

Licence 2 — Info4B

Examen du Lundi 12 Mai 2025

Durée 2h • Documents autorisés : une feuille A4 recto-verso

Le barème est donné à titre indicatif



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. Expliquer ce que permet de réaliser l'abstraction *socket* proposée par le noyau d'un système d'exploitation. Montrer les relations entre les notions de *socket*, de port et de processus (lourd ou léger).
2. En quoi le mécanisme de sérialisation du langage Java simplifie la vie du programmeur ? Donner 3 cas d'usage différents.
3. Expliquer le rôle de la RAM et du mécanisme de *swap*. Donner leurs caractéristiques.
4. Quels sont les problèmes de concurrence sur les fichiers ?

Exercice 2 - 6 pts

On considère une mémoire virtuelle composée de 6 pages (numérotées de 0 à 5) comportant 3 pages mémoire logées en RAM (pages 0 à 2). Chaque page a une taille de 2 048 octets (adresses de 0 à 2 047). La stratégie de gestion de la mémoire est une allocation non contiguë pour laquelle les pages sont allouées à un seul processus. Le mécanisme de *swap* utilise une stratégie de déchargement qui consiste à choisir la page mémoire qui est la moins récemment accédée.

1. Sur votre copie en reprenant le schéma mémoire de l'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).
2. Sur l'annexe, effectuer la simulation (au départ la mémoire est vide), donner les différents états de la mémoire en fonction des opérations de la table 1, remplir les colonnes vides du tableau de l'annexe correspondant. La colonne "Adresse" correspond aux adresses virtuelle, la colonne "Adresse réelle" correspond aux adresses physiques (RAM).
3. Quel est le nombre de défauts de page ? Comment pourrait-on améliorer la stratégie ?

Top	1	2	3	4	5	6	7	8
Opérations	All P_0	All P_2	Acc 10	Acc 4098	All P_3	Acc 6148	All P_1	Acc 2050
Top	9	10	11	12	13	14	15	
Opérations	Acc 6150	All P_4	Acc 8195	Acc 10238	Acc 191	Acc 5000	Acc 2051	

TABLE 1 – Opérations sur la mémoire (All = allocation suivi du numéro de page, Acc = accès suivi d'une adresse en mémoire virtuelle)

Exercice 3 - 8 pts

On souhaite réaliser une application de sauvegarde de fichiers. Chaque utilisateur installe l'application qui fonctionne en permanence sur son poste et il spécifie les répertoires à sauvegarder. Les fichiers seront envoyés périodiquement sur un serveur. À intervalle régulier, l'application s'authentifie auprès du serveur et envoie au serveur les fichiers qui ont été modifiés depuis le dernier envoi. Le serveur récupère les fichiers et les stocke sur le disque dans un répertoire spécifique à chaque utilisateur, en archivant les anciens fichiers dans un sous-répertoire dont le nom comporte la date et l'heure de la modification. Lorsqu'un poste client veut restaurer un fichier, il peut consulter la liste des fichiers sur le serveur, sélectionner celui ou ceux à restaurer et les télécharger dans un répertoire temporaire prévu à cet effet sur son poste.

1. Faire un schéma du système faisant apparaître plusieurs postes et le serveur et préciser :
 - le type de socket que vous allez utiliser pour envoyer ou recevoir les fichiers ;
 - les messages et données échangés entre l'application et le serveur ;
 - la manière dont se structure la communication entre les postes et le serveur ;
 - les structures de données que vous allez utiliser sur le serveur.Justifier vos réponses par des phrases simples et précises.
2. Écrire la portion de code Java du serveur qui accepte une nouvelle connexion et met en place les flux pour communiquer avec un poste client.
3. Écrire la portion de code Java d'une machine pour envoyer un message avec *login* et mot de passe (qu'on suppose chiffré) au serveur.
4. Déclarer côté serveur la *hashtable* qui permet de stocker les informations d'authentification. Écrire les extraits de code correspondant à la sauvegarde de la *hashtable* dans un fichier.
5. Sur le poste client, on crée un thread **Timer** qui doit réveiller un autre thread **SendFiles** chargé d'envoyer les fichiers qui ont été modifiés depuis la dernière sauvegarde sur le serveur. Le thread d'envoi **SendFiles** est en attente. Écrire le thread **Timer** avec période de déclenchement paramétrable à la construction.
6. Écrire la portion de code Java du serveur qui réceptionne un fichier envoyé depuis un poste client et le stocke dans un répertoire.

Numéro d'anonymat :

Compléments à remplir pour l'exercice 2.

0		0		0		0		0		0		0		0		0		0	
1		1		1		1		1		1		1		1		1		1	
2		2		2		2		2		2		2		2		2		2	
3		3		3		3		3		3		3		3		3		3	
4		4		4		4		4		4		4		4		4		4	
5		5		5		5		5		5		5		5		5		5	

Correspondances mémoire virtuelle et adresses réelles

Demande	Opération	Adresse	Page	Offset	Page RAM	Adresse réelle
1	All P_0	-	-	-	0	-
2	All P_2	-	-	-	1	-
3	Acc	10	0	10	0	10
4	Acc	4 098	2	2	1	
5	All P_3	-	-	-		-
6	Acc	6 148				
7	All P_1	-	-	-		-
8	Acc	2 050				
9	Acc	6 150				
10	All P_4	-	-	-		-
11	Acc	8 195				
12	Acc	10 238				
13	Acc	191				
14	Acc	5 000				
15	Acc	2 051				