

Si un exercice vous conduit à faire des hypothèses, indiquez-les clairement sur votre copie.
Rédigez et justifiez précisément les réponses aux questions.

Exercice 1 - 6 pts

1. Quel est le rôle de la MMU (*Memory Management Unit*) ?
2. Expliquer comment la méthode `sleep()` du langage Java agit sur l'ordonnancement. Montrer sa différence avec les primitives `wait()` et `notify()`.
3. La gestion des systèmes de fichiers utilise une allocation non contiguë au moyen de la structure de i-nœud. Expliquer la notion d'allocation non contiguë et la notion de i-nœud ainsi que la limite induite par une gestion des fichiers au moyen de i-nœuds.
4. Expliquer l'intérêt des flux et du mécanisme de sérialisation de Java.

Exercice 2 - 6 pts

On considère une mémoire virtuelle composée de 8 pages (numérotées de 0 à 7) comportant 4 pages mémoire logées en RAM (pages 0 à 3). Chaque page a une taille de 2048 octets (adresses de 0 à 2047).

1. Définir la notion de mémoire virtuelle et expliquer le mécanisme de *swap*.
2. Sur le schéma de la feuille donnée en annexe, préciser la limite de la mémoire physique, et identifier les pages stockées sur le disque. Donner les adresses des limites de chaque page (limite basse, limite haute).
3. On suppose que le noyau du système d'exploitation dispose d'une stratégie d'allocation de mémoire non contiguë. Il réalise les allocations mémoire par page : une page est allouée à un seul processus. Le mécanisme de *swap* utilise une stratégie de déchargement qui consiste à choisir la page mémoire qui est la moins accédée. Le noyau du système d'exploitation possède une table munie d'un compteur pour chaque page de la RAM (pages 0 à 3). Le compteur est incrémenté à chaque fois que la page est accédée. Il est remis à zéro dans le cas d'un échange de pages.

Les deux tableaux suivants donnent les allocations des pages aux processus et les accès mémoire. On suppose que dans l'état initial la mémoire contient les pages présentes dans la fiche jointe, c'est-à-dire (1, 0, 4, 7, 2, 3, vide, 5) et que la table des compteurs d'accès aux pages de la RAM est initialisée comme suit (Page 0 = 1 accès, Page 1 = 2 accès, Page 2 = 5 accès, Page 3 = 4 accès).

- (a) Effectuer la simulation, donner les différents états de la mémoire en fonction des demandes d'accès mémoire de la table 2, remplir les colonnes vides de l'annexe correspondant aux différents états de la mémoire virtuelle. Ajouter et remplir la table des compteurs de pages.
- (b) Pour chaque accès (table 2), remplir le tableau en annexe correspondant afin de donner le couple numéro de page, offset et l'adresse physique accédée correspondant à la page placée en RAM.
- (c) Quel est le nombre de défauts de page ? Comment améliorer la stratégie ?

Processus	P_1	P_2	P_3	P_4
Pages allouées	0,4,5	1,7	2	3

TABLE 1 – Pages initialement allouées à chaque processus

Top d'accès	1	2	3	4	5	6	7
Accès adresse	8197	1221	2049	15 320	9 001	6234	4098
Processus	P_1	P_1	P_2	P_2	P_1	P_4	P_3
Top d'accès	8	9	10	11	12	13	14
Accès adresse	11 240	14 441	8 195	10 243	6 191	7 101	9 218
Processus	P_1	P_2	P_1	P_1	P_4	P_4	P_1

TABLE 2 – Demandes d'accès à la mémoire

Exercice 3 - 8 pts

On souhaite réaliser une application de surveillance de machines connectées au réseau pour avertir les administrateurs en cas de problème de connexion (mécanisme *Heartbeat* - écoute des battements de cœur). L'objectif est de savoir si chacune des machines surveillées est bien connectée au réseau. À intervalle régulier, chaque machine envoie un message au serveur pour lui dire qu'elle est présente (toutes les minutes par exemple). Le serveur récupère les messages et les stocke immédiatement en mémoire et périodiquement sur le disque (toutes les 15 minutes). Le serveur surveille plusieurs centaines de machines.

1. Faire un schéma du système faisant apparaître le serveur et plusieurs machines surveillées. Quel type de socket allez-vous utiliser pour envoyer ou recevoir des messages ? Justifier votre réponse.
2. Comment se structure la communication entre les machines surveillées et le serveur ? Quels sont les messages échangés ?
3. Écrire la portion de code Java du serveur qui récupère un message d'une machine.
4. Écrire la portion de code Java d'une machine pour envoyer un message au serveur.
5. Les messages reçus sont stockés sur le serveur et on doit pouvoir accéder rapidement à l'ensemble des messages d'une machine afin de vérifier si elle a eu des problèmes de connexion. Quelle structure de données utiliser ? Donner l'extrait de code Java pour déclarer la structure de données nécessaire et ajouter un nouveau message reçu d'une machine donnée.
6. On ne peut pas garder en mémoire tous les messages reçus de toutes les machines. Ainsi, on fixe une limite à 300 messages par machine, et un thread est chargé de faire appliquer cette limite. Ce mécanisme peut-il faire apparaître des problèmes de concurrence ?
7. Écrire l'extrait de code permettant de réaliser la sauvegarde périodique sur disque de la structure de stockage des messages.

Numéro d'anonymat :

Compléments à remplir pour l'exercice 2.

0	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1
2	4	2	2	2	2	2	2	2	2
3	7	3	3	3	3	3	3	3	3
4	2	4	4	4	4	4	4	4	4
5	3	5	5	5	5	5	5	5	5
6		6	6	6	6	6	6	6	6
7	5	7	7	7	7	7	7	7	7

Correspondances mémoire virtuelle et adresses réelles

Demande	Processus	Adresse	Page	Offset	Page RAM	Adresse réelle
1		8 197	4	5	2	
2		1 221				
3		2 049				
4		15 320				
5		9 001				
6		6 234				
7		4 098				
8		11 240				
9		14 441				
10		8 195				
11		10 243				
12		6 191				
13		7 101				
14		9 218				