

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Netwerkschijf aankoppelen.
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-nas
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2014
13:
14: VERSION_NUMBER=13.00.07
15: VERSION_DATE=2021-09-21
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=true
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $DISPLAY_NAME $OPTIONS_USAGE_COMMON"
26: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...]"
27:
28: Netwerkschijf aankoppelen.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: readonly NAS_LOGIN=/usr/local/etc/${PROGRAM_NAME}login
34: readonly NAS_MOUNT_POINT=/mnt/$PROGRAM_NAME
35: NAS_FIXED_IP_ADDRESS="
36: Zorg dat de NAS een vast IP-adres heeft.
37:
38: Hiervoor is nodig:
39:
40: - het MAC-adres van de NAS \
41: (zoals $(ip -oneline link | awk '/ UP /{print $17}'))
42: - toegang tot de NAS via een webbrowser
43: - toegang tot de router via een webbrowser
44:
45: 1. Kies in de router met het MAC-adres van de NAS een
46: IP-adres buiten het bereik van DHCP die automatisch
47: de IP-adressen uitdeelt (vaak 192.168.0.200).
48: 2. Kies in de NAS het vaste IP-adres en zet DHCP uit."
49: readonly NAS_FIXED_IP_ADDRESS
50: readonly SYSTEMD_MOUNT_DIR=/etc/systemd/system
51: readonly SYSTEMD_MOUNT_FILE=mnt-nas.mount
52: readonly SYSTEMD_MOUNT=$SYSTEMD_MOUNT_DIR/$SYSTEMD_MOUNT_FILE
53:
54:
55: #####
56: # Global variables
57: #####
58:
59: declare NAS_GEBRUIKERSNAAM=${SUDO_USER:-$USER}
60: declare NAS_IP_ADRES=192.168.0.200
61: declare NAS_SHARENAAM=${SUDO_USER:-$USER}
62: declare NAS_WACHTWOORD='geheim'
63:
64:
65: #####
66: # Functions
67: #####
68:
```

```
69: check_input() {
70:     local -i getopt_rc=0
71:     local parsed=''
72:
73:     parsed=$(
74:         getopt --alternative \
75:                --options "$OPTIONS_SHORT" \
76:                --longoptions "$OPTIONS_LONG" \
77:                --name "$DISPLAY_NAME" \
78:                -- "$@"
79:     ) || getopt_rc=$?
80:     if [[ $getopt_rc -ne $SUCCESS ]]; then
81:         printf '%s\n' "$USAGELINE" >&2
82:         exit $ERROR
83:     fi
84:     eval set -- "$parsed"
85:     process_common_options "$@"
86:
87:     while true; do
88:         case $1 in
89:             --)
90:                 shift
91:                 break
92:                 ;;
93:             *)
94:                 shift
95:                 ;;
96:         esac
97:     done
98:
99:     if [[ "$*" ]]; then
100:         TEXT='geen argumenten opgeven'
101:         printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
102:         exit $ERROR
103:     fi
104:
105:     check_user
106:     # Na check_user voor toegang tot media.
107:     request_input
108: }
109:
110:
111: request_input() {
112:     local fstab=''
113:     local quest1='Deze regels verwijderen?'
114:     local quest2="Heeft de NAS een vast IP-adres zoals $NAS_IP_ADRES?"
115:     local quest3='IP-adres van de NAS'
116:     local quest4='Gebruikersnaam op de NAS'
117:     local quest5='Wachtwoord van deze gebruiker op de NAS'
118:     local quest6='Share-naam van deze gebruiker op de NAS'
119:     local -i zenity_rc=0
120:     local warning=''
121:
122:     fstab=$(
123:         if ! grep --regexp="$PROGRAM_NAME" /etc/fstab; then
124:             printf '\n'
125:         fi
126:     )
127:     if [[ $fstab ]]; then
128:         warning="Onderstaande regels zijn al aanwezig in de koppeltabel:
129: $fstab"
130:         if $OPTION_GUI; then
131:             # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
132:             # Voorbeeld: Unable to init server: Kon niet verbinden:
133:             #                 Verbinding is geweigerd
134:             #                 en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
135:             #                 cannot open display:
136:             zenity --question \
```

