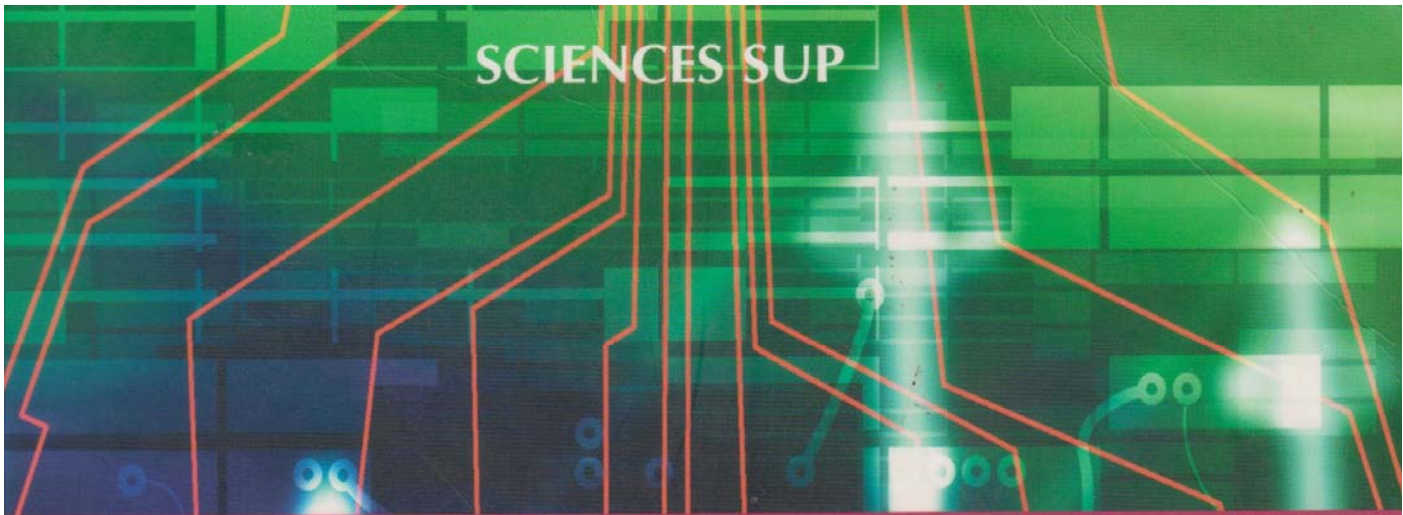




SCIENCES SUP

Cours et exercices corrigés

Licence 3^e année • Master • Écoles d'ingénieurs

108

LES SYSTÈMES D'EXPLOITATION

Unix, Linux et Windows XP
avec C et Java

B-103

Samia Bouzefrane

DUNOD

2-005-676-3



2-005-676-1

LES SYSTÈMES D'EXPLOITATION

Unix, Linux et Windows XP
avec C et Java

Samia Bouzefrane

Maître de conférences au CNAM

DUNOD

Table des matières

CHAPITRE 1 • GESTION DES PROCESSUS ET DES THREADS

1.1	Notion de processus	2
1.2	Contexte d'un processus	2
1.3	Droits d'un processus	4
1.4	Processus dans le système Unix	5
1.4.1	Définitions	5
1.4.2	Le processus et ses propriétaires	7
1.4.3	Les ressources de base d'un processus	9
1.4.4	L'arrêt d'un processus	12
1.4.5	Les états d'un processus Unix	12
1.4.6	L'ordonnancement des processus	13
1.5	Création de processus Unix	14
1.5.1	Utilisation de fork	14
1.5.2	Héritage du fils	15
1.6	Synchronisation père-fils	17
1.6.1	Les processus zombis	17
1.6.2	Les primitives wait et waitpid	18
1.7	Variables d'environnement et paramètres	25
1.7.1	Le passage de paramètres	25
1.7.2	Les variables d'environnement	27

