

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Back-up terugzetten.
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-restore
11: DISPLAY_NAME=${PROGRAM_NAME/kz-/kz }
12: RELEASE_YEAR=2007
13:
14: VERSION_NUMBER=39.01.01
15: VERSION_DATE=2021-10-08
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=true
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON'e:s:t:'
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON',source:,target:,exclude:'
25: readonly USAGE="Gebruik: $DISPLAY_NAME [-e|--exclude=UITSLUITEN]... \
26: [-s|--source=BRON] [-t|--target=DOEL]
27:             $OPTIONS_USAGE_COMMON
28:             [SELECTIE...]"
29: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...] [SELECTIE...]"
30:
31: Back-up terugzetten.
32:
33: Opties:
34:   -e --exclude=UITSLUITEN
35:           sluit uit bestand of map in UITSLUITEN
36:   -s --source=BRONMAP
37:           zoek back-up vanaf de BRONMAP
38:   -t --target=DOELMAP
39:           zet back-up terug in de DOELMAP
40: $OPTIONS_HELP_COMMON
41:
42: Argumenten:
43:   SELECTIE      zet opgegeven mappen en bestanden terug"
44:
45: readonly EXCLUDEFILE_DEFAULT=/usr/local/etc/$PROGRAM_NAME-exclude-dflt
46: EXCLUDEFILE_DEFAULT_CONTENT="$(\
47:   find      /etc/skel                \
48:           -type f                    \
49:           -exec basename '{}' \;    |
50:   awk      '$0="/home/*/"$0'
51:   )"; readonly EXCLUDEFILE_DEFAULT_CONTENT
52: # Hieboven worden de bestanden opgenomen die bij het aanmaken van een
53: # gebruiker worden neergezet in $HOME. Meestal zijn dit de bestanden
54: # .bash_logout, .bashrc, examples.desktop, en .profile (bij Ubuntu).
55: # Deze bestanden terugzetten bij een nieuwe versie van Linux betekent dat
56: # deze bestanden worden teruggezeten naar de oude versie; niet wenselijk.
57: readonly EXCLUDEFILE_OPTIONAL=/usr/local/etc/$PROGRAM_NAME-exclude-opt
58: readonly SELECTION_DEFAULT=''
59: readonly SOURCE_DEFAULT=/media/
60: readonly TARGET_DEFAULT=/
61:
62:
63: #####
64: # Global variables
65: #####
66:
67: declare -a EXCLUDE_ARGUMENT=()
68: declare     ARGUMENT_SELECTION=false
```

```
69: declare -a SELECTION=()
70: declare -a SELECTION_ARGUMENT=()
71: declare BACKUP=''
72: declare BACKUPSIZE_HUMAN=''
73: declare -i BACKUPSIZE_MACHINE=0
74: declare -i TAR_RC=0
75: declare OPTION_EXCLUDE=false
76: declare OPTION_SOURCE=false
77: declare OPTION_TARGET=false
78: declare SOURCE=''
79: declare SOURCE_ARGUMENT=''
80: declare -i SYNC_RC=0
81: declare TARGET=''
82: declare TARGET_ARGUMENT=''
83:
84:
85: #####
86: # Functions
87: #####
88:
89: check_input() {
90:     local -i exclude_arg_num=0
91:     local -i select_arg_num=0
92:     local -i select_num=0
93:     local -i getopt_rc=0
94:     local parsed=''
95:
96:     parsed=$(
97:         getopt --alternative \
98:                --options "$OPTIONS_SHORT" \
99:                --longoptions "$OPTIONS_LONG" \
100:                --name "$DISPLAY_NAME" \
101:                -- "$@"
102:     ) || getopt_rc=$?
103:     if [[ $getopt_rc -ne $SUCCESS ]]; then
104:         printf '%s\n' "$USAGELINE" >&2
105:         exit $ERROR
106:     fi
107:     eval set -- "$parsed"
108:     process_common_options "$@"
109:
110:     while true; do
111:         case $1 in
112:             -e|--exclude)
113:                 OPTION_EXCLUDE=true
114:                 EXCLUDE_ARGUMENT[$exclude_arg_num]=$2
115:                 ((++exclude_arg_num))
116:                 shift 2
117:                 ;;
118:             -s|--source)
119:                 if $OPTION_SOURCE; then
120:                     TEXT="optie '$1' eenmaal opgeven"
121:                     printf "$DISPLAY_NAME: %s\n%s\n" \
122:                            "$TEXT" \
123:                            "$USAGELINE" >&2
124:                     exit $ERROR
125:                 else
126:                     OPTION_SOURCE=true
127:                     SOURCE_ARGUMENT=$2
128:                 fi
129:                 shift 2
130:                 ;;
131:             -t|--target)
132:                 if $OPTION_TARGET; then
133:                     TEXT="optie '$1' eenmaal opgeven"
134:                     printf "$DISPLAY_NAME: %s\n%s\n" \
135:                            "$TEXT" \
136:                            "$USAGELINE" >&2
```

