

Choix Multiples

Question 1

1 / 1 point

Laquelle des transitions d'état suivantes n'est pas possible pendant la durée de vie d'un processus ?

- ☐ Nouveau -> Prêt
- ☒ En attente -> En exécution
- ☐ En exécution -> Terminé
- ☐ En exécution -> En attente

Question 2

1 / 1 point

La transition en exécution → prêt se produit lorsque

- ☐ Le processus est terminé
- ☒ Une interruption a été causée par un événement indépendant du processus
- ☐ Le processus est en état d'attente occupée.
- ☐ Le processus est en attente d'un événement d'entrée/sortie

Question 3

0 / 1 point

Laquelle des affirmations suivantes est correcte à propos de l'ordonnanceur à court et à long terme ?

- ☐ L'ordonnanceur à court terme est très rarement invoqué
- ☒ L'ordonnanceur à long terme est très souvent invoqué
- ☐ L'ordonnanceur à court terme est très souvent invoqué
- ☐ L'ordonnanceur à long terme est invoqué aussi souvent que l'ordonnanceur à court terme

Question 4**1 / 1 point**

L'appel système `fork()` sous Unix

- ☐ crée un nouveau processus avec le `process_id` dupliqué du processus parent
- ☐ crée un nouveau processus avec une mémoire partagée avec le processus parent
- ☒ crée un nouveau processus avec une copie de l'espace d'adresse du parent
- ☐ tout ce qui précède

Question 5**0 / 1 point**

Combien de processus sont créés à la fin de la boucle suivante ?

```
for(i=0 ; i< 5 ; i++) fork() ;
```

- ☒ 32
- ☐ 24
- ☒ 31
- ☐ 5

Question 6**1 / 1 point**

Quelle est la motivation des fils (threads) qui ne s'applique pas aux processus ?

- ☐ De nombreux threads peuvent s'exécuter en parallèle sur plusieurs CPU
- ☐ Tout ce qui précède
- ☐ Un fil s'occupe de l'interaction avec l'utilisateur tandis que l'autre fait le travail de fond (background)
- ☒ Faible surcharge de travail (overhead) pour passer d'un fil à l'autre

Question 7**1 / 1 point**

Le temps de réponse est le temps

- ☐ nécessaire à l'exécution d'un processus particulier
- ☐ que l'utilisation de l'UCT est minimale
- ☐ qu'un processus est en attente dans la file d'attente « prêt »
- ☒ qui s'étend du moment où une demande a été soumise jusqu'à la première action

Question 8**1 / 1 point**

Un processus tributaire des entrées/sorties (I/O bound)

- ☐ demande rarement des opérations d'entrée/sortie et passe moins de temps à effectuer des travaux de calcul
- ☐ demande fréquemment des opérations d'entrée/sortie et passe la plupart de son temps à effectuer des travaux de calcul
- ☐ demande rarement des opérations d'entrée/sortie et passe plus de temps à effectuer des travaux de calcul
- ☒ demande fréquemment des opérations d'entrée/sortie et passe moins de temps à effectuer des travaux de calcul

Question 9**1 / 1 point**

Le débit est

- ☐ le nombre de processus liés aux entrées/sorties (I/O bound) qui utilisent le CPU par unité de temps
- ☐ le nombre de processus qui passent de l'état prêt à l'état en cours
- ☐ le nombre de processus qui sont mis en file d'attente par unité de temps
- ☒ le nombre de processus qui achèvent leur exécution par unité de temps

Question 10**0 / 1 point**

Lequel des énoncés suivants définit le « plus court d'abord préemptif (SJF) » ?

- ☐ Si un nouveau processus arrive avec une priorité plus élevée que le processus en cours d'exécution, interrompre le processus en exécution .
- ☒ Si un nouveau processus arrive avec une longueur de rafale de CPU inférieure au temps restant du processus en cours d'exécution, interrompre le processus en exécution .
- ☐ Si un nouveau processus arrive avec une priorité plus élevée que le processus en cours d'exécution et une durée de rafale de CPU plus courte, interrompre le processus en exécution
- ☒ Si un nouveau processus arrive avec une longueur de rafale CPU (CPU Burst) inférieure à la longueur de rafale CPU du processus en cours d'exécution, interrompre le processus en exécution .

Question 11**1 / 1 point**

Dans l'ordonnancement tourniquet (Round-Robin), la longueur du quantum

- ☐ Doit être plus grande que la longueur de rafale typique du CPU
- ☐ Doit être au moins la moitié de la longueur de rafale typique
- ☒ Doit être égale à la rafale de CPU (CPU burst) typique
- ☐ Doit être plus courte que la durée de rafale typique du CPU

Question 12**1 / 1 point**

Laquelle des affirmations suivantes concernant l'ordonnancement avec des files d'attente à plusieurs niveaux n'est pas vraie ?