1.8	Primitives de recouvrement	29
1.8.1	L'exécution d'un fichier binaire	29
1.8.2	La description des primitives exec	31
1.9	Threads d'Unix	33
1.9.1	Création et activation d'une thread	34
1.9.2	Terminaison d'une thread	34
1.9.3	Libération des ressources d'une thread	35
1.9.4	Libération du processeur	35
1.9.5	Attente de la terminaison d'une thread	35
1.9.6	Comportement vis-à-vis des cancels	35
1.9.7	Exécution d'une routine d'initialisation	36
1.9.8	Récupérer son propre identificateur	36
1.10	Threads dans la machine virtuelle Java	46
1.10.1	La création d'une thread	46
1.10.2	Trouver la thread courante	48
1.10.3	Méthodes de nommage des threads	49
1.10.4	Méthodes start et isAlive	50
1.10.5	Méthode sleep	51
1.10.6	Threads démons	52
1.10.7	Groupes de threads	54
CHAPITRE 2 • INTERRUPTIONS ET SIGNAUX		61
2.1	Niveaux d'interruption et priorités	62
2.2	Déroulement d'une interruption dans un système non hiérarchisé	63
2.3	Déroulement d'une interruption dans un système d'interruptions hiérarchisé	64
2.4	Masquage des interruptions	65
2.5	Conditions de prise en compte d'une interruption	65
2.6	Types d'interruptions	66
2.7	Analogie et différences entre interruption et déroutement	67
2.8	États d'un niveau d'interruption	68
2.9	Exemples d'utilisation des interruptions	68
2.9.1	Prélèvement de mesures en contrôle de processus	69
2.9.2	Délai de garde (Time Out)	69
2.9.3	Partage du temps CPU (Time Sharing)	69
2.10	Exemples d'utilisation de déroutements	70
2.10.1	Mesure de la taille d'une mémoire	70
2.10.2	Personnalisation du traitement des déroutements	70
2.10.3	Simulation d'une instruction manquante	70
2.11	Signaux sous Unix	71

2.12	Types de signaux	72
2.13	Envoi d'un signal : la primitive <code>kill</code>	74
2.14	Attente d'un signal	74
2.14.1	Primitive <code>pause</code>	74
2.14.2	Primitive <code>sigsuspend</code>	75
2.15	Traitement par défaut et traitement spécifique	75
2.16	Installation de nouveaux handlers	75
2.16.1	Structure <code>sigaction</code>	75
2.16.2	Primitive <code>sigaction</code>	76
2.16.3	Exemple : installation des handlers <code>SIG_IGN</code> et <code>SIG_DFL</code> avec la primitive <code>sigaction</code>	77
2.16.4	Exemple illustrant l'utilisation de la primitive <code>sigaction</code> et du signal <code>SIGUSR1</code>	78
2.16.5	Exemple illustrant l'utilisation de <code>SIGUSR1</code> : perte de signaux	80
2.16.6	Réalisation d'un rendez-vous entre deux processus à l'aide de signaux	81
2.17	Opérations sur les ensembles de signaux	83
2.17.1	Masquage et démasquage d'un ensemble de signaux	84
2.17.2	Liste des signaux pendants	84
2.18	Signaux particuliers	86
2.18.1	Signal <code>SIGHUP</code>	86
2.18.2	Signaux <code>SIGINT</code> et <code>SIGQUIT</code>	86
2.18.3	Signal <code>SIGSEGV</code>	87
2.18.4	Signal <code>SIGALRM</code> et primitives <code>alarm</code> et <code>ualarm</code>	89
2.19	Système d'interruptions des processeurs Intel	92
2.19.1	Différents types d'interruptions	93
2.19.2	Masquage des interruptions	94
2.19.3	Déroulement d'une interruption	94
2.19.4	Emplacements réservés	94
2.19.5	Fonctionnement du PIC 8259	95
2.19.6	Schéma de principe d'un codeur de priorité	96
2.19.7	Appels superviseur	97
2.19.8	Contrôle du temps	97
2.20	Sauvegarde et restauration de contexte	100
2.20.1	Primitives <code>setjmp</code> et <code>longjmp</code>	100
2.20.2	Exemple d'utilisation de <code>setjmp</code> et <code>longjmp</code> pour intercepter <code>SIGFPE</code>	100
2.20.3	Utilisation de l'horloge temps réel (Real Time Clock ou RTC)	102
2.21	Interruption des threads dans la MVI	108
2.21.1	Interrompre une thread active	108
2.21.2	Interrompre une thread endormie	109
2.21.3	Interrompre un groupe de threads	109
2.21.4	Interruptions pendantes	110

