

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Apps installeren.
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-install
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2009
13:
14: VERSION_NUMBER=59.04.01
15: VERSION_DATE=2021-10-01
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=true
23: readonly OPTIONS_SHORT=${OPTIONS_SHORT_COMMON}'ace:filrs'
24: readonly OPTIONS_LONG=${OPTIONS_LONG_COMMON}",apps,cat,edition:,files,info,list,\
25: remove,simulate"
26: readonly USAGE="Gebruik: $DISPLAY_NAME [-a|--apps] [-c|--cat] \
27: [-e|--edition=EDITIE] [-f|--files]
28:                 [-i|--info] [-l|--list] [-r|--remove] [-s|--simulate]
29:                 $OPTIONS_USAGE_COMMON
30:                 [APP...] [BESTAND...]"
31: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...] [APP...] [BESTAND...]"
32:
33: Apps installeren.
34:
35: Opties:
36:   -a --apps      toon lijst van te installeren apps
37:   -c --cat       toon inhoud van installatiebestanden
38:   -e --edition=EDITIE
39:                 gebruik opgegeven editie
40:   -f --files     toon lijst van beschikbare installatiebestanden
41:   -i --info      toon aanvullende instructies na het installeren van apps
42:   -l --list      maak applijst aan
43:   -r --remove    toon instructies voor het verwijderen van apps
44:   -s --simulate  geen actie, simuleer installatie, en toon opdracht die
45:                 uitgevoerd zou worden
46: $OPTIONS_HELP_COMMON
47:
48: Argumenten:
49:   APP           opgegeven apps installeren
50:   BESTAND       opgegeven installatiebestanden verwerken"
51:
52: DASHES=$(printf '%.0s=' {1..79}); readonly DASHES
53: readonly EDITION_DEFAULT=desktop
54: SPACES=$(printf '%.0s ' {1..79}); readonly SPACES
55: VERSION=$(lsb_release --release --short); readonly VERSION
56: readonly LIST_FILE=/usr/local/etc/${PROGRAM_NAME}-applist
57: CMDS_FILE=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.cmds"); readonly CMDS_FILE
58: TEXT_FILE=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.text"); readonly TEXT_FILE
59: TEMP_LIST=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.list"); readonly TEMP_LIST
60:
61: # Less-opties, overgenomen (en aangepast, zie 'man less', zoek PROMPTS)
62: # van:
63: # 1. systemctl en journalctl, zie bijv. 'man systemctl', zoek LESS
64: #    ("FRSXMK")
65: # 2. man, zie 'mam man', zoek LESS
66: readonly LESS_OPTIONS="--LONG-PROMPT --no-init --quit-if-one-screen \
67: --quit-on-intr --RAW-CONTROL-CHARS --prompt=M Tekstuitvoer $PROGRAM_NAME \
68: ?ltregel %lt?L/%L:.byte %bB?s/%s..?e (EINDE) :?pB %pB\%. .(druk h voor \
```

```
69: hulp of q voor stoppen) "
70:
71:
72: #####
73: # Global variables
74: #####
75:
76: declare -a APP_ARGUMENT=()
77: declare -a FILE_ARGUMENT=()
78: declare -a FILE_TO_PROCESS=()
79: declare ARGUMENT_APP=false
80: declare ARGUMENT_FILE=false
81: declare COMMANDS_FOUND=false
82: declare EDITION=$EDITION_DEFAULT
83: declare EDITION_ARGUMENT=''
84: declare EXECUTE_COMMANDS=true
85: declare -i MAXRC=0
86: declare INFO_FOUND=false
87: declare OPTION_APPS=false
88: declare OPTION_CAT=false
89: declare OPTION_EDITION=false
90: declare OPTION_FILES=false
91: declare OPTION_INFO=false
92: declare OPTION_LIST=false
93: declare OPTION_REMOVE=false
94: declare OPTION_SIMULATE=false
95: declare REMOVE_FOUND=false
96:
97: # Terminalattributen, zie 'man terminfo'. Gebruik ${<variabele-naam>}.
98: declare CARRIAGE_RETURN=''
99: declare BLINK=''
100: declare CURSOR_INVISIBLE=''
101: declare CURSOR_VISIBLE=''
102:
103:
104: #####
105: # Functions
106: #####
107:
108: check_input() {
109:     local -i app_arg_num=0
110:     local -i file_arg_num=0
111:     local -i getopt_rc=0
112:     local parsed=''
113:
114:     parsed=$(
115:         getopt --alternative \
116:                --options "$OPTIONS_SHORT" \
117:                --longoptions "$OPTIONS_LONG" \
118:                --name "$DISPLAY_NAME" \
119:                -- "$@"
120:     ) || getopt_rc=$?
121:     if [[ $getopt_rc -ne $SUCCESS ]]; then
122:         printf '%s\n' "$USAGELINE" >&2
123:         exit $ERROR
124:     fi
125:     eval set -- "$parsed"
126:     process_common_options "$@"
127:     STATUS_ERROR="[ ${RED}FOUT${NORMAL} ]"
128:     STATUS_SUCCESS="[ ${GREEN}GOED${NORMAL} ]"
129:
130:     while true; do
131:         case $1 in
132:             -a|--apps)
133:                 OPTION_APPS=true
134:                 shift
135:                 ;;
136:             -c|--cat)
```

