



```
4 start=1      # Valeur de départ par défaut
5 end=20       # Valeur de fin par défaut
6 increment=1  # Incrément par défaut
7
8 # Attribution des paramètres aux variables, selon leur nombre
9 case $# in
10  0)
11    ;; # Aucun paramètre: les valeurs par défaut sont déjà définies
12  1)
13    start=$1
14    end=20
15    ;; # Un seul paramètre: il définit la valeur de départ, fin est 20
16  2)
17    start=$1
18    end=$2
19    ;; # Deux paramètres: valeurs de début et de fin
20  3)
21    start=$1
22    end=$2
23    increment=$3
24    ;; # Trois paramètres: valeurs de début, de fin et incrément
25  *)
26    echo "Erreur: Trop de paramètres."
27    exit 1
28    ;; # Plus de trois paramètres: affiche un message d'erreur et quitte
29 esac
30
31 # Boucle pour afficher la séquence des nombres
32 for ((i=start; i<=end; i+=increment)); do
33   echo $i
34 done
35
```

Compte rendu TP6 :

Systeme d'exploitation Unix/Linux



Réalise par :

Mohammed kharchafi

Exercice 1:

```
1 #!/bin/bash
2
3 # Définition des valeurs par défaut
4 start=1      # Valeur de départ par défaut
5 end=20       # Valeur de fin par défaut
6 increment=1  # Incrément par défaut
7
8 # Attribution des paramètres aux variables, selon leur nombre
9 case $# in
10  0)
11     ;; # Aucun paramètre: les valeurs par défaut sont déjà définies
12  1)
13     start=$1
14     end=20
15     ;; # Un seul paramètre: il définit la valeur de départ, fin est 20
16  2)
17     start=$1
18     end=$2
19     ;; # Deux paramètres: valeurs de début et de fin
20  3)
21     start=$1
22     end=$2
23     increment=$3
24     ;; # Trois paramètres: valeurs de début, de fin et incrément
25  *)
26     echo "Erreur: Trop de paramètres."
27     exit 1
28     ;; # Plus de trois paramètres: affiche un message d'erreur et quitte
29 esac
30
31 # Boucle pour afficher la séquence des nombres
32 for ((i=start; i<=end; i+=increment)); do
33     echo $i
```

```

4 start=1      # Valeur de départ par défaut
5 end=20       # Valeur de fin par défaut
6 increment=1  # Incrément par défaut
7
8 # Attribution des paramètres aux variables, selon leur nombre
9 case $# in
10  0)
11    ;; # Aucun paramètre: les valeurs par défaut sont déjà définies
12  1)
13    start=$1
14    end=20
15    ;; # Un seul paramètre: il définit la valeur de départ, fin est 20
16  2)
17    start=$1
18    end=$2
19    ;; # Deux paramètres: valeurs de début et de fin
20  3)
21    start=$1
22    end=$2
23    increment=$3
24    ;; # Trois paramètres: valeurs de début, de fin et incrément
25  *)
26    echo "Erreur: Trop de paramètres."
27    exit 1
28    ;; # Plus de trois paramètres: affiche un message d'erreur et quitte
29 esac
30
31 # Boucle pour afficher la séquence des nombres
32 for ((i=start; i<=end; i+=increment)); do
33   echo $i
34 done
35

```

A terminal window titled "kharchafi@kharchafi-virtual-machine: ~" with standard window controls. The terminal shows the execution of a script named "script3.sh".

```

4
5
6
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh 1 6 2
1
3
5
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh 6
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
kharchafi@kharchafi-virtual-machine:~$ w

```

```
kharchafi@kharchafi-virtual-machine: ~  
4  
5  
6  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh 1 6 2  
1  
3  
5  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh 6  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
kharchafi@kharchafi-virtual-machine:~$ w
```

EXERCICE2

```
script3.sh  
~/  
1#!/bin/bash  
2  
3# Vérification de la présence d'un paramètre  
4if [ $# -eq 0 ]; then  
5    echo "Erreur: paramètre manquant."  
6    exit 1  
7fi  
8  
9# Récupération du nom de fichier depuis le paramètre  
10filename="$1"  
11  
12# Vérification de l'existence du fichier  
13if [ ! -f "$filename" ]; then  
14    echo "Erreur: Le fichier '$filename' n'existe pas."  
15    exit 1  
16fi  
17  
18# Affichage des droits d'accès du fichier  
19permissions=$(ls -l "$filename" | cut -d ' ' -f 1)  
20echo "Les droits d'accès pour '$filename' sont: $permissions"  
21
```

```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ touch fichier45  
kharchafi@kharchafi-virtual-machine:~$ ls -l fichier45  
-rw-rw-r-- 1 kharchafi kharchafi 0 11:43 20 أبريل fichier45  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier45  
Les droits d'accès pour 'fichier45' sont: -rw-rw-r--  
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE3

