

TP n°6

Les files de messages

Le but de ce TP est de manipuler les files de messages System V.

1 Premier pas avec les files de messages

Nous souhaitons écrire un programme qui permet d'envoyer ou de lire une chaîne de caractères dans ou depuis une file de messages, ou de supprimer une file de messages. Le programme prend en argument la clé de la file de messages, un ordre (-R pour réception, -E pour envoi et -S pour supprimer la file), puis la chaîne de caractères (seulement pour l'envoi). Pour l'envoi, la chaîne de caractères est envoyée dans la file de messages. Pour la réception, le programme se contente d'afficher à l'écran la chaîne de caractères contenue dans le premier message de la file de messages. Pour la suppression, le programme supprime simplement la file de messages.



Le programme doit impérativement prendre en arguments les différentes valeurs.

Questions

1. Nous supposons que les messages envoyés dans la file de messages contiennent seulement une chaîne de caractères dont la taille maximale est fixée en constante pour l'application. Proposez une structure en C.
2. Écrivez le programme et faites les différents tests.
3. Nous souhaitons modifier maintenant le programme pour coupler le message texte à un identifiant. Le programme ne lit que les messages correspondant à cet identifiant. Maintenant, le programme prend la clé de la file de messages, l'identifiant, l'ordre et la chaîne de caractère (pour l'envoi). Modifiez la structure et le programme.

2 Application multi-clients

Nous souhaitons écrire une application (que nous appellerons *trieur*) permettant de mettre en relation différents clients via une unique file de messages. Pour pouvoir envoyer ou recevoir des messages, les clients doivent s'enregistrer auprès du *trieur*. Lorsqu'un client envoie un message dans la file, le message est lu par le trieur qui vérifie que le client de destination existe. Le message est ensuite placé dans la file pour le client de destination. Lorsqu'un client se termine, il prévient le trieur via un autre message. Nous distinguons donc trois types de messages : le premier permettant d'enregistrer un client ou d'avertir de son arrêt, le second est un message à envoyer par un client au trieur et le troisième est un message envoyé par le trieur pour un client.



Cet exercice est préparatoire au projet n°2.

Questions

1. Représentez tous les échanges possibles dans l'application.
2. Comment est-il possible d'envoyer un message pour une application spécifique dans une file de messages ? Proposez une solution pour cet exercice.
3. Proposez une structure pour les différents messages.
4. Écrivez le trieur (il enregistrera les clients dans un simple tableau).
5. Un client est un programme qui demande à l'utilisateur s'il souhaite envoyer un message, recevoir un message ou s'arrêter. Pour l'envoi, le programme demande de taper un texte puis un client de destination (en fonction de votre réponse à la question 2.).
6. Faites en sorte que le trieur gère l'arrêt via un CTRL+C. Qu'est-ce que cela change dans vos applications ?