```
137:             exit    $ERROR
138:         else
139:             OPTION_TARGET=true
140:             TARGET_ARGUMENT=$2
141:         fi
142:     shift 2
143:     ;;
144: --)
145:     shift
146:     break
147:     ;;
148: *)
149:     shift
150:     ;;
151: esac
152: done
153:
154: if $OPTION_SOURCE; then
155:     SOURCE=$SOURCE_ARGUMENT
156:     if ! [[ -d $SOURCE ]]; then
157:         TEXT="map '$SOURCE' bestaat niet"
158:         printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
159:         exit $ERROR
160:     fi
161: else
162:     SOURCE=$SOURCE_DEFAULT
163: fi
164:
165: if $OPTION_TARGET; then
166:     if ! [[ -d $TARGET_ARGUMENT ]]; then
167:         TEXT="map '$TARGET_ARGUMENT' bestaat niet"
168:         printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
169:         exit $ERROR
170:     fi
171:     TARGET=$TARGET_ARGUMENT
172: else
173:     TARGET=$TARGET_DEFAULT
174: fi
175:
176: while [[ "$*" ]]; do
177:     ARGUMENT_SELECTION=true
178:     SELECTION_ARGUMENT[$select_arg_num]=$1
179:     ((++select_arg_num))
180:     shift
181: done
182:
183: check_user
184:
185: # Na check_user voor toegang tot media.
186: if $ARGUMENT_SELECTION; then
187:     for select_num in "${!SELECTION_ARGUMENT[@]}; do
188:         # Verwijder slash aan het begin en voeg * toe aan begin en einde
189:         # voor optie wildcards tar extract.
190:         if [[ ${SELECTION_ARGUMENT[$select_num]:0:1} = '/' ]]; then
191:             SELECTION[$select_num]="*${SELECTION_ARGUMENT[$select_num]:1}*"
192:         else
193:             SELECTION[$select_num]="*${SELECTION_ARGUMENT[$select_num]}*"
194:         fi
195:     done
196: else
197:     SELECTION[0]=$SELECTION_DEFAULT
198: fi
199:
200: request_input
201: }
202:
203:
204: request_input() {
```

```
205:     if $OPTION_GUI; then
206:         request_input_gui
207:     else
208:         request_input_tui
209:     fi
210: }
211:
212:
213: request_input_gui() {
214:     local filename="${SOURCE_ARGUMENT:-/media}"
215:     local -i file_selected=0
216:
217:     TITLE='Kies de terug te zetten back-up'
218:     # Constructie '2> >($LOGCMD)' om stderr naar de log te krijgen.
219:     # Voorbeeld: Unable to init server: Kon niet verbinden:
220:     #             Verbinding is geweigerd
221:     #             en: (zenity:47712): Gtk-WARNING **: 10:35:49.339:
222:     #             cannot open display:
223:     BACKUP="$(
224:         zenity --file-selection          \
225:             --width          600        \
226:             --height         50         \
227:             --title          "$TITLE"   \
228:             --filename       "$filename" \
229:             --file-filter    '*.tar'    2> >($LOGCMD)
230:     )" || file_selected=?
231:     if [[ $file_selected -ne 0 ]]; then
232:         warning 'Geen back-up geselecteerd.
233:
234: Geen back-up teruggezet!'
235:         exit $WARNING
236:     fi
237: }
238:
239:
240: request_input_tui() {
241:     local range=''
242:     local -a buparray
243:     local -i bupnum=0
244:
245:     info "Er wordt naar back-ups gezocht vanaf $SOURCE.
246: Het medium waarop de back-up staat dient aangekoppeld te zijn.
247: Voorbeelden van een medium zijn een (externe) (netwerk)schijf,
248: USB-stick, CD, of DVD.
249: Zoek nu naar back-ups van deze computer (kan even duren)..."
250:
251:     show_backups "$HOSTNAME"
252:
253:     if [[ $bupnum -eq 0 ]]; then
254:         warning "Geen back-ups gevonden van deze computer in $SOURCE."
255:         REPLY='A'
256:     fi
257:
258:     if [[ $bupnum -gt 0 ]]; then
259:         printf '\n%s\n' "Kies nummer, of A om te zoeken naar alle back-ups, \
260: en S om te stoppen."
261:         if [[ $bupnum -eq 1 ]]; then
262:             range=''
263:         else
264:             range="-${bupnum}"
265:         fi
266:         while true; do
267:             read -rp "Terugzetten back-up nummer [1$range, of A, of S, \
268: standaard 1]: "
269:             case $REPLY in
270:                 '')
271:                     BACKUP=${buparray[1]}
272:                     break
```