```
1 #!/bin/bash  
2  
3 # Vérification de la présence d'un paramètre  
4 if [ $# -eq 0 ]; then  
5     echo "Erreur: Aucun paramètre spécifié."  
6     exit 1  
7 fi  
8  
9 # Assignment du paramètre à la variable directory  
10 directory=$1  
11  
12 # Vérification de l'existence du répertoire  
13 if [ ! -d "$directory" ]; then  
14     echo "Erreur: Le répertoire '$directory' n'existe pas."  
15     exit 1  
16 fi  
17  
18 # Vérification si le répertoire est vide  
19 if [ -z "$(ls -A "$directory")" ]; then  
20     # Le répertoire est vide, on peut le supprimer  
21     rmdir "$directory"  
22     echo "Le répertoire '$directory' a été supprimé car il était vide."  
23 else  
24     echo "Le répertoire '$directory' n'est pas vide."  
25 fi  
26
```

```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh doss1  
Le répertoire 'doss1' a été supprimé car il était vide.  
kharchafi@kharchafi-virtual-machine:~$
```


EXERCICE4

```
1#!/bin/bash
2
3# Vérifier si des paramètres ont été fournis
4if [ $# -eq 0 ]; then
5    echo "Erreur: Aucun fichier spécifié."
6    exit 1
7fi
8
9# Boucle sur tous les paramètres passés au script
10for file in "$@"; do
11    if [ -e "$file" ]; then # Vérifier que le fichier ou le répertoire existe
12        if [ -f "$file" ]; then
13            # C'est un fichier ordinaire, ajouter le droit de lecture s'il n'est pas déjà présent
14            if [ ! -r "$file" ]; then
15                chmod u+r "$file"
16                echo "Droit de lecture ajouté à '$file'."
17            else
18                echo "'$file' a déjà le droit de lecture."
19            fi
20        elif [ -d "$file" ]; then
21            # C'est un répertoire, ajouter le droit d'exécution
22            if [ ! -x "$file" ]; then
23                chmod u+x "$file"
24                echo "Droit d'exécution ajouté à '$file'."
25            else
26                echo "'$file' a déjà le droit d'exécution."
27            fi
28        else
29            # Ni fichier ordinaire ni répertoire, réinitialiser les permissions pour le propriétaire
30            chmod u=rwx "$file"
31            echo "Permissions réinitialisées à 'rwx' pour '$file'."
32        fi
33    fi
34done
```

Saving file "/home/kharchafi/script3.sh"...

sh Tab Width: 8 Ln 37, Col 1

16°C Ciel couvert 12:08 20/04/2024

```
14    if [ ! -r "$file" ]; then
15        chmod u+r "$file"
16        echo "Droit de lecture ajouté à '$file'."
17    else
18        echo "'$file' a déjà le droit de lecture."
19    fi
20elif [ -d "$file" ]; then
21    # C'est un répertoire, ajouter le droit d'exécution
22    if [ ! -x "$file" ]; then
23        chmod u+x "$file"
24        echo "Droit d'exécution ajouté à '$file'."
25    else
26        echo "'$file' a déjà le droit d'exécution."
27    fi
28else
29    # Ni fichier ordinaire ni répertoire, réinitialiser les permissions pour le propriétaire
30    chmod u=rwx "$file"
31    echo "Permissions réinitialisées à 'rwx' pour '$file'."
32fi
33else
34    echo "Le fichier ou répertoire '$file' n'existe pas."
35fi
36done
37
```

sh Tab Width: 8