```
137:         OPTION_CAT=true
138:         shift
139:         ;;
140:     -e|--edition)
141:         if $OPTION_EDITION; then
142:             TEXT="optie '$1' eenmaal opgeven"
143:             printf "$DISPLAY_NAME: %s\n%s\n" \
144:                 "$TEXT" \
145:                 "$USAGELINE" >&2
146:             exit $ERROR
147:         else
148:             OPTION_EDITION=true
149:             EDITION_ARGUMENT=$2
150:             EDITION=$EDITION_ARGUMENT
151:         fi
152:         shift 2
153:         ;;
154:     -f|--files)
155:         OPTION_FILES=true
156:         shift
157:         ;;
158:     -i|--info)
159:         OPTION_INFO=true
160:         EXECUTE_COMMANDS=false
161:         shift
162:         ;;
163:     -l|--list)
164:         OPTION_LIST=true
165:         EXECUTE_COMMANDS=false
166:         shift
167:         ;;
168:     -r|--remove)
169:         OPTION_REMOVE=true
170:         EXECUTE_COMMANDS=false
171:         shift
172:         ;;
173:     -s|--simulate)
174:         OPTION_SIMULATE=true
175:         EXECUTE_COMMANDS=false
176:         shift
177:         ;;
178:     --)
179:         shift
180:         break
181:         ;;
182:     *)
183:         shift
184:         ;;
185:     esac
186: done
187:
188: if ! $OPTION_GUI && [[ -t 1 ]]; then
189:     # Tekstuitvoer non-gui naar de terminal.
190:     CARRIAGE_RETURN=$(tput cr)
191:     BLINK=$(tput blink)
192:     CURSOR_INVISIBLE=$(tput civis)
193:     CURSOR_VISIBLE=$(tput cvvis)
194: fi
195: STATUS_BUSY=" [${BLINK}${BOLD}WERK${NORMAL} ] "
196:
197: # Verwerk argumenten.
198: while [[ "$*" ]]; do
199:     if [[ "$(basename "$1")" = $PROGRAM_NAME-*.sh ]]; then
200:         ARGUMENT_FILE=true
201:         FILE_ARGUMENT[${file_arg_num}]=$1
202:         ((++file_arg_num))
203:         shift
204:     else
```

```
205:         ARGUMENT_APP=true
206:         APP_ARGUMENT[$app_arg_num]=$1
207:         ((++app_arg_num))
208:         shift
209:     fi
210: done
211: }
212:
213:
214: process_input() {
215:     local file=''
216:     local file_notfound=false
217:     local -i app_seq_num=0
218:     local -i app_tot_num=0
219:     local -i file_num=0
220:     local skip_line=false
221:
222:     if $OPTION_LIST; then
223:         check_user
224:         process_option_list
225:         exit $SUCCESS
226:     fi
227:     if $ARGUMENT_APP; then
228:         process_argument_app
229:     fi
230:     if $ARGUMENT_FILE; then
231:         process_argument_file
232:     fi
233:
234:     # Als argument APP of FILE is opgegeven, geen standaardbestanden verwerken.
235:     if ! ($ARGUMENT_APP || $ARGUMENT_FILE); then
236:         process_dfltfiles
237:     fi
238:
239:     # Verwerk opties.
240:     if $OPTION_APPS; then
241:         process_option_apps
242:         exit $SUCCESS
243:     elif $OPTION_FILES; then
244:         process_option_files
245:         exit $SUCCESS
246:     fi
247:
248:     # Ergens hiervoor ging het niet helemaal goed.
249:     if $file_notfound; then
250:         exit $SUCCESS
251:     fi
252:
253:     # Optie cat is afhankelijk van argumenten APP en FILE.
254:     if $OPTION_CAT; then
255:         process_option_cat
256:         exit $SUCCESS
257:     fi
258:
259:     # We gaan opdrachten uitvoeren, mogen we dat wel?
260:     if $EXECUTE_COMMANDS; then
261:         check_user
262:         check_on_ac_power
263:     fi
264:
265:     # We gaan opdrachten uitvoeren of simuleren.
266:     for file in "${FILE_TO_PROCESS[@]}; do
267:         cat "$file" >> "$CMDS_FILE"
268:     done
269:
270:     # 'if ! grep --regexp='^#2'...; then...' werkt niet; grep geeft 0
271:     # terug als niets gevonden, is OK, maar dan ook rc=1, is niet OK.
272:     app_tot_num=$(
```

