2020 - 2021

TONNELLE Nathan

INFO0601-TD

Système d'exploitation - avancé

Table des matières

[TD 1 2](#_Toc59000386)

[Exercice 1 - Table d’adresses 2](#_Toc59000387)

[Question 1 2](#_Toc59000388)

[Question 2 2](#_Toc59000389)

[Question 3 2](#_Toc59000390)

[Question 4 2](#_Toc59000391)

[Question 5 2](#_Toc59000392)

[Question 6 2](#_Toc59000393)

[Question 7 2](#_Toc59000394)

[Exercice 2 - Fichiers à trous (1) 2](#_Toc59000395)

[Question 1 2](#_Toc59000396)

[Question 2 2](#_Toc59000397)

[Question 3 2](#_Toc59000398)

[Question 4 3](#_Toc59000399)

[Exercice 2 - Fichiers à trous (2) 3](#_Toc59000400)

[Question 1 3](#_Toc59000401)

[Question 2 3](#_Toc59000402)

[Question 3 3](#_Toc59000403)

[Question 4 3](#_Toc59000404)

[Question 5 3](#_Toc59000405)

[Question 6 3](#_Toc59000406)

[Exercice 2 - Fichiers à trous (3) 3](#_Toc59000407)

[Question 1 3](#_Toc59000408)

[Question 2 3](#_Toc59000409)

[Question 3 3](#_Toc59000410)

[Question 4 3](#_Toc59000411)

[Question 5 3](#_Toc59000412)

[Exercice 2 - Fichiers à trous (4) 4](#_Toc59000413)

[Question 1 4](#_Toc59000414)

[Question 2 4](#_Toc59000415)

[Question 3 4](#_Toc59000416)

[Question 4 4](#_Toc59000417)

# TD 1

## Exercice 1 - Table d’adresses

Nous supposons qu’un système de fichiers est basé sur la notion de table d’adresses (comme un i-node) pour mémoriser les blocs utilisés pour un fichier.

### Question 1

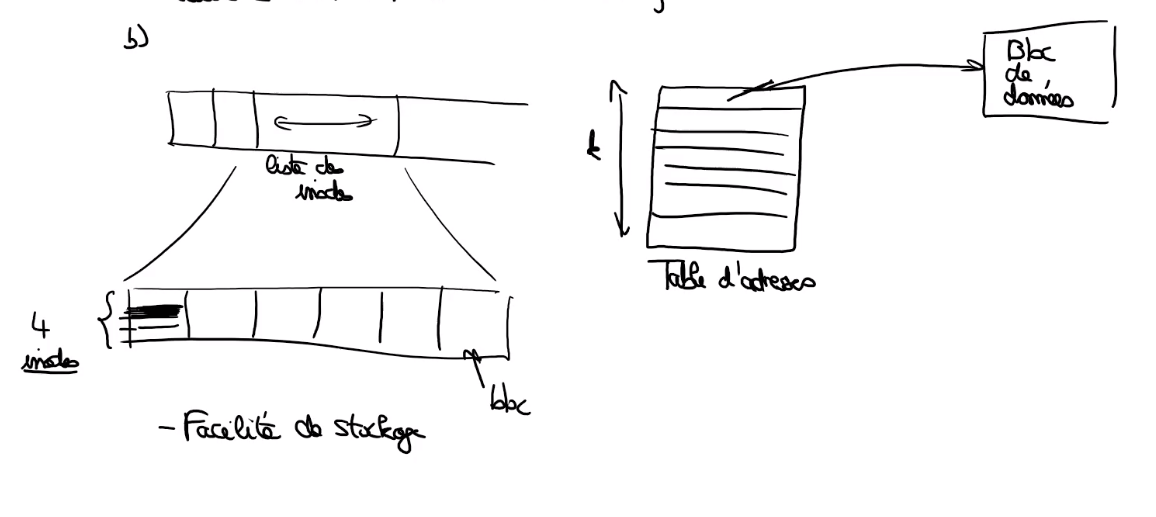
Sans niveau d’indirection et en considérant qu’une table possède un nombre fixe k d’enregistrements, quelle est la taille maximale d’un fichier ?

