

EXAMEN FINAL

Instructions : – Une feuille aide-mémoire recto verso manuscrite est permise ;
 – Durée de l'examen : 1 h 50.

Pondération : Cet examen compte pour 25% de la note finale.

Aide-mémoire

Combinatoire

- Factorielle de n : $n! = 1 \times 2 \times 3 \times \dots \times n$ ($0! = 1$)
- Choisir k parmi n , selon un certain ordre (arrangement) : $A_n^k = \frac{n!}{(n-k)!}$
- Choisir k parmi n , sans ordre (combinaison) : $\binom{n}{k} = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!}$

Loi exponentielle

- Densité de probabilité : $p(t) = \lambda \exp(-\lambda t)$, pour $t \geq 0$
- Loi de répartition : $P(X \leq t) = 1 - \exp(-\lambda t)$, pour $t \geq 0$
- Espérance : $\mathbb{E}(t) = \frac{1}{\lambda}$
- Variance : $\text{Var}(t) = \frac{1}{\lambda^2}$

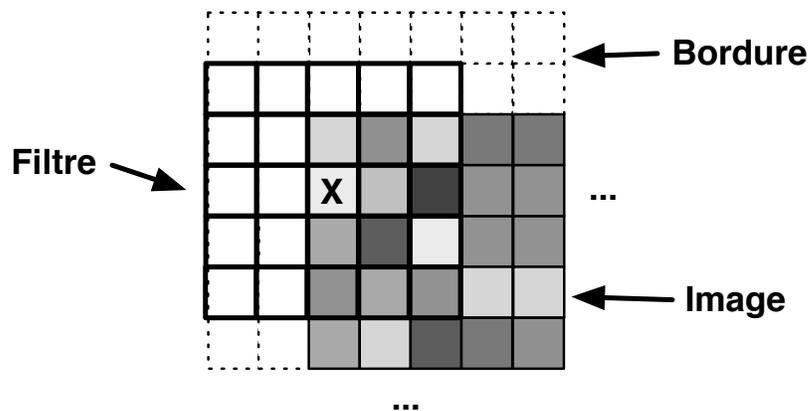
Théorie des files d'attente

Modèle	$M/M/1$	$M/M/m$	$M/M/\infty$	$M/M/1/K$
Utilisation (ρ)	$\frac{\lambda}{\mu}$	$\frac{\lambda}{m\mu}$	$\frac{\lambda}{\mu}$	$\frac{\lambda}{\mu}$
Probabilité d'aucune tâche (π_0)	$1 - \frac{\lambda}{\mu} = 1 - \rho$	$\left[1 + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho}\right]^{-1}$	$\exp(-\rho)$	$\left[1 + \sum_{n=1}^K (\lambda/\mu)^n\right]^{-1} = \frac{1-\rho}{1-\rho^{K+1}}$
Probabilité de n tâches (π_n)	$(1-\rho)\rho^n$	$\pi_0 \frac{(m\rho)^n}{n!}$ pour $1 \leq n \leq m$ $\pi_0 \frac{m^m}{m!} \rho^n$ pour $n > m$	$\exp(-\rho) \frac{\rho^n}{n!}$	$\frac{1-\rho}{1-\rho^{K+1}} \rho^n$ pour $0 \leq n \leq K$ 0 pour $n > K$
Taille moyenne de la file ($\mathbb{E}(X)$)	$\frac{\rho}{1-\rho}$	$\sum_{n=0}^{\infty} n\pi_n = m\rho + \frac{(m\rho)^m}{m!} \frac{\rho}{(1-\rho)^2} \pi_0$	ρ	$\frac{\rho}{1-\rho^{K+1}} \left[\frac{1-\rho^K}{1-\rho} - K\rho^K \right]$
Temps système moyen ($\mathbb{E}(S)$)	$\frac{1/\mu}{1-\rho}$	$\frac{1}{\mu} + \frac{1}{\mu} \frac{(m\rho)^m}{m!} \frac{\pi_0}{m(1-\rho)^2}$	$\frac{1}{\mu}$	$\frac{\mathbb{E}(X)}{\lambda(1-\pi_K)}$

Question 1 (20 points sur 100)

Supposons que vous travaillez à la conception d'un système de conduite de véhicules autonomes. Le système doit effectuer un traitement en temps réel, en traitant l'information reçue par les différents capteurs du véhicule et prenant des décisions de conduite à intervalle régulier.

