



A. Bertails, N.A. Diago, T. Guidini, J.A. Lorenzo	Système d'exploitation
ING1 Informatique-Mathématiques appliquées ING1 Informatique-Mathématiques Apprentissage	Année 2022–2023

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.

QCM à répondre sur votre copie et non pas sur le sujet : (5 points)

- a) Quel est le rôle du MMU (Memory Management Unit) ?
- Elle gère les processus
 - Elle implémente le compteur ordinal
 - Elle transforme une adresse virtuelle en une adresse réelle
 - Elle gère la mémoire secondaire comme les disques et les clés usb
- b) Le processus A de priorité 7 s'exécute. Le processus B de priorité 5 se réveille. Le plus petit chiffre code la priorité la plus forte. Quelles sont les propositions justes :
- B interrompt l'exécution de A car B est plus prioritaire et l'ordonnancement est préemptif
 - A continue son exécution car il est plus prioritaire et l'ordonnancement est préemptif
 - A continue son exécution car l'ordonnancement est non préemptif
 - B interrompt l'exécution de A car B est plus prioritaire et l'ordonnancement est non préemptif

- c) Quelles sont les informations stockées dans un i-node disque classique en Unix ?
- Les attributs du fichier et 13 adresses de blocs contenant le fichier (10x direct, 3x indirect)
 - Le nom du fichier et 10 adresses de blocs contenant le fichier (7x direct, 3x indirect)
 - Les attributs du fichier et 16 adresses de blocs contenant le fichier (13x direct, 3x indirect)
 - Le nom du fichier et 13 adresses de blocs contenant le fichier (10x direct, 3x indirect)
- d) De quelles deux manières pouvons-nous créer une image Docker ?
- Avec *Docker Hub* à partir d'un Dockerfile, ou avec *docker build* à partir d'un conteneur existant
 - Avec *docker build* à partir à partir d'un conteneur existant, ou avec *docker commit* à partir d'un Dockerfile
 - Avec *Docker Hub* à partir d'un conteneur existant, ou avec *docker build* à partir d'un Dockerfile
 - Avec *docker commit* à partir d'un conteneur existant, ou avec *docker build* à partir d'un Dockerfile
- e) Quel est le noyau du système d'exploitation?
- Script
 - Commandes
 - Kernel
 - Shell

Questions de cours : (4 points)

- a) Est-ce que c'est possible de chercher un paquet avec *apt* sur notre machine linux si nous n'avons pas de connexion à Internet ?
- b) Nous avons un processus en état "running" (actif). Prise en compte le cycle de vie d'un processus, quels sont les états auxquels il peut passer ensuite ?
- c) Sous quelle condition un ordonnanceur Round Robin aura un comportement identique à celui d'un ordonnanceur FIFO (First In First Out) ?
- d) Mentionner, au moins, un système de fichiers propre à Linux et un autre de Windows.

Exercice 1 : Programmation de processus (2.5 points)

Supposons le code ci-dessous

```
int main()
{
    if (fork() == 0) {
        printf("a");
    }else {
        printf("b");
        wait(&status);
    }
    printf("c");
    exit(0);
}
```

Répondre aux questions suivantes :

- a) Quels messages sont affichés par le père et lesquels par le fils ?
- b) Quels sont les ordres possibles de ces messages ?

Exercice 2 : Le système de fichiers (2.0 points)

Supposons un nœud d'information (i-node) en Unix (10 adresses de bloc directs et 3 indirects : 1 indirect simple, 1 indirect double et 1 indirect triple.) contenant un fichier. Le système de fichiers utilise des blocs de données de taille fixe 3Ko. Chaque adresse de bloc est représentée sur 64 bits.

- a) Quelle est la taille maximale d'un fichier ?
- b) On considère un fichier contenant 200,000 octets. Combien de blocs de données sont-ils nécessaires (au total) pour représenter ce fichier sur disque ?

Exercice 3 : Ordonnancement (4 points)

Considérons cinq processus P1, P2, P3 et P4, dont les temps d'exécution et leurs temps d'arrivée respectifs sont donnés dans le tableau ci-dessous.

Processus	Temps d'exécution	Temps d'arrivée
P1	3	0
P2	6	1
P3	4	4
P4	2	6

- a) Faire un schéma qui illustre l'exécution de ces différents processus pour chaque technique d'ordonnancement : FCFS (First Come First Served), SJF (Shortest Job First, sans préemption), SJF (Shortest Job First, avec préemption) et RR (Round Robin, quantum = 2 unités de temps).
- b) Calculer le temps de séjour de chaque processus (intervalle de temps entre la soumission du processus et son achèvement), le temps moyen de séjour, le temps d'attente (temps de séjour - temps d'exécution du processus) et le temps moyen d'attente, pour chaque technique d'ordonnancement.

