

```
1  #!/bin/bash
2  #####
3  # Bestand:  bbackup.sh                                     #
4  # Doel:    Script voor het maken van een back-up van een back-up.      #
5  # Gebruik: [sudo] ./bbackup.sh [opties]                          #
6  # Gebruikt: script-common.sh   (algemene variabelen en functies)      #
7  # Auteur:  Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
8  # -----#
9  # Auteursrecht © 2010-2014 Karel Zimmer.                          #
10 #                                                                 #
11 # Dit programma is vrije software: u mag het herdistribueren en/of wijzigen #
12 # onder de voorwaarden van de GNU Algemene Publieke Licentie zoals      #
13 # gepubliceerd door de Free Software Foundation, onder versie 3 van de  #
14 # Licentie of (naar Uw keuze) elke latere versie.                      #
15 #                                                                 #
16 # Dit programma is gedistribueerd in de hoop dat het nuttig zal zijn maar #
17 # ZONDER ENIGE GARANTIE; zelfs zonder de impliciete garanties die      #
18 # GEBRUIKELIJK ZIJN IN DE HANDEL of voor BRUIKBAARHEID VOOR EEN SPECIFIEK #
19 # DOEL. Zie de GNU Algemene Publieke Licentie voor meer details.       #
20 #                                                                 #
21 # U hoort een kopie van de GNU Algemene Publieke Licentie te hebben     #
22 # ontvangen samen met dit programma. Als dat niet het geval is, zie    #
23 # http://www.gnu.org/licenses/. #
24 # -----#
25 # Versies:  1.0.0    2010-08-02  Eerste versie.                      #
26 #           2.0.0    2010-08-27  Meer berichten naar de gebruiker.   #
27 #           3.0.0    2010-09-22  Meerdere back-uplocaties mogelijk.  #
28 #           4.0.0    2010-10-15  Meer generaties en betere toelichting.#
29 #           5.0.0    2012-02-18  Schijven afkoppelen wanneer klaar.   #
30 #           6.0.0    2013-11-28  Controle op schijfruimte toegevoegd. #
31 #           7.0.0    2014-10-05  Configuratiebestand en optie reset   #
32 #                                     toegevoegd.                      #
33 #           8.0.0    2014-10-23  Configuratiebestand en optie reset   #
34 #                                     verwijderd, harddisks opzoeken en verwerken.#
35 #####
36 readonly VERSION_NUMBER=8.1.0
37 readonly RELEASE_DATE=2014-11-05
38
39 #####
40 # Instellingen.                                               #
41 #####
42
43 #-----#
44 # Algemene instellingen                                       #
45 # -----#
46 # Lees de algemene variabelen en functies in.                 #
47 # -----#
48 source script-common.sh 2> /dev/null || {
49     echo "Het algemeen scriptbestand 'script-common.sh'"      >&2
50     echo 'is niet gevonden of bevat fouten.'                 >&2
51     echo 'Dit algemeen scriptbestand wordt gewoonlijk door'
52     echo "script getscripts (gs) gedownload en in '/tmp' geplaatst."
53     echo "Is 'cd /tmp; wget karelzimmer.nl/gs; bash gs' uitgevoerd?"
54     echo 'Voor scripts zie http://karelzimmer.nl, klik op Linux.'
55     exit 1
56 }
57
58 #-----#
59 # Globale constanten                                          #
60 # -----#
61 #----- ALGEMEEN -----#
62 readonly SCRIPT_NEEDS_SUDO=true          # Uitvoeren als beheerder
63 readonly FIRST_COPYRIGHT_YEAR=2010       # Eerste auteursrechtjaar
64
65 #----- FROM -----#
66 readonly STORAGEDIR=/var/scripts/backup # Back-upbmap, bron. STORAGEDIR ivm
```

```

67                                     # gebruik in backup en script-common.
