POLYTECH TOURS

DI3 - 2022-2023
M. Delalandre

Contrôle Terminal
« Systèmes d'exploitation »

La durée est de 1h30 min. Les notes de cours sont autorisées, ainsi que la calculatrice. Une annexe est donnée en fin de sujet rappelant les notions de cours nécessaires à la composition de cet examen. Un barème sur 20 pts est donné à titre indicatif.

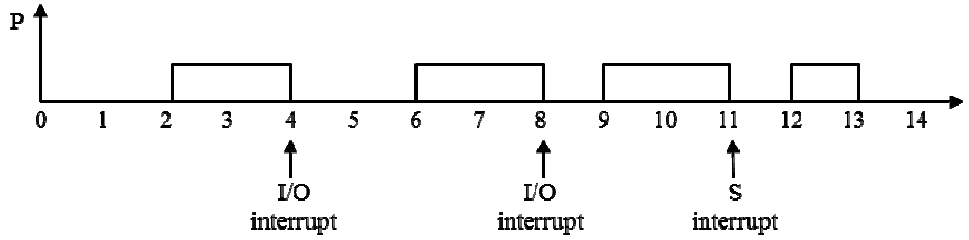
1. Question générale (2 pt)

Q11. Expliquez, selon vos propres termes, ce que sont les notions de parallélisme et de synchronisation. Vous pourrez expliquer les raisons pour lesquelles ces notions / propriétés ont été développées / considérées au sein des premiers systèmes d'exploitation.

2. Ordonnancement optimal (8 pts)

Q21. Soit le diagramme d'occupation du processeur suivant décrivant un cycle complet d'exécution d'un processus P au sein d'un système d'exploitation. On considéra ici la valeur de date de réveil comme étant égale à w_0=1. Sur ce diagramme, les interruptions ordonnanceur et en entrée / sortie (i.e. passage par la queue de blocage) sont notées respectivement « S » et « I/O ». Sur l'intervalle de temps considéré, indiquez quels sont:

- les dates de démarrage et de fin d'exécution,
- les valeurs de capacité, les temps de réponse et d'attente,
- les valeurs de la capacité résiduelle C(t), du temps d'exécution T(t), du temps d'attente WT(t) et du temps CPU ouvert (ou âge) E(t),
- les différents « CPU Burst » en précisant leur nombre, positions et durées.



w0	
C	
start	
end	
RT	
WT	

Burst	C (Burst)	C(T)	T(t)	WT(t)	E(t)
…	…	…	…	…	

Q22. On considère le cas où le processus P de la Q21 est exécuté au sein d'un système géré par un ordonnancement optimal SJF préemptif (ou SRT). Considérant les contraintes sur la connaissance a priori des paramètres C(t), l'ordonnancement SRT au sein de ce système est piloté à base de prédiction temporelle. Considérant le chronogramme d'exécution du processus P de la question Q21, indiquez les valeurs de prédiction temporelle pour les différents burst considérant les paramètres suivants: $T_0 = 1$ et $\alpha = 0{,}7$.

P	
ti	**Ti**
…	…

On considère qu'à l'instant t=11 le processus P est en concurrence avec un processus P' arrivant en « ready queue » et ayant une valeur de prédiction temporelle de 1,6 pour son burst. Expliquez ce qu'il se passe entre P, P' à t=11 en cohérence avec le chronogramme de la Q21.

Q23. L'algorithme SRT discuté en Q22 rentre dans le cadre des techniques d'ordonnancement dites optimales. Répondez aux questions suivantes:

a. selon-vous, la capacité d'un processus est-elle toujours prédictible, oui/non? précisez la relation entre les valeurs de prédiction et C(t),
b. quel critère l'algorithme SRT tente-t-il d'optimiser? vous pourrez comparer ce critère à ceux optimisés par d'autres algorithmes comme le HRNN,
c. considérant un système en interaction utilisateur, l'algorithme SRT est-il recommandé, oui/non? donnez alors des exemples d'algorithmes plus adaptés.

Q24. On se propose de mettre en place un ordonnancement à temps partagé de type « Guaranteed Scheduling » (GS). Répondez brièvement, aux questions suivantes:

a. donnez le diagramme d'occupation du processeur pour l'algorithme d'ordonnancement GS sur l'intervalle t=[2,6] pour le jeux de processus ci-dessous. Vous pourrez pour cela calculer les critères R(t) pour les 3 processus A, B et C en mentionnant les valeurs de T(t) et E(t).

	r_0	C
A	2	∞
B	4	∞
C	3	∞

burst		2-3	3-4	4-5	5-6
A	T(t)				
	E(t)				
	R(t)				
B	T(t)				
	E(t)				
	R(t)				
C	T(t)				
	E(t)				
	R(t)				

b. dans le cas de priorités définies par un expert système, quel intérêt présente le FSS au regard d'un algorithme à temps partagé comme le GS ou le Round Robin (RR)? selon vous, à quel niveau intervient le temps partagé dans le calcul des priorités du FSS?