```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ ls  
Desktop    fichier1    hi          Pictures    script3.sh  unix  
docker     fichier2    mon         Projet-socket script.sh   Unix  
dockerr    fichier3    mon_programme Public      snap        Videos  
Documents  fichier45   mon_programme.c repertoire  Templates   y  
Downloads  file1       Music       script1.sh  TP2  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier1 file1 fichier2  
'fichier1' a déjà le droit de lecture.  
'file1' a déjà le droit de lecture.  
'fichier2' a déjà le droit de lecture.  
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE5

```
script3.sh  
~/  
1#!/bin/bash  
2  
3# Vérifier si des paramètres ont été fournis  
4if [ $# -eq 0 ]; then  
5    echo "Erreur: Aucun fichier spécifié."  
6    exit 1  
7fi  
8  
9# Boucle sur chaque fichier passé en paramètre  
10for file in "$@"; do  
11    if [ -f "$file" ]; then # S'assurer que c'est bien un fichier  
12        # Demander confirmation avant d'afficher le contenu du fichier  
13        echo "Voulez-vous afficher le contenu de '$file'? (y/n)"  
14        read response  
15        case $response in  
16            [Yy]* )  
17            echo "Affichage de '$file':"  
18            cat "$file" # Utiliser cat pour afficher le contenu du fichier  
19            echo "-----"  
20            ;;  
21            [Nn]* )  
22            echo "Le fichier '$file' ne sera pas affiché."  
23            ;;  
24            * )  
25            echo "Réponse non valide. Entrez 'y' pour oui ou 'n' pour non."  
26            ;;  
27        esac  
28    else  
29        echo "Le fichier '$file' n'existe pas ou n'est pas un fichier régulier."  
30    fi  
31 done  
32
```



```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier1  
Voulez-vous afficher le contenu de 'fichier1'? (y/n)  
y  
Affichage de 'fichier1':  
hiiiiiiiiiiiiiiiiiii  
-----  
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE6

```
1 #!/bin/bash  
2  
3 # Vérifier si deux fichiers ont été passés en paramètre  
4 if [ $# -ne 2 ]; then  
5     echo "Usage: $0 fichier1 fichier2"  
6     exit 1  
7 fi  
8  
9 # Stocker les noms de fichiers dans des variables  
10 fichier1=$1  
11 fichier2=$2  
12  
13 # Vérifier si les deux fichiers existent  
14 if [ ! -e "$fichier1" ] || [ ! -e "$fichier2" ]; then  
15     echo "Erreur: Au moins un des fichiers n'existe pas."  
16     exit 1  
17 fi  
18  
19 # Utiliser stat pour obtenir les numéros d'inode des fichiers  
20 inode1=$(stat -c "%i" "$fichier1")  
21 inode2=$(stat -c "%i" "$fichier2")  
22  
23 # Comparer les inodes pour déterminer s'ils sont liés  
24 if [ "$inode1" -eq "$inode2" ]; then  
25     echo "Les fichiers '$fichier1' et '$fichier2' sont liés par un lien physique."  
26 else  
27     echo "Les fichiers '$fichier1' et '$fichier2' ne sont pas liés."  
28 fi  
29
```

```
Activities Terminal 12:34 20 أبريل
kharchafi@kharchafi-virtual-machine: ~
kharchafi@kharchafi-virtual-machine:~$ touch fichier5
kharchafi@kharchafi-virtual-machine:~$ ln fichier5 lien
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier5 lien
Voulez-vous afficher le contenu de 'fichier5'? (y/n)
y
Affichage de 'fichier5':
-----
Voulez-vous afficher le contenu de 'lien'? (y/n)
y
Affichage de 'lien':
-----
kharchafi@kharchafi-virtual-machine:~$ gedit script3.sh
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier5 lien7
Erreur: Au moins un des fichiers n'existe pas.
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh fichier5 lien
Les fichiers 'fichier5' et 'lien' sont liés par un lien physique.
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE7

```
Open script3.sh Save
1 #!/bin/bash
2
3 while true; do
4     echo "Choisissez une option:"
5     echo "1 - Afficher le chemin du répertoire courant"
6     echo "2 - Lister le contenu du répertoire courant"
7     echo "3 - Créer deux fichiers"
8     echo "4 - Quitter le programme"
9     read -p "Entrez votre choix: " choix
10
11     case $choix in
12         1)
13             echo "Chemin du répertoire courant : $(pwd)"
14             ;;
15         2)
16             echo "Contenu du répertoire courant : "
17             ls
18             ;;
19         3)
20             read -p "Entrez le nom du premier fichier : " fichier1
21             touch "$fichier1"
22             echo "Fichier '$fichier1' créé."
23             read -p "Entrez le nom du deuxième fichier : " fichier2
24             touch "$fichier2"
25             echo "Fichier '$fichier2' créé."
26             ;;
27         4)
28             echo "Quitter le programme."
29             exit 0
30             ;;
31         *)
32             echo "Erreur: Choix invalide "
```

```
16         echo "Contenu du répertoire courant :"  
17         ls  
18         ;;  
19     3)  
20         read -p "Entrez le nom du premier fichier : " fichier1  
21         touch "$fichier1"  
22         echo "Fichier '$fichier1' créé."  
23         read -p "Entrez le nom du deuxième fichier : " fichier2  
24         touch "$fichier2"  
25         echo "Fichier '$fichier2' créé."  
26         ;;  
27     4)  
28         echo "Quitter le programme."  
29         exit 0  
30         ;;  
31     *)  
32         echo "Erreur: Choix invalide."  
33         ;;  
34     esac  
35     echo  
36 done  
37
```