T = taille d'un bloc

Taille = T\*k , k le nombre d'enregistrement

### Question 2

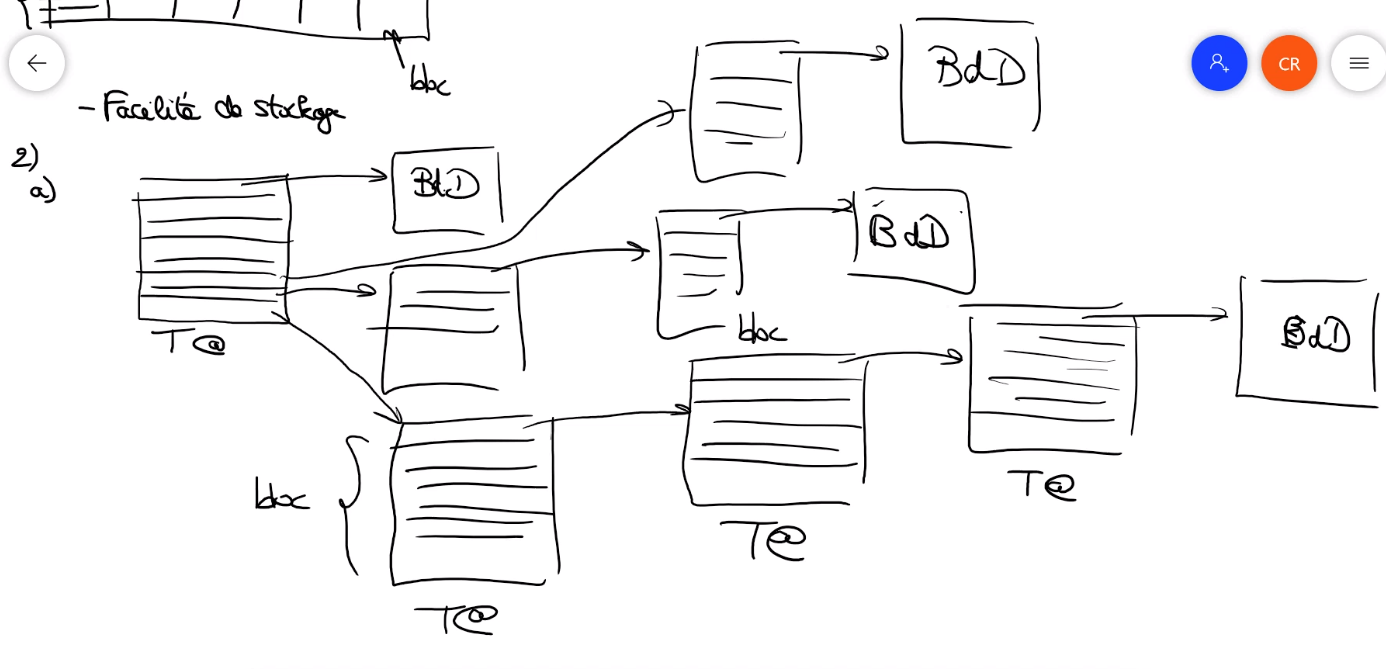
Pourquoi est-il préférable que la table d’adresses possède un nombre d’enregistrements maximum ?



Dans les i-nodes du système ext2, les tables d’adresses possèdent 15 enregistrements, les 3 derniers correspondant à des niveaux d’indirection différents. Les adresses de bloc ont une taille de 4 octets.