3. Synchronisation (10 pts)

Q31. 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. Répondez aux questions suivantes:

a. considérant la séquence d'ordonnancement suivante, ou l'ordonnanceur passe la main tour à tour à 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 » (i = {A, B, C}) et « Lock », détention de la section.

(1) Request the critical section with **P**
(2) do **R** equals CAS **LOCK**, **0**, **1**
(3) while key **R** equals **1**
do something in the critical section …
(4) Release the critical section with **P**
(5) set **LOCK** at 0

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

scheduling		RA	RB	RC	Lock	Section
Process	Code					

Q32. Répondez brièvement et précisément aux questions suivantes:

a. au regard des résultats de la Q31, que pouvez-vous dire de l'ordre d'accès à la section critique entre une approche par instruction matérielle et mutex?
b. considérant le cas d'un code de section critique pour un accès en écriture à large volume de données sur un buffer, quelles recommandations feriez-vous entre les approches par instruction matérielle et mutex? que pouvez dire alors de l'impact du temps d'accès au buffer sur celui du temps d'attente active?

Q33. On se propose de mettre en œuvre une synchronisation par sémaphore au travers d'un problème de type producteur / consommateur, exploitant les algorithmes détaillés ci-dessous ou chaque instruction (1-3) est dite atomique. Définir la synchronisation entre le processus P (Producteur) et C (Consommateur) au travers de la séquence d'ordonnancement donnée ci-dessous. On considérera, à l'initialisation du système les informations suivantes:

- taille maximum du buffer = 2,
- les valeurs à t=0 de « fill » et « empty » sont respectivement de 2 et 0,
- la taille du buffer à t=0 est de 2 (i.e. au maximum),
- P à t=0 est dans la queue du sémaphore « empty ».

fill = 0, **empty** = n are semaphores
buffer is the data structure

consumer
loop
(1) down **fill**
(2) pop **item** from **buffer**
(3) up **empty**

producer
loop
(1) down **empty**
(2) push a new **item** in **buffer**
(3) up **fill**

Nb instruction	5	1	2	5
Processus	C	P	C	P

scheduling		buffer	fill		empty	
Process	Code		value	Q	value	Q

Q34. On considère un problème de synchronisation par moniteur entre 3 producteurs et 3 consommateurs. A un instant t, l'état du moniteur (buffer, queue d'entrée, variables conditionnelles) est donné ci-dessous. Considérant que la taille maximale du buffer est n=2 (i.e. le buffer est plien à t) et que l'ordonnanceur passe la main pour 4 instructions au seul processus en état « ready » (i.e. non bloqué) dans la queue d'entrée du moniteur, indiquez l'état de la synchronisation à l'instant t+1 (processus, instructions réalisées, état du buffer et des variables conditionnelles) dans le cas des implémentations Mesa puis Hoare.

Mesa

scheduling		buffer	count	Conditions			By	entry queue
Process	Code			full	empty	signal		
		2	2	P3				P1 C3 C2 P2 C1

Hoare

scheduling		buffer	count	Conditions			By	entry queue
Process	Code			full	empty	signal		
		2	2	P3				P1 C3 C2 P2 C1

Q35. Répondez brièvement aux questions suivantes:

a. selon-vous, les instructions pop et push, telles qu'illustrées en Q33 et Q34, sont-elles atomiques? oui, non, expliquez-en la ou les raisons.
b. qu'elle est alors la contrainte sur un problème de coordination comme le producteur / consommateur, dans quelle mesure le problème doit-il être étendu?
c. on souhaite mettre en place un code producteur / consommateur par sémaphore / mutex considérant les contraintes suivantes (i) N>1 producteurs P (ii) N>1 consommateur C (iii) sans victime de concurrence. Quelles recommandations feriez-vous pour la réécriture du code de la Q33?

Consumer

Producer

d. comparez le nombre de sémaphores / mutex mis en œuvre dans la question c. au nombre de variables conditionnelles utilisées dans les architectures moniteur de la Q34, que pouvez-vous en dire?
e. expliquez la raison de la modification du code P/C entre les moniteurs Mesa et Hoare, en particulier le code de l'instruction (1) (passage « while » en « if »),
f. selon-vous, pour quelle raison le moniteur Hoare a-t-il été architecturé avec une queue s.q adossée à sa queue d'entrée e.q? expliquez alors quelles stratégies d'accès équitables peut-être appliquées aux processus stockés en « s.q ».