

	<b>Examen Session Rattrapage</b>	
L. Alouache - T. Gherbi - J.A. Lorenzo - S. Yassa	Système d'exploitation	
ING1-GI-GM	Année 2018–2019	

### 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.
- 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)

Ecrivez un programme qui crée un processus père. Ce dernier crée quatre processus fils. Chaque processus fils calculera le carré de son PID et affichera le résultat ainsi que la valeur du PID. Ensuite, chaque fils renvoie via exit la valeur calculée à son père. Le père attendra la terminaison de tous ses fils et montrera la somme des valeurs retournées.

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

Soit un système de fichiers Unix basé sur les i-nodes, formé de blocs de taille 2 Ko et utilisant des numéros de blocs sur 16 bits.

Calculez la taille maximale que peut prendre un fichier. Pour rappel, un i-node contient, en plus des propriétés du fichier, 10 adresses de bloc directes, et 3 adresses de bloc indirectes. Rappelez-vous que la taille maximale d'un fichier est le nombre maximum de blocs multiplié par la taille d'un bloc.

### Exercice 3 : Gestion de processus - ordonnancement (3 points)

Soient cinq processus prêts A, B, C, D et E, 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	6	0
B	2	3
C	2	4
D	4	6
E	1	8

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

- 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. FCFS (First Come First Served)
2. SRT (Shortest Remaining Time)

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

Soit une machine ayant 256 KBytes (1 Byte = 1 octet = 8 bits) de mémoire physique, divisée en pages de taille 512 Bytes, et adressable via une plage d'adressage virtuelle de 16 bits.

- a) Dans l'adresse virtuelle, combien de bits sont nécessaires pour le numéro de page virtuelle ?
- b) Quelle est la taille maximale de mémoire virtuelle qui peut être gérée ?

#### Questions de cours : (5 points)

- a) Que fait la commande *docker ps -a* ?
- b) Citez quatre segments mémoire du processus en indiquant leurs contenus.
- c) Qu'est-ce qu'une table de partitionnement et à quoi sert-elle ?
- d) Expliquez dans quelles conditions un processus peut devenir un processus zombie.
- e) Dans le cadre de la gestion d'une mémoire virtuelle paginée, quand pouvons-nous trouver un défaut de page ?

#### QCM : (5 points)

- a) Dans quel dossier nous pourrions trouver le fichier texte de configuration d'un programme installé dans notre ordinateur ?
  - ☐ /etc
  - ☐ /var
  - ☐ /lib
  - ☐ /cfg
- b) À quoi sert le fichier de configuration `/etc/apt/sources.list` ?
  - ☐ Contient la liste des paquets prêts à installer.
  - ☐ Contient la liste des « sources » à partir desquelles les paquets peuvent être obtenus.
  - ☐ Contient la liste des paquets installés.
  - ☐ Contient la liste des paquets à partir desquelles les « sources » peuvent être obtenus.
- c) L'allocation chaînée d'un fichier sur le disque...
  - ☐ ne permet pas l'accès aléatoire.
  - ☐ ne permet pas l'accès séquentiel.
  - ☐ ne permet pas l'accroissement.
  - ☐ ne permet pas les modifications.

d) Que fait la commande apt-get upgrade ?

- ☐ Met à jour les paquets déjà installés.
- ☐ Met à jour le noyau.
- ☐ Met à jour le fichier des liens des dépôts officiels.
- ☐ Met à jour le bios.

e) La multiprogrammation...

- ☐ est le processus d'écriture du code par plusieurs développeurs.
- ☐ est l'exécution de plusieurs processus en même temps.
- ☐ utilise toujours un seul compteur de programme.
- ☐ Aucune des réponses n'est valide.

