

Examen 2021-2022 - CYBER1 (1h)

Système d'Exploitation

NOM :

PRÉNOM :

1 Systèmes de fichiers (8 points)

	FAT	Clusters	
3	0x0006	0 1 2 3 4 5 6 7 8 9	
4	0xFFFF	1 0 1 1 1 2 1 3 1 4	
5	0xFFF7	1 5 1 6 1 7 1 8 1 9	
6	0x0004	2 0 2 1 2 2 2 3 2 4	
7	0x0009	2 5 2 6 2 7 2 8 2 9	
8	0xFFFF	3 0 3 1 3 2 3 3 3 4	
9	0x0008	3 5 3 6 3 7 3 8 3 9	

[direntry]

Name: **DOCUMENT**
Attribute: **Subdirectory**
Creation Time: **25 Jan. 1987**
First Cluster: **0x003**

1.1 (1 point) Quelle séquence de chiffres correspond à « DOCUMENT » ?

1.2 (1 point) En admettant qu'un fichier est stocké dans cette FAT (en plus de « DOCUMENT »), à quel cluster semble-t-il démarrer ?

1.3 (0,5 point) En admettant que les clusters et secteurs font 4.096 octets, et que la *direntry* du fichier indique en size 11.042 octets. Combien d'octets faut-il ignorer dans le dernier cluster ?

1.4 (1 point) Un fichier contenant 10.000 caractères est stocké sur un système de fichiers dont les clusters font 4.096 octets et les secteurs font 512 octets. Quelle taille en octets consomme-t-il sur le support physique et combien de secteurs consomme-t-il ?

- 1.5 (2 points) Dans quelle situation ou action peut-on se rendre compte qu'une telle FAT ne fonctionne pas/n'est pas viable pour stocker des données ? (selon les règles que vous avez apprises) Et pourquoi n'est-elle pas viable ?

FAT

3	0x0004
4	0x0005
5	0xFFFF
6	0x0005

- 1.6 (1 point) En quoi les *i-nodes* et *dirent* d'ext2 sont-ils différents ?
- 1.7 (0,5 point) Dans quelle(s) structure(s) trouve-t-on le nom des fichiers ?
- 1.8 (1 point) Expliquer la différence entre « *hardlink* » et « *symbolic link* ».

2 Ordonnancement et Processus (4 points)

- 2.1 (1 point) Citer les états dans lesquels un processus peut se trouver.
- 2.2 (1 point) Expliquez succinctement pourquoi on ne peut pas tuer avec SIGKILL un processus en état "Zombie".

2.3 (2 point) Dessinez la hiérarchie de processus créée par ce programme C :

```
1 int main(void)
2 {
3     pid_t p1, p2, p3;
4
5     p1 = fork();
6     if (p1 > 0) {
7         // Process A
8         p2 = fork();
9         if (p2 > 0) {
10            // Process B
11        } else { // Process C }
12    } else {
13        // Process D
14        p3 = fork();
15        if (p3 > 0) {
16            // Process E
17        } else { // Process F }
18    }
19 }
```

3 Script Shell et Commandes (8 points)

3.1 (3 points) Expliquez ce que font ces lignes de commandes :

```
ps aux | head -n 20 | tail -n 10 > file1.txt
```

```
touch file3.txt && cat /etc/passwd | grep "metaldan" > file2.txt
```

```
cat file1.txt < file2.txt 1> file3.txt 2> file4.txt
```

3.2 (5 point) Indiquez à quoi sert chaque opérateur, et donnez un exemple.

Opérateur	Exemple et explication
>	
>>	
<	
<<	