



Durée : 2h — Documents autorisés

■ ■ ■ Unix & Gestion des processus — (3 points)

- 1– a. Sur quels mécanismes matériels repose l'**ordonnement** des processus dans Unix et pourquoi un processus ne peut pas bloquer, de manière accidentelle ou malveillante, l'exécution des autres processus ? **3pts**
- b. Est-ce qu'un utilisateur malveillant peut prendre le contrôle de la machine depuis l'écriture de son programme ?
Vous donnerez différents scénarios possibles et les mécanismes de protection.

■ ■ ■ Réseaux et protocoles — (3 points)

- 2– a. Comment peut-on identifier la nature des communications dans un réseau (Web, mail etc.) ? **3pts**
- b. Décrivez comment une attaque «Man-In-The-Middle» peut avoir lieu dans un réseau local. Est-ce possible dans un réseau global ? Si oui, comment ? Si non pourquoi ?

■ ■ ■ Programmation Python — (8 points)

Les deux exercices partageant un même objectif, vous êtes libre de donner un seul programme pour répondre à ces deux exercices.

- 3– On veut écrire un programme permettant de coder/décoder le code Morse uniquement pour les chiffres : **4pts**
- | | |
|---|--------------------|
| 1 | • — — — — |
| 2 | •• — — — — |
| 3 | ••• — — — — |
| 4 | •••• — — — — |
| 5 | ••••• — — — — |
| 6 | •••••• — — — — |
| 7 | ••••••• — — — — |
| 8 | •••••••• — — — — |
| 9 | ••••••••• — — — — |
| 0 | •••••••••• — — — — |
- a. Écrire un programme Python réalisant le codage de la manière suivante : 4532 ⇒ *****- ***** ***--- **----
 - b. Écrire un programme Python réalisant le décodage.

- 4– Écrivez en Python un programme réseau en mode **serveur**, permettant de réaliser les opérations de codage/décodage du code Morse au travers d'une connexion TCP suivant le protocole : **4pts**

o une requête de codage :

| Client | Serveur | Description |
|----------|--------------------|----------------------------|
| E 565765 | | transmission de la requête |
| | -..... | réponse du serveur |

o Une requête de décodage :

| Client | Serveur | Description |
|----------------------|---------|----------------------------|
| D -..... | | transmission de la requête |
| | 565765 | réponse du serveur |

Remarques :

- Une transaction requête/réponse correspond à une connexion TCP.
- Si vous n'avez pas fait l'exercice 3, vous pouvez considérer que vous disposez des commandes `codage_morse` et `decodage_morse`, marchant de la façon suivante :

```

pef@darkstar
$ codage_morse 565765
..... -..... .....
```

■ ■ ■ Segmentation & Pagination — (6 points)

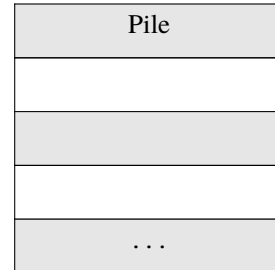
5– Gestion de la pile :

3pts

```
1 #include <stdio.h>
2
3 int add(int);
4
5 int mult(int a)
6 {     int v = a*2;
7       if (v<11) v = add(v);
8       return v;
9 }
10
11 int add(int b)
12 {     int v = b+3;
13       if (v<11) v = mult(v);
14       return v;
15 }
16
17 int main()
18 {     int valeur_initiale = 1;
19
20       printf("%d\n", add(valeur_initiale));
21 }
```

Détaillez l'utilisation de la pile lors de l'exécution du programme.

Vous pourrez utiliser une ou plusieurs tables représentant chacune un état de la pile.



6– Soit le morceau de programme suivant :

3pts

```
1 #include <stdlib.h>
2 #include <malloc.h>
3
4 int valeurs[10] = { 1, 2, 3, 4, 5, 6,
5 7 8, 9, 10 };
6
7 int *initialiser_tableau(int coeff)
8 { int i;
9   int tab[10];
10
11   for (i=0; i < 10; i++)
12     { tab[i] = valeurs[i]*coeff; }
13
14   return tab;
15 }
16
17 int main()
18 {
19   int *donnees = initialiser_tableau(10);
20   ...
21 }
```

a. Donnez la localisation de chacune des variables dans les différents **segments** du processus exécutant le programme.

b. Est-ce que le programme est **correct** dans son utilisation de la mémoire ? Expliquez pourquoi ?

c. Est-ce que l'utilisation du **tas**, «*heap*», est nécessaire ?

Donnez un programme modifié exploitant le tas.

Est-ce qu'il fonctionne mieux ?