```
273:     grep --word-regexp --regexp='^#2' --count "$CMDS_FILE" || true
274:   )
275:
276:   if $EXECUTE_COMMANDS && $OPTION_GUI; then
277:     TITLE='Apps installeren'
278:     TEXT=''
279:     # Met "|& zenity --progress" worden globale variabelen uit
280:     # aangeroepen functies niet doorgegeven, vandaar de
281:     # 'process substitution' met "> >(zenity...)".
282:     process_file > >(
283:         zenity --progress          \
284:                --pulsate          \
285:                --auto-close       \
286:                --no-cancel        \
287:                --width            700 \
288:                --height           50  \
289:                --title            "$TITLE" \
290:                --text              "$TEXT"  2> >($LOGCMD)
291:     )
292:   else
293:     process_file
294:   fi
295: }
296:
297:
298: process_option_list() {
299:   local text0='Opdrachten worden verwerkt'
300:   local text1="[1/3] Bepaal welke pakketten handmatig zijn geïnstalleerd"
301:   local text3="[2/3] Bepaal welke snaps zijn geïnstalleerd"
302:   local text2="[3/3] Bepaal welke pakketbronnen zijn toegevoegd"
303:
304:   TITLE='Applijst maken'
305:   if $OPTION_GUI; then
306:     (
307:       printf '%s\n' "$text1"
308:       create_list_manual_installed #1
309:       printf '%s\n' "$text2"
310:       create_list_snaps #2
311:       printf '%s\n' "$text3"
312:       create_list_added_repos #3
313:     ) |&
314:     zenity --progress          \
315:            --pulsate          \
316:            --auto-close       \
317:            --no-cancel        \
318:            --width            600 \
319:            --height           50  \
320:            --title            "$TITLE" \
321:            --text              "$text0"  2> >($LOGCMD)
322:   else
323:     create_list_manual_installed #1
324:     create_list_snaps #2
325:     create_list_added_repos #3
326:   fi
327:
328:   info "
329: De applijst is vastgelegd in bestand $LIST_FILE.
330: Dit bestand bevatten de eerder geïnstalleerde apps.
331: Dit bestand kan worden gebruikt voor het nalopen van de installatie."
332: }
333:
334:
335: create_list_manual_installed() { #1
336:   local header="1. Deze pakketten zijn handmatig geïnstalleerd via \
337: Software of
338: met een opdracht zoals 'sudo apt-get --install <pakket|bestand>':"
339:   local footer=''
340:   local -i number_of_lines=0
```