68 readonly BACKUPFILE=backup*          # Back-upbestand
69 readonly LOGFILE_TO_COPY=$(ls $STORAGEDIR/$BACKUPFILE.log 2>> "$LOG")
70 readonly LOGFILE_TO_COPY_BASE=$(basename "$LOGFILE_TO_COPY")
71                                     # Te kopiëren logboek
72 readonly TARFILE_TO_COPY=$(ls $STORAGEDIR/$BACKUPFILE.tar 2>> "$LOG")
73 readonly TARFILE_TO_COPY_BASE=$(basename "$TARFILE_TO_COPY")
74                                     # Te kopiëren backup
75 readonly FILES_TO_COPY="$LOGFILE_TO_COPY $TARFILE_TO_COPY"
76                                     # Te kopiëren bestanden
77 readonly NEED_BYTES=$(du $FILES_TO_COPY \
78 --summarize \
79 --total \
80 --apparent-size \
81 --block-size=1 \
82 2>> "$LOG" |
83 tail --lines=1 |
84 awk '{print $1}')
85                                     # Back-upgrootte in bytes
86 readonly NEED_HUMAN=$(du $FILES_TO_COPY \
87 --summarize \
88 --total \
89 --apparent-size \
90 --human-readable \
91 2>> "$LOG" |
92 tail --lines=1 |
93 awk '{print $1}')
94                                     # Back-upgrootte in leesbare vorm
95
96 #----- TO -----
97 readonly NUM_TO_FILES_OKE='0, 2, 4, 6, of 8'
98                                     # Correct aantal BACKUPDIR bestanden
99 readonly NUM_TO_FILES_MAX=${NUM_TO_FILES_OKE:${#NUM_TO_FILES_OKE} - 1:1}
100                                     # Maximum aantal BACKUPDIR bestanden
101 readonly NUM_GEN=$(( NUM_TO_FILES_MAX / 2 ))
102                                     # Aantal bewaarde generaties
103 readonly -A HARDDISK_STATUS_TXT=(
104 [PRESENT]='- aanwezig'
105 [NO-BACKUPS-FOLDER]="- map 'Backups' niet aanwezig"
106 [NO-COMPNAME-FOLDER]="- map '$COMPNAME' niet aanwezig"
107 )
108                                     # Harddisk status-teksten
109
110 #----- FOUTCODES -----
111 readonly E_STORAGEDIR_NOT_FOUND=64 # Foutcodes
112 readonly E_STORAGEDIR_NUMBER_OF_FILES_ERROR=65
113 readonly E_BACKUPDIR_NUMBER_OF_FILES_ERROR=66
114 readonly E_SIZES_ARE_DIFFERENT=67
115 #-----#
116 # Globale variabelen #
117 #-----#
118 declare BUPFILES_COPIED=0 # Aantal gekopieerde BUPFILES
119 declare BUPFILES_DELETED=0 # Aantal verwijderde BUPFILES
120 declare MAX_BACKUPDIR_LEN=0 # Maximale lengte string BACKUPDIR
121
122 declare DISKSPACE_OK=true # Voldoende back-up ruimte
123 declare NO_HARDDISK_FOUND=true # Geen enkele harddisk aangekoppeld
124 declare SOME_BACKUPDIR_NOT_FOUND=false # Enkele BACKUPDIRs niet gevonden
125
126 #----- TO -----
127 declare -A HARDDISK # Harddisk koppelpunt/namen
128 declare -A HARDDISK_STATUS # Harddisk status
129 declare -A BACKUPDIR # Back-upmappen, doel
130
131 #####
132 # Functies (op volgorde van uitvoering) #

```

```
133 #####
134
135 #-Functie-----#
136 # Naam: toon_hulp #
137 # Doel: Uitleg werking script. #
138 # Arg.: Geen argumenten. #
139 #-----#
140 toon_hulp() {
141     toon_gebruik "$SCRIPT_NEEDS_SUDO" "$OPTION_USAGE"
142     cat << HULP
143
144 Beschrijving:
145     Script voor het maken van een back-up van een back-up.
146
147     Dit script maakt een extra back-up van een eerder gemaakt back-up als
148     deze lokaal is opgeslagen (standaard op $STORAGEDIR).
149     Met andere woorden: kopieer een back-up, bijvoorbeeld gemaakt met script
150     backup, naar een andere plek zoals een (externe) (netwerk)schijf.
151     Er wordt gezocht naar aangekoppelde schijven op /media/ en /mnt/.
152     Voorwaarden:
153     - Bron dient 2 bestanden te bevatten, back-up en back-uplogboek.
154     - Doel dient $NUM_TO_FILES_OKE bestanden te bevatten.
