	Examen Session Rattrapage **20/02/2025**
M. Allouch, T. Guidini, A. Larbi, J.A. Lorenzo,	Système d'exploitation
ING1 Informatique-Mathématiques Appliquées	Année 2024–2025

Nom :

Prénom :

Numéro carte étudiant :

Filière (GI ou GM) :

Groupe :

Remarque : N'oubliez pas de préciser en haut votre nom, votre prénom, GI ou GM et votre numéro de groupe.

Modalités

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- **Les réponses devront être fournies sur le sujet lui-même.**
- 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.
- **Le QCM peut contenir une ou plusieurs réponses valides.**
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.

QCM : Cocher ☒ les bonnes réponses: (5 points)

a. Les fonctions wait() et waitpid()...

❑ attendent la terminaison d'un processus en particulier
❑ arrêtent l'exécution d'un fils
❑ arrêtent l'exécution du processus qui fait l'appel
❑ Aucune des réponses n'est valide.

b. Ignorer temporairement la prise en compte des interruptions d'un niveau…

❑ est un désarmement des signaux
❑ est un masquage des signaux
❑ invalide le numéro de priorité par niveau d'interruption
❑ Aucune des réponses n'est valide

c. Le PCB (Process Control Block)...

❑ contient un indicateur spécifique pour le masquage des signaux
❑ contient des informations-clés sur le fonctionnement du processeur
❑ est utilisé par le système d'exploitation pour connaître l'état d'un processus
❑ Aucune des réponses n'est valide

d. Un processus en attente d'une ressource est un processus…

❑ Actif (running)
❑ Prêt (waiting)
❑ Terminé (zombie)
❑ Aucune des réponses n'est valide

e. La localité temporelle dans la mémoire cache…

❑ est la tendance à réutiliser des données voisines d'autres données récemment accédées
❑ est la tendance à réutiliser des données récemment accédées, comme les instructions d'une boucle
❑ est gérée par le système d'exploitation
❑ Aucune des réponses n'est valide

f. Dans les systèmes multitâches…

❑ l'utilisation de partitions de taille variable améliore l'usage de la mémoire
❑ un processus ne doit pas pouvoir accéder à la mémoire d'un autre processus

❏ la Loi de Parkinson nous assure que tous les processus auront une place dans la RAM.

❏ Aucune des réponses n'est valide

g. La Table de Pages…

❏ traduit l'offset de la page virtuelle vers le cadre de page

❏ retourne la valeur du bit de présence/absence

❏ indique la correspondance entre une page d'un processus et un cadre de page

❏ Aucune des réponses n'est valide.

h. Dans la gestion de la mémoire, la segmentation…

❏ remplace la pagination si le processus peut être divisé en espaces d'adressage indépendants

❏ permet aux programmes d'être divisés en espaces d'adressage indépendants

❏ permet la division du espace d'adressage en pages de tailles différentes

❏ Aucune des réponses n'est valide.

i. Quels systèmes de fichiers sont journalisés ?

❏ FAT16, ISO9660

❏ Btrfs, FAT16

❏ NTFS, Ext4

❏ Aucune des réponses n'est valide.

j. Dans un système UEFI…

❏ le bootloader est enregistré sous la forme d'un fichier .efi

❏ le bootloader est contenu dans le MBR du disque dur

❏ le bootloader est contenu dans le premier bloc de la partition active

❏ Aucune des réponses n'est valide.

Exercice 1 : Programmation de processus (2.5 points)

Considérons le programme suivant :