```
273:             ;;
274:             a|A)
275:                 break
276:             ;;
277:             s|S)
278:                 exit $SUCCESS
279:             ;;
280:             *[^[:digit:]]*)
281:                 continue
282:             ;;
283:             *)
284:                 if [[ $REPLY -ge 1 && $REPLY -le $bupnum ]] && /dev/null
285:                 then
286:                     BACKUP=${buparray[$REPLY]}
287:                     break
288:                 else
289:                     continue
290:                 fi
291:             ;;
292:         esac
293:     done
294: fi
295:
296: if [[ $REPLY = [A,a] ]]; then
297:     printf '\n%s\n' "Zoek nu naar back-ups van alle computers (dit kan \
298: even duren)..."
299:     bupnum=0
300:     show_backups ''
301:     if [[ $bupnum -eq 0 ]]; then
302:         warning "Geen back-ups gevonden in $SOURCE."
303:         info "
304: Gebruik optie source om op te geven waar de back-up gevonden kan worden.
305: Voorbeeld: $DISPLAY_NAME --source BRONMAP
306: $USAGELINE" >&2
307:         exit $WARNING
308:     elif [[ $bupnum -eq 1 ]]; then
309:         range=''
310:     else
311:         range="-${bupnum}"
312:     fi
313:     printf '\n%s\n' "Kies nummer, of S om te stoppen."
314:     while true; do
315:         read -rp "Terugzetten back-up nummer [1$range, of S, \
316: standaard S]: "
317:         case $REPLY in
318:             s|S|'')
319:                 exit $SUCCESS
320:             ;;
321:             *[^[:digit:]]*)
322:                 continue
323:             ;;
324:             *)
325:                 if [[ $REPLY -ge 1 && $REPLY -le $bupnum ]] && /dev/null
326:                 then
327:                     BACKUP=${buparray[$REPLY]}
328:                     break
329:                 else
330:                     continue
331:                 fi
332:             ;;
333:         esac
334:     done
335: fi
336: }
337:
338:
339: show_backups() {
340:     local search4=${1-search4?}
```

