Question n°3

Il existe deux méthodes pour gérer les accès concurrentiels sur cet objet.

La méthode UDP, utilise un serveur UDP qui possède une adresse IP. Ce serveur envoie une requête via la passerelle qui est destinée au client.  
Lorsque le client veut interroger le serveur, la passerelle ne sait pas où renvoyer cette nouvelle requête qui est destinée au serveur UDP.  
  
Cette méthode n'est donc pas appropriée car la requête ne parvient pas au destinateur.  
  
La méthode TCP  
Met en relation un client est un serveur qui fait fonctionner les capteurs,  
Le client peut interroger le serveur sur les différents capteurs situés dans les différents processus.  
  
Multicast: le client interroge un à un les capteurs et a la possibilité de lecture  
  
Multithreading: lance simultanément plusieurs processus de traitement.

3 options s'offrent donc à nous:  
  
1) sémaphore IPC

-----???????

2) sémaphore POSIX  
→ accès multicast

--------????????  
3) serveur queue IPC

------????????  
Message Queue  
pas de sémaphore.

Question n°4

Le serveur étant le soubassement de notre projet, nous avons décidé de mettre en place un protocole d’interrogation afin de l'exploiter au mieux notre.

Pourquoi les accès concurrentiels ?

La création d’un protocole serveur est nécessaire pour notre objet.

En effet, il est impératif de prévoir un certain nombre de fonctionnalités pour la connexion d’un client sur le serveur.

À la connexion, le client aura la possibilité de récupérer des informations (CPU, mémoire, flux entrants et sortants sur les connecteur virtuels Ethernet) :

* Lire les taux d’occupation
* Modifier les taux d’occupation

La modification de l’objet ne peut pas être fait en même temps par deux clients.

Pour gérer cela on utilise un Mutex (exclusion mutuelle).

Sachant que nous avons plusieurs manières de mettre en place notre système d’exclusion mutuelle:

* Le sémaphore
* message queue (la file d’attente de message) IPC
* ??????????????????