```

137:         --no-markup           \
138:         --width      400       \
139:         --height     100       \
140:         --title      'Waarschuwing' \
141:         --text       "$warning\n\n$quest1" \
142:         2> >($LOGCMD)           || zenity_rc=$?
143:     if [[ $zenity_rc -eq $SUCCESS ]]; then
144:         verwijder_regels
145:     fi
146: else
147:     printf '%s\n' "$warning"
148:     read -rp "$quest1 [j/N]: "
149:     while true; do
150:         case $REPLY in
151:             j*|J*)
152:                 verwijder_regels
153:                 break
154:                 ;;
155:             n*|N*|'')
156:                 info 'Ok, we gaan verder.'
157:                 break
158:                 ;;
159:             *)
160:                 continue
161:                 ;;
162:         esac
163:     done
164: fi
165:
166: if $OPTION_GUI; then
167:     if ! zenity --question           \
168:         --no-markup                 \
169:         --width      400             \
170:         --height     100            \
171:         --title      'Vraag'        \
172:         --text       "$quest2"      2> >($LOGCMD); then
173:         zenity --info                 \
174:             --no-markup                 \
175:             --width      500           \
176:             --height     100          \
177:             --title      'Melding'    \
178:             --text       "$NAS_FIXED_IP_ADDRESS" 2> >($LOGCMD) || true
179:         exit $SUCCESS
180:     fi
181: else
182:     while true; do
183:         read -rp "$quest2 [j/N]: "
184:         case $REPLY in
185:             j*|J*)
186:                 info 'Ok, we gaan verder.'
187:                 break
188:                 ;;
189:             n*|N*|'')
190:                 printf '%s\n' "$NAS_FIXED_IP_ADDRESS"
191:                 exit $SUCCESS
192:                 ;;
193:             *)
194:                 continue
195:                 ;;
196:         esac
197:     done
198: fi
199:
200: TEXT='Beantwoord alstublieft de volgende vragen'
201: if $OPTION_GUI; then
202:     REPLY=$(
203:         zenity --forms
204:     )

```

```

205:         --title           'Vragen'           \
206:         --text            "$TEXT"            \
207:         --separator       ", "              \
208:         --add-entry       "$quest3"         \
209:         --add-entry       "$quest4"         \
210:         --add-password   "$quest5"         \
211:         --add-entry       "$quest6"         2> >($LOGCMD)
212:     ) || zenity_rc=$?
213:     if [[ $zenity_rc -ne $SUCCESS ]]; then
214:         exit $SUCCESS
215:     fi
216:     NAS_IP_ADRES=$( printf '%s' "$REPLY" | awk -F, '{print $1}')
217:     NAS_GEBRUIKERSNAAM=$( printf '%s' "$REPLY" | awk -F, '{print $2}')
218:     NAS_WACHTWOORD=$( printf '%s' "$REPLY" | awk -F, '{print $3}')
219:     NAS_SHARENAAM=$( printf '%s' "$REPLY" | awk -F, '{print $4}')
220:     else
221:         printf '%s\n%s\n' "$TEXT." "Als u niets opgeeft, wordt de waarde \
222: tussen [] gebruikt als antwoord."
223:         read -rp "
224: $quest3? [$NAS_IP_ADRES]: "
225:         if [[ $REPLY ]]; then
226:             NAS_IP_ADRES=$REPLY
227:         fi
228:         read -rp "
229: $quest4? [$NAS_GEBRUIKERSNAAM]: "
230:         if [[ $REPLY ]]; then
231:             NAS_GEBRUIKERSNAAM=$REPLY
232:         fi
233:         read -rsp "
234: $quest5? [$NAS_WACHTWOORD]: "
235:         if [[ $REPLY ]]; then
236:             NAS_WACHTWOORD=$REPLY
237:         fi
238:         printf '\n'
239:         read -rp "
240: $quest6? [$NAS_SHARENAAM]: "
241:         if [[ $REPLY ]]; then
242:             NAS_SHARENAAM=$REPLY
243:         fi
244:     fi
245: }
246:
247:
248: process_input() {
249:     local text0='Opdrachten worden verwerkt'
250:     local text1='Maak koppelpunt...'
251:     local text2='Maak NAS login...'
252:     local text3='Maak systemd...'
253:     local text4='Koppel NAS aan...'
254:
255:     TITLE='Netwerkschijf aankoppelen'
256:     if $OPTION_GUI; then
257:         (
258:             printf '%s\n' "$text1"
259:             create_mountpoint
260:             printf '%s\n' "$text2"
261:             create_nas_login
262:             printf '%s\n' "$text3"
263:             create_systemd_unit_file
264:             printf '%s\n' "$text4"
265:             mount_nas
266:         ) | &
267:         zenity --progress           \
268:               --pulsate            \
269:               --auto-close         \
270:               --no-cancel          \
271:               --width              600 \
272:               --height             50  \

```

