

Controllo su una coda di messaggi

domenica 7 maggio 2023 11:17

Ora vogliamo eliminare una message queues ed eliminare anche i messaggi che sono correntemente all'interno di questo deposito. Questo è fondamentale quando vogliamo permettere al kernel di fare "Garbage collection" e di recuperare memoria. Per far questo possiamo utilizzare l'API message control.

Controllo su una coda di messaggi

```
int msgctl(int ds_coda, int cmd, struct msqid_ds *buff)
```

Descrizione invoca l'esecuzione di un comando su una coda di messaggi

Parametri

- 1) ds_coda: descrittore della coda su cui si vuole operare
- 2) cmd: specifica del comando da eseguire (IPC_RMID, IPC_STAT, IPC_SET)
- 3) buff: puntatore al buffer con eventuali parametri per il comando

Descrizione -1 in caso di fallimento

Quando vogliamo eseguire un comando sulla coda di messaggi dobbiamo passare il "descrittore" della coda come primo parametro (le operazioni sulle code di messaggi si fanno sul descrittore) quindi NON la chiave, perché la chiave è solo un nome che ci serve per poter eventualmente permettere ad altri processi di aprire la stessa coda, utilizzando message get con lo stesso nome e quindi con la stessa chiave. Come secondo parametro dobbiamo specificare il comando che vogliamo eseguire su quella message queue. Uno dei comandi possibili è associato alla macro IPC_RMID, ossia il comando di RIMOZIONE DELLA MESSAGE QUEUES.

IPC_RMID invoca la rimozione della coda dal sistema

Abbiamo anche la possibilità di richiamare la richiesta di informazioni con IPC_STAT, quindi lo stato di questa message queues, oppure possiamo chiamare una macro in cui vogliamo settare informazioni con IPC_SET riguardanti questa message queues. L'informazione che noi possiamo sicuramente richiedere o cambiare nella gestione di questa message queues, sono i suoi permessi di accesso. Percui ad un certo punto, data una message queues con certi permessi di accesso A, vogliamo portare i permessi d'accesso ad essere B, e lo facciamo eseguendo un settaggio di informazioni con IPC_SET, per la message queues con quel descrittore, e le informazioni che vogliamo settare le scriviamo all'interno di un'area di memoria (terzo parametro) che è una struttura dati di cui passiamo il puntatore. Quindi andiamo in memoria a scrivere, all'interno dei campi di quella struttura, tutte le informazioni che noi vogliamo che vengano settate per gestire quella coda. Questa struttura qui noi la passiamo tramite pointer ad un area di memoria che ospita questa struttura, anche quando chiediamo informazioni sullo stato attuale della message queues. Chiaramente possiamo avere un fallimento di questa funzione quando passiamo un descrittore di coda di messaggio, che non ci porta su una coda di messaggio.

N.B: Quando noi andiamo in rimozione, il terzo parametro è "don't care". Abbiamo un'irrilevanza di questo terzo parametro dovuto al fatto che non vengono considerati informazioni che noi abbiamo scritto da qualche parte in memoria e non ci viene nemmeno riconsegnato nulla, viene rimossa la coda ci viene indicato tramite il valore di ritorno e l'operazione è stata eseguita.