

Programmation concurrente et temps réel avec Java

Luigi Zaffalon

```
import javax.realtime.*;
public class ATC {
    static boolean interrompu = false;
    static LectureInterruptible lecture =
        new LectureInterruptible();
    static OneShotTimer compteRebours;
    static AsyncEventHandler actionUnique =
        new AsyncEventHandler(new PriorityParametre(
            null, null, null, null, true, null) {
            public void handleAsyncEvent() {
                aie.fire();
            }
        });
    // -----
    static AsynchronouslyInterruptedException aie =
        new AsynchronouslyInterruptedException();
    static Interruptible inter = new Interruptible() {
        public void run(AsynchronouslyInterruptedException
            throws AsynchronouslyInterruptedException {
            // Code susceptible d'être interrompu:
            try { char car = lecture.lireCar(); }
            catch (InterruptedException ie) {}
            System.err.println("Bouton pressé!");
            aie.clear();
        }
    };
    public void interruptAction(
        AsynchronouslyInterruptedException aie) {
        // Action à entreprendre en cas d'interruption:
        System.err.println("Alarme!");
    }
    // -----
    public static void main(String[] args) {

```



Programmation concurrente et temps réel avec Java

Luigi Zaffalon



Les Presses polytechniques et universitaires romandes sont une fondation académique dont le but est principalement la diffusion des travaux de l'École polytechnique fédérale de Lausanne ainsi que d'autres universités et écoles d'ingénierie francophones. Le catalogue de leurs publications peut être obtenu par courrier aux Presses polytechniques et universitaires romandes: BPFL - Centre Misi, CH-1015 Lausanne, par E-Mail à pun@epfl.ch par téléphone au (0)21 693 41 40, ou par fax au (0)21 693 40 22.

www.pun.ch

1^{ère} édition
ISBN 978-2-88074-698-8

© Presses polytechniques et universitaires romandes, 2007
CH-1015 Lausanne

Tous droits réservés

Reproduction même partielle, sous quelque forme que ce soit, sans l'accord écrit de l'éditeur.

PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES

TABLE DES MATIÈRES

AVANT-PROPOS		XV
Chapitre 1 INTRODUCTION		1
1.1	Systèmes réactifs et temps réel - - - - -	1
1.2	Compléments - - - - -	3
1.3	Architecture fonctionnelle des systèmes temps réel - - - - -	4
1.4	Programmation parallèle ou concurrente - - - - -	8
1.5	Etats et transitions des threads et des processus - - - - -	12
1.6	Coroutines - - - - -	16
1.7	Contexte d'exécution et noyau- - - - -	18
1.8	Exemple d'ordonnancement de processus - - - - -	21
1.9	Processus lourds et processus légers - - - - -	23
1.10	Formalismes pour la programmation concurrente - - - - -	23
1.11	Exclusion mutuelle - - - - -	32
1.12	Gestion de ressources- - - - -	35
1.13	Gestion du temps- - - - -	36
1.14	Synchronisme versus asynchronisme - - - - -	42
1.15	Propriétés usuelles des langages dédiés à la programmation temps réel- - - - -	43
1.16	Langages pour le développement d'applications embarquées et temps réel - - - - -	45
1.17	Remarques et commentaires - - - - -	47
1.18	Exercices - - - - -	48
Chapitre 2 THREADS JAVA		51
2.1	Threads Java- - - - -	51
2.2	Création de threads Java basée sur la classe Thread- - - - -	57
2.3	Création de threads Java basée sur l'interface Runnable- - - - -	63
2.4	Orientation objet - - - - -	67
2.5	Crible d'Eratosthène - - - - -	69
2.6	Gestion de locomotives sur un simulateur de maquettes de trains miniatures - - - - -	71
2.7	Exemple d'activation récursive de threads- - - - -	76

