



Ressourcesinformatiques

# Programmation **shell** sous **Unix/Linux** **ksh, bash, sh** (avec exercices corrigés)

5<sup>e</sup> édition

Christine DEFFAIX RÉMY

Fichiers complémentaires  
à télécharger



Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **RI5PRO** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1 Introduction

- 1. Définition du shell ..... 21
- 2. Caractéristiques d'un interpréteur de commandes ..... 21
- 3. Interpréteurs de commandes (shells) ..... 22
  - 3.1 Historique ..... 22
  - 3.2 Avec quel shell faut-il programmer ? ..... 23
    - 3.2.1 Scripts de démarrage ..... 23
    - 3.2.2 Autres scripts ..... 23
  - 3.3 Nom des exécutable sous Unix ..... 24
- 4. Shells abordés dans cet ouvrage ..... 24

### Chapitre 2 Mécanismes essentiels du shell

- 1. Commandes internes et externes ..... 27
  - 1.1 Les commandes externes ..... 27
  - 1.2 Les commandes internes ..... 29
  - 1.3 Implémentation interne et implémentation externe ..... 30
- 2. Affichage à l'écran ..... 31
  - 2.1 La commande echo ..... 31
    - 2.1.1 Le caractère \n ..... 32
    - 2.1.2 Le caractère \c ..... 32



# 2 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

2.1.3	Le caractère \t	33
2.1.4	Liste des caractères d'échappement	33
2.2	Les commandes print et printf	34
3.	Le caractère ~ (tilde)	34
4.	La commande interne cd	35
5.	Substitution de noms de fichiers	36
5.1	Expressions basiques	36
5.1.1	Le caractère *	36
5.1.2	Le caractère ?	37
5.1.3	Les caractères [ ]	37
5.2	Expressions complexes	38
5.2.1	?(expression)	39
5.2.2	*(expression)	39
5.2.3	+(expression)	39
5.2.4	@(expression)	40
5.2.5	!(expression)	40
5.2.6	Alternatives	40
5.3	Interprétation du shell	41
6.	Séparateur de commandes	42
7.	Redirections	43
7.1	Entrée et sorties standards des processus	43
7.1.1	Entrée standard	43
7.1.2	Sortie standard	43
7.1.3	Sortie d'erreur standard	44
7.2	Héritage	44
7.3	Redirection des sorties en écriture	45
7.3.1	Sortie standard	45
7.3.2	Sortie d'erreur standard	46
7.3.3	Sortie standard et sortie d'erreur standard	47
7.3.4	Se protéger d'un écrasement involontaire de fichier	48
7.3.5	Éliminer les affichages	49
7.3.6	Mécanisme interne	49

7.4	Redirection de l'entrée standard	50
7.5	Redirections avancées	52
7.5.1	Rediriger les descripteurs 1 et 2 vers le même fichier	52
7.5.2	La double redirection en lecture	58
7.5.3	Fermeture d'un descripteur	59
8.	Tubes de communication	59
8.1	Commandes ne lisant pas leur entrée standard	61
8.2	Commandes lisant leur entrée standard	61
8.2.1	Exemples triviaux	62
8.2.2	Cas des filtres	62
8.3	Compléments	66
8.3.1	Enchaîner des tubes	66
8.3.2	Dupliquer les sorties	66
8.3.3	Envoyer la sortie standard et la sortie d'erreur standard dans le tube	67
9.	Regroupement de commandes	68
9.1	Les parenthèses	69
9.2	Les accolades	73
9.3	Conclusion	77
10.	Processus en arrière-plan	77
11.	Exercices	78
11.1	Fonctionnalités diverses	78
11.1.1	Exercice 1 : commandes internes et externes	78
11.1.2	Exercice 2 : génération de noms de fichiers	78
11.1.3	Exercice 3 : séparateur de commandes	79
11.2	Redirections	79
11.2.1	Exercice 1	79
11.2.2	Exercice 2	79
11.2.3	Exercice 3	79
11.2.4	Exercice 4	80
11.2.5	Exercice 5	80
11.2.6	Exercice 6	80

# 4 \_\_\_\_\_ Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

11.3 Tubes de communication .....	80
11.3.1 Exercice 1 .....	80
11.3.2 Exercice 2 .....	80
11.3.3 Exercice 3 .....	80
11.3.4 Exercice 4 .....	80

