

# IF-3-SYS : Examen final

**NOM Prénom :**

Durée : 1h30.

Répondez directement sur le sujet.

Remplissez votre nom tout de suite. Vraiment.

Les réponses incorrectes seront comptées négativement.

Tous documents papier autorisés. Pas de téléphone ni autre appareil.

Prenez la peine d'écrire lisiblement. Utilisez un brouillon plutôt que de faire des ratures.

## 1 Noyau et processus

**Question 1** Quel(s) appel(s) système faut-il invoquer pour charger un programme depuis le disque dur et le démarrer ? Cochez vrai pour «nécessaire» et faux pour «pas nécessaire».

<input type="checkbox"/>	<input type="checkbox"/>	clone
<input type="checkbox"/>	<input type="checkbox"/>	exec
<input type="checkbox"/>	<input type="checkbox"/>	fork
<input type="checkbox"/>	<input type="checkbox"/>	load
<input type="checkbox"/>	<input type="checkbox"/>	open
<input type="checkbox"/>	<input type="checkbox"/>	pthread_create
<input type="checkbox"/>	<input type="checkbox"/>	read
<input type="checkbox"/>	<input type="checkbox"/>	wait

**Question 2** On s'intéresse au fragment de programme ci-dessous. On note u,v les valeurs affichées par le processus parent et x,y les valeurs affichées par le processus enfant. Indiquez si les affirmations ci-dessous sont vraies ou fausses :

<input type="checkbox"/>	<input type="checkbox"/>	$u = x + 10$
<input type="checkbox"/>	<input type="checkbox"/>	$x = u + 10$
<input type="checkbox"/>	<input type="checkbox"/>	$y = v$
<input type="checkbox"/>	<input type="checkbox"/>	$y \neq v$

```
if (fork() == 0)
{
    a = a + 5;
    printf("%d,%d\n", a, &a);
}
else
{
    a = a - 5;
    printf("%d,%d\n", a, &a);
}
```

**Question 3** Parmi les éléments ci-dessous, le ou lesquels appartiennent au noyau du système d'exploitation ? Cochez «vrai» pour «dans le noyau» et faux sinon.

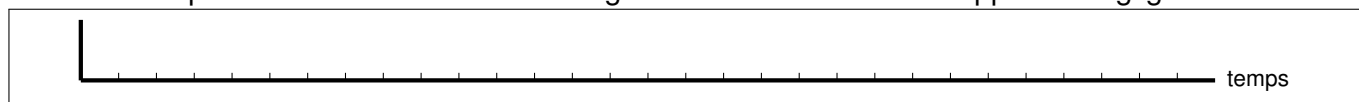
<input type="checkbox"/>	<input type="checkbox"/>	bibliothèque standard
<input type="checkbox"/>	<input type="checkbox"/>	compilateur
<input type="checkbox"/>	<input type="checkbox"/>	dispatcheur
<input type="checkbox"/>	<input type="checkbox"/>	mémoire cache
<input type="checkbox"/>	<input type="checkbox"/>	ordonnanceur
<input type="checkbox"/>	<input type="checkbox"/>	routine de traitement d'interruption
<input type="checkbox"/>	<input type="checkbox"/>	shell
<input type="checkbox"/>	<input type="checkbox"/>	système de fichiers

## 2 Ordonnancement

On suppose dans cette question un système monoprocesseur à ordonnancement circulaire (Round Robin) avec un quantum de 4 unités de temps. On s'intéresse à trois processus purement calculatoires A, B, et C aux caractéristiques suivantes :

tâche	A	B	C
instant d'arrivée	0	1	2
durée d'exécution	10	5	7

**Question 4** Sur le schéma ci-dessous, dessinez un chronogramme indiquant la succession des tâches sur le processeur. La durée des changements de contexte est supposée négligeable.



