****

**INF3610 –Systèmes embarqués**

**Automne 2021**

**TP No. 1**

**Groupe 2**

**XXXXXXX – Guillaume Rochon-Vear**

**2028284 – Florence Cloutier**

**Soumis à : M. Guy Bois**

**22 septembre 2021**

# Question 1

Expliquez la structure de la fonction TaskForwarding. Pourquoi un if then else et pas un switch ?

On veut aller chercher les paquets dans l’ordre High – Medium – Low priorités des files. Le premier if est fait sur le message d’erreur retourné en essayant d’aller chercher dans la file de haute priorité. S’il y a une erreur, on va chercher dans la file de priorité moyenne. Finalement, s’il y a encore une erreur, on va chercher dans la file de priorité basse. Un switch est utilisé quand une variable prend plusieurs valeurs différentes. Dans ce cas-ci, la condition des if reste toujours pareille, car on vérifie seulement s’il y a une erreur ou non.

Expliquez. Également, pourquoi l’option OS\_OPT\_PEND\_NON\_BLOCKING dans OSQPend ?

L’option OS\_OPT\_PEND\_NON\_BLOCKING est utilisée dans OSQPend pour éviter d’attendre à l’infini dans la file de haute priorité pour un paquet, tandis qu’il y a peut-être des paquets dans les files moins prioritaires. Si cette option n’était pas utilisée, on n’attendrait que des paquets dans la file de priorité élevée.

# Question 2

Justifiez le choix des priorités de chaque tâche (lignes 24 à 28 de routeur.h). Auriez-vous fait différemment ? Au besoin, proposez des alternatives (que vous pouvez rapidement valider).

TaskStats a la plus haute priorité parce qu’on veut qu’il affiche les statistiques à toutes les 30 secondes, sans avoir à attendre après une autre tâche.

TaskGenerate est le deuxième plus prioritaire, car c’est cette tâche qui produit les paquets, qui sont ensuite consommés par TaskComputing. Si TaskGenerate ne s’exécute pas avant TaskComputing, TaskComputing n’aura pas de paquets à «compute ». De plus, dans TaskGenerate, il y a un boolean isGenPhase qui permet de sortir de cette tâche quand il atteint un maximum de 255 paquets. Donc, même si cette tâche est prioritaire aux subséquentes, elle laissera éventuellement sa place aux tâches moins prioritaires, si elle atteint 255 paquets générés. Ce qui évite la famine dans les tâches moins prioritaires.

TaskOutputPort est le 3e plus prioritaire, car …

TaskComputing est le 4e plus prioritarie, car …

TaskForwarding est le moins prioritaire, car …

Nous aurions mis en ordre du plus prioritaire au moins prioritaire :

TaskStats > TaskOutputPort > TaskForwarding > TaskComputing > TaskGenerate

Avec cet ordre de priorité, on évite la famine, sans avoir à utiliser de booléen isGenPhase dans TaskGenerate.

TaskStats reste le plus prioritaire, car on souhaite qu’il s’exécute tous les 30 secondes sans avoir à attendre après une autre tâche. Ensuite, si TaskOutputPort, la tâche la plus prioritaire après TaskStats, n’a rien à output, la prochaine tâche la plus prioritaire est TaskForwarding. Si TaskForwarding n’a rien à forward, la prochaine tâche la plus prioritaire est TaskComputing. Si TaskComputing n’a aucun paquet à “compute”, alors TaskGenerate pourra s’exécuter. Cette ordre de priorité permet de print les statistiques tous les 30 secondes et évite aussi la famine de certaines tâches.

# Question 3

Présentez vos résultats (4 captures d’écran) d’analyse de performance réalisés plus haut et indiquez s’il y a une différence entre 100Hz et 1000Hz. Dans un cas comme dans l’autre, justifiez.

Q: On a genre 200 paquets rejetés, mais seulement 50 qui sont print et des id se répètent.

# Question 4

Expliquez la nécessité de faire un passage de paramètres lors de la création des tâches TaskOutputPortTCB (lignes 591 à 594). Et pourquoi ici utiliser un tableau de tâches ?