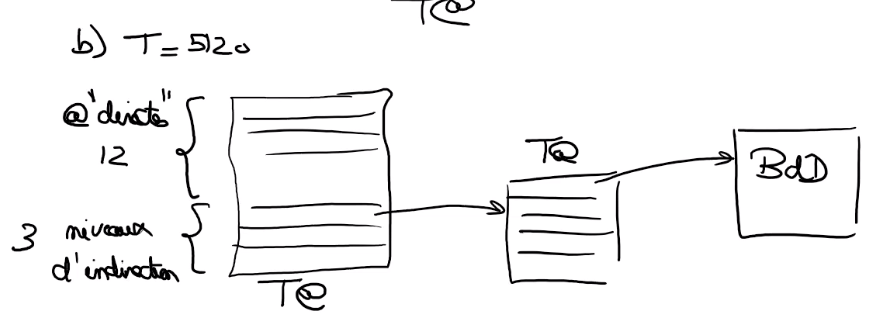
### Question 3

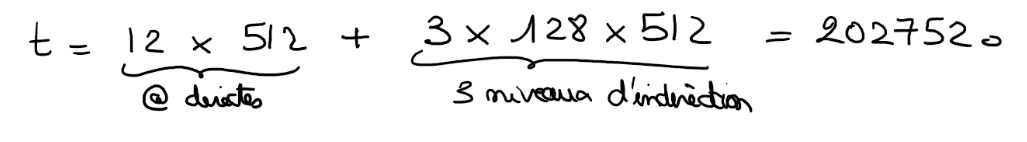
Rappelez le principe de l’indirection. Illustrez votre réponse par un schéma.



### Question 4

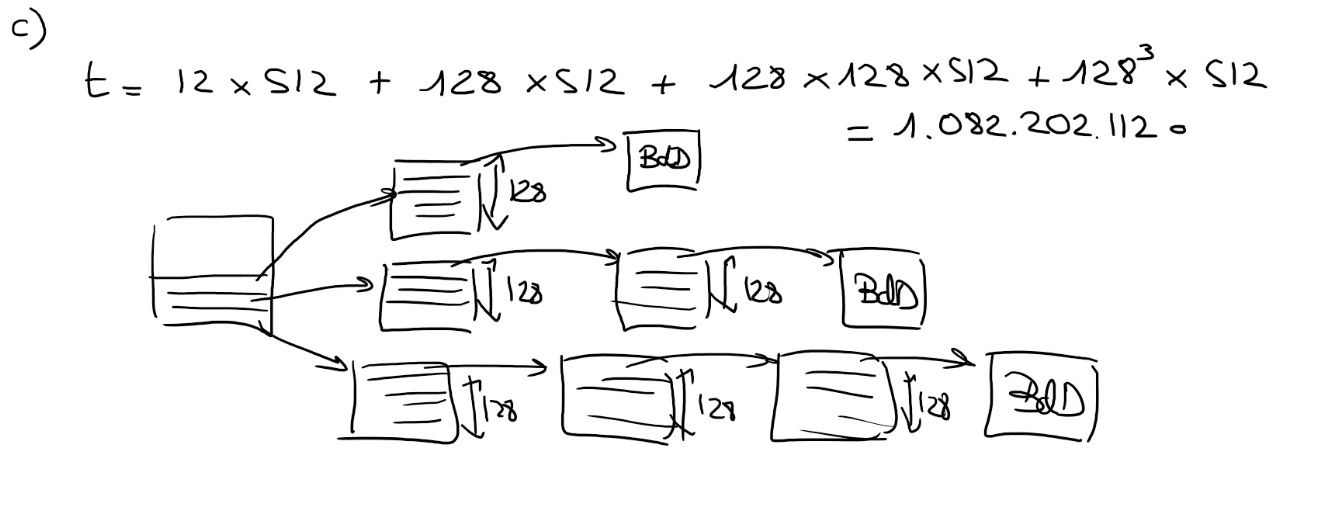
Avec des blocs de 512o et en supposant que les 3 derniers enregistrements des i-nodes permettent un seul niveau d’indirection, quelle est la taille maximale d’un fichier ?





### Question 5

En réalité, le treizième enregistrement possède un niveau d’indirection, le quatorzième en possède deux et le quinzième en possède trois. Quelle est la taille maximale d’un fichier ?



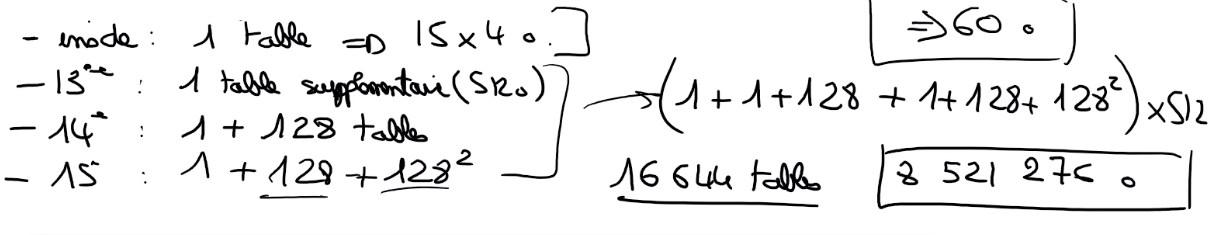
### Question 6

Pourtant ext2, qui est basé sur ce principe, permet des tailles de fichier de 2TiB maximum. Comment l’expliquer ?