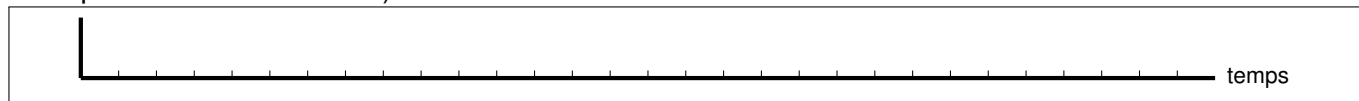
On suppose à présent que A et B (mais pas C) ont chacun une section critique, exécutée en exclusion mutuelle. Le tableau ci-dessous indique le détail de l'exécution de chaque processus. Un triplet (x, y, z) signifie que :

- le processus réalise un calcul ordinaire nécessitant x unités de temps processeur
- puis il demande à entrer en section critique
- pour exécuter et quitter sa section critique, il a besoin de y unités de temps processeur
- après sa section critique, il s'exécute encore pendant une durée z puis se termine.

Bien sûr le tableau n'inclut pas les temps d'attente éventuels avant l'accès aux sections critiques.

tâche	A	B	C
instant d'arrivée	0	1	2
durée d'exécution	10 (3, 6, 1)	5 (2, 2, 1)	7

**Question 5** Sur le schéma ci-dessous, dessinez un chronogramme indiquant la succession des tâches sur le processeur. Faites apparaître les moments où un processus est en section critique (par exemple avec des hachures).



## 3 Gestion mémoire

**Question 6** Citer les deux types de localité d'accès et expliquer en quoi l'un est un cas particulier de l'autre.

---



---



---



---



---



---

**Question 7** Le programme ci-dessous affiche l'adresse de sa variable i. S'agit-il d'une adresse physique ou d'une adresse virtuelle ?

- ☐ V ☐ F adresse physique  
☐ V ☐ F adresse virtuelle

```
#include <stdio.h>

int main(void)
{
    int i = 42;
    printf("adresse de i : %p\n", &i);
    return 0;
}
```

**Question 8** Lors d'un changement de contexte entre deux threads, à quelle condition faut-il invalider le contenu du *Translation Lookaside Buffer* ? Justifiez votre réponse.

---

---

---

---

---

---

**Question 9** Sur toutes les architectures matérielles, la taille des pages mémoire est toujours une puissance de 2. Pourquoi ?

---

---

---

---

---

---

**Question 10** Dans un allocateur dynamique, découper un bloc libre permet-il de réduire la fragmentation interne et/ou la fragmentation externe ?

- ☐ V ☐ F réduire la fragmentation interne  
☐ V ☐ F réduire la fragmentation externe

## 4 Systèmes de fichiers

Dans les questions suivantes, on s'intéresse à un système de fichiers basé sur une indexation multi-niveaux, avec des des blocs (secteurs) de 16kB ( $=2^{14}$  octets) identifiés par des numéros de 32 bits. Chaque inode contient 5 entrées pour des blocs directs, 1 entrée pour un bloc indirect et 1 entrée pour de bloc doublement indirect.

**Question 11** Quelle est la taille maximale d'un volume formaté avec ce système de fichiers ?

---

**Question 12** Quelle est la taille maximale d'un fichier ? Répondez par la puissance de deux la plus proche.

---

**Question 13** On suppose que le système d'exploitation vient de charger l'inode d'un fichier et qu'il veut maintenant sauter les 128 premiers méga-octets du fichier pour accéder directement à l'octet situé à la position 128MB ( $=134217728$ ).

Combien de blocs devra-t-il lire avant d'accéder à la donnée voulue ? Expliquez pour chaque bloc lu, ce qu'il contient et pourquoi sa lecture est nécessaire.

---



---



---



---



---



---

## 5 Synchronisation

**Question 14** Quelle est la signification du terme «attente active» ? En quoi est-ce généralement considéré comme une mauvaise façon d'attendre ?

---



---



---



---

---

---

**Question 15** Un programme crée 4 threads qui exécutent la même fonction `Inc()`. Cette fonction consiste en une incrémentation de 2 d'un même compteur global. Le programme attend ensuite la fin des 4 threads, affiche la valeur du compteur puis se termine. La valeur initiale du compteur est 0.

Ce programme est-il déterministe ? Si vous répondez oui, donnez la valeur affichée à l'écran. Si vous répondez non, donnez la plus petite valeur qui pourrait s'afficher. Dans les deux cas, justifiez votre réponse.