- (8) (a) Supposons que le système peut effectuer une certaine charge de travail sur un seul processeur en 250 ms. 75 % de ce traitement peut être parfaitement parallélisé, les éléments restants du traitement devant être exécutés sur un seul processeur. Déterminez le nombre de processeurs que l'on doit prévoir pour que le système puisse effectuer un traitement en temps réel pour chaque charge de travail en 100 ms ou moins.
- (12) (b) Supposons maintenant que l'on doit traiter sur un GPU des images en tons de gris de 256×256 pixels provenant d'une caméra de la voiture. Le traitement consiste en la convolution d'un filtre de 5×5 , qui s'applique à chaque pixel de l'image. L'application du filtre sur des pixels en bordure de l'image implique de définir une bordure de deux pixels de large de chaque côté de l'image tel qu'illustré ici-bas, où le filtre est appliqué au pixel de la deuxième ligne et première colonne de l'image.



On veut paralléliser les opérations en découpant l'image en sous-images de plus petite taille, qui seront traitées en parallèle sur les unités de traitement du GPU. Pour chaque image, une bordure de deux pixels de large des quatre côtés de l'image doit être utilisée, en utilisant les pixels voisins de la sous-image lorsque la bordure de celle-ci ne correspond pas à la bordure de l'image. L'application du filtre sur un pixel implique 25 multiplications, 24 additions et une écriture, donc 50 opérations au total. Le coût du chargement d'une sous-image dans la mémoire d'une unité de traitement correspond à une opération par pixel. On doit inclure dans ce calcul le coût de chargement des pixels correspondant aux bordures supplémentaires et aux pixels pertinents des sous-images voisines. On suppose que l'opération de chargement en mémoire des unités de traitement des sous-images doit être effectuée **séquentiellement** sur le GPU.

Déterminez l'équation générale correspondant au temps de traitement parallèle d'un tel modèle, en nombre d'opérations.

Déterminez également la configuration la plus efficace en termes de temps de traitement parallèle selon les différentes configurations suivantes :

- 16 unités de traitement traitant des sous-images de 64×64 pixels ;
- 64 unités de traitement traitant des sous-images de 32×32 pixels ;
- 256 unités de traitement traitant des sous-images de 16×16 pixels ;
- 1024 unités de traitement traitant des sous-images de 8×8 pixels.

Justifiez votre réponse avec calculs.

Question 2 (20 points sur 100)

Supposons maintenant que le véhicule autonome inclut un système de vision numérique capable de faire le suivi des déplacements d'entités d'intérêt. Ces entités se situent dans l'environnement immédiat autour de la voiture, par exemple d'autres véhicules, des piétons ou des cyclistes. Le système doit faire le suivi de chaque entité d'intérêt visible par les capteurs de la voiture. Une analyse du nombre d'entités à suivre a été effectuée par un modèle de files d'attente, où il a été déterminé qu'une nouvelle entité d'intérêt devient visible aux capteurs en moyenne à chaque 10 s et reste visible pour le suivi pendant 4 s, en moyenne. Cette modélisation fait l'hypothèse que les arrivées et les départs suivent des processus aléatoires de Poisson, sans mémoire. Le système est conçu pour pouvoir suivre simultanément jusqu'à quatre entités d'intérêt.