CHAPITRE 3 • GESTION DE LA MÉMOIRE CENTRALE	113
3.1 Inconvénients de la mémoire unique	114
3.2 Caractéristiques d'une mémoire hiérarchisée	114
3.3 Mécanismes d'adressage	114
3.4 Allocation dans une mémoire hiérarchisée : technique du swapping	115
3.4.1 Swapping global	116
3.4.2 Swapping à la demande	116
3.5 Gestion de la mémoire par zones	116
3.5.1 Représentation des zones libres	117
3.5.2 Sélection d'une zone à allouer	117
3.5.3 Libération d'une zone occupée	118
3.5.4 Cas d'aucune zone convenable	118
3.6 Gestion de la mémoire par pages	118
3.6.1 Représentation des espaces réel et virtuel	119
3.6.2 Stratégies de chargement	120
3.6.3 Pagination à un niveau	120
3.6.4 Pagination à plusieurs niveaux	126
3.7 Segmentation	127
3.7.1 Segmentation sans pagination	127
3.7.2 Segmentation avec pagination	129
3.8 Segmentation et pagination dans le Pentium	129
3.8.1 Segmentation sans pagination	129
3.8.2 Pagination	131
3.9 La pagination dans le système Linux	133
3.9.1 Le partage des pages	134
3.9.2 Les descripteurs de pages	135
3.9.3 Opérations sur les pages mémoires Linux	136
3.10 Techniques de ramasse-miettes	144
3.10.1 Techniques de comptage	145
3.10.2 Technique du parcours du graphe	146
3.11 Gestion mémoire dans la machine virtuelle Java	147
3.11.1 Niveaux d'accessibilité	148
3.11.2 Forcer la finalisation et le ramassage de miettes	152
CHAPITRE 4 • GESTION DES RESSOURCES ET ORDONNANCEMENT	155
4.1 Allocation d'une ressource	155
4.2 Traitement des demandes de ressources	156

4.3	Stratégies d'allocation	158
4.3.1	Stratégies non préemptives	158
4.3.2	Stratégies préemptives	159
4.4	Ordonnancement dans le système Linux	163
4.4.1	Les politiques d'ordonnancement	163
4.4.2	Changer de politique d'ordonnancement	165
4.5	Ordonnancement des threads sous Linux	174
4.5.1	Définir des attributs à la création d'une thread	174
4.5.2	Changer de politique et/ou de paramètres d'ordonnancement	175
4.5.3	Changer l'attribut d'une thread	176
4.6	Ordonnancement dans la MVJ	181
4.6.1	Constantes de priorité	181
4.6.2	Obtenir la priorité d'une thread	182
4.6.3	Changer la priorité d'une thread	182
4.6.4	États d'une thread	183
4.6.5	Libération volontaire du processeur	185
4.6.6	Priorité d'un groupe de threads	185
4.6.7	Classe <code>java.util.Timer</code>	186
4.6.8	Classe <code>java.util.TimerTask</code>	188
CHAPITRE 5 • GESTION DES FICHIERS		193
5.1	Types de fichiers	194
5.2	Système de fichiers	195
5.2.1	Mise en œuvre du système de fichiers	195
5.2.2	Conséquence sur les temps d'accès aux fichiers	201
5.3	Fast File System de BSD 4.2-4.3	201
5.4	Primitives système de gestion de fichiers	203
5.4.1	Opérations sur les fichiers	203
5.4.2	Ouverture de fichiers	203
5.4.3	Fermeture de fichiers	207
5.4.4	Opérations de lecture/écriture de données	207
5.4.5	Accès direct à un fichier	207
5.4.6	Sauvegarde des données modifiées de manière synchrone	208
5.5	Manipulation de fichiers	208
5.5.1	Création de liens symboliques	208
5.5.2	Suppression d'un fichier	209
5.5.3	Changement de nom pour un fichier	209
5.5.4	Changement de taille pour un fichier	209
5.5.5	Droits d'accès à un fichier	209
5.5.6	Modification du propriétaire d'un fichier	210

