

	EX - Examen n°1	
T. Gherbi - J.A. Lorenzo	Système d'exploitation	
ING1-GI	Année 2017-2018	

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.
- Une seule feuille manuscrite (pas de photocopies) est autorisée.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.
- Aucun échange, de quelque nature que ce soit, n'est possible.

Exercice 1 : Programmation de processus (3 points)

Écrivez un programme C qui crée un fils. Chaque processus doit afficher son PID à l'écran. Ensuite, le père doit attendre la terminaison du fils. Lorsque le fils termine, il enverra un code de retour, qui doit être récupéré par le père et affiché à l'écran.

Exercice 2 : Gestion des fichiers (4 points)

Supposons un noeud d'information (*i-node*) en Unix (10 adresses de bloc directs et 3 indirects) contenant un fichier. Quelle est la taille maximale d'un fichier si nous avons des blocs de 2 Kbytes et le numéro de bloc est donné sur 16 bits ? (**Note** : Vous n'avez pas de calculatrice, donc il suffit de proposer la réponse même si vous ne faites pas toutes les opérations).

Exercice 3 : Ordonnanceur (4 points)

Soient trois processus A, B et C prêts tels que A est arrivé en premier suivi de B, 2 unités de temps après et C, 1 unité de temps après B. Les temps nécessaires pour l'exécution des processus A, B et C sont respectivement 8, 4 et 2 unités de temps. Le temps de commutation est supposé nul. Calculer :

- le **temps de séjour** de chaque processus.
- le **temps moyen de séjour**.
- le **temps d'attente** : temps de séjour - temps d'exécution du travail.
- le **temps moyen d'attente**.
- le nombre de **changements de contexte** (Note : un changement de contexte se produit à chaque fois qu'un processus acquiert le processeur)

en utilisant les techniques :

1. SRT (Shortest Remaining Time)
2. Round robin (quantum = 3 unités de temps)

Processus	Temps d'exécution	Temps d'arrivage
A	8	0
B	4	2
C	2	3

Exercice 4 : Conteneurs (3 points)

Supposons que je démarre un conteneur Docker avec la commande `docker run -it ubuntu /bin/bash`. Ensuite je modifie un fichier texte à l'intérieur du conteneur et, finalement, je sors du conteneur avec `exit`. Plus tard, je redémarre le conteneur avec la même commande. Le fichier texte contiendra-t-il les modifications apportées ? Si oui, justifiez. Si non, expliquez pourquoi et dites s'il est possible de retrouver la version modifiée du fichier texte.

Questions courtes (6 points)

- 1 | Dans le code suivant, combien de processus sont créés lorsque le programme est exécuté ? ☐

```
int main() {  
    fork();  
    fork();  
    exit();  
}
```
- 2 | Quelles sont les avantages et les inconvénients d'un système d'allocation chaînée dans un disque dur ? ☐
- 3 | Dans quelle région en mémoire sont placées les données allouées avec `malloc()` ? ☐
- 4 | Quelle est la différence entre un ordonnanceur sans réquisition (OSR) et avec réquisition (OAR) ? ☐
- 5 | Un programme contient une variable *a*. Après avoir appelé `fork()`, le fils contiendra une copie de la même variable, qui pourra être modifiée indépendamment de celle du père. Si on vérifie l'adresse de cette variable dans le père et dans le fils, nous trouvons qu'elle est la même. Comment peut-on avoir deux variables avec la même adresse en mémoire mais avec des valeurs différentes ? ☐
- 6 | C'est quoi un *défaut de page* et à quoi sert le *bit V* (valid-bit) de la table de pages ? ☐