

Rapport Projet-Système

Azibi Mounia – Oudin Victor

- Choix de la structure de queue :

```
struct queue
{
    unsigned int size_buf; // size of queue
    int nbr_msg;           // number of messages in queue
    long occupied;        // occupied space in queue
    size_t * msg_size;     // table of size of messages in queue
    void * buf;            // queue

    pthread_mutex_t mutex;
    pthread_cond_t full;
    pthread_cond_t empty;
};
```

« size_buf » représente la taille de la queue.

« nbr_msg » représente le nombre de message dans la queue.

« occupied » représente l'espace occupé dans la queue.

« msg_size » est un tableau de la taille de chaque message dans la queue.

« buf » est la queue sous forme de tableau.

« mutex » est la primitive de synchronisation, « full » et « empty » sont les variables de condition.

- queue_new : `struct queue *queue_new(size_t size){`

On crée une structure dont on alloue l'espace mémoire, on initialise toutes les variables, on alloue l'espace mémoire pour les deux tableaux et on initialise le mutex et les conditions, et on renvoie la structure créée.

- queue_destroy : `void queue_destroy(struct queue *queue){`

On détruit le mutex et les conditions, on libère l'espace mémoire des deux tableaux et enfin on libère l'espace mémoire de la structure.

- enqueue et enqueue_multiple :

Dans les deux « enqueue » on commence par verrouiller le mutex, puis on teste la variable du mode bloquant, ensuite on teste la taille restante dans le tableau de la queue, puis on enfile le buffer. La seule différence entre le simple et le multiple est la boucle « while » autour du « pthread_cond_wait » pour le multiple pour enfile le premier message du vecteur.

- dequeue et dequeue_multiple :

Dans les deux « dequeue » on commence par verrouiller le mutex, puis on teste la taille du buffer d'arrivée, après on teste que la file ne soit pas vide, puis on défile la queue. La seule différence entre le simple et le multiple est la boucle « while » autour du « pthread_cond_wait » pour le multiple pour

défiler le premier message de la queue.

- Extention : Allocation dynamique

« enqueue_multiple_dynamic » fonctionne comme « enqueue_multiple » mais au lieu de bloquer quand l'espace mémoire est insuffisant, il ré-alloue la double d'espace mémoire.