Systèmes d'exploitation **IFT-2001**
A20

RAPPORT Travail Pratique no 3
Implémentation d'un mini système de fichier UFS

fait par: Claire BOUTTES 536793110

Implémentation du système de fichier UFS.

a) Une description de votre solution. Expliquer brièvement chacune des fonctions ajoutées (pas celles qui sont demandées) et donner éventuellement les problèmes rencontrés.
J'ai séparé la classe Disque Virtuel en 4 sous parties.
La première partie concerne les i-nodes. Elle contient 4 fonctions. create est la fonction de création et initialisation des i-nodes. take et free sont 2 fonctions qui permettent respectivement de prendre et de libérer un i-node. Enfin, findFirstFree permet de trouver le premier i-node libre.
La seconde partie concerne les blocs. Les fonctions take et free permettent de prendre et libérer un bloc, et findFirstFree permet de trouver le premier bloc libre.
Ces deux premières parties simules la gestion des bitmaps. Aucune opération sur des blocs ou des i-nodes ne devraient être faites or ces fonctions.
La troisième partie concerne toute la gestion des fichiers en elle-même (leur creation, suppression, récupération ou encore le système de lien entre eux).
Enfin la quatrième et dernière partie concerne l'interface utilisateur avec les fonctions demandées pour le tp.

Chacune de ces parties peuvent être reconnue par un préfixe qui précèdent chacune de leur fonction.

Le principal problème que j'ai pu rencontrer concernait le destructeur de Block. Je n'ai finalement pas pu delete les i-nodes. Pour une raison inconnue, mon destructeur était appelé en plus d'un autre, créant un double free. Même en initialisant, toutes mes variables à nullptr. J'ai donc laissé des variables non libérées à la fin du programme. Ce qui est la raison pour laquelle à la fin du programme, il reste de la mémoire non libérée. Je n'ai pas trouvé de solution qui résolvait correctement mon problème.

b) Sortie d'écran du programme pour `le script Test1.`
Expliquer les différences, s'il y a lieu, entre votre sortie et celle donnée dans le TP.
Pour la commande format, on peut voir que la saisie de l'i-node 0 avec aucun bloc affilié. C'est un bloc à laisser libre. Or pour indiquer que cet i-node n'est pas libre, je fais un appel à inode_take pour éviter de dupliquer du code et cette fonction indique à chaque appel son action. Ça me paraissait le plus logique de laisser de la sorte.

```
===== Commande format =====
UFS: saisir i-node 0
UFS: saisir i-node 1
UFS: saisir bloc 24

===== Commande ls / =====
/
d              . Size:   56 inode:    1 nlink:    2
d             .. Size:   56 inode:    1 nlink:    2
```

```
===== Commande mkdir /doc =====
UFS: saisir i-node 2
UFS: saisir bloc 25

===== Commande ls / =====
/
d              . Size:   84 inode:    1 nlink:    3
d             .. Size:   84 inode:    1 nlink:    3
d            doc Size:   56 inode:    2 nlink:    2

===== Commande mkdir /tmp =====
UFS: saisir i-node 3
UFS: saisir bloc 26

===== Commande ls / =====
/
d              . Size:  112 inode:    1 nlink:    4
d             .. Size:  112 inode:    1 nlink:    4
d            doc Size:   56 inode:    2 nlink:    2
d            tmp Size:   56 inode:    3 nlink:    2

===== Commande mkdir /tmp/lib =====
UFS: saisir i-node 4
UFS: saisir bloc 27

===== Commande ls / =====
/
d              . Size:  112 inode:    1 nlink:    4
d             .. Size:  112 inode:    1 nlink:    4
d            doc Size:   56 inode:    2 nlink:    2
d            tmp Size:   84 inode:    3 nlink:    3

===== Commande ls /tmp =====
/tmp
d              . Size:   84 inode:    3 nlink:    3
d             .. Size:  112 inode:    1 nlink:    4
d            lib Size:   56 inode:    4 nlink:    2

===== Commande mkdir /tmp/lib/deep =====
UFS: saisir i-node 5
UFS: saisir bloc 28

===== Commande ls /tmp/lib/deep =====
/tmp/lib/deep
d              . Size:   56 inode:    5 nlink:    2
d             .. Size:   84 inode:    4 nlink:    3

===== Commande rm /tmp/lib/deep =====
UFS: Relache i-node 5
UFS: Relache bloc 28

===== Commande rm /tmp/lib =====
UFS: Relache i-node 4
UFS: Relache bloc 27

===== Commande rm /tmp =====
```

```
UFS: Relache i-node 3
UFS: Relache bloc 26

===== Commande ls / =====
/
d              . Size:   84 inode:    1 nlink:    3
d             .. Size:   84 inode:    1 nlink:    3
d            doc Size:   56 inode:    2 nlink:    2

===== Commande create /a.txt =====
UFS: saisir i-node 3
UFS: saisir bloc 26

===== Commande ls / =====
/
d              . Size:  112 inode:    1 nlink:    3
d             .. Size:  112 inode:    1 nlink:    3
d            doc Size:   56 inode:    2 nlink:    2
-          a.txt Size:    0 inode:    3 nlink:    1

===== Commande rm /a.txt =====
UFS: Relache i-node 3
UFS: Relache bloc 26

===== Commande ls / =====
/
d              . Size:   84 inode:    1 nlink:    3
d             .. Size:   84 inode:    1 nlink:    3
d            doc Size:   56 inode:    2 nlink:    2

===== Commande format =====
UFS: saisir i-node 0
UFS: saisir i-node 1
UFS: saisir bloc 24

===== Commande ls / =====
/
d              . Size:   56 inode:    1 nlink:    2
d             .. Size:   56 inode:    1 nlink:    2
```