155       Als het doel $NUM_TO_FILES_MAX bestanden bevat en er zijn bestanden
156       te kopiëren, worden de oudste 2 bestanden verwijderd.
157       Dit correspondeert met \
158     overgrootvader/grootvader/vader/zoon-strategie; $NUM_GEN generaties.
159
160     Als de back-up van een back-up is uitgevoerd, worden de lokale back-up en
161     logboek verwijderd.
162
163     Indien nodig wordt het beheerderswachtwoord gevraagd.
164
165 $OPTIONS_HELP_SC
166 $PART_OF_INSTALL_HELPTEXT
167 HULP
168 }
169
170
171 #-Functie-----#
172 # Naam: controleer_invoer #
173 # Doel: Initiële controles en/of acties. #
174 # Arg.: Geen argumenten. #
175 #-----#
176 controleer_invoer() {
177     local harddisk
178     local -i num_from_files
179     local -i to_dir_len
180
181     #-----#
182     # Controleer opslagmap. #
183     #-----#
184     if [[ ! -d $STORAGEDIR ]]; then
185         error "Opslagmap $STORAGEDIR bestaat niet"
186         normal 'Is script backup uitgevoerd?'
187         warning "Voer dit script niet uit als de back-up extern is \
188 opgeslagen."
189         exit $E_STORAGEDIR_NOT_FOUND
190     fi
191
192     #-----#
193     # Controleer aantal bestanden in opslagmap. #
194     #-----#
195     num_from_files=$(ls "$STORAGEDIR"/$BACKUPFILE 2>> "$LOG" | wc --lines)
196     if [[ $num_from_files -ne 2 ]]; then
197         error 'Het aantal gevonden bestanden is niet 2.'
198         warning 'Verwacht 2 bestanden, namelijk backup en logboek.'
```

```

199     normal "Het aantal gevonden bestanden is $num_from_files."
200     normal "Controleer de $BACKUPFILE bestanden in $STORAGEDIR."
201     normal 'Is script backup (teveel) uitgevoerd?'
202     exit $E_STORAGEDIR_NUMBER_OF_FILES_ERROR
203 fi
204
205 #-----#
206 # Vul associatief array met harddisknamen. #
207 #-----#
208 while read record; do
209     HARDDISK[$record]="$record"
210 done < <(mount |
211     grep --regex='/media/' \
212         --regex='/mnt/' |
213     awk '{print $3}')
214 readonly HARDDISK
215
216 #-----#
217 # Verwerk aangekoppelde harddisks. #
218 #-----#
219 for harddisk in "${HARDDISK[@]}; do
220
221     NO_HARDDISK_FOUND=false
222
223     #-----#
224     # Vul overig associatief array. #
225     #-----#
226     BACKUPDIR[$harddisk]="${HARDDISK[$harddisk]}/Backups/$COMPNAME"
227
228     #-----#
229     # Bepaal grootste breedte voor melding. #
230     #-----#
231     to_dir_len=${#BACKUPDIR[$harddisk]}
232     if [[ $to_dir_len -gt $MAX_BACKUPDIR_LEN ]]; then
233         MAX_BACKUPDIR_LEN=$to_dir_len
234     fi
235
236     #-----#
237     # Controleer of map 'Backups' aanwezig is. #
238     #-----#
239     if [[ ! -d ${HARDDISK[$harddisk]}/Backups ]]; then
240         SOME_BACKUPDIR_NOT_FOUND=true
241         HARDDISK_STATUS[$harddisk]=\
242 "${HARDDISK_STATUS_TXT[NO-BACKUPS-FOLDER]}"
243         continue
244     fi
245
246     #-----#
247     # Controleer of map '<computernaam>' aanwezig is. #
248     #-----#
249     if [[ ! -d ${BACKUPDIR[$harddisk]} ]]; then
250         SOME_BACKUPDIR_NOT_FOUND=true
251         HARDDISK_STATUS[$harddisk]=\
252 "${HARDDISK_STATUS_TXT[NO-COMPNAME-FOLDER]}"
253         continue
254     else
255         HARDDISK_STATUS[$harddisk]="${HARDDISK_STATUS_TXT[PRESENT]}"
256     fi
257
258     #-----#
259     # Controleer het aantal bestanden. #
260     #-----#
261     controleer_aantal_bestanden
262 done
263 }
264

```

```

265 #-Functie-----#
266 # Naam: controleer_aantal_bestanden #
267 # Doel: Controleer aantal back-up/logbestanden. #
268 # Arg.: Geen argumenten. #
269 #-----#
270 controleer_aantal_bestanden() {
271 # local harddisk is gedefinieerd in aanroepende functie
272 local num_to_files=$(ls "${BACKUPDIR[$harddisk]}/$BACKUPFILE \
273 2>> "$LOG" |
274 wc --lines)
275
276 if ! echo "$NUM_TO_FILES_OKE" | grep --quiet $num_to_files; then
277 error "Het aantal gevonden bestanden is niet $NUM_TO_FILES_OKE."