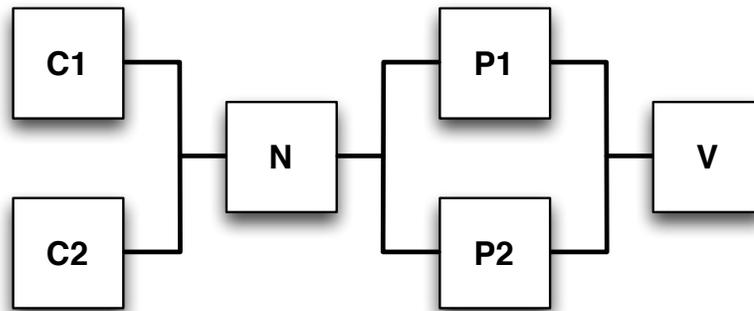
- (14) (a) Calculez la probabilité instantanée, en régime permanent, que le nombre d'entités pour lesquelles le suivi doit être effectué dépasse la limite du système, soit de pouvoir suivre au maximum quatre entités simultanément.
- (6) (b) Est-ce que les hypothèses faites pour la modélisation par file d'attente des entités d'intérêt devant être suivies apparaissent raisonnables en pratique, pour ce problème de suivi ? Justifiez de façon convaincante votre réponse.

Question 3 (20 points sur 100)

Supposons maintenant que l'on s'intéresse à la fiabilité de composantes du véhicule autonome.

- (10) (a) Supposons que la voiture comporte un dispositif de trois capteurs permettant de mesurer la distance d'objets externes. On suppose que la fiabilité individuelle des capteurs de la matrice est identique pour tous les capteurs, soit une durée de vie suivant une loi exponentielle avec un temps espéré (moyen) de fonctionnement de 100 000 heures. De plus, on fait l'hypothèse que les défaillances des capteurs sont indépendantes et que les mesures de distance sont utilisables tant qu'au moins deux capteurs sont fonctionnels. Donnez la probabilité de défaillance du dispositif ne soit plus en mesure de retourner des mesures de distance utilisables au temps t .

- (10) (b) Supposons maintenant la modélisation suivante d'un sous-système de la voiture.



Voici les caractéristiques des composants de ce sous-système :

- Les composants C1 et C2 sont des capteurs fonctionnant en parallèle – le système fonctionne tant qu’au moins un capteur fonctionne – dont la fiabilité individuelle est R_C ;
- Le composant N désigne l’interconnexion entre les capteurs C1 et C2 et les processeurs P1 et P2, dont la fiabilité est R_N ;
- Les composants P1 et P2 correspondent à deux processeurs redondants, dont la fiabilité individuelle est R_P ;
- Le composant V correspond au contrôle du véhicule selon les traitements faits sur les processeurs P1 et P2, dont la fiabilité est R_V .

Déterminez la fiabilité globale du sous-système modélisé ainsi.

Question 4 (40 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

- (4) (a) Expliquez la différence entre la moitié supérieure (*top-half*) et la moitié inférieure (*bottom-half*) d’un pilote dans le noyau Linux.
- (4) (b) Présentez la différence entre la classe des dispositifs de caractères (*character devices*) et la classe des dispositifs de blocs (*block devices*) des périphériques du noyau Linux, ainsi que des exemples de dispositifs de ces deux classes.
- (4) (c) Indiquez où peut-on récupérer les messages écrits via l’appel à la fonction `printk` dans le noyau Linux.
- (4) (d) Expliquez pourquoi les systèmes temps réel et la communication sur réseaux TCP/IP ne font pas bon ménage.
- (4) (e) Expliquez à quoi sert le protocole NTP (*Network Time Protocol*).
- (4) (f) Expliquez la différence entre le délai (latence) et la variabilité (*jitter*) dans un contexte de communication avec qualité de service (*QoS*).

- (4) (g) Dans la notation $A/B/m/K$ des files d'attente, indiquez à quoi correspond le K .
- (4) (h) Indiquez le principal avantage pour le programmeur de travailler avec un système multi-processeur à **mémoire partagée**.
- (4) (i) Dans le contexte du problème des généraux byzantins, expliquez en vos mots pourquoi faut-il au moins quatre généraux pour que l'on puisse fonctionner correctement, malgré la présence d'un général traître.
- (4) (j) Donnez les différences entre l'utilisation des fonctions `shm_open` et `mkfifo` pour la communication entre processus d'une application temps réels.