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

int main() {
  pid_t myPid;
  myPid = fork();
  switch (myPid) {
  case 0 :
     printf("mon PID est %d\n ", getpid ( ) );
     break;
  case -1 :
     printf ("Erreur de création de processus avec fork ");
     break;
  default :
     printf ("mon PID est %d\n ", getpid ( ) );
     break;
  }
  return 0;
}
```

Supposons que le système d'exploitation donne un PID = 100 au père et un PID = 101 au fils.

a) Donnez et expliquez la sortie à l'écran de ce programme pour le processus père et pour le processus fils.

Réponse :

b) Donnez la valeur de la variable myPid pour chacun.

Réponse :

Exercice 2 : Ordonnancement (4 points)

Soient quatre processus A, B, C et D avec les informations sur les temps d'arrivée et d'exécution indiquées dans le tableau ci-dessous :

Processus	Temps d'arrivée	Temps d'exécution
A	0	6
B	2	4
C	3	5
D	6	3

- Faire un schéma qui illustre l'exécution de ces différents processus pour chaque technique d'ordonnancement :

  NON Préemptif avec SJF (Shortest Job First) ;
  Round robin (quantum = 3 unités de temps)

- Calculer le temps de séjour de chaque processus, le temps moyen de séjour, le temps d'attente et le temps moyen d'attente et le nombre de changements de contexte pour chaque technique d'ordonnancement.

OBS : Chaque case correspond à une unité de temps d'exécution. La première case correspond à la première unité de temps d'exécution.

Réponse :

Donnez les formules pour calculer :

Temps de séjour =

Temps d'attente =

a) SJF NON Préemptif

- ***Réponse pour le schéma (0.5 point)***

Processus	***SJF NON Préemptif***																
A																	
B																	
C																	
D																	

- ***Réponse : Calcul de temps de séjour et temps d'attente (1 point)***

Processus	Temps de séjour	Temps d'attente
A		
B		
C		
D		

Temps de séjour moyen =

Temps d'attente moyen =

- ***Réponse : Calcul du nombre de changements de contexte (0.5 point)***

***b)* Round robin (quantum = 3 unités de temps)**

- ***Réponse pour le schéma (0.5 point)***

Processus	**Round robin (3 unités de temps)**																
A																	
B																	
C																	
D																	

- ***Réponse : Calcul de temps de séjour et temps d'attente (1 point)***

Processus	Temps de séjour	Temps d'attente
A		
B		
C		
D		

Temps de séjour moyen =

Temps d'attente moyen =

- ***Réponse : Calcul du nombre de changements de contexte (0.5 point)***

Exercice 3 : Mémoire virtuelle (4 points)

Soit une machine ayant 1 KBytes (1 Byte = 1 octet) de mémoire physique, divisée en pages de taille 512 Bytes, et adressable via une plage d'adressage virtuelle de 12 bits. Répondez aux questions suivantes :

a) Dans l'adresse virtuelle, combien de bits sont nécessaires pour le numéro de page virtuelle ?
Réponse:

b) Quelle est la taille maximale de mémoire virtuelle qui peut être gérée ?
Réponse:

c) En utilisant la table de pages ci-dessous, traduisez l'adresse virtuelle suivante en adresse réelle : 0 1 0 1 0 0 1 0 0 0 1 0

	Page number	Valid bit
7	00	1
6	01	0
5	11	1
4	01	1
3	10	0
2	01	0
1	10	1
0	11	0

Réponse:

d) Faut-il remplacer la page en mémoire physique ? Pourquoi ?
Réponse:

Exercice 4 : Le système de fichiers (2 points)

Supposons un nœud 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 ?

Réponse:

Exercice 5 : Docker (2.5 points)

Nous exécutons la commande suivante dans un terminal de notre ordinateur (désormais appelé *le host*):

$ docker run −i −t ubuntu /bin/bash

Sachant que c'est la première fois que docker est utilisé sur la machine, répondre les questions suivantes :

a) Une fois à l'intérieur du conteneur, je crée un fichier appelé *toto*:

root@9f5c175eb28b:/# touch toto

Est-ce que ce fichier sera aussi disponible dans le système de fichiers du host ? Pourquoi ? (0.75 pt)
Réponse :

b) J'arrête l'exécution du conteneur avec **docker stop 9f5c175eb28b**. Si j'exécute à nouveau la commande

$ docker run −i −t ubuntu /bin/bash

Sera le fichier *toto* disponible sur le conteneur démarré ? Pourquoi ? (0.75 pt)
Réponse :

c) Quel(s) commande(s) devrai-je exécuter pour récupérer le fichier *toto* ? (1 pt.)

Réponse :