- ☒ Les processus interactifs ont une priorité plus élevée que les processus système
- ☐ Les processus par lots (batch) ont une priorité moindre que les processus système
- ☐ Les processus système ont une priorité plus élevée que les processus interactifs
- ☐ Les processus interactifs ont une priorité plus élevée que les processus par lots

Question 13**0 / 1 point**

Parmi les paramètres suivants, lesquels ne sont pas utilisés pour définir un ordonnanceur de file d'attente à plusieurs niveaux ?

- ☐ nombre de files d'attente
- ☐ algorithmes d'ordonnancement pour chaque file d'attente
- ☒ méthode utilisée pour déterminer quand il faut améliorer ou rétrograder un processus
- ☒ Aucune de ces réponses

Question 14**1 / 1 point**

Dans un algorithme d'ordonnancement du CPU, trois files d'attente sont utilisées (Q2, Q1 et Q0). Q0 a la priorité la plus élevée tandis que Q2 a la priorité la plus faible. Si l'algorithme d'ordonnancement est une file d'attente à plusieurs niveaux, lequel des énoncés suivants est correct ?

- ☐ Si Q2 est vide, les processus de Q0 sont sélectionnés.
- ☐ Si Q2 est vide, les processus de Q1 sont sélectionnés.
- ☒ Si Q0 est vide, les processus de Q1 sont sélectionnés.
- ☐ Si Q1 est vide, les processus de Q0 sont sélectionnés.

Question 15**0 / 1 point**

Dans la programmation multiprocesseur, l'affinité avec le processeur peut être garantie en exécutant les processus

- ☐ sur l'UCT physique avec le débit le plus faible
- ☒ toujours sur la même UCT physique
- ☒ sur l'UCT physique ayant le débit le plus élevé
- ☐ en alternance sur les multiples UCT physiques

Question 16**1 / 1 point**

Dans l'ordonnancement avec priorités

- ☐ Les processus ayant le même niveau de priorité peuvent être traités par l'ordonnancement "premier arrivé, premier servi".
- ☐ le traitement en temps réel est autorisé
- ☐ Les processus ayant le même niveau de priorité peuvent être traités par l'ordonnancement tourniquet
- ☒ Tout ce qui précède

Question 17**1 / 1 point**

Dans l'ordonnancement, le terme "vieillessement" implique

- ☐ les processus à priorité élevée empêchant les processus à faible priorité d'obtenir le CPU
- ☐ les processus bloqués dans des files d'attente si longues qu'ils meurent
- ☒ d'augmenter progressivement la priorité d'un processus de manière à ce qu'il soit finalement exécuté
- ☐ des processus prêts à exécuter mais bloqués en attendant indéfiniment le CPU

Question 18**1 / 1 point**

Dans l'ordonnancement du CPU, le terme "temps d'attente" désigne le temps

- ☐ qu'il s'écoule entre le moment où une demande a été soumise et celui où la première réponse est produite
- ☐ qu'un processus est en attente dans l'état d'attente
- ☐ entre la soumission du travail et son achèvement
- ☒ qu'un processus est en attente dans la file d'attente des prêts

Question 19**1 / 1 point**

Dans l'ordonnancement tourniquet, le quantum de temps ne doit pas être _____ temps de changement de contexte.

- ☐ grand par rapport au
- ☐ de la même taille que le
- ☐ sans rapport avec le
- ☒ petit par rapport au

Question 20

1 / 1 point

Le changement de contexte entre les processus est effectué par le

- ☐ gestionnaire d'interruptions
- ☒ expéditeur (dispatcher)
- ☐ ordonnanceur à court terme
- ☐ gestionnaire de fils

Résolution de problèmes

Examinez l'ensemble des processus suivants, la durée de la rafale du CPU (CPU Burst) étant donnée en millisecondes. Dans la dernière colonne, 0 indique la priorité la plus élevée tandis que 2 indique le niveau de priorité le plus bas.

Processus	Temps d'arrivée (ms)	Temps de rafale (burst time) (ms)	Priorité
P1	0	10	2
P2	3	3	1
P3	4	2	0
P4	5	3	0
P5	6	5	1

Dessinez trois diagrammes de Gantt qui illustrent l'exécution de ces processus à l'aide des algorithmes FCFS, plus court d'abord (SJF) préemptif, et de l'ordonnancement tourniquet avec priorités (quantum = 2).