278 normal "Het aantal gevonden bestanden is $num_to_files."
279 normal "Controleer de $BACKUPFILE bestanden in"
280 normal "${BACKUPDIR[$harddisk]}."
281 exit $E_BACKUPDIR_NUMBER_OF_FILES_ERROR
282 fi
283 }
284
285 #-Functie-----#
286 # Naam: toon_invoer #
287 # Doel: Toon wat het script gaat doen. #
288 # Arg.: Geen argumenten. #
289 #-----#
290 toon_invoer() {
291 local filler
292 local msg
293 local harddisk
294 local -i to_dir_len
295
296 clear
297 log "$DASHES"
298 normal "$HEADER"
299 normal
300 normal 'Back-up: '$BACKUPFILE
301 normal 'Van: '$STORAGEDIR"
302 msg='Naar: '
303 for harddisk in "${!HARDDISK[@]}"; do
304
305 #-----#
306 # Filler voor uitlijnen back-upmap en doel aanwezig tekst. #
307 #-----#
308 to_dir_len=${#BACKUPDIR[$harddisk]}
309 filler=${SPACES:0:$MAX_BACKUPDIR_LEN-$to_dir_len+1}
310
311 normal "$msg${BACKUPDIR[$harddisk]}$filler\
312 ${HARDDISK_STATUS[$harddisk]}"
313 msg=' '
314 done
315 normal "Generaties: $NUM_GEN"
316 normal 'Snelheid: ≈ 2,0 GB/min'
317 normal "Logboek: $LOG"
318 log "$DASHES"
319 normal
320 }
321
322 #-Functie-----#
323 # Naam: verwerk_backup_backup #
324 # Doel: Bepaal wat er gekopieerd/verwijderd moet worden. #
325 # Arg.: Geen argumenten. #
326 #-----#
327 verwerk_backup_backup() {
328 local bupfilecopy
329 local bupfiledelete
330 local filler

```

```

331     local    harddisk
332     local -i  to_dir_len
333
334     for harddisk in "${!HARDDISK[@]}; do
335
336         #-----#
337         # Filler voor uitlijnen back-upmap en doel aanwezig tekst.           #
338         #-----#
339         to_dir_len=${#BACKUPDIR[$harddisk]}
340         filler=${SPACES:0:$MAX_BACKUPDIR_LEN-$to_dir_len+1}
341
342         if [[ ${HARDDISK_STATUS[$harddisk]} = \
343 ${HARDDISK_STATUS_TXT[PRESENT]} ]]; then
344             normal "Verwerk..... \
345 ${BACKUPDIR[$harddisk]}$filler${HARDDISK_STATUS[$harddisk]}"
346             bupfilecopy=false
347             controleer_kopieren_nodig
348             if ! $bupfilecopy; then
349                 controleer_gekopieerde_bestanden
350             fi
351
352             bupfiledelete=false
353             if $bupfilecopy; then
354                 controleer_verwijderen_nodig
355             fi
356
357             if $bupfilecopy && $bupfiledelete; then
358                 verwijder_bestanden
359             fi
360
361             if $bupfilecopy; then
362                 controleer_schijfruimte
363                 if $DISKSPACE_OK; then
364                     kopieer_bestanden
365                     controleer_gekopieerde_bestanden
366                 fi
367             fi
368         else
369             normal "Sla over.... \
370 ${BACKUPDIR[$harddisk]}$filler${HARDDISK_STATUS[$harddisk]}"
371             normal ' Kan geen bestanden verwijderen.'
372             normal ' Kan geen bestanden kopiëren.'