## Chapitre 3

### Paramétrage de l'environnement de travail

1. Variables d'environnement .....	81
1.1 Liste des variables .....	81
1.2 Affichage de la valeur d'une variable .....	82
1.3 Modification de la valeur d'une variable .....	82
1.4 Principales variables .....	83
1.4.1 HOME .....	83
1.4.2 PATH .....	83
1.4.3 PWD .....	85
1.4.4 PS1 .....	85
1.4.5 PS2 .....	89
1.4.6 TMOUT .....	89
1.4.7 TERM .....	90
1.4.8 LOGNAME .....	90
1.4.9 Processus et variables d'environnement .....	90
1.5 Exportation des variables .....	91
1.5.1 Liste des variables exportées .....	91
1.5.2 Variables devant être exportées .....	91
1.5.3 Exporter une variable .....	92
2. Les options du shell .....	95
2.1 Activer et désactiver une option du shell .....	95
2.2 Visualiser la liste des options .....	95
2.3 Principales options .....	96
2.3.1 ignoreeof .....	96
2.3.2 noclobber .....	96



2.3.3	emacs et vi	97
2.3.4	xtrace	98
3.	Les alias	98
3.1	Définir un alias	98
3.2	Visualiser la liste des alias	99
3.2.1	Visualiser tous les alias	99
3.2.2	Visualiser un alias en particulier	99
3.3	Supprimer un alias	99
4.	Historique de commandes	100
4.1	Paramétrer le rappel de commandes en ksh	101
4.1.1	Option vi	101
4.1.2	Option emacs	102
4.2	Paramétrer le rappel de commandes en bash	106
4.3	La complétion de noms de fichiers	106
4.3.1	La complétion du bash	106
4.3.2	La complétion du ksh	107
4.3.3	Tableau récapitulatif	109
4.4	Autres complétions	109
5.	Les fichiers d'environnement	110
5.1	Caractéristiques des fichiers d'environnement	110
5.1.1	Shell de connexion	111
5.1.2	Fichiers d'environnement lus par le shell de connexion	111
5.2	Session utilisant un Bourne Shell	115
5.3	Session utilisant un Korn Shell	116
5.4	Session utilisant un Bourne Again Shell	118
6.	Exercices	120
6.1	Variables d'environnement	120
6.1.1	Exercice 1	120
6.1.2	Exercice 2	120
6.2	Alias de commandes	121
6.2.1	Exercice 1	121
6.2.2	Exercice 2	121

# 6 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

## Chapitre 4

### Les bases de la programmation shell

1. Les variables utilisateur . . . . .	123
1.1 Nommer une variable. . . . .	123
1.2 Définir une variable . . . . .	123
1.2.1 Affecter une valeur à une variable . . . . .	124
1.2.2 Affecter une valeur contenant au moins un espace . . . . .	124
1.2.3 Variable indéfinie. . . . .	124
1.2.4 Retirer la définition d'une variable . . . . .	125
1.2.5 Isoler le nom d'une variable . . . . .	125
1.2.6 Variables numériques . . . . .	126
1.2.7 Variables complexes . . . . .	127
1.3 Substitution de variables . . . . .	128
2. Substitution de commandes . . . . .	130
3. Caractères de protection . . . . .	131
3.1 Les simples quotes . . . . .	131
3.2 Le caractère \. . . . .	133
3.3 Les guillemets . . . . .	134
4. Récapitulatif des caractères spéciaux . . . . .	134
5. Interprétation d'une ligne de commande . . . . .	135
6. Écriture et lancement d'un script shell . . . . .	136
6.1 Définition . . . . .	136
6.2 Exécution d'un script par un shell enfant . . . . .	137
6.3 Exécution d'un script par le shell courant . . . . .	143
6.4 Commentaires . . . . .	145
7. Variables réservées du shell . . . . .	146
7.1 Les paramètres positionnels. . . . .	146
7.2 La commande shift . . . . .	148
7.2.1 Syntaxe. . . . .	148
7.2.2 Principe. . . . .	148