Exercice 4 : Mémoire virtuelle (2.5 points)

Considérons un ordinateur de 32 bits avec un système de mémoire virtuelle qui utilise des pages de 16KB et possède une mémoire physique de 1 GB. Calculer, en expliquant la démarche utilisée :

- a) Le format de l'adresse virtuelle.
- b) Le nombre maximum de pages dans cet ordinateur.
- c) Le nombre de cadres de page.

CORRIGÉ

Questions QCM (5 points)

- a) Quel est le rôle du MMU (Memory Management Unit) ?
- Elle transforme une adresse virtuelle en une adresse réelle
- b) Le processus A de priorité 7 s'exécute. Le processus B de priorité 5 se réveille. Le plus petit chiffre code la priorité la plus forte. Quelles sont les propositions justes :
- B interrompt l'exécution de A car B est plus prioritaire et l'ordonnancement est préemptif
- c) Quelles sont les informations stockées dans un i-node disque classique en Unix ?
- Les attributs du fichier et 13 adresses de blocs contenant le fichier (10x direct, 3x indirect)
- d) De quelles deux manières pouvons-nous créer une image Docker ?
- Avec *docker commit* à partir d'un conteneur existant, ou avec *docker build* à partir d'un Dockerfile
- e) Quel est le noyau du système d'exploitation ?
- Kernel

Questions de cours : (4 points)

- a) Est-ce que c'est possible de chercher un paquet avec *apt* sur notre machine linux si nous n'avons pas de connexion à Internet ?
Oui, la liste de paquets est stockée localement sur la machine, prise en compte que nous avons exécuté *apt update* précédemment. En revanche, nous ne pourrons pas l'installer sans une connexion à internet.
- b) Nous avons un processus en état “*running*” (actif). Prise en compte le cycle de vie d'un processus, quels sont les états auxquels il peut passer ensuite ?
Prêt, endormi ou zombie s'il a terminé.
- c) Sous quelle condition un ordonnanceur Round Robin aura un comportement identique à celui d'un ordonnanceur FIFO ?
Lorsque le quantum du Round Robin est égal ou supérieur au temps d'exécution du processus le plus long de la queue.
- d) Mentionner, au moins, un système de fichiers propre à Linux et un autre de Windows
EXT4 ou Btrfs pour Linux, et FAT (16,32) ou NTFS pour Windows.

Exercice 1 : Programmation de processus (2.5 points)

- a) Le père affiche b et, ensuite, c. Le fils affiche a et, ensuite, c.
- b) Nous ne pouvons rien supposer concernant l'ordre d'affichage des variables du fils par rapport au père, donc nous pourrons avoir : acbc, abcc ou bacc.

Exercice 2 : Le système de fichiers (2.0 points)

a) Bloc de données de taille 3Ko (3*1024 octets) et chaque adresse de bloc est représentée sur 64 bits (8 octets). Alors $3\text{Ko} / 8 = 3 \cdot 1024 / 8 = 384$ numéros de blocs dans un bloc.

blocs directs : 10 blocs

1 bloc indirect : 384 blocs

2 bloc indirect : $384 \cdot 384$ blocs

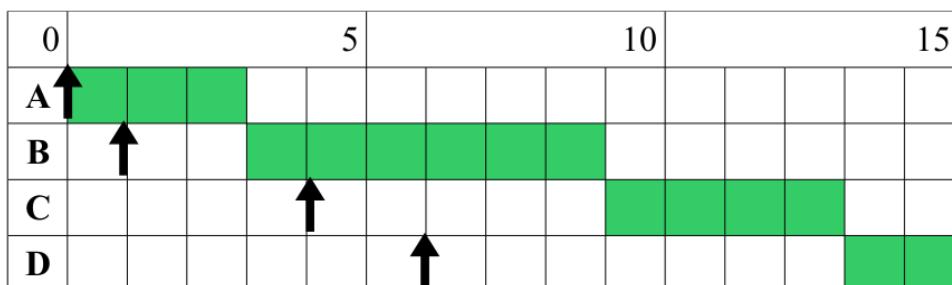
3 bloc indirect : $384 \cdot 384 \cdot 384$ blocs

Taille maximale d'un fichier = $(10 + 384 + 384 \cdot 384 + 384 \cdot 384 \cdot 384) \cdot 3\text{Ko}$

b) Taille du fichier divisé par la taille d'un bloc = $200000 / 3 \cdot 1024 \sim 65$ blocs