```
341: local plural='geinstalleerde pakketten'
342:
343: printf '%s\n' "$text1..."
344: printf '%b\n' "$header\n$DASHES" > "$TEMP_LIST"
345:
346: apt list --manual-installed 2> >($LOGCMD) >> "$TEMP_LIST"
347:
348: number_of_lines=$(wc --lines < "$TEMP_LIST")
349: number_of_lines=$((number_of_lines - 4))
350: if [[ $number_of_lines -eq 1 ]]; then
351:     plural='geinstalleerd pakket'
352: fi
353: footer="Dit overzicht bevat $number_of_lines handmatig $plural."
354: printf '%b\n' "$DASHES\n$footer\n" >> "$TEMP_LIST"
355:
356: cat "$TEMP_LIST" > "$LIST_FILE"
357: }
358:
359:
360: create_list_snaps() { #2
361:     local header="2. Deze snaps zijn geinstalleerd met '[sudo] install \
362: <snap>' of met Software:"
363:     local footer=''
364:     local -i number_of_lines=0
365:     local plural='snaps'
366:
367:     printf '%s\n' "$text3..."
368:     printf '%b\n' "$header\n$DASHES" > "$TEMP_LIST"
369:
370:     if ! snap list 2> /dev/null >> "$TEMP_LIST"; then
371:         TEXT='snap is niet geinstalleerd op dit systeem'
372:         printf '%s\n' "$TEXT" >> "$TEMP_LIST"
373:     fi
374:
375:     number_of_lines=$(wc --lines < "$TEMP_LIST")
376:     number_of_lines=$((number_of_lines - 3))
377:     if [[ $number_of_lines -eq 1 ]]; then
378:         plural='snap'
379:     fi
380:     footer="Dit overzicht bevat $number_of_lines $plural."
381:     printf '%b\n' "$DASHES\n$footer\n" >> "$TEMP_LIST"
382:
383:     cat "$TEMP_LIST" >> "$LIST_FILE"
384: }
385:
386:
387: create_list_added_repos() { #3
388:     local header="3. Deze pakketbronnen zijn toegevoegd met de \
389: installatie van een pakket of
390: met de opdracht 'sudo add-apt-repository ppa:<ppa-gebruiker>/<ppa-naam>':"
391:     local footer=''
392:     local -i number_of_lines=0
393:     local plural='pakketbronnen'
394:     local search=''
395:
396:     printf '%s\n' "$text2..."
397:     printf '%b\n' "$header\n$DASHES" > "$TEMP_LIST"
398:
399:     search=/etc/apt/sources.list
400:     cd $search.d || exit $ERROR
401:     if ! grep --recursive \
402:         --no-filename \
403:         --regexp='^deb ' \
404:         --recursive \
405:         /etc/apt/sources.list \
406:         /etc/apt/sources.list.d |
407:         grep --invert-match \
408:         --regexp='ubuntu.com' |
```

```
409:         sort      --unique          >> "$TEMP_LIST"; then
410:         true
411:     fi
412:
413:     number_of_lines=$(wc --lines < "$TEMP_LIST")
414:     number_of_lines=$((number_of_lines - 3))
415:     if [[ $number_of_lines -eq 1 ]]; then
416:         plural='pakketbron'
417:     fi
418:     footer="Dit overzicht bevat $number_of_lines toegevoegde $plural."
419:     printf '%b\n' "$DASHES\n$header" >> "$TEMP_LIST"
420:
421:     cat "$TEMP_LIST" >> "$LIST_FILE"
422: }
423:
424:
425: process_argument_app() {
426:     local      app=''
427:     local      file=''
428:     local  -a  files=()
429:     local      search=''
430:     local      cmds_file=''
431:
432:     cmds_file=$(mktemp -t "$PROGRAM_NAME-XXXXXXXXXX.cmds")
433:     # Install zoekt naar apps in alle kz-install-<distro>-<versie>-bestanden
434:     # en stopt bij de eerstgevonden (installatie is computer-onafhankelijk),
435:     # setup zoekt naar apps in kz-install-<distro>-<versie>-<edition>.sh en
436:     # kz-install-<distro>-<versie>-<edition>-<computernaam>.sh.
437:     search=$PROGRAM_NAME-$DISTRO-$VERSION
438:     for file in "$PROGRAM_PATH/$search"*.sh; do
439:         files+=("$file")
440:     done
441:     for app in "${APP_ARGUMENT[@]}; do
442:         file=$(
443:             if ! grep          --files-with-matches  \
444:                 --line-regexp \
445:                 --regexp="#1 $app" \
446:                 "${files[@]}" \
447:                 2> /dev/null \
448:                 head --lines=1; then
449:                 printf '\n'
450:             fi
451:         )
452:         if [[ $file ]]; then
453:             process_argument_app_file
454:         else
455:             info "App '$app' niet gevonden."
456:             file_notfound=true
457:         fi
458:     done
459:     if $file_notfound; then
460:         info "Typ '$DISPLAY_NAME --apps' voor beschikbare apps."
461:     else
462:         if ! grep --quiet --line-regexp --regexp='# EOF' "$cmds_file"
463:         then
464:             printf '%s\n' '# EOF' >> "$cmds_file"
465:         fi
466:         FILE_TO_PROCESS[$file_num]=$cmds_file
467:         ((++file_num))
468:     fi
469: }
470:
471:
472: process_argument_app_file() {
473:     local app_found=false
474:     local description_found=false
475:     local record=''
476:
```