## Corrigé de l'examen

### Exercice 1 : Programmation de processus (3 points)

Pseudocode :

```
for (i == 0 ; i < Nfils ; i++){
    pid[i] = fork() ;
    if (pid[i] == 0)
        break ;
}

// père
if(i == Nfils){
    for (i == 0 ; i < Nfils ; i++){
        wait( &status ) ;
        result += status ;
    }
    printf("Addition des carrées :%d\n", result) ;
    exit(0) ;
} else
{ // fils
    monpid = getpid() ;
    printf("Mon pid: %d. Valeur carrée :%d\n", monpid, monpid*monpid) ;
    return(monpid*monpid) ;
}
```

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

2 KB/ 2 bytes (16 bits) =  $2048 / 2 = 1024$  numéros de bloc dans chaque bloc

blocs directs = 10 blocs

bloc indirect\_1 = 1024 blocs

bloc indirect\_2 =  $1024^2$  blocs

bloc indirect\_3 =  $1024^3$  blocs

Nombre max. de blocs =  $10 + 1024 + 1024^2 + 1024^3 = 1074791434$  blocs

Taille maximale =  $1074791434 \text{ blocs} * 2 \text{ KB/bloc} = 2 \text{ TBytes}$

### Exercice 3 : Gestion de processus - ordonnancement (3 points)

#### 1. FCFS

0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17  
18

A	A	A	A	A	A	B	B	C	C	D	D	D	D	E			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

A arrive

B arrive C arrive

D arrive

E arrive

Processus	Temps séjour	Temps d'attente
A	6	6-6=0
B	8-3=5	5-2=3
C	10-4=6	6-2=4
D	14-6=8	8-4=4
E	15-8=7	7-1=6

$$T_{\text{moyen\_sejour}} = (6+5+6+8+7) / 5 = 6,4$$

$$T_{\text{moyen\_attente}} = (0+3+4+4+6) / 5 = 3,4$$

5 changements de contexte (en comptant le changement initial)

### 1. SRT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

A	A	A	B	B	C	C	A	E	A	A	D	D	D	D			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

A arrive

B arrive C arrive

D arrive

E arrive

Processus	Temps séjour	Temps d'attente
A	11-0=11	11-6=5
B	5-3=2	2-2=0
C	7-4=3	3-2=1
D	15-6=9	9-4=5
E	9-8=1	1-1=0

$$T_{\text{moyen\_sejour}} = (11+2+3+9+1) / 5 = 5,2$$

$$T_{\text{moyen\_attente}} = (5+0+1+5+0) / 5 = 2,2$$

7 changements de contexte (en comptant le changement initial)

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

Soit une machine ayant 256 KBytes (1 Byte = 1 octet = 8 bits) de mémoire physique, divisée en pages de taille 512 Bytes, et adressable via une plage d'adressage virtuelle de 16 bits.

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

Offset :  $2^9 = 512$ , donc 9 bits

Num. page virtuelle =  $16 - 9 = 7$  bits

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

$2^{16} = 65536$  bytes, ou 64 KBytes

Également :  $2^7$  pages \* 512 bytes/page

#### Questions de cours : (5 points)

- a) Que fait la commande *docker ps -a* ?  
Elle montre tous les conteneurs dans le host, même ceux qui ne sont pas en exécution.
- b) Citez quatre segments mémoire du processus en indiquant leurs contenus.  
Tas ou heap, stack ou pile, région des données et région du code (slide 7 séance 4).
- c) Qu'est-ce qu'une table de partitionnement et à quoi sert-elle ?
- d) Expliquez dans quelles conditions un processus peut devenir un processus zombie.
- e) Dans le cadre de la gestion d'une mémoire virtuelle paginée, quand pouvons-nous trouver un défaut de page ?

#### QCM : (5 points)

- f) Dans quel dossier nous pourrions trouver le fichier texte de configuration d'un programme installé dans notre ordinateur ?  
☐ /etc
- g) À quoi sert le fichier de configuration /etc/apt/sources.list  
☐ Contient la liste des « sources » à partir desquelles les paquets peuvent être obtenus.
- h) L'allocation chaînée d'un fichier sur le disque...  
☐ Ne permet pas l'accès aléatoire.
- i) Que fait la commande *apt-get upgrade* ?  
☐ Met à jour les paquets déjà installés.
- j) La multiprogrammation est...  
☐ L'activation de plusieurs processus en même temps.