7.3	Code de retour d'une commande . . . . .	150
7.3.1	La variable \$? . . . . .	150
7.3.2	La commande exit . . . . .	151
7.4	Autres variables spéciales . . . . .	152
7.4.1	PID du shell interpréteur . . . . .	152
7.4.2	PID du dernier processus lancé en arrière-plan . . . . .	153
8.	La commande read . . . . .	155
8.1	Syntaxe . . . . .	155
8.2	Lectures au clavier . . . . .	155
8.3	Code de retour . . . . .	157
8.4	La variable IFS . . . . .	158
9.	Exécution de tests . . . . .	159
9.1	Introduction . . . . .	159
9.2	La commande test . . . . .	159
9.2.1	Syntaxe . . . . .	160
9.2.2	Tests sur les fichiers . . . . .	160
9.2.3	Tests sur les chaînes de caractères . . . . .	163
9.2.4	Tests sur les nombres . . . . .	165
9.2.5	Les opérateurs . . . . .	166
9.2.6	Exemple concret d'utilisation . . . . .	167
9.3	La commande [[ ]] . . . . .	168
10.	Les opérateurs du shell . . . . .	172
10.1	Évaluation de l'opérateur && . . . . .	173
10.2	Évaluation de l'opérateur    . . . . .	174
11.	L'arithmétique . . . . .	175
11.1	La commande expr . . . . .	175
11.1.1	Syntaxe . . . . .	175
11.1.2	Opérateurs . . . . .	175
11.2	La commande (( )) . . . . .	179
11.2.1	Syntaxe . . . . .	179
11.2.2	Utilisation . . . . .	179
11.3	La commande let . . . . .	182



# 8 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

11.4 Arithmétique sur les flottants . . . . .	182
11.4.1 ksh93 . . . . .	182
11.4.2 Autres shells . . . . .	183
12. Substitution d'expressions arithmétiques . . . . .	184
13. Mise au point d'un script . . . . .	185
13.1 Option -x . . . . .	185
13.2 Autres options . . . . .	188
14. Les structures de contrôle . . . . .	189
14.1 if . . . . .	189
14.2 case . . . . .	193
14.2.1 Syntaxe . . . . .	193
14.2.2 Principe . . . . .	193
14.2.3 Utilisation . . . . .	195
14.3 Boucle for . . . . .	198
14.4 Boucle while . . . . .	202
14.4.1 Syntaxe . . . . .	202
14.4.2 Utilisation . . . . .	202
14.4.3 Boucle infinie . . . . .	203
14.5 until . . . . .	206
14.5.1 Syntaxe . . . . .	206
14.5.2 Utilisation . . . . .	206
14.6 break et continue . . . . .	210
15. Exercices . . . . .	212
15.1 Variables, caractères spéciaux . . . . .	212
15.1.1 Exercice 1 : variables . . . . .	212
15.1.2 Exercice 2 : variables . . . . .	212
15.1.3 Exercice 3 : substitution de commande . . . . .	213
15.1.4 Exercice 4 : caractères de protection . . . . .	213
15.2 Variables, affichages et lectures clavier . . . . .	214
15.2.1 Exercice 1 : variables . . . . .	214
15.2.2 Exercice 2 : paramètres positionnels . . . . .	214
15.2.3 Exercice 3 : lectures clavier . . . . .	214

- 15.3 Tests et arithmétique . . . . . 215
  - 15.3.1 Exercice 1 : tests sur des fichiers . . . . . 215
  - 15.3.2 Exercice 2 : tests de chaînes de caractères . . . . . 215
  - 15.3.3 Exercice 3 : tests numériques . . . . . 216
  - 15.3.4 Exercice 4 : arithmétique. . . . . 216
  - 15.3.5 Exercice 5 : opérateurs logiques des commandes  
 [ ], [[ ]] et opérateurs logiques du shell . . . . . 216
- 15.4 Structures de contrôle if, case, boucle for . . . . . 217
  - 15.4.1 Exercice 1 : les commandes [ ] et [[ ]], la structure  
 de contrôle if . . . . . 217
  - 15.4.2 Exercice 2 : structures de contrôle case, boucle for . . . . 217
- 15.5 Boucles. . . . . 218
  - 15.5.1 Exercice 1 : boucle for, commande tr . . . . . 218
  - 15.5.2 Exercice 2 : boucle for, arithmétique . . . . . 218
  - 15.5.3 Exercice 3 : boucles for, while . . . . . 219

**Chapitre 5**  
**Aspects avancés de la programmation shell**

- 1. Comparatif des variables \$\* et @\$ . . . . . 221
  - 1.1 Utilisation de \$\* et de @\$ . . . . . 221
  - 1.2 Utilisation de "\$\*" . . . . . 222
  - 1.3 Utilisation de "\$@" . . . . . 223
- 2. Substitution de variables . . . . . 225
  - 2.1 Longueur de la valeur contenue dans une variable. . . . . 225
  - 2.2 Manipulation de chaînes de caractères . . . . . 225
    - 2.2.1 Retirer le plus petit fragment à gauche . . . . . 225
    - 2.2.2 Retirer le plus grand fragment à gauche . . . . . 226
    - 2.2.3 Retirer le plus petit fragment à droite . . . . . 226
    - 2.2.4 Retirer le plus grand fragment à droite . . . . . 227

