Programmation Système

Travaux Pratiques (7), Licence 2 Informatique Les IPC SYSTEM V - Les files de messages

Les exercices suivants ont pour but de vous familiariser avec les IPC (Inter-Process Communication) SYSTEM V. Cette partie se consacre aux files de messages.

Il vous est recommandé de consulter les pages man des primitives relatives aux IPC pour de plus amples informations sur leur syntaxe, leur sémantique et les éventuelles options qu'elles offrent.

Les instructions des exercices se repèrent par des icônes, qui sont les suivantes :

i	Information	Information concernant l'usage ou le rôle d'une commande, par exemple. Dans certains cas, il s'agit d'une information sur ce que vous êtes en train de faire ou sur ce qui se passe.
		, &

	Exemple	Exemple d'utilisation.
--	---------	------------------------

Contrôle	Vérifier le résultat d'une (ou plusieurs) action(s
Controle	vermer le resultat d'une (ou plusieurs) action(

Action	7700
Action	Effectuer la ou les action(s) décrite(s).

Question	Questions auxquelles vous devez répondre.
	1 1

De plus, un texte en police courier correspond soit à une sortie écran soit à des noms spécifiques (menus, fenêtre, icône, processus, commandes...).

Un **texte en police times gras** correspond à ce que l'utilisateur doit introduire comme valeur de paramètre, ou encore, est utilisé pour attirer l'attention de l'utilisateur.

Création d'une file de messages







La file créée existe-t-elle encore après la terminaison du programme ? Qu'en déduisez-vous ?

Recherche de l'identifiant d'une file de messages et envoi d'un message

	Modifier	le	programme	précédent	de	façon	qu'il	recherche	l'identifiant	de	la	file
précédemment créée et qu'il y envoie un message.												

Quel problème rencontrez-vous ? Expliquer !

Dans quel cas peut-on utiliser une clé privée ? Expliquer !

Modifier le premier programme de façon qu'il crée une file de message avec une clé que vous aurez construit (utiliser la fonction ftok).

S'assurer que la file est bien créée.

Modifier le programme précédent de façon qu'il recherche l'identifiant de la file précédemment créée et qu'il y envoie un message.

S'assurer qu'une nouvelle file n'a pas été créée.

Recherche de l'identifiant d'une file de messages et extraction d'un message

Écrire un nouveau programme qui recherche l'identifiant de la file précédemment créée et qui y lise le message contenu en tête de file.

Est-ce le même message que celui émis par le précédent programme ? Qu'en déduisez-vous ?

Opérations de contrôle sur les files de messages

Extraction et modification des caractéristiques d'une file de messages

La primitive int msgctl(int msg_id, int cmd, msqid_ds *buf); permet l'extraction des caractéristiques d'une file de messages.

Écrire un programme qui affiches certaines caractéristiques (créateur, propriétaire, ...) d'une file existante.

La primitive précédente, permet également de modifier certaines caractéristiques d'une file existante.

Modifier le programme précédent de façon à ce qu'il modifie le propriétaire d'une file déjà existante.

Suppression d'une file de messages



La primitive int msgctl(int msg_id, int cmd, msqid_ds *buf); permet enfin la suppression d'une file de messages existante.



Modifier le programme précédent de façon qu'il supprime la file de messages une fois terminé.

Communication et synchronisation entre processus



Nous donnons l'exemple de communication entre deux processus, un écrivain et un lecteur.

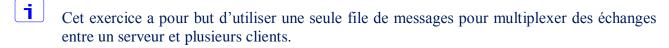
L'écrivain a le comportement suivant :

- il appelle la primitive ftok, pour composer une clé unique, en spécifiant un path="filemsg" (fichier existant et accessible) et un identificateur code='Q',
- il crée une file de messages avec la clé rendue par ftok,
- il lit des chaînes de caractères, à partir de l'entrée standard, qu'il envoie dans la file comme étant des messages de type 1,
- sur lecture d'une chaîne commençant par un point (qui traduit une fin de fichier), il envoie un message de type 2 dans la file pour notifier au lecteur que la saisie est terminée, et que lui même peut procéder à l'extraction des messages,
- il se met en attente d'arrivée du message de type 3, qui traduira que le lecteur a extrait tous les messages, et il détruit la file.

Le lecteur a le comportement suivant :

- il appelle la primitive ftok, pour composer une clé unique, en spécifiant un path="filemsg" (fichier existant et accessible) et un identificateur code='Q',
- il recherche l'identifiant de la file associée à la clé rendue par ftok,
- il se met en attente d'arrivée du message de type 2,
- il consulte la structure associée à la file, et prend connaissance du nombre de messages présents,
- il extrait tous les messages,
- il envoie un message de type 3 à l'écrivain pour lui notifier qu'il a terminé.

Multiplexage dans une file de messages



Écrire deux programmes, un serveur et un client. Le processus serveur reçoit du client, à travers une file de messages, des requêtes auxquelles il répond par l'envoie de données (par exemple, des nombres aléatoires). Considérer le cas d'un processus serveur et de plusieurs processus client (par exemple 5). Pour que le serveur puisse distinguer les clients, ces derniers envoie leur pid avec la requête. La valeur de leur pid est utilisée par le serveur comme type dans le message de réponse. Un client ne lit donc de la file que les message de type égal à son pid. Les requêtes au serveur envoyées par les clients sont de type égal à 1.

Commandes shell relatives aux IPC SYSTEM V

i	La commande ipcs	permet d'afficher	les ressources IF	C existantes.
	La commande ipos	permet a arrient	100 10000 011 000 11	C CHIBCAILCON.

- Utiliser la commande ipcs pour prendre connaissances des différentes files de messages existantes. Essayer quelques options permises.
- La commande iperm permet de supprimer les ressources IPC existantes.
- Utiliser la commande iperm pour supprimer des files de messages existantes. Essayer quelques options permises.