---

---

---

---

---

---

## Programmation à sémaphores

Un étudiant qui se spécialise en anthropologie et accessoirement en informatique s'est embarqué dans un projet de recherche pour voir s'il était possible d'enseigner les interblocages aux babouins d'Afrique. Il repère un profond canyon et y jette une corde au travers, de sorte que les babouins puissent le traverser à bout de bras. Plusieurs babouins ont le droit de traverser en même temps, pourvu qu'ils aillent tous dans la même direction. Si des babouins qui se dirigent vers l'est et d'autres vers l'ouest se trouvent sur la corde au même moment, cela conduit à un interblocage (les babouins sont bloqués au point de rencontre sur la corde) : en effet, ils n'ont pas la possibilité de passer les uns par-dessus les autres alors qu'ils sont suspendus au-dessus du canyon. Si un babouin souhaite traverser le canyon, il doit vérifier qu'aucun autre babouin ne traverse en sens inverse.

On modélise ce scénario par le programme ci-dessous, exécuté par une quantité arbitraire de threads :

babouins allant vers l'est

```
entree_ouest()  
traverser_d_ouest_en_est()  
sortie_est()
```

babouins allant vers l'ouest

```
entree_est()  
traverser_d_est_en_ouest()  
sortie_ouest()
```

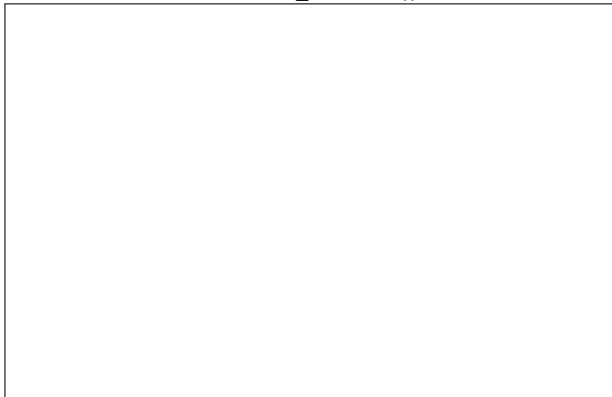
L'exercice va consister à écrire les synchronisations `entree_...()` et `sortie_...()` pour éviter l'interblocage au milieu du pont de singe.

**Question 16** Une solution simpliste consiste à n'autoriser qu'un seul singe à être sur la corde à chaque instant, indifféremment de sa direction. Implémentez ci-dessous les synchronisations nécessaires à l'aide de sémaphores et/ou de variables partagées.

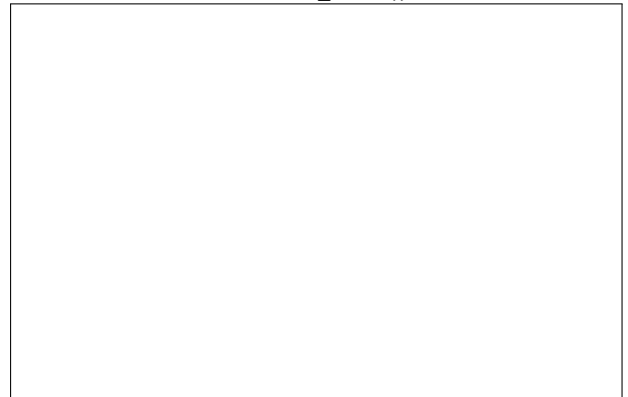
Conditions initiales



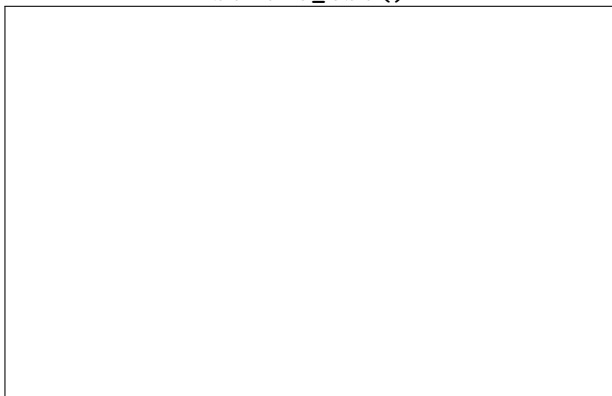
entree\_ouest()



