

## Licence 2 — Info4B

Examen du Mercredi 5 Mai 2021

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

Les réponses aux questions de cet exercice sont limitées à 1/4 page de texte par question (hors schémas et/ou extraits de code).

1. Comment gérer la concurrence pour une variable de classe qui est modifiée uniquement dans le constructeur de la classe en question ? Expliquer le problème sur un exemple de code et montrez comment le résoudre.
2. Définir la notion de *buffer* (tampon) et décrire son utilité dans le noyau du système d'exploitation ainsi que dans les programmes utilisateur. Par exemple, dans le cas où une ressource de type tampon est utilisée par plusieurs threads quel sont les modèles d'interaction ?
3. Un programmeur a écrit le code ci dessous, qu'en pensez vous ? Justifier votre réponse.

```
1 class TestThread extends Thread{
  public int a;
  ...
  public synchronized void run(){ ... }
  public synchronized void setA(int val){...}
}
7
public class Lancer{
9  public static void main(String arg[]){
  Thread t = new TestThread();
11  t.run();
  t.setA(2);
13  ...}
}
```

### Exercice 2 - 6 pts

On considère une mémoire de 16 pages (numérotées de 0 à 15) comportant 11 pages de mémoire virtuelle simulée au moyen du disque dur. Chaque page a une taille de 512 octets ayant des adresses de 1 à 512.

1. Faire un schéma de l'organisation de la mémoire, préciser la limite de la mémoire physique, numéroter les pages et donner les adresses des limites de chaque page.
2. Quel composant matériel intervient dans la gestion de la mémoire ? Le nommer et préciser son rôle.
3. On suppose que le mécanisme de *swap* utilise une stratégie LRU (*Least Recently Used*), c'est-à-dire moins récemment utilisé. Donner les différents états de la mémoire en fonction des demandes d'accès mémoire du tableau 1. Si une valeur de page n'apparaît pas c'est qu'il n'y a pas eu encore de demande. On suppose que dans l'état initial la mémoire physique contient les pages présentes dans la fiche jointe, c'est-à-dire (4,3,1,2,5) et entrées dans cet ordre (4 est la page la plus anciennement utilisée).
4. Pour chaque accès (table 1), donner le couple numéro de page, offset et le couple correspondant à la page placée en mémoire physique et remplir le tableau correspondant (table 2).
5. Quel est le nombre de défauts de page ?

demande	1	2	3	4	5	6	7
accès adresse	516	2 600	2 410	6 300	4 410	2 123	530
demande	8	9	10	11	12	13	14
accès adresse	1 603	6 400	7 200	3 121	1 538	6 146	8 196

TABLE 1 – Demandes d'accès à la mémoire

### Exercice 3 - 8 pts

On développe une application réseau pour permettre à des utilisateurs de sauvegarder des fichiers depuis leur poste de travail sur un serveur. Le programme client est installé sur le poste de travail et on lui indique le nom d'un répertoire qui contient les fichiers à envoyer sur le serveur. Il le fait à intervalle régulier (par exemple toutes les 2 heures). Pour ce faire, il demande au serveur la liste des fichiers déjà sauvegardés et n'envoie les fichiers que si ils sont plus récents que ceux déjà présents sur le serveur. Le serveur quant à lui garde une copie de toutes les versions des fichiers d'un utilisateur et les classe en différents types en fonction de l'extension des noms de fichiers : bureautique, image, vidéo, son, données, programmes, etc. Lorsqu'un utilisateur veut restaurer ses fichiers, il peut explorer son catalogue, par type de fichier, par nom de fichier, en obtenir la liste triée par date, etc. Le serveur doit supporter plusieurs clients simultanés.

1. Faire un schéma et y faire figurer le serveur, les clients, les threads, les structures de données partagées, les fichiers, etc. Décrire et justifier les constructions Java que vous allez utiliser pour implanter votre système.
2. Quel type de socket utilisez-vous pour les connexions entre les clients et le serveur ?
3. Comment faire pour que le client déclenche automatiquement une sauvegarde toutes les 2 heures ?
4. Écrire le squelette de la classe `ClientEnvoiFichier` (attributs et nom des méthodes, paramètres, sans le corps des méthodes) qui sert à lire un fichier et à l'envoyer au serveur avec le nom de l'utilisateur, la date, le type. Donner le corps de la méthode qui permet de se connecter au serveur et d'envoyer un fichier.
5. Écrire le code de la méthode de la classe `ServeurSauvegarde` qui permet d'accepter une connexion, de récupérer un fichier et de l'ajouter dans le catalogue. Le serveur doit pouvoir prendre en charge plusieurs clients et réceptionner des fichiers.
6. Définir la classe `CatalogueFichiers` du programme serveur qui permet de gérer et d'organiser les fichiers utilisateur (ajouter, rechercher par nom d'utilisateur, par type). Vous ne donnerez que les attributs, les noms des méthodes, leurs paramètres et leurs modifieurs (le code des méthodes n'est pas demandé). Indications : vous devez utiliser des tables de hachage, la classe `CatalogueFichiers` ne contient qu'un objet.
7. Expliquer les problèmes de concurrence éventuels côté serveur et comment les résoudre.
8. Comment enregistrer le catalogue des fichiers c'est-à-dire un objet `CatalogueFichiers` ?

**Numero d'anonymat :**

Compléments à remplir pour l'exercice 2.

0	4	0		0		0		0		0		0		0	
1	3	1		1		1		1		1		1		1	
2	1	2		2		2		2		2		2		2	
3	2	3		3		3		3		3		3		3	
4	5	4		4		4		4		4		4		4	
5		5		5		5		5		5		5		5	
6		6		6		6		6		6		6		6	
7		7		7		7		7		7		7		7	
8		8		8		8		8		8		8		8	
9		9		9		9		9		9		9		9	
10		10		10		10		10		10		10		10	
11		11		11		11		11		11		11		11	
12		12		12		12		12		12		12		12	
13		13		13		13		13		13		13		13	
14		14		14		14		14		14		14		14	
15		15		15		15		15		15		15		15	

Correspondances mémoire virtuelle, mémoire et adresses réelles

Demande	Adresse	Page	Offset	Page RAM	Adresse réelle
1	516	1	4	2	1028
2	2 600				
3	2 410				
4	6 300				
5	4 410				
6	2 123				
7	530				
8	1 603				
9	6 400				
10	7 200				
11	3 121				
12	1 538				
13	6 146				
14	8 196				