'
24: readonly OPTIONS_LONG=${OPTIONS_LONG_COMMON},source:,target:'
25: readonly USAGE="Gebruik: $DISPLAY_NAME [-s|--source=BRON] \
26: [-t|--target=DOEL]
27:     $OPTIONS_USAGE_COMMON"
28: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...]"
29:
30: USB-sticks maken.
31:
32: Apties:
33:     -s --source=BRONMAP
34:         gebruik beeldbestand (.iso) in de BRONMAP
35:     -t --target=APPARAAT
36:         gebruik de opgegeven USB-stick-device APPARAAT
37: $OPTIONS_HELP_COMMON"
38:
39: readonly DEVICE_DEFAULT=/dev/sdb
40: readonly SOURCE_DEFAULT=/home/${SUDO_USER:-$USER}/Downloads
41:
42:
43: #####
44: # Global variables
45: #####
46:
47: declare DEVICE=''
48: declare NO_FILES_FOUND=true
49: declare OPTION_SOURCE=false
50: declare OPTION_TARGET=false
51: declare SOURCEDIR=''
52: declare SOURCE_ARGUMENT=''
53: declare TARGET_ARGUMENT=''
54:
55: #####
56: # Functions
57: #####
58:
59: check_input() {
60:     local -i getopt_rc=0
61:     local parsed=''
62:
63:     parsed=$(
64:         getopt --alternative \
65:             --options "$OPTIONS_SHORT" \
66:             --longoptions "$OPTIONS_LONG" \
67:             --name "$DISPLAY_NAME" \
68:             -- "$@"
```

```
69:     )          || getopt_rc=$?
70:   if [[ $getopt_rc -ne $SUCCESS ]]; then
71:     printf '%s\n' "$USAGELINE" >&2
72:     exit $ERROR
73:   fi
74:   eval set -- "$parsed"
75:   process_common_options "$@"
76:
77:   while true; do
78:     case $1 in
79:       -s|--source)
80:         if $OPTION_SOURCE; then
81:           TEXT="optie '$1' eenmaal opgeven"
82:           printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" \
83:             "$TEXT" \
84:             "$USAGELINE" >&2
85:           exit $ERROR
86:         else
87:           OPTION_SOURCE=true
88:           SOURCE_ARGUMENT=$2
89:         fi
90:         shift 2
91:         ;;
92:       -t|--target)
93:         if $OPTION_TARGET; then
94:           TEXT="optie '$1' eenmaal opgeven"
95:           printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" \
96:             "$TEXT" \
97:             "$USAGELINE" >&2
98:           exit $ERROR
99:         else
100:          OPTION_TARGET=true
101:          TARGET_ARGUMENT=$2
102:        fi
103:        shift 2
104:        ;;
105:       --)
106:         shift
107:         break
108:        ;;
109:       *)
110:         shift
111:        ;;
112:     esac
113:   done
114:
115:   if [[ "$*" ]]; then
116:     TEXT='geen argumenten opgeven'
117:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
118:     exit $ERROR
119:   fi
120:
121:   if $OPTION_SOURCE; then
122:     SOURCEDIR=$SOURCE_ARGUMENT
123:   else
124:     SOURCEDIR=$SOURCE_DEFAULT
125:   fi
126:   if ! [[ -d $SOURCEDIR ]]; then
127:     TEXT="map '$SOURCEDIR' bestaat niet"
128:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
129:     exit $ERROR
130:   fi
131:
132:   if $OPTION_TARGET; then
133:     DEVICE=$TARGET_ARGUMENT
134:   else
135:     DEVICE=$DEVICE_DEFAULT
136:   fi
```

