



# THESE

*par*

Olivier Potonniée

*Pour obtenir le titre de*

DOCTEUR DE L'UNIVERSITE PARIS VI

Spécialité : INFORMATIQUE

**Etude et prototypage en ESTEREL de  
la gestion de processus d'un micro-  
noyau de système d'exploitation  
réparti avec garantie de service**

Soutenu le 02 avril 1996, devant le jury composé de :

Président	Claude GIRAULT
Rapporteurs	Roland BALTER Jean-Paul RIGAULT
Examineurs	Charles ANDRE Michel GIEN François HORN Sacha KRAKOWIAK Jean-Bernard STEFANI

## Résumé

L'utilisation de l'approche synchrone dans la programmation des systèmes d'exploitation permet d'obtenir une plate-forme déterministe et prouvée pour bâtir des applications temps-réel.

Cette thèse présente une réalisation modulaire de la gestion des tâches à l'aide du langage synchrone ESTEREL. Nous montrons quelques types de propriétés non-triviales qui ont pu être vérifiées automatiquement sur le code ESTEREL. Nous définissons la machine d'exécution nécessaire à son fonctionnement, et en décrivons une implantation expérimentale.

Nous montrons comment, dans une configuration répartie, plusieurs exemplaires de cette gestion des tâches peuvent être synchronisés, ceci afin d'offrir un exécutif synchrone réparti, qui facilite la réalisation des applications temps-réel distribuées.

**Mots clefs :** Programmation synchrone, systèmes répartis, gestion de tâches, systèmes déterministes, systèmes temps-réel

## Abstract

*Study and prototyping in Esterel of process management in a distributed operating system's microkernel with service guarantees.*

The use of the synchronous programming approach in operating systems implementation allows the construction of provable, deterministic platforms to support real-time applications.

This thesis describes a modular process management written in Esterel, a synchronous programming language. We present non-trivial properties that have been verified automatically on the Esterel code. The execution machine necessary to support its correct behavior is defined, and we describe an implementation of it.

We also present how several instances of this process management can be synchronized in a distributed context, yielding a synchronous platform that should facilitate the development of distributed real-time applications.

**Key words :** Synchronous programming, distributed systems, process management, real-time systems, deterministic systems

# Table des matières

<b>Table des matières.....</b>	<b>v</b>
<b>Introduction .....</b>	<b>1</b>
<b>1. Objectifs et état de l'art .....</b>	<b>7</b>
1.1. Présentation d'ODP .....	7
1.1.1. Définitions .....	8
1.1.2. Les points de vue de RM-ODP .....	9
1.1.2.1. Le point de vue des traitements .....	9
1.1.2.2. Le point de vue d'ingénierie.....	11
1.1.3. Les fonctions ODP .....	11
1.1.4. Les bienfaits du modèle.....	12
1.2. Réalisation d'un plate-forme ODP avec garantie de service.....	12
1.2.1. Gestion de la QoS .....	12
1.2.1.1. Du point de vue des traitements : Spécification de QoS	13
1.2.1.2. Aspect d'ingénierie : traitement des contraintes de QoS	15
1.2.2. Gestion du contrôle.....	17
1.2.2.1. Du point de vue des traitements : l'approche synchrone	17
1.2.2.2. Aspect d'ingénierie : exécution des objets de contrôle....	20
1.2.2.3. Aspect technologique : choix du langage .....	21
1.2.3. Les micro noyaux : Evolutifs et modulaires.....	21
1.3. Comparaison avec d'autres travaux.....	23
1.3.1. Mars .....	24
1.3.2. Arts .....	25
1.3.3. Spring.....	26
1.3.4. Maruti.....	28
1.3.5. Delta-4 .....	29
1.3.6. CHAOSarc .....	29
1.3.7. Bilan.....	30
1.4. Problématique et plan .....	31
1.4.1. Gestion synchrone des tâches .....	31
1.4.2. Synchronisation répartie.....	33
<b>2. Gestion synchrone des tâches.....</b>	<b>37</b>
2.1. Portée de l'étude .....	37