```
341: local backup=''
342: local -i bupnumlen=2
343: local dirname=''
344: local filename=''
345: local filesize_human=''
346:
347: # Zoek vanaf bronmap ("${SOURCE}") naar back-ups. Find zoals hier gebruikt
348: # levert, gescheiden door een TAB, back-upnaam en directory waarin de
349: # back-up zich bevindt. De sort sorteert op back-upnaam aflopend, dat is
350: # van nieuw naar oud.
351:
352: while read -r backup; do
353:     filename=$(printf '%s' "$backup" | cut --fields=1)
354:     dirname=$( printf '%s' "$backup" | cut --fields=2)
355:     filesize_human=$(
356:         du --apparent-size          \
357:            --block-size=1           \
358:            --human-readable         \
359:            "$dirname/$filename"    |
360:         awk '{print $1}'
361:     )
362:
363:     ((++bupnum))
364:     buparray[$bupnum]=$dirname/$filename
365:
366:     if [[ $bupnum -eq 1 ]]; then
367:         if [[ -z $search4 ]]; then
368:             printf '\n%s\n' "De volgende back-ups zijn gevonden:"
369:         else
370:             printf '\n%s\n' "De volgende back-ups van $search4 zijn \
371: gevonden, van nieuw naar oud:"
372:             fi
373:             fi
374:
375:             printf '%+*s %s\t%s\t%s\n' $bupnumlen "$bupnum" "${basename \
376: "$filename" .tar}" "$filesize_human" "iB" "$dirname"
377:             done <<(
378:                 find      "${SOURCE}"          \
379:                    -noleaf          \
380:                    -type f          \
381:                    -name '*backup*"$search4"*.*tar' \
382:                    -printf '%f\t%h\n' \
383:                    2> >($LOGCMD)      |
384:                 sort      --reverse
385:             )
386: }
387:
388:
389: process_input() {
390:     local select=''
391:     local size_num=0
392:     local size_unit=''
393:
394:     printf '%s\n' "$EXCLUDEFILE_DEFAULT_CONTENT" > "$EXCLUDEFILE_DEFAULT"
395:
396:     if $OPTION_EXCLUDE; then
397:         for exclude in "${EXCLUDE_ARGUMENT[@]}; do
398:             printf '%s\n' "$exclude" >> "$EXCLUDEFILE_OPTIONAL"
399:         done
400:     else
401:         printf '\n' > "$EXCLUDEFILE_OPTIONAL"
402:     fi
403:
404:     TITLE='Back-up terugzetten'
405:     TEXT='Controleer beschikbare schijfruimte (kan even duren)...'
406:     if $OPTION_GUI; then
407:         # Met "|& zenity --progress" worden globale variabelen uit
408:         # aangeroepen functies niet doorgegeven, vandaar de
```

```
409:      # 'process substitution' met "> >(zenity...)".
410:      controleer_space > >(
411:          zenity --progress          \
412:                --pulsate           \
413:                --auto-close        \
414:                --no-cancel         \
415:                --width 600         \
416:                --height 50         \
417:                --title "$TITLE"    \
418:                --text "$TEXT"      2> >($LOGCMD)
419:      )
420:  else
421:      info "$TEXT"
422:      controleer_space
423:  fi
424:
425:  determine_file_size
426:
427:  if $ARGUMENT_SELECTION; then
428:      select=${SELECTION[*]}
429:  else
430:      select="alles"
431:  fi
432:  if [[ $TARGET = '/' ]]; then
433:      target="originele plek"
434:  else
435:      target=$TARGET
436:  fi
437:  size_num=${BACKUPSIZE_HUMAN%?}
438:  size_unit=${BACKUPSIZE_HUMAN: -1}
439:  TEXT="Selectie $select uit backup $(basename "$BACKUP") \
440: [$size_num ${size_unit}iB] naar $target terugzetten..."
441:  check_on_ac_power
442:  restore_backup
443: }
444:
445:
446: controleer_space() {
447:     local filesys=''
448:     local free_human=''
449:     local -i free_bytes=0
450:     local -i need_bytes=0
451:     local mounted=''
452:     local need_human=''
453:
454:     need_bytes=$(
455:         du --apparent-size          \
456:            --block-size=1           \
457:            --exclude-from="$EXCLUDEFILE_DEFAULT" \
458:            --exclude-from="$EXCLUDEFILE_OPTIONAL" \
459:            --summarize              \
460:            --total                  \
461:            "$BACKUP"                |
462:         awk 'END{print $1}'
463:     )
464:     free_bytes=$(df --block-size=1 "$TARGET" | awk 'END{print $4}')
465:
466:     if [[ $need_bytes -gt $free_bytes ]]; then
467:         filesys=$(df "$TARGET" | awk 'END{print $1}')
468:         mounted=$(df "$TARGET" | awk 'END{print $6}')
469:         need_human=$(
470:             du --apparent-size      \
471:                --block-size=1      \
472:                --exclude-from="$EXCLUDEFILE_DEFAULT" \
473:                --exclude-from="$EXCLUDEFILE_OPTIONAL" \
474:                --human-readable    \
475:                --summarize         \
476:                --total             \

```