373         fi
374     done
375 }
376
377 #-Functie-----#
378 # Naam: controleer_kopieren_nodig                                           #
379 # Doel: Bepaal of er gekopieerd moet worden.                               #
380 # Arg.: Geen argumenten.                                                   #
381 #-----#
382 controleer_kopieren_nodig() {
383 # local bupfilecopy is gedefinieerd in aanroepende functie
384 # local harddisk is gedefinieerd in aanroepende functie
385     local first_fromfilename=$(ls "$STORAGEDIR" \
386                               2>> "$LOG" |
387                               head --lines=1)
388
389     if [[ -f ${BACKUPDIR[$harddisk]}/$first_fromfilename ]]; then
390         normal ' Geen bestanden te verwijderen.'
391         normal " Bestanden zijn al gekopieerd ($(basename \
392 $first_fromfilename .log))."
393     else
394         bupfilecopy=true
395     fi
396 }

```

```

397
398 #-Functie-----#
399 # Naam: controleer_gekopieerde_bestanden #
400 # Doel: Controleer de gekopieerde bestanden op grootte. #
401 # Arg.: Geen argumenten. #
402 #-----#
403 controleer_gekopieerde_bestanden() {
404     local -i copied_bytes=\
405     $(du    ${BACKUPDIR[$harddisk]}/${STARFILE_TO_COPY_BASE} \
406           ${BACKUPDIR[$harddisk]}/${LOGFILE_TO_COPY_BASE} \
407           --summarize \
408           --total \
409           --apparent-size \
410           --block-size=1 \
411           2>> "$LOG" |
412           tail --lines=1 |
413           awk '{print $1}')
414
415     if [[ copied_bytes -ne NEED_BYTES ]]; then
416         error 'De gekopieerde backup is niet even groot als de back-up.'
417         error 'Verwijder de gekopieerde back-up van de externe harddisk,'
418         error "zie hierboven, en start $PROGNAME opnieuw."
419         exit $E_SIZES_ARE_DIFFERENT
420     fi
421 }
422
423 #-Functie-----#
424 # Naam: controleer_verwijderen_nodig #
425 # Doel: Bepaal of er verwijderd moet worden. #
426 # Arg.: Geen argumenten. #
427 #-----#
428 controleer_verwijderen_nodig() {
429     # local bupfiledelete is gedefinieerd in aanroepende functie
430     # local harddisk is gedefinieerd in aanroepende functie
431     local num_to_files=$(ls "${BACKUPDIR[$harddisk]}/${BACKUPFILE} \
432                           2>> "$LOG" |
433                           wc --lines)
434
435     if [[ $num_to_files -eq $NUM_TO_FILES_MAX ]]; then
436         bupfiledelete=true
437     else
438         normal ' Geen bestanden te verwijderen.'
439     fi
440 }
441
442 #-Functie-----#
443 # Naam: verwijder_bestanden #
444 # Doel: Verwijder oudste back-up en logboek. #
445 # Arg.: Geen argumenten. #
446 #-----#
447 verwijder_bestanden() {
448     # local harddisk is gedefinieerd in aanroepende functie
449     local file_to_delete
450     local file_to_delete_base
451     local -i rm_rc
452     local msg
453     local files_to_delete=$(ls "${BACKUPDIR[$harddisk]}/${BACKUPFILE} \
454                             2>> "$LOG" |
455                             head --lines=2)
456
457     normal ' Twee oudste bestanden worden nu verwijderd...'
458     msg=' Verwijder. '
459     for file_to_delete in $files_to_delete; do
460         bepaal_bestandsgrootte "$file_to_delete" filesize
461         file_to_delete_base=$(basename $file_to_delete)
462

```

```

463     #-----#
464     # Filler voor uitlijnen te verwijderen back-up en grootte.      #
465     #-----#
466     file_to_delete_base_len=${#file_to_delete_base}
467     filler=${SPACES:0:$file_to_delete_base_len-109}
468
469     normal "msg$file_to_delete_base$filler - ${filesize}B"
470     msg='                en. '
471     done
472
473     spinner 'aan'
474
475     TO_DELETE="$files_to_delete"    # Voor eventueel afbreken van het script
476     rm --verbose \
477        $files_to_delete \
478        &>> "$LOG"
479     rm_rc=$?