# 10 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

3.	Tableaux	227
3.1	Assigner un élément	227
3.2	Référencer un élément	228
3.3	Assignment globale d'un tableau	228
3.4	Référencer tous les éléments d'un tableau	229
3.5	Obtenir le nombre d'éléments d'un tableau	229
3.6	Obtenir la longueur d'un élément d'un tableau	230
3.7	Tableaux associatifs	230
4.	Initialisation des paramètres positionnels avec set	231
5.	Les fonctions	232
5.1	Définition d'une fonction	232
5.2	Code de retour d'une fonction	234
5.3	Portée des variables	236
5.4	Définition de variables locales	237
5.5	Passage d'arguments	239
5.6	Exploiter l'affichage d'une fonction	241
5.7	Programme complet de l'exemple	242
6.	Commandes d'affichage	244
6.1	La commande print	244
6.1.1	Utilisation simple	244
6.1.2	Suppression du saut de ligne naturel de print	244
6.1.3	Afficher des arguments commençant par le caractère "-"	244
6.1.4	Écrire sur un descripteur particulier	245
6.2	La commande printf	246
7.	Gestion des entrées/sorties d'un script	247
7.1	Redirection des entrées/sorties standards	247
7.2	Gestion de fichiers	251
7.2.1	Ouverture de fichier	251
7.2.2	Lecture à partir d'un fichier	251
7.2.3	Écriture dans un fichier	251
7.2.4	Fermeture d'un fichier	252



- 7.3 Traitement d'un fichier ..... 253
  - 7.3.1 Informations préalables..... 253
  - 7.3.2 Les différentes façons d'exploiter un fichier..... 254
  - 7.3.3 Découper une ligne en champs..... 259
  - 7.3.4 Modifier le séparateur de ligne ..... 261
- 8. La commande eval ..... 262
- 9. Gestion des signaux ..... 264
  - 9.1 Principaux signaux ..... 264
  - 9.2 Ignorer un signal..... 265
  - 9.3 Modifier le traitement associé à un signal ..... 266
  - 9.4 Repositionner le traitement par défaut  
du shell vis-à-vis d'un signal..... 267
  - 9.5 Utiliser trap à partir d'un script shell ..... 268
- 10. Gestion de menus avec select..... 269
- 11. Analyse des options d'un script avec getopt ..... 271
- 12. Gestion d'un processus en arrière-plan ..... 277
- 13. Script d'archivage incrémental et transfert sftp automatique..... 279
  - 13.1 Objectif ..... 279
  - 13.2 Le fichier uploadBackup.sh ..... 282
  - 13.3 Le fichier fonctions.inc.sh ..... 284
- 14. Exercices ..... 288
  - 14.1 Fonctions..... 288
    - 14.1.1 Exercice 1 : fonctions simples ..... 288
    - 14.1.2 Exercice 2 : fonctions simples, statut de retour..... 289
    - 14.1.3 Exercice 3 : passage de paramètres, retour de valeur... 290
    - 14.1.4 Exercice 4 : fichiers ..... 291
    - 14.1.5 Exercice 5 : fichiers, fonctions, menu select..... 292
    - 14.1.6 Exercice 6 : fichiers, tableaux  
associatifs (bash 4, ksh93)..... 293

# 12 \_\_\_\_\_ Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

## Chapitre 6

### Les expressions régulières

1. Introduction . . . . .	295
2. Caractères communs aux ERb et ERe . . . . .	296
3. Caractères spécifiques aux ERb . . . . .	300
4. Caractères spécifiques aux ERe . . . . .	300
5. Exploitation des expressions régulières par les commandes . . . . .	302
5.1 La commande vi . . . . .	302
5.2 La commande grep . . . . .	303
5.3 La commande expr . . . . .	305
5.4 sed et awk . . . . .	308
6. Exercices . . . . .	308
6.1 Expressions régulières. . . . .	309
6.1.1 Exercice 1 : expressions régulières avec vi . . . . .	309
6.1.2 Exercice 2 : grep . . . . .	310

## Chapitre 7

### La commande sed

1. Utilisation de la commande sed. . . . .	311
2. Exemples . . . . .	314
2.1 Utilisation de sed en ligne de commande . . . . .	314
2.1.1 La commande d (delete) . . . . .	314
2.1.2 La commande p (print) . . . . .	315
2.1.3 La commande w (write) . . . . .	316
2.1.4 Négation d'une commande (!) . . . . .	316
2.1.5 La commande s (substitution) . . . . .	317
2.2 Script sed. . . . .	318