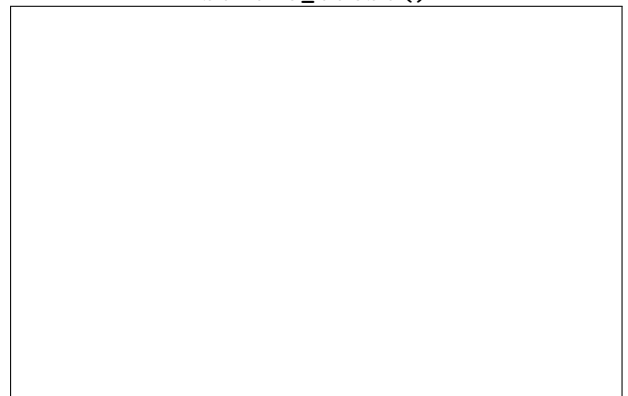
entree\_est()



sortie\_est()



sortie\_ouest()



N'hésitez pas à ajouter des explications pour aider votre correcteur à comprendre votre approche :

---

---

---

---

---

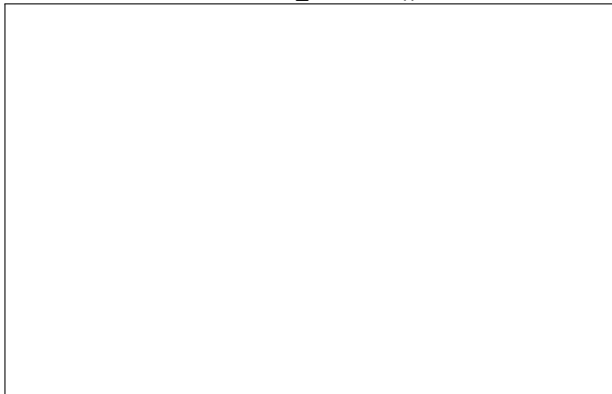
---

**Question 17** Proposez maintenant une solution permettant à plusieurs singes dans la même direction d'avancer simultanément. Ne vous préoccupez pas du cas d'un groupe infini de babouins se déplaçant dans un même sens et interdisant tout passage à ceux qui se déplacent dans l'autre sens.

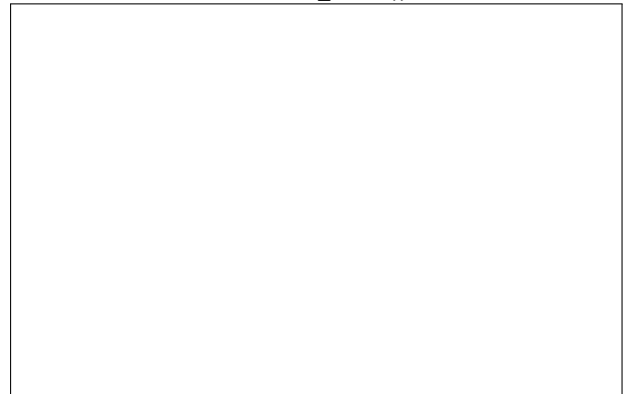
Conditions initiales



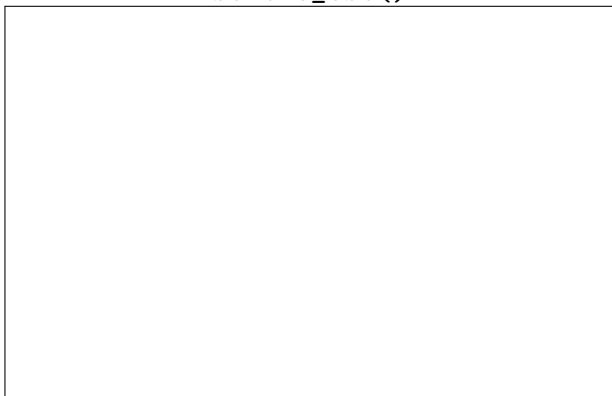
entree\_ouest()



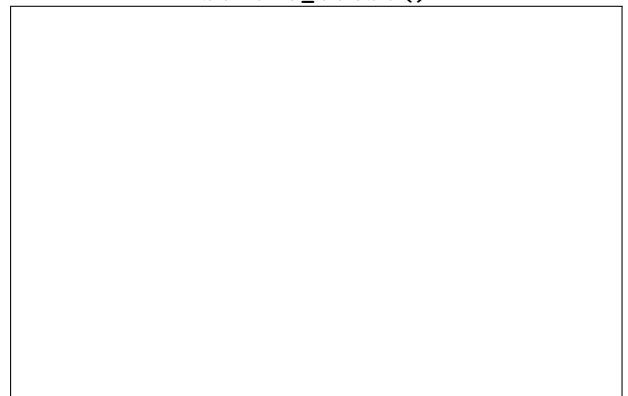
entree\_est()



sortie\_est()



sortie\_ouest()