sh Tab Width: 8 Ln 37, Col 1

Taper ici pour rechercher

Pluie aujourd'hui 12:48 20/04/20

```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ gedit script3.sh  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh  
Choisissez une option:  
1 - Afficher le chemin du répertoire courant  
2 - Lister le contenu du répertoire courant  
3 - Créer deux fichiers  
4 - Quitter le programme  
Entrez votre choix: 1  
Chemin du répertoire courant : /home/kharchafi  
  
Choisissez une option:  
1 - Afficher le chemin du répertoire courant  
2 - Lister le contenu du répertoire courant  
3 - Créer deux fichiers  
4 - Quitter le programme  
Entrez votre choix: 3  
Entrez le nom du premier fichier : fi  
Fichier 'fi' créé.  
Entrez le nom du deuxième fichier : de  
Fichier 'de' créé.  
  
Choisissez une option:  
1 - Afficher le chemin du répertoire courant  
2 - Lister le contenu du répertoire courant
```

EXERCICE8

Open ▾



script3.sh

~/

```
1 #!/bin/bash
2
3 # Vérifier si le nombre correct de paramètres a été fourni
4 if [ $# -ne 2 ]; then
5     echo "Usage: $0 source_directory target_directory"
6     exit 1
7 fi
8
9 # Stocker les chemins source et cible dans des variables
10 source_directory=$1
11 target_directory=$2
12
13 # Vérifier si le répertoire source existe
14 if [ ! -d "$source_directory" ]; then
15     echo "Erreur: Le répertoire source '$source_directory' n'existe pas."
16     exit 1
17 fi
18
19 # Vérifier si le répertoire cible existe, sinon le créer
20 if [ ! -d "$target_directory" ]; then
21     echo "Le répertoire cible '$target_directory' n'existe pas, il va être créé."
22     mkdir -p "$target_directory"
23     if [ $? -ne 0 ]; then
24         echo "Erreur: La création du répertoire cible a échoué."
25         exit 1
26     fi
27 fi
28
29 # Copier la sous-arborescence du répertoire source vers le répertoire cible
30 cp -a "$source_directory"/. "$target_directory"
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ gedit script3.sh  
kharchafi@kharchafi-virtual-machine:~$ ls  
de          fi          fichiers5    mon_programme.c  script1.sh  unix  
Desktop     fichier1     file1        Music             script3.sh  Unix  
docker      fichier1.save hi           Pictures          script.sh   Videos  
dockerr     fichier2     lien         Projet-socket     snap        y  
Documents   fichier3     mon          Public            Templates  
Downloads   fichier45    mon_programme repertoire        TP2  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh pictures index  
Erreur: Le répertoire source 'pictures' n'existe pas.  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh Pictures index  
Le répertoire cible 'index' n'existe pas, il va être créé.  
La copie a été réalisée avec succès.  
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE9

```
1#!/bin/bash  
2  
3# Check if the correct number of arguments is provided  
4if [ $# -ne 1 ]; then  
5    echo "Usage: $0 <username>"  
6    exit 1  
7fi  
8  
9# Get the username from the argument  
10 username=$1  
11  
12# Check if the user exists  
13if id "$username" &>/dev/null; then  
14    # Check if the user is currently logged in  
15    if who | grep -q "^$username "; then  
16        # User is logged in, echo "bonjour" to the terminal  
17        echo "bonjour"  
18    else  
19        # User is not logged in, send "bonjour" via email  
20        echo "bonjour" | mail -s "Message from script" $username  
21    fi  
22else  
23    # User does not exist  
24    echo "User $username does not exist."  
25fi
```



```
kharchafi@kharchafi-virtual-machine: ~  
kharchafi@kharchafi-virtual-machine:~$ gedit script3.sh  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh kharchafi  
bonjour  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh kharchafii  
User kharchafii does not exist.  
kharchafi@kharchafi-virtual-machine:~$ ./script3.sh abdellah  
kharchafi@kharchafi-virtual-machine:~$
```

EXERCICE

```
1 #!/bin/bash  
2  
3 if [ $# -ne 1 ]; then  
4     echo "Usage incorrect."  
5     exit 1  
6 fi  
7  
8 nombre_utilisateurs=$1  
9  
10 for (( i=1; i<=$nombre_utilisateurs; i++ )); do  
11     username="utilisateur$i"  
12     sudo useradd -m $username  
13 done  
14  
15 echo "$nombre_utilisateurs utilisateurs créés avec succès."
```