L'algorithme SJF n'utilise pas la priorité pour faire sa décision.

Pour éviter toute confusion ici : Si un processus de la plus haute priorité est actuellement en exécution et qu'un autre processus de la plus haute priorité arrive (i.e., leur priorité est la même), le processus nouvellement arrivé **DOIT ATTENDRE JUSQU'À LA FIN DU QUANTUM DU PROCESSUS ACTIF**. Si un processus d'une classe de priorité inférieure est actuellement en exécution et qu'un processus d'une classe de priorité supérieure est arrivé, le processus nouvellement arrivé peut interrompre le processus actif sans attendre jusqu'à la fin du QUANTUM.

no priority for scfc

priority for RR

no priority for SJF

Question 21**2 / 2 points**

Dans l'algorithme FCFS, indiquez le temps quand chaque processus se termine.

- | | |
|--------------|--------|
| | 1. 1 |
| | 2. 2 |
| | 3. 3 |
| | 4. 4 |
| | 5. 5 |
| | 6. 6 |
| | 7. 7 |
| | 8. 8 |
| | 9. 9 |
| <u>23</u> P5 | 10. 10 |
| <u>15</u> P3 | 11. 11 |
| <u>18</u> P4 | 12. 12 |
| <u>13</u> P2 | 13. 13 |
| <u>10</u> P1 | 14. 14 |
| | 15. 15 |
| | 16. 16 |
| | 17. 17 |
| | 18. 18 |
| | 19. 19 |
| | 20. 20 |
| | 21. 21 |
| | 22. 22 |
| | 23. 23 |

Question 22

2 / 2 points

Dans l'algorithme SJF avec préemption, indiquez le temps quand chaque processus se termine.

- | | |
|--------------|--------|
| | 1. 1 |
| | 2. 2 |
| | 3. 3 |
| | 4. 4 |
| | 5. 5 |
| | 6. 6 |
| | 7. 7 |
| | 8. 8 |
| | 9. 9 |
| <u>16</u> P5 | 10. 10 |
| <u>8</u> P3 | 11. 11 |
| <u>11</u> P4 | 12. 12 |
| <u>23</u> P1 | 13. 13 |
| <u>6</u> P2 | 14. 14 |
| | 15. 15 |
| | 16. 16 |
| | 17. 17 |
| | 18. 18 |
| | 19. 19 |
| | 20. 20 |
| | 21. 21 |
| | 22. 22 |
| | 23. 23 |

Question 23

2 / 2 points

Dans l'algorithme RR avec priorité, indiquez le temps quand chaque processus se termine.

- | | |
|--------------|--------|
| | 1. 1 |
| | 2. 2 |
| | 3. 3 |
| | 4. 4 |
| | 5. 5 |
| | 6. 6 |
| | 7. 7 |
| | 8. 8 |
| | 9. 9 |
| <u>9</u> P4 | 10. 10 |
| <u>6</u> P3 | 11. 11 |
| <u>23</u> P1 | 12. 12 |
| <u>16</u> P5 | 13. 13 |
| <u>11</u> P2 | 14. 14 |
| | 15. 15 |
| | 16. 16 |
| | 17. 17 |
| | 18. 18 |
| | 19. 19 |
| | 20. 20 |
| | 21. 21 |
| | 22. 22 |
| | 23. 23 |

Question 24**2 / 2 points**

Quel est le temps de rotation de P4 pour chacun des algorithmes d'ordonnancement:

- | | |
|------------------------------|--------|
| | 1. 1 |
| | 2. 2 |
| | 3. 3 |
| | 4. 4 |
| | 5. 5 |
| | 6. 6 |
| <u>6</u> SJF avec préemption | 7. 7 |
| <u>4</u> RR avec priorité | 8. 8 |
| <u>13</u> FCFS | 9. 9 |
| | 10. 10 |
| | 11. 11 |
| | 12. 12 |
| | 13. 13 |
| | 14. 14 |
| | 15. 15 |

Question 25**2 / 2 points**

Quel est le temps d'attente de P2 pour chacun des algorithmes d'ordonnancement:

*Notez que les choix commencent à 1=0ms d'attente

- | | |
|-------|-------------------------|
| | 1. 0 |
| | 2. 1 |
| | 3. 2 |
| | 4. 3 |
| __6__ | RR avec priorité |
| | 5. 4 |
| __8__ | FCFS |
| | 6. 5 |
| __1__ | SJF avec préemption |
| | 7. 6 |
| | 8. 7 |
| | 9. 8 |
| | 10. Aucune des réponses |
-