- 3. Exercices ..... 320
  - 3.1 Expressions régulières..... 320
    - 3.1.1 Exercice 1 : insertion de balises dans un fichier..... 320
    - 3.1.2 Exercice 2 : formatage de fichier ..... 321

**Chapitre 8**  
**Le langage de programmation awk**

- 1. Principe ..... 323
  - 1.1 Syntaxe ..... 323
  - 1.2 Variables spéciales ..... 324
    - 1.2.1 Variables prédéfinies dès le lancement de awk ..... 324
    - 1.2.2 Variables initialisées lors du traitement d'une ligne ... 325
    - 1.2.3 Exemples simples..... 326
  - 1.3 Critères de sélection ..... 327
    - 1.3.1 Expressions régulières ..... 328
    - 1.3.2 Tests logiques ..... 329
    - 1.3.3 Intervalles de lignes..... 330
  - 1.4 Structure d'un script awk..... 330
    - 1.4.1 BEGIN ..... 330
    - 1.4.2 Sections intermédiaires..... 330
    - 1.4.3 END ..... 330
    - 1.4.4 Commentaires ..... 331
    - 1.4.5 Variables..... 331
    - 1.4.6 Exemple ..... 331
- 2. Opérateurs..... 333
- 3. La fonction printf ..... 334
- 4. Redirections ..... 335
- 5. Lecture de la ligne suivante : next ..... 337
- 6. Structures de contrôle ..... 339
  - 6.1 if..... 339
  - 6.2 switch ..... 339

# 14 \_\_\_\_\_ Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

6.3	for	341
6.4	while	342
6.5	do-while	342
6.6	break	342
6.7	continue	342
7.	Terminer un script	343
8.	Tableaux	343
8.1	Tableaux indicés par un nombre	343
8.2	Tableaux indicés par une chaîne de caractères	344
8.3	Tester l'existence d'un élément	346
8.4	Supprimer un élément	347
8.5	Nombre d'éléments d'un tableau	349
8.6	Ordre de parcours des éléments	349
8.7	Trier un tableau	351
9.	Tableaux multidimensionnels	353
9.1	Simulation de tableaux multidimensionnels	353
9.2	Vrais tableaux multidimensionnels	355
9.3	Tester si une variable est un tableau	356
10.	Les arguments de la ligne de commande	357
11.	Fonctions intégrées	359
11.1	Fonctions travaillant sur les chaînes	359
11.2	Fonctions mathématiques	360
11.3	Fonctions liées aux tableaux	360
11.4	Autres fonctions	361
11.4.1	La fonction getline	361
11.4.2	La fonction close	365
11.4.3	La fonction system	366
12.	Fonctions utilisateur	367
13.	Inclusions de fichiers	369

- 14. Exercices ..... 369
  - 14.1 awk en ligne de commande ..... 370
    - 14.1.1 Exercice 1 : awk et autres filtres ..... 370
    - 14.1.2 Exercice 2 : critères de sélection ..... 370
    - 14.1.3 Exercice 3 : critères de sélection, affichage de champs, sections BEGIN et END ..... 371
  - 14.2 Scripts awk ..... 372
    - 14.2.1 Exercice 4 : fonctions ..... 372
    - 14.2.2 Exercice 5 : analyse d'un fichier de log ..... 373
    - 14.2.3 Exercice 6 : génération d'un fichier d'étiquettes ..... 375

**Chapitre 9**  
**Les commandes filtres**

- 1. Introduction ..... 377
- 2. Syntaxe d'appel des commandes filtres ..... 377
- 3. Visualisation de données ..... 378
  - 3.1 Consultation de données, création de fichiers : cat ..... 378
  - 3.2 Valeur des octets d'un flux de données : od ..... 380
  - 3.3 Filtrage de lignes : grep ..... 381
  - 3.4 Dernières lignes d'un flux de données : tail ..... 384
  - 3.5 Premières lignes d'un flux de données : head ..... 385
  - 3.6 Duplication de la sortie standard : tee ..... 386
  - 3.7 Numérotation de lignes : nl ..... 387
  - 3.8 Présentation d'un flux de données : pr ..... 388
- 4. Traitement de données ..... 390
  - 4.1 Comptage de lignes, de mots et de caractères : wc ..... 390
  - 4.2 Extraction de caractères : cut ..... 392
  - 4.3 Tri de données : sort ..... 393
  - 4.4 paste ..... 396
  - 4.5 split ..... 397
  - 4.6 Transformation de caractères : tr ..... 399
  - 4.7 Dédoublonnage : uniq ..... 400