```
137:
138: # Een non-gui script gestart met optie gui.
139: if $OPTION_GUI; then
140:     OPTION_GUI=false
141:     DESKTOP_TERMINAL=false
142: fi
143:
144: check_user
145: request_input
146: }
147:
148:
149: request_input() {
150:     warning 'De volgende /dev/sd schijven en partities zijn aanwezig:'
151:     TEXT='Gebruik optie --target om een ander device op te geven.'
152:     if findmnt --list --output=SOURCE,TARGET | grep --regexp='/dev/sd'; then
153:         warning "
154: $DEVICE wordt gebruikt als doel en OVERSCHREVEN!"
155:         while true; do
156:             read -rp 'Is dit OK? [j/N]: '
157:             case $REPLY in
158:                 j*|J*)
159:                 break
160:                 ;;
161:                 n*|N*|'')
162:                 printf '\n%s\n%s\n' "$TEXT" "$USAGELINE"
163:                 exit $SUCCESS
164:                 ;;
165:                 *)
166:                 continue
167:                 ;;
168:             esac
169:         done
170:     else
171:         warning "
172: Geen $DEVICE gevonden om te gebruiken als doel!"
173:         printf '%s\n%s\n' "$TEXT" "$USAGELINE"
174:         exit $WARNING
175:     fi
176: }
177:
178:
179: process_input() {
180:     local -i isosize=0
181:     local     usbname=''
182:     local     file=''
183:
184:     NO_FILES_FOUND=true
185:
186:     for file in "$SOURCEDIR"/*.iso; do
187:
188:         if ! [[ -f "$file" ]]; then
189:             continue
190:         fi
191:
192:         NO_FILES_FOUND=false
193:         usbname=$(basename "$file" .iso | tr '-' ' ')
194:
195:         printf "\n${BOLD}%s${NORMAL}\n" "$usbname"
196:
197:         while true; do
198:             read -rp 'maken/Overslaan/stoppen? [m/O/s]: '
199:             case $REPLY in
200:                 m*|M*)
201:                     isosize=$(
202:                         du --apparent-size \
203:                             --block-size=1 \
204:                             "$file" |
```

```
205:         awk '{print $1}'
206:     )
207:     check_on_ac_power
208:     create_usb_stick "$file"
209:     while true; do
210:         read -rp "Nog een USB-stick '$susname' maken? [j/N]: "
211:         case $REPLY in
212:             j*|J*)
213:                 check_on_ac_power
214:                 create_usb_stick "$file"
215:                 continue
216:             ;;
217:             n*|N*|'')
218:                 printf '%s\n' 'Klaar.'
219:                 break
220:             ;;
221:             *)
222:                 continue
223:             ;;
224:         esac
225:     done
226:     break
227:     ;;
228:     o*|O*|'')
229:         printf '%s\n' 'Overgeslagen.'
230:         break
231:     ;;
232:     s*|S*)
233:         exit $SUCCESS
234:     ;;
235:     *)
236:         continue
237:     ;;
238: esac
239: done
240: done
241:
242: if $NO_FILES_FOUND; then
243:     warning "Geen beeldbestanden (.iso) gevonden in map '$SOURCEDIR'."
244:     exit $WARNING
245: fi
246: }
247:
248:
249: create_usb_stick() {
250:     local dd_from=${1:-dd_from?}
251:
252:     read -rp "Plaats USB-stick '$susname', gevolgd door Enter [Enter]: "
253:     if [[ -e $DEVICE ]]; then
254:         info "USB-stick $susname wordt gemaakt..."
255:
256:         dd if="$dd_from" \
257:            status=none \
258:         pv --size="$sisosize" \
259:         dd of=$DEVICE \
260:            bs=4M \
261:            status=none
262:
263:         info "Gegevens schrijven naar $susname (kan even duren)..."
264:         sync
265:
266:         info "USB-stick '$susname' is gemaakt."
267:
268:         info "Ter controle is er een Kernel-based Virtuele Machine (KVM) \
269: gestart."
270:         if ! qemu-system-x86_64 -enable-kvm \
271:            -m 2048 \
272:            -hda $DEVICE |& $LOGCMD; then
```

```
273:         true
274:     fi
275:
276:     if ! umount "$DEVICE" 2> /dev/null; then
277:         read -rp "Ontkoppel de USB-stick '$usbname', gevolgd door Enter \
278: [Enter]: "
279:     fi
280:     read -rp "Verwijder de USB-stick '$usbname', gevolgd door Enter \
281: [Enter]: "
282:     else
283:         warning "USB-stick '$usbname' is niet aanwezig op $DEVICE."
284:         exit $WARNING
285:     fi
286: }
287:
288:
289: term_script() {
290:     exit $SUCCESS
291: }
292:
293:
294: #####
295: # Main line
296: #####
297:
298: main() {
299:     init_script "$@"
300:     check_input "$@"
301:     process_input
302:     term_script
303: }
304:
305:
306: main "$@"
307:
308:
309: # EOF
```