```
273:             --title      "$TITLE"      \
274:             --text       "$text0"      2> >($LOGCMD)
275:     else
276:         create_mountpoint
277:         create_nas_login
278:         create_systemd_unit_file
279:         mount_nas
280:     fi
281: }
282:
283:
284: create_mountpoint() {
285:     if [[ -d $NAS_MOUNT_POINT ]]; then
286:         return $SUCCESS
287:     fi
288:
289:     printf '%s\n' 'Maak NAS koppelpunt...'
290:
291:     mkdir "$NAS_MOUNT_POINT" |& $LOGCMD
292:     chmod '777' "$NAS_MOUNT_POINT" |& $LOGCMD
293: }
294:
295:
296: create_nas_login() {
297:     printf '%s\n' 'Maak NAS login...'
298:     TEXT="username=$NAS_GEbruikersnaam\npassword=$NAS_Wachtwoord"
299:     printf '%b\n' "$TEXT" > "$NAS_LOGIN"
300:     chmod '600' "$NAS_LOGIN" |& $LOGCMD
301: }
302:
303:
304: create_systemd_unit_file() {
305:     printf '%s\n' 'Maak systemd mount-bestand...'
306:     printf '%s\n' "[Unit]
307: Description=Mount NAS Directory
308:
309: [Mount]
310: What=//$NAS_IP_ADRES/$NAS_SHARENAAM
311: Where=/mnt/nas
312: Type=cifs
313: Options=credentials=$NAS_LOGIN,sec=ntlm,vers=1.0,rw,noperm,icharset=utf8,\
314: dir_mode=0777,file_mode=0777
315:
316: [Install]
317: WantedBy=multi-user.target" > "$SYSTEMD_MOUNT"
318: }
319:
320:
321: mount_nas() {
322:     printf '%s\n' 'Koppel NAS aan...'
323:     systemctl enable "$SYSTEMD_MOUNT_FILE" |& $LOGCMD
324:     if ! systemctl start "$SYSTEMD_MOUNT_FILE" |& $LOGCMD; then
325:         true
326:     fi
327:     systemctl daemon-reload |& $LOGCMD
328: }
329:
330:
331: term_script() {
332:     if ! systemctl status "$SYSTEMD_MOUNT_FILE"; then
333:         true
334:     fi
335:     info "Het systemd mount-bestand is geactiveerd.
336: De NAS zou beschikbaar moeten zijn op '$NAS_MOUNT_POINT' .
337:
338: De-actieveer het systemd mount-besta2nd met de opdracht:
339: ${BLUE}sudo systemctl disable mnt-nas.mount${NORMAL}"
340:     exit $SUCCESS
```

```
341: }
342:
343:
344: #####
345: # Main line
346: #####
347:
348: main() {
349:     init_script "$@"
350:     check_input "$@"
351:     process_input
352:     term_script
353: }
354:
355:
356:
357: main "$@"
358:
359:
360: # EOF
```