2.8	Synchronisation de threads - - - - -	79
2.9	Mécanismes de transfert du contrôle - - - - -	82
2.10	Mécanismes de réordonnancement- - - - -	86
2.11	Compléments relatifs à la classe Thread - - - - -	86
2.12	Threads système - - - - -	87
2.13	Threads temps réel RTSJ - - - - -	89
2.14	Threads et exceptions - - - - -	94
2.15	Remarques et commentaires - - - - -	99
2.16	Exercices - - - - -	99
Chapitre 3	MÉCANISMES D'EXCLUSION MUTUELLE PAR ATTENTE ACTIVE	103
3.1	Section critique et exclusion mutuelle - - - - -	103
3.2	Exemple trivial en Java - - - - -	107
3.3	Contrôle d'accès aux variables partagées en Java - - - - -	110
3.4	Algorithmes d'exclusion mutuelle - - - - -	114
3.5	Algorithme de Fischer - - - - -	130
3.6	Algorithme rapide de Lamport - - - - -	133
3.7	Remarques relatives aux algorithmes d'exclusion mutuelle - -	133
3.8	Solution matérielle au problème de l'exclusion mutuelle - - -	134
3.9	Algorithmes basés sur des instructions matérielles spéciales -	136
3.10	Variables atomiques Java 5- - - - -	142
3.11	Remarques et commentaires - - - - -	144
3.12	Exercices - - - - -	145
Chapitre 4	VERROUS ET SÉMAPHORES	151
4.1	Exclusion mutuelle basée sur l'attente passive - - - - -	151
4.2	Verrous - - - - -	153
4.3	Gestion de ressources basée sur les verrous - - - - -	156
4.4	Sémaphores - - - - -	163
4.5	Contrôle de la concurrence - - - - -	166
4.6	Gestion de ressources basée sur les sémaphores - - - - -	169
4.7	Définition faible ou forte des sémaphores - - - - -	170
4.8	Sémaphores privés - - - - -	171
4.9	Barrière de synchronisation - - - - -	177
4.10	Remarques et commentaires - - - - -	178
4.11	Exercices - - - - -	179
Chapitre 5	PRODUCTEURS-CONSOMMATEURS	185
5.1	Introduction - - - - -	185
5.2	Tampon double- - - - -	187
5.3	Tampon borné - - - - -	190
5.4	Généralisation à plusieurs producteurs et plusieurs consommateurs- - - - -	194

5.5	Amélioration des performances - - - - -	196
5.6	Tampon borné sans primitives de synchronisation - - - - -	199
5.7	Remarques et commentaires - - - - -	200
5.8	Exercices - - - - -	201
Chapitre 6	LECTEURS-RÉDACTEURS	205
6.1	Introduction - - - - -	205
6.2	Comportements des threads et priorités d'accès - - - - -	206
6.3	Solution avec priorité aux lecteurs - - - - -	208
6.4	Solution avec priorités égales - - - - -	215
6.5	Solution avec priorité aux rédacteurs - - - - -	218
6.6	Généralisation du problème à n classes - - - - -	222
6.7	Remarques et commentaires - - - - -	226
6.8	Exercices - - - - -	226
Chapitre 7	EXCLUSION MUTUELLE JAVA	231
7.1	Méthodes mutuellement exclusives - - - - -	231
7.2	Méthodes synchronisées ou non d'un même objet- - - - -	232
7.3	Synchronisation d'objet ou de bloc- - - - -	234
7.4	Contrôle centralisé de la synchronisation- - - - -	236
7.5	Accès synchronisé à des éléments statiques - - - - -	236
7.6	Synchronisation de classe - - - - -	237
7.7	Variables privées et méthodes statiques- - - - -	238
7.8	Singletons et verrouillage - - - - -	240
7.9	Restitution et allocation du verrou de protection- - - - -	242
7.10	Emboîtements - - - - -	243
7.11	Interblocages - - - - -	243
7.12	Tri multiple - - - - -	246
7.13	Verrous et famine - - - - -	252
7.14	Verrous Java 5 - - - - -	255
7.15	Exemple de gestion d'une ressource critique simple - - - - -	259
7.16	Exceptions et finalisation - - - - -	260
7.17	Remarques et commentaires - - - - -	261
7.18	Exercices - - - - -	262
Chapitre 8	MONITEURS JAVA	267
8.1	Introduction - - - - -	267
8.2	Mécanismes de synchronisation par objet - - - - -	268
8.3	Barrière de synchronisation équitable - - - - -	275
8.4	Implémentation de sémaphores faibles - - - - -	277
8.5	Implémentation de sémaphores forts- - - - -	279
8.6	Modèle producteur-consommateur - - - - -	283
8.7	Gestion de ressources identiques - - - - -	289