```
477: while read -r record; do
478:     case $record in
479:         '#1 '* )
480:             if [[ $record = '#1 '$app ]]; then
481:                 if $app_found; then
482:                     info "Dubbele app-naam $app."
483:                 else
484:                     # Gezochte app-naam.
485:                     # Regels doorschrijven naar tijdelijk bestand tot
486:                     # eerstvolgende app-naam of beschrijving (verplicht),
487:                     # of tot EOF.
488:                     app_found=true
489:                 fi
490:             elif $app_found; then
491:                 # Volgende app-naam.
492:                 break
493:             fi
494:             ;;
495:         '#2'* )
496:             # Beschrijving.
497:             if $description_found; then
498:                 # Volgende beschrijving.
499:                 break
500:             elif $app_found; then
501:                 printf '%b\n' "$record" >> "$cmds_file"
502:                 description_found=true
503:             fi
504:             ;;
505:         *)
506:             # Overige regels.
507:             if $app_found; then
508:                 printf '%s\n' "$record" >> "$cmds_file"
509:             fi
510:             ;;
511:     esac
512: done < "$file"
513: }
514:
515:
516: process_argument_file() {
517:     local file=''
518:
519:     for file in "${FILE_ARGUMENT[@]}; do
520:         if [[ -e $file ]]; then
521:             FILE_TO_PROCESS[$file_num]=$file
522:             ((++file_num))
523:         else
524:             info "Bestand $file niet gevonden."
525:             file_notfound=true
526:         fi
527:     done
528:     if $file_notfound; then
529:         info "Typ '$DISPLAY_NAME --files' voor beschikbare \
530: installatiebestanden."
531:     fi
532: }
533:
534:
535: process_dfltfiles() {
536:     local dfltfile_1=''
537:     local dfltfile_2=''
538:
539:     dfltfile_1=$PROGRAM_PATH/$PROGRAM_NAME-$DISTRO-$VERSION-$EDITION.sh
540:     dfltfile_2=$PROGRAM_PATH/$PROGRAM_NAME-$DISTRO-$VERSION-$EDITION-\
541: $HOSTNAME.sh
542:
543:     if [[ -e $dfltfile_1 ]]; then
544:         FILE_TO_PROCESS[$file_num]=$dfltfile_1
```



```

545:     ((++file_num))
546:   else
547:     file_notfound=true
548:   fi
549:   # Niet verplicht aanwezig.
550:   if [[ -e $dfltfile_2 ]]; then
551:     FILE_TO_PROCESS[$file_num]=$dfltfile_2
552:     ((++file_num))
553:   fi
554:   if $file_notfound; then
555:     info "Geen installatiebestanden gevonden voor $DISTRO $VERSION \
556: $EDITION."
557:     exit $WARNING
558:   fi
559: }
560:
561:
562: process_option_apps() {
563:   local -a files=()
564:   local search=''
565:
566:   # Install zoekt naar apps in alle kz-install-<distro>-<versie>-bestanden
567:   # en stopt bij de eerstgevonden (installatie is computer-onafhankelijk),
568:   # setup zoekt naar apps in kz-install-<distro>-<versie>-<edition>.sh en
569:   # kz-install-<distro>-<versie>-<edition>-<computernaam>.sh.
570:   search=$PROGRAM_NAME-$DISTRO-$VERSION
571:   for file in "$PROGRAM_PATH/$search"*.sh; do
572:     files+=("$file")
573:   done
574:   printf '%s\n' "De volgende apps zijn beschikbaar:
575:
576: NUMMER APP
577: $(
578:   if ! grep --no-messages \
579:     --regexp='^#1 ' \
580:     "${files[@]}" \
581:     awk '{print $2}' \
582:     sort --unique \
583:     nl --number-width=6 \
584:     --number-format=rn \
585:     --number-separator=' ' \
586:     --body-numbering=a; then
587:     printf '%s\n' " 0 Geen apps gevonden voor $DISTRO $VERSION."
588:   else
589:     printf '%s\n' "
590: Om APPs te bekijken voer uit:
591:   ${BLUE}$DISPLAY_NAME --sim APP...${NORMAL}
592:   ${BLUE}$DISPLAY_NAME --cat APP...${NORMAL}
593: Om APPs te installeren voer uit:
594:   ${BLUE}$DISPLAY_NAME APP...${NORMAL}"
595:   fi
596: )"
597: }
598:
599:
600: process_option_files() {
601:   printf '%s\n' "De volgende installatiebestanden zijn beschikbaar:
602:
603: NUMMER BESTAND
604: $(
605:   if ! find "${FILE_TO_PROCESS[@]}" \
606:     2> /dev/null \
607:     nl --number-width=6 \
608:     --number-format=rn \
609:     --number-separator=' ' \
610:     --body-numbering=a; then
611:     printf '%s\n' " 0 Geen installatiebestanden gevonden voor $DISTRO \
612: $VERSION $EDITION."

```