# 16 \_\_\_\_\_ Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

5.	Compressions, archivages et conversions . . . . .	402
5.1	Compressions : gzip, bzip2 . . . . .	402
5.2	Archives tar . . . . .	404
5.3	Archives cpio . . . . .	406
5.4	Copie physique, transformations : dd . . . . .	408
5.5	Changement d'encodage : iconv . . . . .	410
6.	Commandes réseau sécurisées . . . . .	411
6.1	Connexion à distance : ssh . . . . .	411
6.2	Transfert de fichier : sftp . . . . .	413
6.2.1	Commandes de sftp s'exécutant sur la machine locale . . . . .	414
6.2.2	Commandes s'exécutant sur la machine distante . . . . .	415
6.2.3	Commandes de transfert . . . . .	416
6.2.4	Connexion automatique sans mot de passe . . . . .	417
7.	Autres commandes . . . . .	419
7.1	La commande xargs . . . . .	419
7.2	Comparer deux fichiers : cmp . . . . .	421
7.3	Lignes communes à deux fichiers : comm . . . . .	422

## Chapitre 10

### Comment faire ... (synthèse)

1.	Introduction . . . . .	425
2.	Informations . . . . .	425
3.	Modèles de noms de fichiers . . . . .	426
4.	Tester le code retour d'une commande . . . . .	427
5.	Test d'égalité entre deux chaînes . . . . .	428
6.	Comparaison d'une variable par rapport à un modèle . . . . .	429
7.	Retirer le début ou la fin d'une chaîne . . . . .	430
8.	Récupérer le champ d'une ligne . . . . .	432
9.	Ne pas afficher certaines lignes d'un flux . . . . .	434
10.	Tests numériques . . . . .	435

- 11. Calculs . . . . . 436
- 12. Lancer un script d'un autre langage à partir d'un shell . . . . . 437
- 13. Accéder à une base MySQL à partir d'un shell . . . . . 438

**Chapitre 11**  
**Solutions des exercices**

- 1. Solutions du chapitre Mécanismes essentiels du shell . . . . . 441
  - 1.1 Fonctionnalités diverses . . . . . 441
    - 1.1.1 Exercice 1 : commandes internes et externes . . . . . 441
    - 1.1.2 Exercice 2 : génération de noms de fichiers . . . . . 441
    - 1.1.3 Exercice 3 : séparateur de commandes . . . . . 443
  - 1.2 Redirections . . . . . 443
    - 1.2.1 Exercice 1 . . . . . 443
    - 1.2.2 Exercice 2 . . . . . 443
    - 1.2.3 Exercice 3 . . . . . 444
    - 1.2.4 Exercice 4 . . . . . 444
    - 1.2.5 Exercice 5 . . . . . 444
    - 1.2.6 Exercice 6 . . . . . 444
  - 1.3 Tubes de communication . . . . . 445
    - 1.3.1 Exercice 1 . . . . . 445
    - 1.3.2 Exercice 2 . . . . . 445
    - 1.3.3 Exercice 3 . . . . . 445
    - 1.3.4 Exercice 4 . . . . . 445
- 2. Solutions du chapitre Paramétrage de l'environnement de travail . 446
  - 2.1 Variables d'environnement . . . . . 446
    - 2.1.1 Exercice 1 . . . . . 446
    - 2.1.2 Exercice 2 . . . . . 446
  - 2.2 Alias de commandes . . . . . 447
    - 2.2.1 Exercice 1 . . . . . 447
    - 2.2.2 Exercice 2 . . . . . 448

