

**Modalités**

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document n'est autorisé.
- La calculatrice est autorisée.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune sortie n'est autorisée avant une durée incompréhensible d'une heure.
- Aucun déplacement n'est autorisé.
- Aucun échange, de quelque nature que ce soit, n'est possible.

**QCM à répondre sur votre copie et non pas sur ce sujet (5 points)**

- a) Le Système d'Exploitation...
- est une couche matérielle qui gère le logiciel de présentation.
  - est responsable du partage des ressources matérielles de l'ordinateur.
  - est stocké dans le secteur 0 du MBR.
  - Aucune des réponses n'est valide.
- b) Un appel système...
- constitue une interface entre l'interface graphique et la ligne de commande.
  - s'exécute au niveau utilisateur.
  - entre dans le kernel.
  - Aucune des réponses n'est valide.
- c) La commande *parted*...
- est un outil de partitionnement.
  - permet d'étendre la taille du système de fichiers d'une partition.
  - permet d'entrer dans le mode de récupération du système d'exploitation.
  - Aucune des réponses n'est valide.

- d) Un tableau alloué avec malloc() sera stocké dans
- la région du code (text segment).
  - la région des données (data segment).
  - la pile (stack).
  - le tas (heap).
- e) Pour créer une image à partir d'un conteneur Docker en exécution nous utiliserons
- docker build
  - docker ps
  - docker commit
  - docker-compose

### Questions de cours (5 points)

- a. Décrire le processus de démarrage de Linux.
- b. Expliquer la différence entre l'allocation contiguë et l'allocation chaînée de fichiers, en précisant les avantages et inconvénients de chacune.
- c. Que contient le MBR (Master Boot Record) ?
- d. Expliquez le rôle de la zone de swap.
- e. Expliquez comment un processus peut devenir zombie.

### Exercice 1 : Programmation de processus (4 (2+2) points )

1- Combien de processus sont créés par le programme suivant ? Expliquez.

```
# include <unistd.h>
# include <stdio.h>

int main() {
    int pid, i;
    for (i=0; i<2; i++)
    {
        pid = fork();
        if (pid < 0)
        {
            printf("echec de fork\n");
            exit(-1);
        }
    }
    return 0 ;
}
```

2- Ecrire un programme qui crée un processus fils. Ce dernier envoie à son père la valeur 20. Le père utilise waitpid pour attendre son fils, récupère la valeur envoyée par le fils et l'affiche.

## Exercice 2 : Le système de fichiers (2 points)

Considérons un système de fichiers de type Unix dans lequel l'information concernant les blocs de données de chaque fichier est accessible à partir du i-noeud.

Supposons que :

- Le système de fichiers utilise des blocs de données de taille fixe 2K (2048 bytes).
- L'i-noeud de chaque fichier (ou répertoire) contient 12 pointeurs directs sur des blocs de données, 1 pointeur indirect simple, 1 pointeur indirect double et 1 pointeur indirect triple.
- Chaque pointeur (numéro de bloc) est représenté sur 4 bytes.

a) Quelle est la plus grande taille de fichier que peut supporter ce système de fichiers?

b) Considérons un fichier de taille 1 MByte (1024 KB). Combien de blocs de données pointés par les **blocs d'indirection** sont-ils nécessaires pour représenter ce fichier sur le disque ?

## Exercice 3 : Gestion de processus - Ordonnancement (2 points)

Soient quatre processus prêts A, B, C et D avec les temps d'arrivée au système et les temps d'exécution indiqués dans le tableau ci-dessous :

Processus	Temps d'exécution	Temps d'arrivée
A	3	0
B	3	1
C	2	3
D	2	7

En supposant que le temps de commutation est nul, calculer :

- le temps de séjour de chaque processus.
- le temps moyen de séjour.
- le temps d'attente : temps de séjour - temps d'exécution.
- le temps moyen d'attente.
- le nombre de changements de contexte

en utilisant les techniques :

1. *First-come, first-served (FCFS)*.

2. *Round-robin* avec un quantum de 2.

Pour chaque technique, dessiner la séquence des processus qui s'exécutent à chaque moment.

#### Exercice 4 : Mémoire virtuelle (2 points)

Dans une architecture hypothétique où les adresses virtuelles sont sur 14 bits, les 4 bits à gauche dans l'adresse dénotent la page et les 10 bits à droite dénotent le déplacement par rapport au début de la page (offset). En sachant que les premières entrées de la table des pages (correspondance « pages virtuelles – cadres réelles ») sont :

0	3
1	6
2	7
3	1
	:

Transformez les deux adresses virtuelles suivantes (exprimées en notation binaire), en adresses physiques qui sont sur 13 bits.

- 101010101010
- 1