```

613: Typ 'man $DISPLAY_NAME' voor meer informatie."
614:     else
615:         printf '%s\n' "
616: Om BESTANDen te bekijken voer uit:
617:     ${BLUE}$DISPLAY_NAME --sim BESTAND...${NORMAL}
618:     ${BLUE}$DISPLAY_NAME --cat BESTAND...${NORMAL}
619: Om BESTANDen te gebruiken voer uit:
620:     ${BLUE}$DISPLAY_NAME BESTAND...${NORMAL}"
621:     fi
622: )"
623: }
624:
625:
626: process_option_cat() {
627:     for file in "${FILE_TO_PROCESS[@]}; do
628:         {
629:             printf '%s\n' "${BLUE}# $file${NORMAL}"
630:             cat "$file"
631:             printf '\n'
632:         } >> "$CMDS_FILE"
633:     done
634:     if $OPTION_GUI; then
635:         TITLE='Inhoud installatiebestanden'
636:         if ! zenity --text-info \
637:             --width 850 \
638:             --height 400 \
639:             --title "$TITLE" \
640:             --filename "$CMDS_FILE" \
641:             --font 'Ubuntu Mono 12' \
642:             --ok-label 'OkÃ©' \
643:             --cancel-label 'Annuleren' 2> >($LOGCMD); then
644:             true
645:         fi
646:     else
647:         less "$LESS_OPTIONS" "$CMDS_FILE"
648:     fi
649: }
650:
651:
652: process_file() {
653:     local description=''
654:     local -i cmd_seq_num=0
655:     local prev_recordtype=''
656:     local record=''
657:     local recordtype=''
658:     local write_description_line=true
659:
660:     if $EXECUTE_COMMANDS; then
661:         printf '%b' "${CURSOR_INVISABLE}"
662:     fi
663:     $skip_line && printf '\n'; skip_line=true
664:
665:     while read -r record; do
666:         recordtype=${record:0:2}
667:         case $recordtype in
668:             ''|#1')
669:                 # Lege regel of lege app-naam.
670:                 continue
671:                 ;;
672:             '#2')
673:                 process_description "$record"
674:                 ;;
675:             '#3')
676:                 process_install_instruction "$record"
677:                 ;;
678:             '#4')
679:                 process_remove_instruction "$record"
680:                 ;;

```

```
681:         '#')
682:         # Commentaarregel.
683:         continue
684:         ;;
685:     *)
686:         process_command "$record"
687:         ;;
688:     esac
689:     prev_recordtype=$recordtype
690: done < "$CMDS_FILE"
691:
692: if $EXECUTE_COMMANDS; then
693:     printf '%b' "${CURSOR_VISABLE}"
694: fi
695: }
696:
697:
698: process_description() {
699:     local record=${1:-record?}
700:
701:     description="${record:3}"
702:     if ! $OPTION_SIMULATE && ! $EXECUTE_COMMANDS; then
703:         return $SUCCESS
704:     fi
705:
706:     write_description_line=true
707:     ((++app_seq_num))
708:     cmd_seq_num=0
709: }
710:
711:
712: process_install_instruction() {
713:     local record=${1:-record?}
714:     local filler=''
715:     local -i len=0
716:
717:     INFO_FOUND=true
718:     if ! $OPTION_INFO; then
719:         return $SUCCESS
720:     fi
721:     filler=${SPACES:0:8}
722:     if [[ $prev_recordtype = '#3' ]]; then
723:         printf '%s%s\n\n' "$filler" "${record:3}" >> "$TEXT_FILE"
724:     else
725:         ((++app_seq_num))
726:         len=${#description}
727:         # ?? wordt in term_script vervangen door aantal te installeren apps.
728:         printf "\n[%2d/??] %s\n%s%.${len}s\n\n%s%s\n\n" \
729:             "$app_seq_num" "$description" \
730:             "$filler" "$DASHES" \
731:             "$filler" "${record:3}" >> "$TEXT_FILE"
732:     fi
733: }
734:
735:
736: process_remove_instruction() {
737:     local record=${1:-record?}
738:     local filler=''
739:     local -i len=0
740:
741:     REMOVE_FOUND=true
742:     if ! $OPTION_REMOVE; then
743:         return $SUCCESS
744:     fi
745:     filler=${SPACES:0:8}
746:     if [[ $prev_recordtype = '#4' ]]; then
747:         printf '%s%s\n\n' "$filler" "${record:3}" >> "$TEXT_FILE"
748:     else
```

