IPC, SYNCHRONISATION ET EXCLUSION MUTUELLE

Il est recommandé d'utiliser un tableur pour réaliser les différents calculs. On rappelle que les supports sont disponibles à http://mathieu.delalandre.free.fr/teachings/operating1.html

1. Conditions de concurrence

On considère un cas de concurrence entre 4 processus A, B, C et D pour la gestion d'un répertoire de « Spooling », ou A-C travaillent en écriture et D est le processus « Daemon ». On considère le code suivant pour le traitement de ce répertoire, ou chaque instruction (1-7) est dite atomique (non interruptible par l'ordonnanceur).

P	(1)	P.in=in
	(2)	S[P.in] = P.name
	(3)	in = P.in+1
D	(4)	D.out=out
	(5)	D.name=S[D.out]
	(6)	out = D.out+1
	(7)	print

Notation

S	the spooling directory
in	current writing index of S
out	current reading index of S
P	a process
D	the printer daemon process
X.a	A data a part of a process X

Considérant la séquence d'ordonnancement ci-dessous, ou l'ordonnanceur passe la main tour à tour aux processus A, B, C et D pour un nombre d'instruction donné (i.e. au premier tour C exécute 1 instruction, B 2 instructions, etc.). Indiquez pour chacun des changements de contexte les informations suivantes: instructions exécutées, états des variables « in » et « out » (du répertoire de « Spooling » et des processus A, B, C et D), état de la variable « name » du « Daemon », état de la pile sur au moins deux niveaux consécutifs (i.e. S[in] et S[in+1]).

Pour les besoins de l'exemple, on considéra un cas de buffer vide à t_0 avec « in = out ». Vous pourrez définir une valeur d'initialisation pour « in » et « out » (e.g. 6). De manière à rendre votre analyse plus claire il est conseillé d'adopter une écriture condensée (i.e. une ligne par jeu d'instruction à chaque changement de contexte).

Nb d'instruction	1	2	1	1	2	4	2	4
Processus	C	B	A	B	A	D	C	D

Indiquez alors les problèmes rencontrés sur cette séquence en l'absence de dispositif de synchronisation. Vous pourrez vous aider de l'exemple présenté en cours.

2. Synchronisation et exclusion mutuelle - méthodes par réveil/activation

On considère un cas de synchronisation par méthode de réveil/activation entre 4 processus A, B, C et D par utilisation d'un sémaphore binaire type mutex. Soit le code suivant ou chaque instruction (1-5) est dite atomique (non interruptible par l'ordonnanceur).

sem is a semaphore, **P** is the process, (1) to (5) the instructions

(1) before the request
    do something ….
(2) down **sem**
(3) run in the critical section with **P**
    do something ….
(4) before the release
    do something ….
(5) up **sem**

Considérant la séquence d'ordonnancement suivante, ou le l'ordonnanceur passe la main tour à tour aux processus A, B, C et D pour un nombre d'instruction donné. Indiquez pour chacun des changements de contexte les informations suivantes: instructions exécutées, état du mutex (valeur et état de la queue d'exploitation), détention de la section critique, états des processus A, B, C et D.

Nb d'instruction	2	1	2	3	1	2	3	3	3
Processus	B	D	A	B	D	C	A	D	C

Sur l'ensemble de la synchronisation, indiquez les temps de réponse pour l'accès à la section, pour chacun des processus.

3. Synchronisation et exclusion mutuelle - méthodes par attente active

3.1. Synchronisation matérielle

On considère un cas de synchronisation matérielle entre 3 processus A, B et C par utilisation de l'instruction « Compare and Swap (CAS) ». Soit le code suivant ou chaque instruction (1-5) est dite atomique (non interruptible par l'ordonnanceur).

Procédure 1 (requête sur la section critique)

(1) Request the critical section with **P**
(2) do **R** equals CAS **LOCK**, **1**, **0**
(3) while key **R** equals 0

Procédure 2 (exécution dans la section critique)
pas d'instruction …

Procédure 3 (sortir de la section critique)
(4) Release the critical section with **P**
(5) set **LOCK** at 1

Considérant la séquence d'ordonnancement suivante, ou le l'ordonnanceur passe la main tour à tour aux processus A, B et C pour un nombre d'instruction donné. Indiquez pour chaque changement de contexte les informations suivantes: instructions exécutées, états des variables « R_i » pour chacun des processus et « Lock », détention de la section critique.

Nb d'instruction	1	2	4	4	2	2	2	1	4	5
Processus	B	A	C	B	A	B	C	A	B	C

Sur l'ensemble de la synchronisation, indiquez les temps de réponse et temps d'attente active pour l'accès à la section, pour chacun des processus.

3.2. Synchronisation logicielle

Soit 2 processus A et B se synchronisant via l'algorithme de Peterson. On considère le code suivant de l'algorithme, ou chaque instruction (1-7) est dite atomique (non interruptible par l'ordonnanceur).

Procédure 1 (requête sur la section critique)

(1) Request the critical section with $\mathbf{P_i}$
(2) **flag[i]** = true
(3) **turn** = **j**
(4) while (**flag[j]** == **true**) && (**turn** == **j**)
(5) busy-waiting

Procédure 2 (exécution dans la section critique)
pas d'instruction …

Procédure 3 (sortir de la section critique)

(6) Release the critical section with $\mathbf{P_i}$
(7) **flag[i]** = **false**

Considérant la séquence d'ordonnancement suivante, ou l'ordonnanceur passe la main tour à tour aux processus A, B pour un nombre d'instruction donné. Indiquez pour chaque changement de contexte les informations suivantes : instructions exécutées, état de la variable « turn », états des « flags », détention de la section critique.

Nb d'instruction	2	3	3	1	4	2	3	6
Processus	A	B	A	B	A	B	A	B

Sur l'ensemble de la synchronisation, indiquez les temps de réponse et temps d'attente active pour l'accès à la section, pour chacun des processus.