c) Indiquez le nombre total de lignes pour votre code se trouvant dans votre fichier disqueVirtuel.cpp. Pensez-vous que ce nombre est assez raisonnable par rapport au travail demandé ?

355. Je dirai qu'il y en a beaucoup pour le travail demandé. Cela amène vite un code à être illisible. Pour autant, ces lignes inclues des explications pour chaque fonction (ou presque). De plus, j'ai pris le soin de séparer mon code en pas mal de fonction, premièrement par propreté. Pour que chaque fonction soit facilement lisible indépendamment et qu'aucune ne finisse en couteau suisse, à avoir trop de fonctionnalité. Deuxièmement, la raison de cette séparation est dans une idée d'ajout de fonctionnalité si un jour il me vient l'envie de retravailler dessus.

d) Sortie d'écran du programme pour `le script Test2.`

```
===== Commande format =====
UFS: saisir i-node 0
UFS: saisir i-node 1
```

```
UFS: saisir bloc 24

===== Commande ls / =====
/
d              . Size:   56 inode:    1 nlink:    2
d             .. Size:   56 inode:    1 nlink:    2

===== Commande ls /doc =====
doc: File not found.

===== Commande mkdir /new =====
UFS: saisir i-node 2
UFS: saisir bloc 25

===== Commande mkdir /empty =====
UFS: saisir i-node 3
UFS: saisir bloc 26

===== Commande mkdir /notempty =====
UFS: saisir i-node 4
UFS: saisir bloc 27

===== Commande mkdir /existepas/new =====
existepas: File not found.

===== Commande create /notempty/c.txt =====
UFS: saisir i-node 5
UFS: saisir bloc 28

===== Commande create /b.txt =====
UFS: saisir i-node 6
UFS: saisir bloc 29

===== Commande create /existepas/b.txt =====
existepas: File not found.

===== Commande ls / =====
/
d              . Size:  168 inode:    1 nlink:    5
d             .. Size:  168 inode:    1 nlink:    5
d            new Size:   56 inode:    2 nlink:    2
d          empty Size:   56 inode:    3 nlink:    2
d       notempty Size:   84 inode:    4 nlink:    2
-          b.txt Size:    0 inode:    6 nlink:    1

===== Commande ls /notempty =====
/notempty
d              . Size:   84 inode:    4 nlink:    2
d             .. Size:  168 inode:    1 nlink:    5
-          c.txt Size:    0 inode:    5 nlink:    1

===== Commande rm /notempty =====
/notempty: Not empty.

===== Commande ls / =====
/
d              . Size:  168 inode:    1 nlink:    5
```

```
d             .. Size:  168 inode:    1 nlink:    5
d            new Size:   56 inode:    2 nlink:    2
d          empty Size:   56 inode:    3 nlink:    2
d       notempty Size:   84 inode:    4 nlink:    2
-          b.txt Size:    0 inode:    6 nlink:    1

===== Commande rm /notempty/c.txt =====
UFS: Relache i-node 5
UFS: Relache bloc 28

===== Commande rm /notempty =====
UFS: Relache i-node 4
UFS: Relache bloc 27

===== Commande ls / =====
/
d              . Size:  140 inode:    1 nlink:    4
d             .. Size:  140 inode:    1 nlink:    4
d            new Size:   56 inode:    2 nlink:    2
d          empty Size:   56 inode:    3 nlink:    2
-          b.txt Size:    0 inode:    6 nlink:    1

===== Commande rm /a.txt =====
a.txt: File not found.

===== Commande ls / =====
/
d              . Size:  140 inode:    1 nlink:    4
d             .. Size:  140 inode:    1 nlink:    4
d            new Size:   56 inode:    2 nlink:    2
d          empty Size:   56 inode:    3 nlink:    2
-          b.txt Size:    0 inode:    6 nlink:    1

===== Commande format =====
UFS: saisir i-node 0
UFS: saisir i-node 1
UFS: saisir bloc 24

===== Commande ls / =====
/
d              . Size:   56 inode:    1 nlink:    2
d             .. Size:   56 inode:    1 nlink:    2
```

e) Explication du résultat. Par exemple, si votre programme plante ou subit des échecs lors du déroulement, veuillez en indiquer la cause probable.
Je n’ai pas d’explication à rajouter.