```

749:      ((++app_seq_num))
750:      len=${#description}
751:      # ?? wordt in term_script vervangen door aantal te te verwijderen apps.
752:      printf "\n[%2d/??] %s\n%s%.${len}s\n\n%s%\n\n" \
753:             "$app_seq_num" "$description" \
754:             "$filler" "$DASHES" \
755:             "$filler" "${record:3}" >> "$TEXT_FILE"
756:   fi
757: }
758:
759:
760: process_command() {
761:     local cmd=${1:-cmd?}
762:     local -i cmd_rc=0
763:     local description_line=''
764:     local -i desc_maxlen=1000
765:     local desc_printf=''
766:     local status=''
767:
768:     COMMANDS_FOUND=true
769:     ((++cmd_seq_num))
770:
771:     if $OPTION_SIMULATE; then
772:         if $write_description_line; then
773:             printf "[%3d/%-3d] %s\n\n" \
774:                    "$app_seq_num" \
775:                    "$app_tot_num" \
776:                    "$description" >> "$TEXT_FILE"
777:             write_description_line=false
778:         fi
779:         printf '%9s [%2d] %s\n\n' \
780:                " " \
781:                "$cmd_seq_num" \
782:                "${BLUE}$record${NORMAL}" >> "$TEXT_FILE"
783:     elif $EXECUTE_COMMANDS; then
784:         desc_printf=$description
785:         if $OPTION_GUI; then
786:             TEXT="#[$app_seq_num/$app_tot_num] $description\n\n$record"
787:             printf '%s\n' "$TEXT"
788:         else
789:             if [[ -t 1 ]]; then
790:                 # Afbreken in Terminalvenster moet i.v.m. updaten status.
791:                 # <----- tput cols ----->
792:                 # '[WERK] [ 1/1 ] [ 1] Te lange descr...' <=== op 1 regel
793:                 # <----- 22 ----->
794:                 desc_maxlen=$((tput cols) - 22)
795:                 if [[ ${#desc_printf} -gt $desc_maxlen ]]; then
796:                     desc_printf=${desc_printf:0:($desc_maxlen - 3)}'...'
797:                 fi
798:             fi
799:         fi
800:         printf -v description_line \
801:                "[%3d/%-3d] [%2d] %-${desc_maxlen}.${desc_maxlen}s" \
802:                "$app_seq_num" \
803:                "$app_tot_num" \
804:                "$cmd_seq_num" \
805:                "$desc_printf"
806:         execute_command "$record"
807:     fi
808: }
809:
810:
811: execute_command() {
812:     local cmd=${1:-cmd?}
813:     local -i cmd_rc=0
814:     local status=''
815:
816:     check_dpkg

```

```
817:     if ! $OPTION_GUI; then
818:         if [[ $cmd_seq_num -eq 1 ]]; then
819:             printf "$STATUS_BUSY %s${CARRIAGE_RETURN}" "$description_line"
820:         else
821:             printf "$STATUS_BUSY ${SPACES:0:9}%s${CARRIAGE_RETURN}" \
822:                 "${description_line:9}"
823:         fi
824:     fi
825:
826:     log "$description_line"
827:     log "${BLUE}$cmd${NORMAL}"
828:     eval "$cmd" |& $LOGCMD || cmd_rc=$?
829:     if [[ $cmd_rc -gt $MAXRC ]]; then
830:         MAXRC=$cmd_rc
831:     fi
832:     if [[ $cmd_rc -eq $SUCCESS ]]; then
833:         status="$STATUS_SUCCESS"
834:     else
835:         status="$STATUS_ERROR"
836:     fi
837:     log "$status rc=$cmd_rc, maxrc=$MAXRC"
838:     printf '%s\n' "$status"
839: }
840:
841: check_dpkg() {
842:     local -i wait_for_apt=5
843:
844:     if ls /snap/core/*/var/cache/debconf/config.dat &> /dev/null; then
845:         while sudo fuser \
846:             /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
847:             /var/cache/debconf/config.dat \
848:             /snap/core/*/var/cache/debconf/config.dat \
849:             &> /dev/null; do
850:             log 'Wacht tot andere pakketbeheerder klaar is...'
851:             sleep $wait_for_apt
852:         done
853:     else
854:         while sudo fuser \
855:             /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
856:             /var/cache/debconf/config.dat \
857:             &> /dev/null; do
858:             log 'Wacht tot andere pakketbeheerder klaar is...'
859:             sleep $wait_for_apt
860:         done
861:     fi
862: }
863:
864:
865: term_script() {
866:     local -i app_num=0
867:
868:     if $OPTION_SIMULATE; then
869:         if $OPTION_GUI; then
870:             TITLE='Installatie-opdrachten'
871:             if ! zenity --text-info \
872:                 --width 1200 \
873:                 --height 400 \
874:                 --title "$TITLE" \
875:                 --filename "$TEXT_FILE" \
876:                 --font 'Ubuntu Mono 12' \
877:                 --ok-label 'OkÃ©' \
878:                 --cancel-label 'Annuleren' \
879:                 2> >($LOGCMD); then
880:                 true
881:             fi
882:         else
883:             less "$LESS_OPTIONS" "$TEXT_FILE"
884:         fi
885:     fi
886: }
```