```

477:         "$BACKUP" |
478:         awk 'END{print $1}'
479:     )
480:     free_human=$(df --human-readable "$TARGET" | awk 'END{print $4}')
481:
482:     if [[ ${SELECTION[0]} = "$SELECTION_DEFAULT" ]]; then
483:         error "Te weinig schijfruimte op $mounted (bestandssysteem \
484: $filesystem)
485: voor het terugzetten van ALLES in $(basename "$BACKUP") naar $TARGET.
486: Overweeg het gebruik van optie SELECTIE.
487: $USAGELINE"
488:         exit $ERROR
489:     fi
490:     warning "Mogelijk te weinig schijfruimte (hangt af van de selectie)
491: op $mounted (bestandssysteem $filesystem)
492: voor het terugzetten van ${SELECTION[*]} in $(basename "$BACKUP") naar \
493: $TARGET).
494: De HELE back-up is ${need_human}iB groot, beschikbaar is ${free_human}iB."
495:     fi
496: }
497:
498:
499: determine_file_size() {
500:     BACKUPSIZE_HUMAN=$(
501:         du --apparent-size \
502:            --block-size=1 \
503:            --human-readable \
504:            "$BACKUP" |
505:         awk '{print $1}'
506:     )
507:
508:     BACKUPSIZE_MACHINE=$(
509:         du --apparent-size \
510:            --block-size=1 \
511:            "$BACKUP" |
512:         awk '{print $1}'
513:     )
514: }
515:
516:
517: restore_backup() {
518:     if $OPTION_GUI; then
519:         (
520:             pv --size="$BACKUPSIZE_MACHINE" \
521:                --numeric \
522:                < "$BACKUP" |
523:             tar --extract \
524:                 --directory="$TARGET" \
525:                 --exclude-from="$EXCLUDEFILE_DEFAULT" \
526:                 --exclude-from="$EXCLUDEFILE_OPTIONAL" \
527:                 --file=- \
528:                 --wildcards \
529:                 "${SELECTION[@]}" \
530:                 2> >($LOGCMD)
531:         ) 2>&1 |
532:         zenity --progress \
533:                --auto-close \
534:                --no-cancel \
535:                --time-remaining \
536:                --width 600 \
537:                --height 50 \
538:                --title "$TITLE" \
539:                --text "$TEXT" 2> >($LOGCMD) || TAR_RC=$?
540:     else
541:         info "$TEXT"
542:         pv --size="$BACKUPSIZE_MACHINE" \
543:            < "$BACKUP" |
544:         tar --extract \

```



```

545:      --directory="$TARGET"          \
546:      --exclude-from="$EXCLUDEFILE_DEFAULT" \
547:      --exclude-from="$EXCLUDEFILE_OPTIONAL" \
548:      --file=-                        \
549:      --wildcards                      \
550:      "${SELECTION[@]}"                2> >($LOGCMD) || TAR_RC=$?
551:  fi
552:
553:  TEXT='Gegevens uit tijdelijk geheugen wegschrijven (kan even duren)...'
554:  if $OPTION_GUI; then
555:      sync                               |
556:      zenity --progress                  \
557:            --pulsate                    \
558:            --auto-close                 \
559:            --no-cancel                  \
560:            --width 600                  \
561:            --height 50                  \
562:            --title "$TITLE"            \
563:            --text "$TEXT"              2> >($LOGCMD) || SYNC_RC=$?
564:  else
565:      info "$TEXT"
566:      sync || SYNC_RC=$?
567:  fi
568: }
569:
570:
571: term_script() {
572:   TEXT=''
573:   if [[ "$BACKUP" = /media/* ]]; then
574:     TEXT='
575:
576: Koppel zelf (veilig!) het back-up-medium af.'
577:   fi
578:   if [[ $STAR_RC -eq $SUCCESS && $SYNC_RC -eq $SUCCESS ]]; then
579:     info "De back-up is teruggezet.$TEXT"
580:     exit $SUCCESS
581:   else
582:     warning "De back-up is niet of slechts gedeeltelijk teruggezet.
583:
584: Opdracht om de log uit te lezen:
585:   ${BLUE}$LOGCMD_CHECK${NORMAL}
586: $TEXT"
587:     exit $WARNING
588:   fi
589: }
590:
591:
592: #####
593: # Main line
594: #####
595:
596: main() {
597:   init_script "$@"
598:   check_input "$@"
599:   process_input
600:   term_script
601: }
602:
603:
604: main "$@"
605:
606:
607: # EOF

```