2.2.	La gestion des tâches Chorus.....	38
2.2.1.	Les interruptions.....	38
2.2.1.1.	Origine des interruptions.....	38
2.2.1.2.	Synchronisme / asynchronisme des interruptions .....	39
2.2.2.	Les threads, leur vie .....	40
2.2.2.1.	Les états d'un thread Chorus.....	41
2.2.2.2.	Les événements .....	42
2.2.3.	L'ordonnancement .....	44
2.2.4.	La gestion du temps .....	44
2.2.5.	Synchronisation dans le noyau .....	45
2.2.5.1.	Verrouillage du noyau .....	45
2.2.5.2.	Gestion des threads.....	45
2.2.5.3.	Gestion des interruptions .....	46
2.2.5.4.	Gestion du temps.....	47
2.3.	Modélisation .....	47
2.3.1.	Objectifs .....	47
2.3.2.	Définitions .....	48
2.3.3.	Présentation des composants .....	48
2.3.4.	Vérifications.....	50
2.3.4.1.	Vérifications graphiques .....	51
2.3.4.2.	Vérifications formelles .....	51
2.3.5.	Les interruptions : composant Interrupt .....	53
2.3.5.1.	Le module InterruptMgr .....	54
2.3.5.2.	Le module SysMode.....	58
2.3.6.	La gestion du temps : composant TimeMgr.....	59
2.3.6.1.	Gestion des timers : module TimerMgr.....	60
2.3.6.2.	Gestion de l'interruption horloge : module TimeGuard .	61
2.3.7.	Les états d'un thread : Composant Thread .....	62
2.3.7.1.	Gestion des suspensions : module thSuspend .....	63
2.3.7.2.	Gestions des arrêts : module thStop.....	64
2.3.7.3.	Gestion des attentes : module thSleep .....	67
2.3.7.4.	Gestions de l'état global : module thMain .....	70
2.3.7.5.	Gestions du mode d'exécution : module thTrapExc .....	72
2.3.8.	L'ordonnancement : composant Scheduler .....	74
2.3.8.1.	Ordonnancement générique : module SchedBase .....	76
2.3.8.2.	Gestion du temps partagé : module TimeSlice.....	79
2.3.9.	Récapitulatif .....	81
2.4.	Conclusion.....	82
<b>3.</b>	<b>Mise en oeuvre du modèle.....</b>	<b>83</b>
3.1.	Composition des modules .....	83
3.1.1.	Le séquençement synchrone.....	84
3.1.2.	Le parallèle asynchrone .....	85
3.1.3.	Application à la gestion des tâches .....	85
3.2.	Machine d'exécution .....	87

3.2.1.	Collecteur d'événements d'entrée externes .....	88
3.2.2.	Constructeur de l'événement de réaction .....	88
3.2.3.	Exécution d'une transition .....	89
3.2.4.	Collecteur de l'événement sortant .....	91
3.2.5.	Emetteur d'événements de sortie.....	91
3.2.6.	Analyse de la durée des réactions.....	92
3.2.6.1.	Réaction d'un objet synchrone .....	92
3.2.6.2.	Réaction de l'ensemble des objets synchrones.....	93
3.2.7.	Communications.....	94
3.3.	Comment ça se passe ? .....	94
3.3.1.	Un exemple .....	94
3.3.2.	Commentaires.....	95
3.4.	Implantation du système .....	96
3.4.1.	Objectif et choix d'implantation .....	96
3.4.2.	Interfaces de la gestion des tâches.....	98
3.4.2.1.	Interface utilisateur .....	98
3.4.2.2.	Interface avec le noyau.....	100
3.4.3.	Exécution des réactions du noyau synchrone.....	100
3.4.4.	Transmission des interruptions au SVP.....	101
3.4.5.	Performances .....	102
3.4.5.1.	Taille du code.....	102
3.4.5.2.	Temps d'exécution .....	103
3.4.5.3.	Optimisations .....	104
3.5.	Conclusion.....	108
<b>4.</b>	<b>Exécutif synchrone réparti.....</b>	<b>111</b>
4.1.	Les modèles des objets réactifs répartis .....	111
4.2.	Le modèle synchrone faible .....	113
4.2.1.	Le modèle sémantique.....	113
4.2.2.	Le modèle d'exécution .....	114
4.3.	Application à la gestion des tâches .....	118
4.3.1.	Comportement temporel.....	118
4.3.2.	Les communications entre les tâches .....	119
4.3.2.1.	Intégration dans le micro-noyau synchrone .....	120
4.3.2.2.	Intégration dans les SVP .....	122
4.3.2.3.	Interface utilisateur .....	122
4.3.3.	Le déclencheur de réaction.....	123
4.3.3.1.	Le déclencheur de base .....	124
4.3.3.2.	Le déclencheur synchronisé à vague .....	124
4.3.3.3.	Le déclencheur parsemé.....	126
4.4.	Démonstration du synchronisme de l'exécutif.....	127
4.4.1.	Présentation de l'application "Tortues" .....	127

4.4.2. Fonctionnement de la démonstration .....	128
4.4.3. Résultats .....	129
4.4.3.1. Sur un exécutif non synchronisé.....	130
4.4.3.2. Sur un exécutif synchronisé par déclencheur à vague. 131	
4.4.3.3. Sur un exécutif synchronisé par déclencheur clairsemé132	
4.5. Conclusion.....	134
<b>Conclusion et perspectives.....</b>	<b>135</b>
<b>Bibliographie.....</b>	<b>139</b>
<b>Annexes.....</b>	<b>145</b>
1 Chorus.....	145
2 Le langage synchrone Esterel .....	148
2.1 Les entrées d'un programme Esterel.....	148
2.2 Les primitives Esterel .....	148
2.2.1 Les primitives impératives .....	148
2.2.2 Les primitives temporelles.....	149
2.2.3 L'exécution de tâches asynchrones .....	151
2.3 Les paradoxes temporels.....	152
2.4 La compilation d'un programme Esterel .....	152
3 Code Esterel de la gestion des tâches .....	154