480     TO_DELETE=''
481     verwerk_rc "$PROGNAME: $FUNCNAME[$LINENO]: verwijder bestanden" \
482               $rm_rc 0 'abend' MAXRC
483
484     spinner 'uit'
485
486     (( BUPFILES_DELETED = BUPFILES_DELETED + 2 ))
487 }
488
489 #-Functie-----#
490 # Naam: controleer_schijfruimte                                     #
491 # Doel: Controleer of er voldoende schijfruimte beschikbaar is om een #
492 #       back-up van een back-up op te slaan.                       #
493 # Arg.: Geen argumenten.                                          #
494 #-----#
495 controleer_schijfruimte() {
496 #   local harddisk is gedefinieerd in aanroepende functie
497
498     spinner 'aan'
499
500     local free_bytes=$(df "${BACKUPDIR[$harddisk]}" \
501                       --block-size=1 |
502                       tail --lines=1 |
503                       awk '{print $4}')
504     local free_human=$(df "${BACKUPDIR[$harddisk]}" \
505                       --human-readable |
506                       tail --lines=1 |
507                       awk '{print $4}')
508
509     spinner 'uit'
510
511     if [[ $NEED_BYTES -gt $free_bytes ]]
512     then
513         local filesys=$(df ${BACKUPDIR[$harddisk]} |
514                       tail --lines=1 |
515                       awk '{print $1}')
516         local mounted=$(df ${BACKUPDIR[$harddisk]} |
517                       tail --lines=1 |
518                       awk '{print $6}')
519         warning " Te weinig schijfruimte op $mounted (bestandssysteem \
520 $filesys)"
521         warning ' voor het kopiëren van de backup:'
522         warning " $TARFILE_TO_COPY_BASE en $LOGFILE_TO_COPY_BASE."
523         normal " Benodigd is ${NEED_HUMAN}B, beschikbaar is ${free_human}B."
524         normal " Maak ruimte vrij op $mounted (bestandssysteem $filesys),"
525         normal " of gebruik een ander opslagmedium met minimaal \
526 ${NEED_HUMAN}B beschikbaar,"
527         normal " zoals een andere (externe) (netwerk)schijf, USB-stick, \
528 CD, of DVD."

```



```

529     DISKSPACE_OK=false
530     fi
531 }
532
533 #-Functie-----#
534 # Naam: kopieer_bestanden                                     #
535 # Doel: Kopieer nieuwste back-up en logboek.               #
536 # Arg.: Geen argumenten.                                   #
537 #------#
538 kopieer_bestanden() {
539 #   local bupfiledelete is gedefinieerd in aanroepende functie
540 #   local harddisk is gedefinieerd in aanroepende functie
541     local -i    cp_rc
542     local      filesize
543     local      file_to_copy
544     local      file_to_copy_base
545     local      msg
546
547     if $bupfiledelete; then
548         normal ' Twee nieuwe bestanden worden nu gekopieerd...'
549     else
550         normal ' Twee bestanden worden nu gekopieerd...'
551     fi
552     msg=' Kopieer... '
553     for file_to_copy in $FILES_TO_COPY; do
554         bepaal_bestandsgrootte "$file_to_copy" filesize
555         file_to_copy_base="$(basename $file_to_copy)"
556
557         #-----#
558         # Filler voor uitlijnen te kopiëren back-up en grootte.      #
559         #-----#
560         file_to_copy_base_len=${#file_to_copy_base}
561         filler=${SPACES:0:$file_to_copy_base_len-109}
562
563         normal "$msg$file_to_copy_base$filler - ${filesize}B"
564         msg='      en... '
565         # Voor het evtueel afbreken van het script
566         TO_DELETE="$TO_DELETE${BACKUPDIR[$harddisk]}/$file_to_copy_base "
567     done
568
569     spinner 'aan'
570
571     cp --preserve          \
572        --verbose          \
573        $FILES_TO_COPY     \
574        "${BACKUPDIR[$harddisk]}" \
575        &>> "$LOG"
576     cp_rc=$?
