



## ESISAR

### NE449 – Programmation répartie

### TDM numéro 5

#### Table des matières

1 Exercice 1 : Copie d'un fichier octet par octet (sans socket TCP).....	1
2 Exercice 2: Test des performances.....	2
3 Exercice 3: Amélioration des performances.....	2
4 Exercice 4 : Transfert de fichier simple par socket TCP/IP.....	2
5 Exercice 5 : Amélioration du transfert de fichiers.....	3

#### 1 Exercice 1 : Copie d'un fichier octet par octet (sans socket TCP)

Vous allez réaliser un programme simple de copie de fichier : votre programme prend en entrée le nom d'un fichier existant et le nom du nouveau fichier. Votre programme copie le fichier existant vers le nouveau fichier, en copiant octet par octet.

Pour cela, vous devez utiliser les classes `FileInputStream` et `FileOutputStream` (lisez la JavaDoc de ces classes). Avec ces classes, vous allez pouvoir obtenir un `InputStream` sur le fichier à lire, et un `OutputStream` sur le fichier à écrire.

Ensuite, avec ce que vous avez vu en cours, vous copiez octet par octet depuis l'`InputStream` vers l'`OutputStream`.

Faites un test de votre programme avec un petit fichier texte. Vérifiez que votre copie est correct en utilisant l'utilitaire `diff`.

## 2 Exercice 2: Test des performances

Créer un fichier de 1Go avec la commande suivante (nom du fichier : file.bin)

```
dd if=/dev/urandom of=file.bin bs=1000 count=1000000
```

Faites une copie de ce fichier avec le programme précédent.

Mesurer le temps mis pour la copie.

Vérifiez que votre copie est correct en utilisant l'utilitaire diff.

## 3 Exercice 3: Amélioration des performances

Modifiez maintenant votre programme pour utiliser un buffer de 10 octets. Mesurer le temps mis pour la copie du fichier de 1Go.

Faites la même mesure avec un buffer de 100 octets, 1000 octets , 10 000 octets , 100 000 octets , 1 000 000 octets.

Tracez une courbe avec en abscisse la taille du buffer (logarithmique) et en ordonnée le temps mis pour la copie.

## 4 Exercice 4 : Transfert de fichier simple par socket TCP/IP

Ecrire un programme client et un programme serveur qui permettent l'échange de fichier.

Le programme client demande à l'utilisateur la saisie du nom d'un fichier.

Le programme client envoie au serveur ce nom de fichier, et le serveur retourne au client le fichier demandé.

Le programme client sauvegarde le fichier demandé dans le répertoire courant.

Faites des essais avec un petit fichier, puis avec un gros fichier de 1 Go.

Vérifiez que votre copie est correct en utilisant l'utilitaire diff.

Transférer ensuite des fichiers entre votre machine et la machine de votre binôme.

## 5 Exercice 5 : Amélioration du transfert de fichiers

Modifiez le programme précédent pour faire un affichage de l'avancement du téléchargement sur le client.

Le client affichera 1 % téléchargé, puis 2 %, .. jusqu'à 100 %.

Pour cela, il est nécessaire que le serveur retourne au client la taille du fichier avant d'envoyer son contenu.

Nota : la lecture de la taille du fichier dans le flux TCP n'est pas un problème trivial. Vous repenserez au TDM4.