```
885:     elif $OPTION_INFO && $INFO_FOUND; then
886:         printf '\n' >> "$TEXT_FILE"
887:         # Vervang ?? door aantal te installeren apps.
888:         app_num=$(grep --regexp='\??]' ' --count "$TEXT_FILE")
889:         sed -i "s|/??] |/${(printf %-2d "$app_num")} |g" "$TEXT_FILE"
890:         if $OPTION_GUI; then
891:             TITLE='Installatie-instructies'
892:             if ! zenity --text-info \
893:                 --width 850 \
894:                 --height 400 \
895:                 --title "$TITLE" \
896:                 --filename "$TEXT_FILE" \
897:                 --font 'Ubuntu Mono 12' \
898:                 --ok-label 'OkÃ©' \
899:                 --cancel-label 'Annuleren' 2> >($LOGCMD); then
900:                 true
901:             fi
902:         else
903:             less "$LESS_OPTIONS" "$TEXT_FILE"
904:         fi
905:     elif $OPTION_INFO && ! $INFO_FOUND; then
906:         info 'Geen installatie-instructies.'
907:     elif $OPTION_REMOVE && $REMOVE_FOUND; then
908:         printf '\n' >> "$TEXT_FILE"
909:         # Vervang ?? door aantal te verwijderen apps.
910:         app_num=$(grep --regexp='\??]' ' --count "$TEXT_FILE")
911:         sed -i "s|/??] |/${(printf %-2d "$app_num")} |g" "$TEXT_FILE"
912:         # Vervang 'installeren' door 'verwijderen' (even lang).
913:         sed --in-place --expression='s/installeren/verwijderen/g' "$TEXT_FILE"
914:         if $OPTION_GUI; then
915:             TITLE='Verwijder-instructies'
916:             if ! zenity --text-info \
917:                 --width 850 \
918:                 --height 400 \
919:                 --title "$TITLE" \
920:                 --filename "$TEXT_FILE" \
921:                 --font 'Ubuntu Mono 12' \
922:                 --ok-label 'OkÃ©' \
923:                 --cancel-label 'Annuleren' 2> >($LOGCMD); then
924:                 true
925:             fi
926:         else
927:             less "$LESS_OPTIONS" "$TEXT_FILE"
928:         fi
929:     elif $OPTION_REMOVE && ! $REMOVE_FOUND; then
930:         info 'Geen verwijder-instructies.'
931:     elif ! $COMMANDS_FOUND; then
932:         info 'Geen opdrachten.'
933:     fi
934:
935:     if $EXECUTE_COMMANDS && [[ $MAXRC -eq $SUCCESS ]] && $INFO_FOUND &&
936:         ! $OPTION_GUI; then
937:         info "
938: Er zijn aanvullende instructies!
939: Gebruik de opdracht:
940: ${BLUE}$DISPLAY_NAME --info ${APP_ARGUMENT[*]}${NORMAL}"
941:     fi
942:
943:     if $EXECUTE_COMMANDS && $OPTION_GUI; then
944:         if [[ $MAXRC -ne $SUCCESS ]]; then
945:             warning 'Installatie is geÃ«indigd met een waarschuwing.'
946:             exit $WARNING
947:         else
948:             info 'Installatie is voltooid.'
949:             exit $SUCCESS
950:         fi
951:     fi
952: }
```

```
953:
954:
955: #####
956: # Main line
957: #####
958:
959: main() {
960:     init_script "$@"
961:     check_input "$@"
962:     process_input
963:     term_script
964: }
965:
966:
967: main "$@"
968:
969:
970: # EOF
```