STR – Q08

Tel que discuté récemment, la métaphore de communication que préconise QNX avec les fonctions MsgSend(), MsgReceive() et MsgReply() a les particularités suivantes :

- une invocation de MsgSend() bloquera l'émetteur jusqu'à ce que le récepteur ait fait un MsgReply() sur le message envoyé;
- si un destinataire est en attente d'un message sur le canal où a été réalisé un MsgSend(), alors ce destinataire obtiendra le processeur dès que l'émetteur (l'appelant de MsgSend()) aura été suspendu;
- une invocation de MsgReceive() bloquera l'appelant jusqu'à ce que celui-ci soit l'objet d'un envoi de message.

Imaginez un système multiprogrammé où un processus P_0 lit des données sur un port et où un processus P_1 doit réagir lors de la réception de ces données. Remarquez au passage la correspondance entre les 2^e et 3^e paramètres de MsgSend() et les 2^e et 3^e paramètres de MsgReceive(), de même qu'entre les 4^e et 5^e paramètres de MsgSend() et les 3^e et 4^e paramètres de MsgReply(). Le code suit, et s'exécute correctement.

Processus P_0

```
void lire_et_envoyer
   (int canal, unsigned int* port) {
   bool ok = false;
   do {
      short data =  in16(port);
      MsgSend(canal, &data, sizeof data,
              &ok, sizeof ok);
   } while(ok);
}
```

Processus P_1

```
void utiliser(short);
void traiter(int canal) {
   bool ok;
   do {
      short data;
      int rcvid = MsgReceive
         (canal, &data, sizeof data, 0);
      ok = (data & 0x8000) != 0;
      MsgReply(rcvid, canal, &ok, sizeof ok);
      utiliser(data);
   } while(ok);
}
```

La lecture[1] sur le port nommé port avec in16() consomme un entier signé sur 16 bits sur le port en question. L'ordre des *bytes* dans cet entier est indigène (pas de conversion requise suite à la lecture). Considérez que P_0 et P_1 ont tous deux une très haute priorité (les autres processus de l'ordinateur n'interféreront pas avec leur propre capacité d'action).

Les questions apparaissent à la page suivante. **Répondez à cinq des sept questions ci-dessous**, pour un point chacune. *Si vous répondez à plus de cinq questions, les cinq moins bien répondues seulement seront retenues*.

[1] Nous ferons comme si in16() était bloquante, pour simplifier l'illustration; en pratique, il s'agit d'un mécanisme de bien plus bas niveau.

Nom : _____ (_____ /5)

Q08.0 – Le processus P_0 peut-il garantir la brièveté de sa réaction à l'arrivée de données sur le port? Pourquoi? _____

_____

_____

_____

Q08.1 – Le processus P_0 peut-il garantir l'immédiateté de sa réaction à l'arrivée de données sur le port? Pourquoi? _____

_____

_____

_____

Q08.2 – La répétitive du processus P_0 itère-t-elle à rythme régulier? Pourquoi? _____

_____

_____

_____

Q08.3 – À quelle(s) condition(s) P_0 pourrait-il traiter un influx constant de données sur le port? Pourquoi? _____

_____

_____

_____

Q08.4 – Le processus P_1 peut-il garantir la brièveté de sa réaction à l'arrivée de données sur le canal? Pourquoi? _____

_____

_____

_____

Q08.5 – Le processus P_1 peut-il garantir l'immédiateté de sa réaction à l'arrivée de données sur le canal? Pourquoi? _____

_____

_____

_____

Q08.6 – La répétitive du processus P_1 itère-t-elle à rythme régulier? Pourquoi? _____

_____

_____

_____