Tailles des blocs > 512o

### Question 7

Pour un fichier qui possède la taille la plus grande possible (calculée à la question 2.c), combien de tables d’adresses sont nécessaires ? Quelle est la taille de l’ensemble de ces tables ?



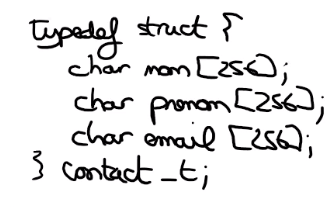
## Exercice 2 - Fichiers à trous (1)

Nous souhaitons réaliser une application permettant d’enregistrer des contacts dans un fichier. Un contact contient un nom, un prénom et une adresse de courriel.

**Première partie.** Nous supposons que chaque champ d’un contact possède une taille fixée.

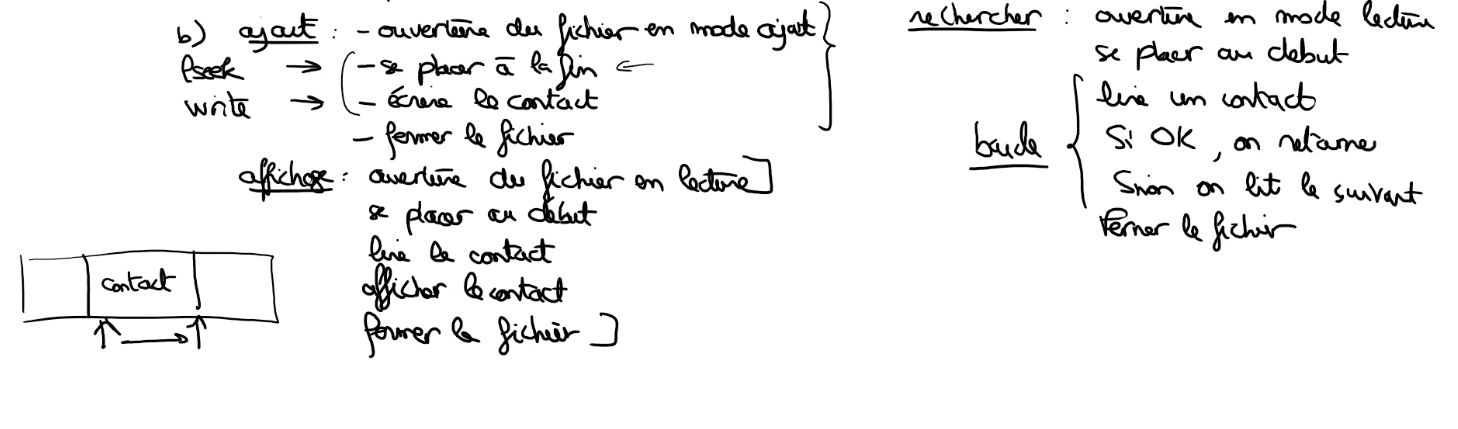
### Question 1

Proposez une structure en C pour représenter un contact (nommée contact\_t).



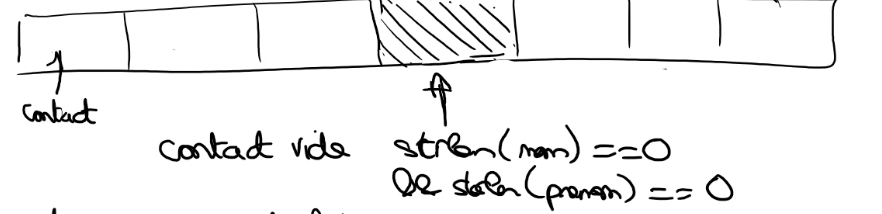
### Question 2

Expliquez comment ajouter un contact dans le fichier (fonction ajout), afficher tous les contacts (fonction affichage) et chercher un contact en fonction d’un nom (fonction recherche).