5.6	Gestion des répertoires	210
5.6.1	Création de répertoires	210
5.6.2	Suppression de répertoires	211
5.6.3	Accès au répertoire courant	211
5.7	Autres opérations sur les fichiers	211
5.7.1	Duplication de descripteurs d'entrée/sortie	211
5.7.2	Attributs de fichiers	212
5.7.3	Dates associées aux fichiers	214
5.7.4	Verrouillage d'un fichier	214
5.8	Gestion des fichiers dans la MVJ	225
5.8.1	Lire un flux de caractères	226
5.8.2	Écrire un flux de caractères	230
5.8.3	Manipuler un flux d'octets	235
5.8.4	Manipuler des fichiers et des répertoires	243
5.8.5	Accès direct à un fichier	245
CHAPITRE 6 • COMMUNICATION DES PROCESSUS ET DES THREADS		247
6.1	La communication par tubes dans le système Unix	248
6.1.1	Caractéristiques d'un tube	248
6.1.2	Tubes anonymes	248
6.1.3	Tubes nommés	259
6.2	La communication par tubes dans la MVJ	265
6.2.1	Classes de manipulation des tubes	265
6.2.2	Échanger des octets <i>via</i> un tube	266
6.2.3	Échanger des caractères <i>via</i> un tube	269
6.3	La communication par messages dans le système Unix	273
6.3.1	Obtention d'une file de messages	274
6.3.2	Envoi d'un message	274
6.3.3	Réception d'un message	275
6.3.4	Primitive <code>msgctl</code>	275
6.4	La communication par les sockets dans le système Unix	280
6.4.1	Processus et numéros de port	280
6.4.2	Mécanismes de mise en œuvre	282
6.4.3	Communication en mode fiable (mode TCP)	283
6.4.4	Communication en mode datagramme (mode UDP)	284
6.4.5	Adressage dans le domaine <code>AF_INET</code>	285
6.4.6	Adressage dans le domaine <code>AF_UNIX</code>	285
6.4.7	Primitives sur les sockets	285
6.4.8	Quelques fonctions auxiliaires	290
6.4.9	Exemple de processus communiquant en mode connecté	293
6.4.10	Exemple de processus communiquant en mode datagramme	299

6.5	La communication par sockets dans la MVJ	303
6.5.1	Manipuler des adresses internet	304
6.5.2	Communiquer en mode datagramme	305
6.5.3	Communiquer en mode flot de données (mode fiable)	310
CHAPITRE 7 • SYNCHRONISATION PAR SÉMAPHORES		319
7.1	Problème de l'exclusion mutuelle	320
7.2	Synchronisation par action directe	321
7.3	Synchronisation par attente active et scrutation de variables communes	322
7.3.1	Réalisation par hard	322
7.3.2	Réalisation par soft	325
7.4	Synchronisation avec mise en file d'attente et partage de variables communes	330
7.4.1	Synchronisation par verrous	330
7.4.2	Synchronisation par événements	331
7.4.3	Synchronisation par sémaphores	332
7.5	Généralisation de l'exclusion mutuelle	350
7.5.1	Exclusion mutuelle entre toutes les classes et à l'intérieur des classes sauf une	350
7.5.2	Exclusion mutuelle entre les classes mais pas à l'intérieur de certaines classes	353
7.5.3	Priorité explicite d'une classe sur les autres	357
7.6	Sémaphores de Vantilborgh	363
7.6.1	Cas où les lecteurs ont la priorité sur les rédacteurs	364
7.6.2	Cas où les rédacteurs ont la priorité sur les lecteurs	365
7.7	Sémaphores de Patil	367
7.8	Sémaphores avec message	369
7.9	Segments de mémoire partagés d'Unix	370
7.9.1	Création d'un segment de mémoire partagé	371
7.9.2	Attachement d'un segment	372
7.9.3	Détachement d'un segment	373
7.9.4	La primitive shmctl	373
7.10	Sémaphores d'Unix	380
7.10.1	Structures semid_ds et sembuf	380
7.10.2	Obtention d'un ensemble de sémaphores	381
7.10.3	Réalisation des opérations P et V : la primitive semop	381
7.10.4	Primitive semctl	382
7.10.5	Application	402
7.11	Commandes de gestion des IPC : ipcs et ipcrm	410
7.12	Exclusion mutuelle dans la MVJ	410
7.12.1	Accéder à une méthode en exclusion mutuelle	411
7.12.2	Accéder à un bloc en exclusion mutuelle	413
7.12.3	Imbrication de méthodes synchronized	414