

BRIAN W. KERNIGHAN et ROB PIKE

LA PROGRAMMATION EN PRATIQUE



CLARTÉ. SIMPLICITÉ. GÉNÉRALITÉ. EFFICACITÉ. PRODUCTIVITÉ
SÉRIE LANGAGES – PROGRAMMATION

Comment gérer spécifications et contraintes, choisir parmi plusieurs conceptions possibles, déboguer et tester, améliorer les performances, assurer la portabilité et la maintenance des logiciels.

Comment optimiser la robustesse et la fiabilité, écrire un code efficace et agréable à lire, choisir les langages et outils qui laisseront la machine réaliser le maximum de tâches.

VUIBERT INFORMATIQUE

2-005-569-1

BRIAN W. KERNIGHAN ET ROB PIKE

La programmation En pratique

Traduction de Thierry Thaureaux



Vuibert Informatique

Table des matières

Preface

Chapitre 1 Style

- 1.1 Noms 3
- 1.2 Expressions et instructions 7
- 1.3 Cohérence et expressions idiomatiques 12
- 1.4 Macros de fonction 20
- 1.5 Les nombres magiques 21
- 1.6 Commentaires 25
- 1.7 Pourquoi se donner du mal ? 31
- Lectures complémentaires 31

Chapitre 2 Algorithmes et structures de données

- 2.1 Recherche 34
- 2.2 Tri 36
- 2.3 Bibliothèques 39
- 2.4 Tri par segmentation en Java 42
- 2.5 Notation O 45
- 2.6 Tableaux dynamiques 47
- 2.7 Listes 50
- 2.8 Arbres 56
- 2.9 Tables de hachage 62
- 2.10 Résumé 66
- Lectures complémentaires 67

Chapitre 3 Conception et implémentation

- 3.1 L'algorithme de la chaîne de Markov 70
- 3.2 Choix d'une structure de données 72

XIII

1

33

69

x	3.3 Construire la structure de données en C	74
	3.4 Générer des données en sortie	78
	3.5 Java	81
	3.6 C++	86
	3.7 Awk et Perl	89
	3.8 Performance	91
	3.9 Leçons	93
	Lectures complémentaires	95

Chapitre 4 Interfaces

97

4.1 Valeurs séparées par des virgules	98
4.2 Une bibliothèque prototype	100
4.3 Une bibliothèque pour les autres	104
4.4 Implémentation en C++	115
4.5 Principes d'une interface	119
4.6 Gestion des ressources	123
4.7 Abandonner, réessayer, échec ?	126
4.8 Interfaces utilisateur	131
Lectures complémentaires	133

Chapitre 5 Débogage

135

5.1 Débogueurs	136
5.2 Bons indices, débogage facile	138
5.3 Aucun indice, bogues difficiles	143
5.4 Derniers recours	148
5.5 Bogues non reproductibles	151
5.6 Outils de débogage	153
5.7 Autres bogues humains	156
5.8 Résumé	158
Lectures complémentaires	159

Chapitre 6 Tests

161

6.1 Testez tout en écrivant votre code	162
6.2 Évaluation systématique	168
6.3 Automatisation des tests	172
6.4 Structures de tests	175
6.5 Tests de charge	180
6.6 Astuces pour les tests	183

6.7 Qui effectue les tests ?	185
6.8 Évaluation du programme de Markov	186
6.9 Résumé	188
Lectures complémentaires	189

Chapitre 7 Performance 191

7.1 Un goulot d'étranglement	192
7.2 Chronométrage et <i>profiling</i>	199
7.3 Stratégies en matière de vitesse	204
7.4 Mise au point du code	208
7.5 Gestion efficace de l'espace mémoire	213
7.6 Estimation	217
7.7 Résumé	220
Lectures complémentaires	220

Chapitre 8 Portabilité 223

8.1 Le langage	224
8.2 Fichiers d'en-tête et bibliothèques	231
8.3 Organisation du programme	234
8.4 Isolation	238
8.5 Échanges de données	240
8.6 Ordre des octets	241
8.7 Portabilité et mise à jour	245
8.8 Internationalisation	248
8.9 Résumé	250
Lectures complémentaires	251

Chapitre 9 Notation 253

9.1 Données de formatage	254
9.2 Expressions régulières	261
9.3 Outils programmables	269
9.4 Interpréteurs, compilateurs et machines virtuelles	272
9.5 Les programmes qui écrivent des programmes	279
9.6 Utilisation de macros pour générer du code	282
9.7 Compilation à la volée	284
Lectures complémentaires	289

Épilogue 291

Récapitulatif des règles 295

Index 299