8.8	Moniteurs de Hoare	293
8.9	Modèle des lecteurs-rédacteurs	298
8.10	Java 5	303
8.11	Synthèse relative au concept de moniteur	308
8.12	Anomalie d'héritage induite par la concurrence	311
8.13	Interblocages	312
8.14	Remarques et commentaires	313
8.15	Exercices	313
Chapitre 9	PROCESSUS SÉQUENTIELS COMMUNICANTS	321
9.1	Présentation	321
9.2	Processus CSP et threads Java	322
9.3	Processus et communication en Occam 2	322
9.4	Composition parallèle de processus JCSP	324
9.5	Composition séquentielle de processus JCSP	327
9.6	Communication entre processus JCSP	328
9.7	Système pipeline	330
9.8	Gestion du non déterminisme	333
9.9	Exemple de concentrateur	340
9.10	Modèle producteur-consommateur	341
9.11	Modèle lecteurs-rédacteurs	345
9.12	Allocateur de ressources segments	348
9.13	Modèle clients-serveur	351
9.14	Sommes partielles avec CTJ	355
9.15	Interblocages	360
9.16	Remarques et commentaires	362
9.17	Exercices	363
Chapitre 10	HORLOGES ET MINUTERIES	369
10.1	Introduction	369
10.2	Appréhension du temps	369
10.3	Sources de temps et modèles de gestion de minuteriers	370
10.4	Expression et manipulation du temps en Java	372
10.5	Minuteriers Java	379
10.6	Gestion du temps avec Java 5	391
10.7	Gestion du temps PERC	391
10.8	Horloges RTSJ	394
10.9	Minuteriers RTSJ	399
10.10	Exemple de minuterier de cuisine	404
10.11	Attente temporisée de haute résolution	407
10.12	Remarques et commentaires	408
10.13	Exercices	409

Chapitre 11	ORDONNANCEMENT ET THREADS TEMPS RÉEL	413
11.1	Introduction	413
11.2	Algorithmes d'ordonnancement	414
11.3	Modèle d'ordonnancement Java	415
11.4	Priorités Java	417
11.5	Exemple d'ordonnancement de threads Java	420
11.6	Modèle d'ordonnancement Java 5	423
11.7	Modèle d'ordonnancement PERC	423
11.8	Modèle d'ordonnancement RTSJ	425
11.9	Phénomène d'inversion de priorité	428
11.10	Protocole d'héritage de priorité	429
11.11	Protocoles à priorité plafond	432
11.12	Protocole SRP	436
11.13	Prévention de l'inversion de priorité en Java	438
11.14	Caractérisation des threads temps réel	444
11.15	Politiques d'ordonnancement temps réel	445
11.16	Politiques d'ordonnancement statique	446
11.17	Ordonnancement à taux monotone	451
11.18	Test sur les bornes d'achèvement avec prise en compte du blocage	457
11.19	Ordonnancement par échéance	458
11.20	Ordonnancement périodique avec RTSJ	462
11.21	Objets ordonnançables RTSJ	463
11.22	Changement d'ordonnanceur	472
11.23	Ordonnancement dynamique	473
11.24	Ordonnancement de threads non périodiques	481
11.25	Paramètres de gestion de la mémoire et des groupes de threads	482
11.26	Remarques et commentaires	483
11.27	Exercices	484
Chapitre 12	ASYNCHRONISME	491
12.1	Introduction	491
12.2	Gestionnaires d'événements asynchrones	492
12.3	Gestionnaires d'événements asynchrones directement associés à des threads	495
12.4	Exemple du bandit manchot	497
12.5	Transfert de contrôle asynchrone	505
12.6	Signaux POSIX	512
12.7	Traitements des activités sporadiques	514
12.8	Remarques et commentaires	514
12.9	Exercices	517

Chapitre 13	GESTION MÉMOIRE	525
13.1	Introduction - - - - -	525
13.2	Gestion mémoire PERC - - - - -	526
13.3	Gestion mémoire RTSJ - - - - -	526
13.4	Threads et domaines - - - - -	531
13.5	Gestion des accès aux domaines mémoire - - - - -	533
13.6	Mécanisme de portail - - - - -	537
13.7	Communications entre threads - - - - -	539
13.8	Remarques et commentaires - - - - -	541
13.9	Exercices - - - - -	542
Chapitre 14	PROGRAMMATION SYSTÈME	545
14.1	Introduction - - - - -	545
14.2	Organisation physique des périphériques - - - - -	547
14.3	Mode de commande synchrone des entrées-sorties - - - - -	548
14.4	Mode de commande asynchrone des entrées-sorties - - - - -	549
14.5	Accès au matériel en Ada 95 - - - - -	555
14.6	Accès au matériel avec Java - - - - -	560
14.7	Méthodes natives Java - - - - -	561
14.8	Accès au matériel avec RTSJ - - - - -	564
14.9	Allocation en mémoire physique - - - - -	567
14.10	Remarques et commentaires - - - - -	569
14.11	Exercices - - - - -	569
	BIBLIOGRAPHIE	573
	INDEX	589