N'hésitez pas à ajouter des explications pour aider votre correcteur à comprendre votre approche :

---

---

---

---

---

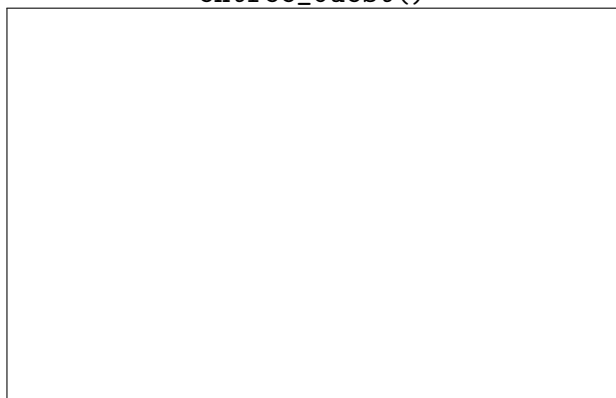
---

**Question 18** Même consigne qu'à l'exercice précédent, en évitant cette fois-ci le risque de famine : Lorsqu'un babouin qui souhaite traverser le canyon vers l'est arrive à la corde et trouve un babouin qui traverse vers l'ouest, il attend jusqu'à ce que la corde soit vide, mais aucun babouin se déplaçant vers l'ouest n'est autorisé à démarrer jusqu'à ce qu'au moins un babouin ait traversé dans l'autre sens.

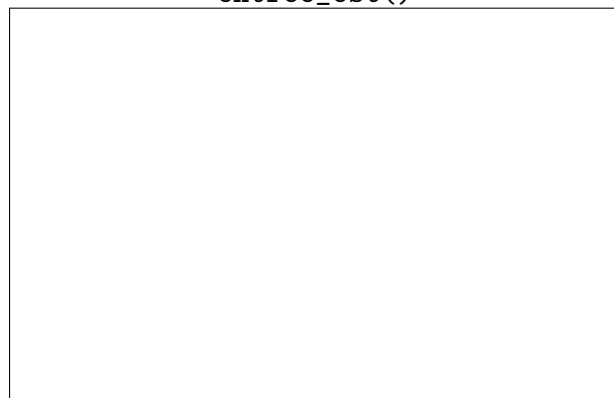
Conditions initiales

A large empty rectangular box intended for drawing the initial state of the system.

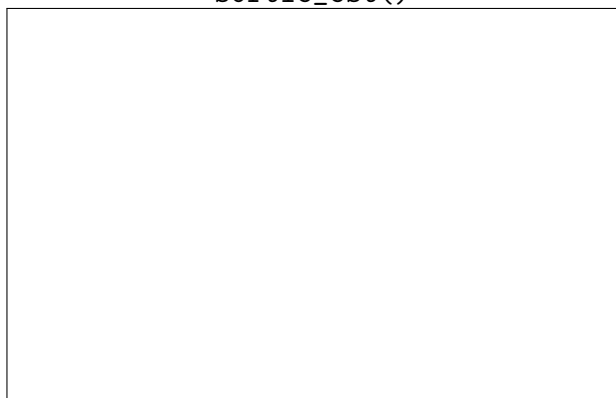
entree\_ouest()

A large empty rectangular box intended for drawing the state of the system after the 'entree\_ouest()' event.

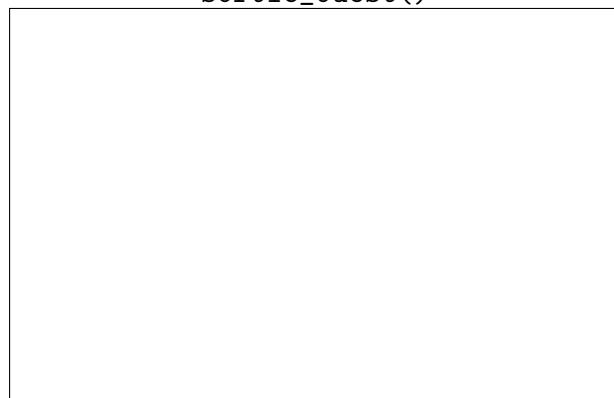
entree\_est()

A large empty rectangular box intended for drawing the state of the system after the 'entree\_est()' event.

sortie\_est()

A large empty rectangular box intended for drawing the state of the system after the 'sortie\_est()' event.

sortie\_ouest()

A large empty rectangular box intended for drawing the state of the system after the 'sortie\_ouest()' event.

