

1. Un processus est une version exécutable d'un programme. R : FAUX
2. Dans quel état est un processus qui est en attente des données devant être lues à partir d'un disque ? R : Bloqué

Quel affichage est impossible à obtenir avec le code suivant :

```
void questionAffichage() {  
  
    printf("A\n");  
    if (fork() == 0) {  
        printf("B\n");  
        exit(0);  
    } else if (fork() == 0) {  
        printf("C\n");  
        exit(0);  
    }  
    printf("D\n");  
    if (fork() == 0) {  
        printf("E\n");  
    }  
    if (fork() != 0) {  
        printf("F\n");  
        while(wait(NULL) > 0);  
        exit(0);  
    }  
    else {  
        printf("G\n");  
        while(wait(NULL) > 0);  
        exit(0);  
    }  
}
```

3. R : ACBDFEGG
4. On peut placer tous les signaux dans le masque de signaux. R : FAUX
5. Les deux commandes suivantes sont équivalentes :
6. `./prog1 2>&1 > tmpfile | ./prog2`
`./prog1 > tmpfile 2> | ./prog2`

R: FAUX

Quel énoncé est faux ?

Veuillez choisir une réponse :

- ☐ a.
Un tube nommé doit être détruit explicitement à l'aide de la fonction `remove()` ou une autre fonction analogue.
- ☒ b.
On peut assigner un gestionnaire de signal à tous les signaux. ✓
- ☐ c. Aucune de ces réponses.
- ☐ d.
Je ne réponds pas à la question (0 point).
- ☐ e.
Un appel `exec` ne change pas le `pid` du processus appelant.

7.

Quel énoncé est faux ?

Veillez choisir une réponse :

- ☒ a. Je ne réponds pas à la question (0 point). ❌
- ☐ b. Aucune de ces réponses.
- ☐ c. Le problème de famine dans le dîner des philosophes peut être solutionné en disant aux philosophes d'essayer de prendre les deux fourchettes en même temps.
- ☐ d. Selon le problème du producteur/consommateur, un problème d'interblocage peut survenir dans un moniteur lorsqu'un processus rédacteur tente d'écrire dans un tampon plein.
- ☐ e. Le problème de famine dans le dîner des philosophes peut être solutionné en disant aux philosophes de prendre les fourchettes une à une avec un ordre imposé.

Votre réponse est incorrecte.

La réponse correcte est : Le problème de famine dans le dîner des philosophes peut être solutionné en disant aux philosophes d'essayer de prendre les deux fourchettes en même temps.

- 8.
9. La sémantique Signal-and-wait utilisée dans les moniteurs implique une file d'attente prioritaire dans laquelle on place une tâche dès que celle-ci tombe en attente d'une variable de condition. R : FAUX
10. La barrière de synchronisation pthread_barrier permet d'attendre un nombre donné de processus avant de continuer un programme. R : FAUX
11. Sauvegarder l'état d'une ressource et le restaurer lorsqu'on a besoin de la réutiliser permet de contrer l'impact de la réquisition lorsqu'on parle du problème de la gestion des ressources. R : VRAI

Le code suivant provoque un interblocage :

```
int main () {
    mkfifo("monTube", 0660);
    if (lfork()) { // fils
        char m[50];
        int i;
        int fd = open("monTube", O_WRONLY);
        printf("Ici writer [%d]\n", getpid());
        // Ecriture dans le pipe
        for(i=0; i<4;i++) {
            sprintf(m, "mess%d de [%d]\n", i,getpid());
            write(fd, m, strlen(m) + 1);
        }
        close(fd);
    }
    else { // parent
        char m;
        int fd = open("monTube", O_RDONLY);
        // lire du pipe
        while (read(fd, &m, 1) > 0) {
            write(1,&m,1);
        }
        close(fd);
        wait(NULL);
    }
    exit(0);
}
```

12. R : FAUX
13. Afin de régler le problème d'attente circulaire, chaque processus doit demander une ressource à la fois, en leur attribuant un poids qui définit l'ordre dans lequel il faut la relâcher. R : VRAI
14. De façon générale, les stratégies de remplacement de page tiennent compte des phénomènes de localité basé sur le fait qu'il y a une forte probabilité que les mêmes pages soient référencées dans un intervalle de temps court. R : VRAI
15. Chaque programme a ses adresses internes (adresses virtuelles) et ses adresses réelles (adresses physiques) en mémoire. R : VRAI
16. Si on a une adresse de 25 bits à séparer en numéro de page et en déplacement et qu'on a des pages (et cadres) de 8192 octets, alors 10 bits sont utilisés pour le numéro de page. R : FAUX
17. Le nombre de pages peut être plus grand que le nombre de cadres. R : VRAI
18. Le MMU (*Memory Management Unit*) est un composant matériel (*hardware*) responsable de traduire une adresse virtuelle en une adresse physique. R : VRAI

On considère un système de pagination à 3 niveaux dans lequel les adresses (virtuelles et physiques) sont codées sur 64 bits. La taille maximale de chaque table de pages (peu importe son niveau) est de 64 KiO. Chaque entrée d'une table de pages est composée de 16 octets. Donnez le format d'une adresse virtuelle. Quelle est la taille d'une page ?