# 18 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

3.	Solutions du chapitre Les bases de la programmation shell. . . . .	449
3.1	Variables, caractères spéciaux . . . . .	449
3.1.1	Exercice 1 : variables . . . . .	449
3.1.2	Exercice 2 : variables . . . . .	449
3.1.3	Exercice 3 : substitution de commande . . . . .	450
3.1.4	Exercice 4 : caractères de protection. . . . .	450
3.2	Variables, affichages et lectures clavier . . . . .	451
3.2.1	Exercice 1 : variables . . . . .	451
3.2.2	Exercice 2 : paramètres positionnels . . . . .	452
3.2.3	Exercice 3 : lectures clavier . . . . .	453
3.3	Tests et arithmétique. . . . .	454
3.3.1	Exercice 1 : tests sur des fichiers. . . . .	454
3.3.2	Exercice 2 : tests de chaînes de caractères . . . . .	455
3.3.3	Exercice 3 : tests numériques . . . . .	456
3.3.4	Exercice 4 : arithmétique. . . . .	457
3.3.5	Exercice 5 : opérateurs logiques des commandes [ ], [[ ]] et opérateurs logiques du shell . . . . .	458
3.4	Structures de contrôle if, case, boucle for . . . . .	459
3.4.1	Exercice 1 : les commandes [ ] et [[ ]], la structure de contrôle if . . . . .	459
3.4.2	Exercice 2 : structures de contrôle case, boucle for . . . .	460
3.5	Boucles. . . . .	461
3.5.1	Exercice 1 : boucle for, commande tr . . . . .	461
3.5.2	Exercice 2 : boucle for, arithmétique . . . . .	462
3.5.3	Exercice 3 : boucles for, while. . . . .	463
4.	Solutions du chapitre Aspects avancés de la programmation shell . . . . .	464
4.1	Fonctions. . . . .	464
4.1.1	Exercice 1 : fonctions simples . . . . .	464
4.1.2	Exercice 2 : fonctions simples, statut de retour. . . . .	466
4.1.3	Exercice 3 : passage de paramètres, retour de valeur. . .	467
4.1.4	Exercice 4 : fichiers . . . . .	469
4.1.5	Exercice 5 : fichiers, fonctions, menu select . . . . .	470



- 4.1.6 Exercice 6 : fichiers, tableaux associatifs  
(bash 4, ksh93).....472
- 5. Solutions du chapitre Les expressions régulières .....473
  - 5.1 Expressions régulières.....473
    - 5.1.1 Exercice 1 : expressions régulières avec vi .....473
    - 5.1.2 Exercice 2 : grep .....475
- 6. Solutions du chapitre La commande sed.....476
  - 6.1 Expressions régulières.....476
    - 6.1.1 Exercice 1 : insertion de balises dans un fichier.....476
    - 6.1.2 Exercice 2 : formatage de fichier .....477
- 7. Solution du chapitre Le langage de programmation awk .....478
  - 7.1 awk en ligne de commande .....478
    - 7.1.1 Exercice 1 : awk et autres filtres .....478
    - 7.1.2 Exercice 2 : critères de sélection .....478
    - 7.1.3 Exercice 3 : critères de sélection, affichage de champs,  
sections BEGIN et END .....478
  - 7.2 Scripts awk .....479
    - 7.2.1 Exercice 4 : fonctions.....479
    - 7.2.2 Exercice 5 : analyse d'un fichier de log .....482
    - 7.2.3 Exercice 6 : génération d'un fichier d'étiquettes .....483

**Annexes**

- 1. Caractères spéciaux du shell .....487
- 2. Commandes internes au shell .....488
- 3. Ordre d'interprétation d'une commande.....491

- Index .....493

---

## Chapitre 3

# Paramétrage de l'environnement de travail

### 1. Variables d'environnement

Les thèmes abordés dans ce chapitre permettront à l'utilisateur de paramétrer son environnement de travail en tenant compte du shell utilisé.

Un certain nombre de variables sont définies dans l'environnement du shell. Elles contiennent des informations nécessaires au fonctionnement de l'interpréteur et/ou des commandes lancées à partir de celui-ci.

#### 1.1 Liste des variables

La commande **set** donne la liste des variables définies dans le shell courant.

##### Exemple

```
$ set
HOME=/home/christie
LOGNAME=christie
PATH=/usr/bin:/bin
PS1='$ '
PS2='> '
TERM=vt100
...
```

## 1.2 Affichage de la valeur d'une variable

Le caractère spécial **\$** du shell permet d'accéder au contenu d'une variable.

### Exemple

```
$ echo $HOME
/home/christie
$
```

## 1.3 Modification de la valeur d'une variable

Le shell permet d'initialiser ou de modifier des variables.

### Exemple

```
$ variable=valeur
$ echo $variable
valeur
$
```

Si la valeur contient des caractères spéciaux du shell (**\$**, **>**, espace...), il faut empêcher le shell d'interpréter ceux-ci en entourant la valeur avec des simples quotes.

### ■ Remarque

*Utiliser des simples quotes est l'une des trois manières de masquer des caractères en shell. Ce point sera détaillé ultérieurement.*

### Exemple

Le symbole **>** (redirection) doit être masqué, l'espace (séparateur de mots sur la ligne de commande) également :

```
$ variable='mot1 mot2 =>'
$ echo $variable
mot1 mot2 =>
$
```

### ■ Remarque

*Il ne faut pas mettre d'espace autour du signe =. Le shell ne comprendrait pas qu'il s'agit d'une affectation.*

## 1.4 Principales variables

Les variables présentées ci-dessous possèdent une valeur au niveau du shell de connexion. D'autres variables peuvent être définies ultérieurement.