N'hésitez pas à ajouter des explications pour aider votre correcteur à comprendre votre approche :

---

---

---

---

---

---



## Annexe : Quelques puissances de 2

$2^0 = 1$	$2^{16} = 65\,536$	$2^{32} = 4\,294\,967\,296$	$2^{48} = 281\,474\,976\,710\,656$
$2^1 = 2$	$2^{17} = 131\,072$	$2^{33} = 8\,589\,934\,592$	$2^{49} = 562\,949\,953\,421\,312$
$2^2 = 4$	$2^{18} = 262\,144$	$2^{34} = 17\,179\,869\,184$	$2^{50} = 1\,125\,899\,906\,842\,624$
$2^3 = 8$	$2^{19} = 524\,288$	$2^{35} = 34\,359\,738\,368$	$2^{51} = 2\,251\,799\,813\,685\,248$
$2^4 = 16$	$2^{20} = 1\,048\,576$	$2^{36} = 68\,719\,476\,736$	$2^{52} = 4\,503\,599\,627\,370\,496$
$2^5 = 32$	$2^{21} = 2\,097\,152$	$2^{37} = 137\,438\,953\,472$	$2^{53} = 9\,007\,199\,254\,740\,992$
$2^6 = 64$	$2^{22} = 4\,194\,304$	$2^{38} = 274\,877\,906\,944$	$2^{54} = 18\,014\,398\,509\,481\,984$
$2^7 = 128$	$2^{23} = 8\,388\,608$	$2^{39} = 549\,755\,813\,888$	$2^{55} = 36\,028\,797\,018\,963\,968$
$2^8 = 256$	$2^{24} = 16\,777\,216$	$2^{40} = 1\,099\,511\,627\,776$	$2^{56} = 72\,057\,594\,037\,927\,936$
$2^9 = 512$	$2^{25} = 33\,554\,432$	$2^{41} = 2\,199\,023\,255\,552$	$2^{57} = 144\,115\,188\,075\,855\,488$
$2^{10} = 1\,024$	$2^{26} = 67\,108\,864$	$2^{42} = 4\,398\,046\,511\,104$	$2^{58} = 288\,230\,376\,151\,711\,744$
$2^{11} = 2\,048$	$2^{27} = 134\,217\,728$	$2^{43} = 8\,796\,093\,022\,208$	$2^{59} = 576\,460\,752\,303\,423\,488$
$2^{12} = 4\,096$	$2^{28} = 268\,435\,456$	$2^{44} = 17\,592\,186\,044\,416$	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
$2^{13} = 8\,192$	$2^{29} = 536\,870\,912$	$2^{45} = 35\,184\,372\,088\,832$	$2^{61} = 2\,305\,843\,009\,213\,693\,952$
$2^{14} = 16\,384$	$2^{30} = 1\,073\,741\,824$	$2^{46} = 70\,368\,744\,177\,664$	$2^{62} = 4\,611\,686\,018\,427\,387\,904$
$2^{15} = 32\,768$	$2^{31} = 2\,147\,483\,648$	$2^{47} = 140\,737\,488\,355\,328$	$2^{63} = 9\,223\,372\,036\,854\,775\,808$
			$2^{64} = 18\,446\,744\,073\,709\,551\,616$

Attention, en fonction du contexte, les lettres k, M, G, T peuvent avoir une signification différente : une taille mémoire de 1MB (un mégaoctet) représente typiquement  $2^{20}$  octets, alors qu'une fréquence de 1MHz (un mégahertz) représente  $10^6$  Hertz.

Pour éviter la confusion, il est possible d'utiliser les *préfixes binaires* ki, Mi, Gi, Ti qui sont non-ambigus. Par exemple, 1 TiB (un tébioctet) représentera toujours  $2^{40}$  octets, c'est à dire 1024 GiB (gibiocets).