Rattrapage 2022-2023 - CYBER1 (1h30)

Architecture des Ordinateurs

NOM : PRÉNOM :

Vous devez respecter les consignes suivantes, sous peine de 0 :
— Lisez le sujet en entier avec attention
— Répondez sur le sujet
— Ne détachez pas les agrafes du sujet
— Écrivez lisiblement vos réponses (si nécessaire en majuscules)
— Les appareils électroniques sont tous interdits (calculatrices également)
— Ne trichez pas

1 Questions (10 points)

1.1 (2 points) Rappelez les 14 premières puissances de 2 :

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192

1.2 (4 points) Convertissez ces nombres en décimaux. Vous donnerez leur interprétation non-signée puis signée sur 12 bits.

	non-signé	signé
% 1010 1110 1011	2795	-1301
% 1001 0111 1001	2425	-1671
$ B01	2817	-1279
$ DB2	3506	-590

1.3 (4 points) Convertissez ces nombres décimaux en binaire sur 12 bits, puis en hexadécimal.

	binaire												hexadécimal			
42	%	0	0	0	0	0	0	1	0	1	0	1	0	$	0	2	A
1122	%	0	1	0	0	0	1	1	0	0	0	1	0	$	4	6	2
1289	%	0	1	0	1	0	0	0	0	1	0	0	1	$	5	0	9
−56	%	1	1	1	1	1	1	0	0	1	0	0	0	$	F	C	8

2 Problème (10 points)

Un étudiant en cybersécurité faisant ses premières expériences en *forensic* (analyse forensique ou investigation numérique en français) a besoin de vous pour convertir plusieurs valeurs et retrouver des informations. Celui-ci a récupéré un disque dur qui servait dans un RAID 1, et il souhaite retrouver les noms de fichiers, le contenu de ces fichiers, mais également analyser quelques programmes stockés dessus.

2.1 (2 points) Première étape : lecture d'une structure

L'étudiant a réussi à extraire un secteur du disque dur formaté en quelque chose de proche de FAT12 qui contenait une liste de fichiers et leurs identifiants uniques associés. Il a isolé 2 *direntries*, les a affiché en hexadécimal, et vous demande de séparer les champs. Utilisez le modèle de structure d'une *direntry* pour séparer les différentes données et remplir les tableaux suivants avec les valeurs hexadécimales.

```
struct direntry {
  char[11] name;
  char     attributes;
  int      first_cluster;
  long     size;
} __attribute__((packed))
```

Les types de données font :
- char : 1 octet (8 bits)
- int : 2 octets (16 bits)
- long : 4 octets (32 bits)

Rappel : char[11] correspond à un tableau de 11 cases (de 0 à 10)

```
direntry 1 :
47 45 4E 53 48 49
4E 00 4A 50 47 1F
00 23 00 00 12 12

direntry 2 :
49 4D 50 41 43 45
00 00 54 58 54 1F
00 2A 00 00 00 0B
```

	direntry[0] (f1)						direntry[1] (f2)					
name	47	45	4E	53	48	49	49	4D	50	41	43	45
	4E	00	4A	50	47		00	00	54	58	54	
attributes	1F						1F					
first_cluster	00 23						00 2A					
size	00 00 12 12						00 00 00 0B					

2.2 (3 points) Deuxième étape : conversion des noms

L'étudiant semble dubitatif et vous demande de lui présenter des valeurs lisibles. Pour cela, il vous fournit une toute petite table de conversion ASCII... mais vous vous apercevez qu'il n'a pas du tout recopié la partie la plus essentielle de la table (probablement car il ne prenait pas assez de notes) : sa table ne dispose que de la correspondance entre des caractères et leurs valeurs en base 10. Il vous faut donc convertir les valeurs hexadécimales à la main (zut alors).

Concernant les noms de fichiers, la norme FAT12 précise que sur les caractères, les 8 premiers servent à coder le nom du fichier, et les 3 derniers servent à coder l'extension. Vous devez donc ajouter un point pour séparer l'extension du nom de fichier.

Dec	Char
48	0
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M

Dec	Char
49	1
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z

f1	G	E	N	S	H	I	N		.	J	P	G
f2	I	M	P	A	C	T			.	T	X	T

2.3 (3 points) Troisième étape : conversion des champs

L'étudiant commence à avoir confiance en vous : vous avez trouvé les bons noms de fichiers (et il reconnait les noms). Afin de pouvoir vous donner les bons clusters à lire, il vous demande de convertir cette fois la taille contenue dans les champs de chaque direntry, ainsi que le numéro du premier cluster, en nombre décimaux (afin qu'il puisse les mettre en paramètre de son outil d'extraction).

Ici, les entiers sont en big endian, c'est-à-dire que les octets sont ordonnés de la même façon que nous représentons les nombres : AUCUNE transformation ou réorganisation n'est nécessaire, vous n'avez qu'à convertir les nombres comme dans la première partie de l'examen.

	taille du fichier	numéro du premier cluster
direntry 1 (f1)	4626	35
direntry 2 (f2)	11	42

2.4 (2 points) Quatrième étape : lecture d'un fichier

L'étudiant est surexcité, vous êtes sur le point de l'aider à retrouver des fichiers et leurs contenus ! L'image est par définition difficile à interpréter en l'état, il vous propose donc plutôt de regarder le fichier texte. Il sait qu'il y a des chiffres et des lettres à l'intérieur.

Convertissez maintenant le contenu du fichier en ASCII, et si possible, de retrouver le numéro codé en binaire dans ce texte. **Souvenez-vous que la taille du fichier est importante pour cette étape : on ne convertit que les octets nécessaires, pas plus.**

```
47 52 41 44 49 55
53 30 30 31 30 31
30 30 31 30 30 31
```

texte :

GRADIUS0010
(1001001)

numéro :

2