La modification d'une variable d'environnement en ligne de commande est valable uniquement dans le shell courant. Pour que les modifications soient prises en compte dans tous les shells, il faut utiliser les fichiers de paramétrage (cf. Les fichiers d'environnement dans ce chapitre).

### 1.4.1 HOME

Cette variable contient la valeur du répertoire d'accueil de l'utilisateur. Elle ne doit pas être modifiée.

### 1.4.2 PATH

La variable PATH contient une liste de répertoires qui sont explorés par le shell lorsqu'il doit lancer une commande externe.

### ■ Remarque

*En aucun cas, une commande n'est recherchée dans le répertoire courant si celui-ci ne figure pas dans la variable PATH.*

### Exemples

```
■ $ echo $PATH
   /usr/bin:/bin
   $
```

La commande `date` est trouvée :

```
■ $ date
   Tue Jan 28 17:51:23 MET 2014
   $
```

# 84 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

En effet, elle se situe dans le répertoire `/usr/bin` :

```
$ find / -name date 2> /dev/null
/usr/bin/date
$
```

La commande `ping` n'est pas trouvée :

```
$ ping localhost
ksh: ping: not found
$
```

La commande est située dans le répertoire `/usr/sbin` qui n'est pas cité dans la variable `PATH` :

```
$ find / -name ping 2> /dev/null
/usr/sbin/ping
$
```

Le répertoire courant n'est pas exploré s'il n'est pas cité dans `PATH` :

```
$ cd /usr/sbin
$ ping localhost
ksh: ping: not found
$
```

Modifier le contenu de la variable `PATH` :

```
$ PATH=$PATH:/usr/sbin
$ echo $PATH
/usr/bin:/bin:/usr/sbin
$
```

La commande `ping` est trouvée :

```
$ ping localhost
localhost is alive
$
```

### Rechercher une commande dans le répertoire courant

Pour qu'une commande soit recherchée dans le répertoire courant, il faut ajouter en fin de variable PATH la chaîne ":" ou simplement le caractère ":".

#### Exemple

```
PATH=/usr/bin:/usr/local/bin:/home/christie/bin:.
```

Équivalent à :

```
PATH=/usr/bin:/usr/local/bin:/home/christie/bin:
```

### 1.4.3 PWD

ksh	bash
-----	------

Cette variable contient la valeur du répertoire courant. Elle est mise à jour par le shell dès que l'utilisateur change de répertoire. Cette variable peut être utilisée en ksh pour faire apparaître la valeur du répertoire courant dans le prompt.

### 1.4.4 PS1

Cette variable contient la chaîne de caractères représentant le prompt principal.

#### Exemple

```
$ echo $PS1
$
$ PS1='Entrez une commande => '
Entrez une commande => date
Thu Jan 30 17:27:51 MET 2014
Entrez une commande =>
```

Avec le ksh et le bash, il est possible de paramétrer son prompt de telle façon qu'il contienne en permanence la valeur du répertoire courant.

# 86 Programmation shell

sous Unix/Linux - ksh, bash, sh (avec exercices corrigés)

## Faire apparaître la valeur du répertoire courant dans le prompt en ksh

Il faut se servir de la variable PWD.

### Exemple

Ici, le prompt est composé de deux caractères : le symbole "\$" suivi d'un espace (cf. figure 1) :

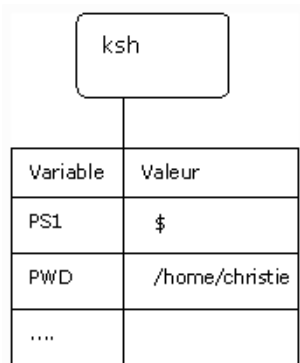


Figure 1 : Initialisation de PS1 avec le répertoire courant (1)

```
$  
$ echo -$PS1-  
-$ -  
$
```

Le répertoire courant est **/home/christie** :

```
$  
$ pwd  
/home/christie  
$
```

Initialisation de PS1 avec la chaîne de caractères '\$PWD\$'; il faut empêcher le shell de substituer \$PWD par sa valeur au moment de l'affectation, donc il faut protéger l'expression avec des quotes (cf. figure 2) :

```
$ PS1=' $PWD$ '
```