Exercice 3 : Ordonnancement (4 points)

FCFS



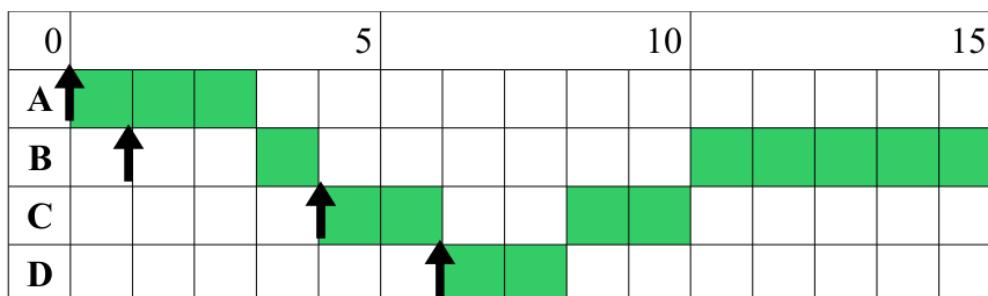
Processus	Temps de séjour	Temps d'attente
P1	3	0
P2	8	2
P3	9	5
P4	9	7
Moyenne	7.25	3.5

SJF non préemptif (SJF choisit de façon prioritaire les processus ayant le plus court temps d'exécution sans réellement tenir compte de leur date d'arrivée)



Processus	Temps de séjour	Temps d'attente
P1	3	0
P2	8	2
P3	11	7
P4	5	3
Moyenne	6.75	3

SJF préemptif (si un processus qui dure moins que le restant du processus courant se présente plus tard, l'UCT est donnée à ce nouveau processus)



Processus	Temps de séjour	Temps d'attente
P1	3	0
P2	14	8
P3	6	2
P4	2	0

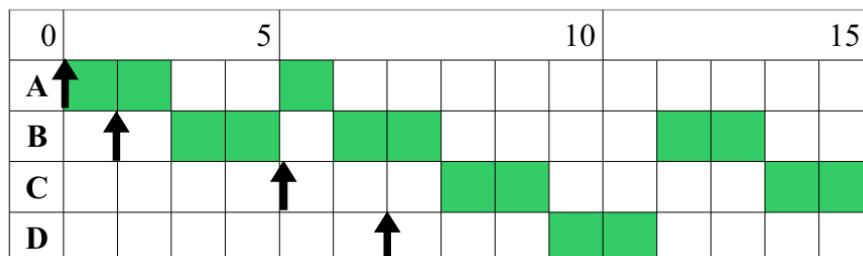
Moyenne	6.25	2.5
---------	------	-----

RR (2 temps)

P1 P1 P2 P2 P3 P3 P4 P4 P1 P2 P2 P3 P3 P2 P2

Processus	Temps de séjour	Temps d'attente
P1	9-0=9	9-3=6
P2	15-2=13	13-6=7
P3	13-4=9	9-4=5
P4	8-6=2	2-2=0
Moyenne	8.25	4.5

RR (q=2)



	Temps de rotation	Temps d'attente	Rendement
A	5	2	0,6
B	12	6	0,5
C	11	7	0,36
D	5	3	0,4
Moy	8,25	4,5	0,47

Exercice 4 : Mémoire virtuelle (2.5 points)

Ordinateur de 32 bits

Mémoire virtuelle avec pages de 16KB = $2^4 * 1024 = 2^4 * 2^{10} = 2^{14}$ bytes

Mémoire physique de 1 GB = 2^{30} bytes)

a) Le format de l'adresse virtuelle (1.0 point)

32 bits (ordinateur) – 14 bits (taille de la page) = 18 bits supérieurs pour le numéro de page

14 bits inférieurs pour l'offset ou déplacement (déterminé par la taille de la page = 16KB = 2^{14} bytes)

b) Le nombre maximum de pages dans cet ordinateur (0.75 point)

On peut adresser $2^{18} = 262144 = 2^8 * 2^{10} = 256$ Kpages de 16KB

c) Le nombre de cadres de page (0.75 point)

La mémoire physique est de 1 GB, ce qui équivaut à 2^{30} bytes. Donc 30 bits en mémoire physique. Chaque page a une taille de 16 KB = 2^{14} bytes, donc 14 bits d'offset. Alors $30 - 14 = 16$ bits pour le nombre de cadres. Alors $2^{16} = 65536$ cadres de page.