Veuillez choisir une réponse :

- ☐ a.
Aucune de ces réponses.
- ☒ b. Je ne réponds pas à la question (0 point). ❌
- ☐ c. 12 bits de 1er niveau; 11 bits de 2e niveau; 11 bits de 3e niveau; 30 bits d'offset. La taille d'une page est de 1 073 741 824 octets.
- ☐ d. 12 bits de 1er niveau; 12 bits de 2e niveau; 12 bits de 3e niveau; 28 bits d'offset. La taille d'une page est de 268 435 456 octets.
- ☐ e. 11 bits de 1er niveau; 11 bits de 2e niveau; 11 bits de 3e niveau; 29 bits d'offset. La taille d'une page est de 536 870 912 octets.
- ☐ f. 12 bits de 1er niveau; 11 bits de 2e niveau; 12 bits de 3e niveau; 29 bits d'offset. La taille d'une page est de 536 870 912 octets.

Votre réponse est incorrecte.

La réponse correcte est : 12 bits de 1er niveau; 12 bits de 2e niveau; 12 bits de 3e niveau; 28 bits d'offset. La taille d'une page est de 268 435 456 octets.

19.

Soit un ordinateur ayant une mémoire contiguë de 16 MiO. Les processus suivants sont alloués en mémoire (dans l'ordre) :

- P1 de taille 3 MiO ;
- P2 de taille 5 MiO ;
- P3 de taille 2 MiO ;
- P4 de taille 2 MiO.

Les processus P1 et P3 se terminent.

À quel endroit en mémoire le processus P5 de taille 2 MiO sera-t-il alloué si l'on utilise l'algorithme d'allocation de mémoire : première allocation (*first fit*) ?

Veuillez choisir une réponse :

- ☐ a. La plage d'adresse 12-14
- ☐ b. Aucune de ces réponses
- ☐ c. Aucune plage disponible
- ☒ d. La plage d'adresse 0-2 ✓
- ☐ e. Je ne réponds pas à la question (0 point)
- ☐ f. La plage d'adresse 8-10

Soit un processus accédant à la suite de pages suivantes : 7, 0, 1, 2, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2 et une mémoire physique de 4 cadres (C1, C2, C3 et C4).

	7	0	1	2	3	0	4	2	3	0	3	2	1	2
C1	7	7	7	7	3	3	3	3	3	3	3	3	1	1
C2	-	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	-	-	1	1	1	1	4	4	4	4	4	4	4	4
C4	-	-	-	2	2	2	2	2	2	2	2	2	2	2

Quel algorithme de remplacement de pages a été utilisé pour produire le tableau ci-dessus ?

Optimal



Combien y a-t-il de défauts de pages ?

7



- 21.
22. Un ordonnanceur non-préemptif est appelé lorsque l'état d'un autre processus bascule vers l'état prêt. R : FAUX
23. Les stratégies d'ordonnancement non-préemptives n'effectuent pas de réquisition. R : VRAI
24. Dans les stratégies d'ordonnancement préemptives, un quantum trop petit provoque trop de commutations de processus et abaisse l'efficacité du processeur. R : VRAI

Soit les 5 processus suivants : P1, P2, P3, P4 et P5. Les caractéristiques d'ordonnancement des processus sont :

Processus	Date d'arrivée	Temps d'exécution
P1	0	3
P2	1	5
P3	3	2
P4	9	5
P5	12	5

Le diagramme de Gantt est :

P1	P1	P1	P3	P3	P2	P2	P2	P2	P2	P4	P4	P4	P4	P4	P5	P5	P5	P5	P5
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Toutes les durées sont en seconde. Les temps de changement de contexte sont supposés nuls.

Quel algorithme d'ordonnancement a été utilisé pour produire le diagramme de Gantt ci-dessus ?

À priorité



Aucune de ces réponses

Quel est le temps moyen de séjour (TMS) ?

5.6



Quel est le temps moyen d'attente (TMA) ?

1.6



25.

Soit un système possédant les caractéristiques suivantes :

- Une mémoire physique de 32 MiO ;
- Une mémoire virtuelle de 64 MiO ;
- La taille d'une page est de 4 KiO.

Quel est le nombre de pages ?

2^{14}



Quel est le nombre de cadres ?

2^{13}



Quel est le nombre de bits nécessaires pour stocker le numéro de cadre ?

13