577     TO_DELETE=''
578     verwerk_rc "$PROGNAME: $FUNCNAME[$LINENO]: kopieer bestanden" \
579                $cp_rc 0 'abend' MAXRC
580
581     spinner 'uit'
582
583     (( BUPFILES_COPIED = BUPFILES_COPIED + 2 ))
584 }
585
586 #-Functie-----#
587 # Naam: bepaal_bestandsgrootte                               #
588 # Doel: Bepaal de grootte van een bestand.                  #
589 # Arg.: Twee verplichte argumenten:                         #
590 #   1. bestandsnaam      string, invoer                    #
591 #   2. variabelenaam      string, uitvoer, krijgt als inhoud de bepaalde #
592 #   bestandsgrootte                                           #
593 # Vb. : bepaal_bestandsgrootte "$file" filesize             #
594 #------#

```

```

595 bepaal_bestandsgrootte() {
596     local file=${1:-/dev/null}
597     local _filesize_variable=${2:-filesize}
598     local _filesize_value=$(du --apparent-size \
599                             --human-readable \
600                             "$file"
601                             cut --fields=1)
602
603     eval $_filesize_variable="$_filesize_value"
604 }
605
606 #-Functie-----#
607 # Naam: toon_afsluiten #
608 # Doel: Afsluitende meldingen en/of acties. #
609 # Arg.: Geen argumenten. #
610 #------#
611 toon_afsluiten() {
612     if $NO_HARDDISK_FOUND; then
613         normal
614         warning 'Geen enkele harddisk is aanwezig:'
615         normal '- koppel een harddisk aan,'
616         normal '- start daarna dit script opnieuw.'
617         normal
618         warning 'De back-up van de back-up is NIET gemaakt.'
619         warning 'De back-up is NIET verwijderd.'
620     elif $SOME_BACKUPDIR_NOT_FOUND; then
621         normal
622         normal "Voor de harddisk met de melding 'map ... niet aanwezig' \
623 (zie hierboven):"
624         normal '- maak op de harddisk de ontbrekende map aan,'
625         normal '- start daarna dit script opnieuw.'
626         normal
627         warning 'De back-up van de back-up is NIET gemaakt.'
628         warning 'De back-up is NIET verwijderd.'
629     else
630         if [[ $BUPFILES_COPIED -eq 0 ]]; then
631             normal 'De back-up van de back-up is al gemaakt.'
632         else
633             success 'De back-up van de back-up is gemaakt.'
634         fi
635
636         if $DISKSPACE_OK; then
637             koppel_schijven_af
638
639             #------#
640             # Verwijder backup, maar niet als deze op externe media staat. #
641             #------#
642             if $DISKSPACE_OK && [[ ${STORAGEDIR:0:6} != /media ]]; then
643                 verwijder_backup_en_logboek
644             fi
645         else
646             normal
647             warning 'De back-up van de back-up is NIET gemaakt.'
648             warning 'De back-up is NIET verwijderd.'
649         fi
650     fi
651 fi
652 }
653
654 #-Functie-----#
655 # Naam: koppel_schijven_af #
656 # Doel: Koppel de aangekoppelde schijven af. #
657 # Arg.: Geen argumenten. #
658 #------#
659 koppel_schijven_af() {
660     normal 'Ontkoppel gebruikte schijven...'

```

```
661
662     spinner 'aan'
663
664     for harddisk in "${!HARDDISK[@]"; do
665         if [[ -e ${HARDDISK[$harddisk]} ]]; then
666             umount "${HARDDISK[$harddisk]}" &>> "$LOG"
667             success "Schijf ${HARDDISK[$harddisk]} kan uitgeschakeld en/of \
668 verwijderd worden."
669         fi
670     done
671
672     spinner 'uit'
673
674     success 'Alle schijven kunnen uitgeschakeld en/of verwijderd worden.'
675 }
676
677 #####
678 # Hoofdlijn #
679 #####
680 # init_script
681 {
682     verwerk_invoer "$@"
683     controleer_gebruiker "$SCRIPT_NEEDS_SUDO"
684 }
685 # verwerk
686 {
687     bepaal_log "$SCRIPT_NEEDS_SUDO" LOGDIR "$LOGFILE" LOG
688     controleer_invoer
689     toon_invoer
690     toon_gestart
691     verwerk_backup_backup
692     toon_gestopt
693 }
694 # afsl_script
695 {
696     toon_afsluiten
697     toon_afsluiten_sc
698     exit $MAXRC
699 }
700
701 # Einde script.
```