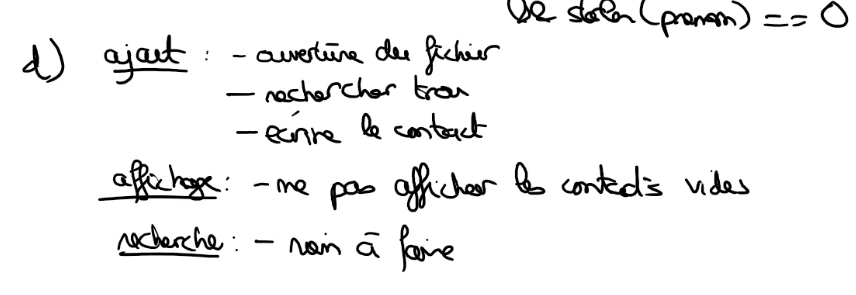
### Question 3

Expliquez quels sont les problèmes rencontrés pour la suppression et donnez une solution qui n’impose pas le déplacement de contacts dans le fichier.



### Question 4

Y-a-t-il des modifications à apporter pour les fonctions ajout, affichage et recherche ?

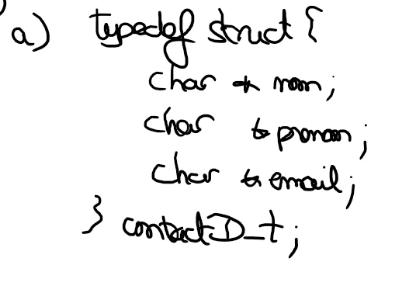


## Exercice 2 - Fichiers à trous (2)

**Deuxième partie.** Nous supposons maintenant que les champs d’un contact possèdent des tailles variables.

### Question 1

Rappelez le problème de la sauvegarde d’une chaîne de caractères dans un fichier et de sa lecture.



### Question 2

Donnez la nouvelle structure en C d’un contact (nommée contactD\_t).

### Question 3

Donnez le code de la procédure void ajouter(int fd ,contactD\_t contact) permettant de sauvegarder un contact dans un fichier dont fd est le descripteur associé. Le contact est sauvegardé à la position en cours.

### Question 4

Donnez maintenant le code de la fonction contactD\_t lire(int fd) permettant de lire le contact situé à la position en cours dans le fichier.

### Question 5

Que se passe-t-il lors d’une suppression d’un contact dans le fichier ? Qu’est-ce que cela change pour les différentes fonctions de l'exercice précédent ajout, affichage et recherche ?

### Question 6

Proposez une solution pour limiter les modifications et précisez les limitations de votre solution.

## Exercice 2 - Fichiers à trous (3)

**Troisième partie.** Nous utilisons une table de positions qui permet de mémoriser la position dans le fichier de chaque contact. Elle est sauvegardée au début du fichier, avant les contacts.

### Question 1

Pourquoi est-il plus simple d’utiliser un nombre fixe d’enregistrements dans cette table ?

### Question 2

Que se passe-t-il si le nombre de contacts dépasse ce maximum? Proposez une solution permettant d’ajouter autant de contacts que souhaité.

### Question 3

Donnez la structure en C de la table de positions et donnez le code en C permettant de la sauvegarder dans un fichier.

### Question 4

Expliquez comment peut-on supprimer un contact dans le fichier, en sachant que la position doit être conservée pour l’ajout de futurs enregistrements. Vous pouvez modifier la structure de la table.

### Question 5

Expliquez comment ajouter un contact dans le fichier, en sachant qu’un contact peut être ajouté soit dans un trou, soit à la fin du fichier.

## Exercice 2 - Fichiers à trous (4)

**Quatrième partie.** Pour simplifier la gestion des trous dans notre fichier, nous exploitons une autre structure que nous appelons la table de vide. Celle-ci permet de mémoriser la position et la taille de chaque trou. Elle fonctionne sur le même principe que la table des positions et elle est sauvegardée après celle-ci dans le fichier.

### Question 1

Donnez la structure en C de cette table.

### Question 2

Expliquez comment ajouter un contact dans le fichier.

### Question 3

Expliquez comment supprimer un contact dans le fichier.

### Question 4

Pour améliorer notre application, nous souhaitons pouvoir associer plusieurs adresses de courriel à un contact. Donnez toutes les structures et détaillez toutes les fonctions.