

Creazione di una coda di messaggi

```
int msgget(key_t key, int flag)
```

Descrizione invoca la creazione una coda di messaggi

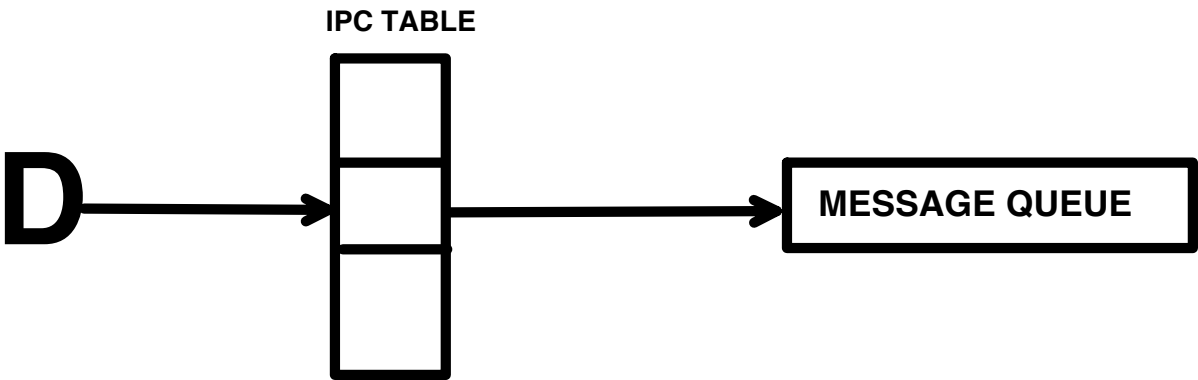
Parametri 1) key: chiave per identificare la coda di messaggi in maniera univoca nel sistema
2) flag: specifica della modalità di creazione (IPC_CREAT, IPC_EXCL, definiti negli header file sys/ipc.h e sys/msg.h) e dei permessi di accesso

Descrizione identificatore numerico per l'accesso alla coda in caso di successo (descrittore di coda), -1 in caso di fallimento

La coda di messaggi viene creata soltanto se noi la chiediamo. Per chiederlo abbiamo un servizio a livello della programmazione di sistema UNIX, che si chiama message get. Come facciamo ad identificare una coda precisa di messaggio? Un deposito già esiste che è gestito dal kernel e noi ne vogliamo creare un altro, dove si appoggeranno più flussi ovviamente. Quindi vogliamo utilizzare un altro deposito per consegnare messaggi al sistema UNIX, in modo tale che questi messaggi vadano verso qualche destinazione in termini di processo richiedente.



Queste code devono essere targate con un nome, e in sistemi UNIX il nome è una key_t, ossia una chiave, cioè un codice NUMERICO. Questa è la chiave del deposito dei messaggi, quindi quando creiamo un deposito dobbiamo specificare la chiave con la quale vogliamo creare questo deposito. Ovviamente quando creiamo uno di questi depositi dobbiamo specificare un flag che va a definire quale tipo di operazione noi stiamo facendo, in particolare possiamo indicare che stiamo eseguendo una creazione (IPC_CREAT). Ma non necessariamente quando invochiamo una creazione stiamo realmente creando un deposito con una certa chiave: perché quel deposito con quella specifica chiave che io sto cercando di utilizzare, potrebbe già esistere, in questo caso l'effetto è semplicemente quello di APRIRE il deposito, ovvero riuscire a poter lavorare esattamente sul deposito già esistente. Quindi se noi passiamo IPC_CREAT e non passiamo altri FLAG, in realtà o creiamo o apriamo un deposito di messaggi già esistente. Se passiamo anche IPC_EXCL vogliamo dire che l'operazione di creazione deve essere realmente eseguita per effetto di questa chiamata, quindi se utilizzo come primo parametro una chiave relativa ad un deposito già esistente, chiaramente avendo passato anche IPC_EXCL, ovviamente ricevo un ERRORE da questa API. Perciò non posso creare un deposito che già è ben definito ed esiste. Sempre in codifica ottale possiamo specificare i permessi di accesso. Se noi chiamiamo questa API e diciamo che vogliamo creare un deposito con una certa chiave, eventualmente specifichiamo i permessi per la gestione di questo oggetto, come facciamo a ritenere che questo oggetto sia aperto per noi - perché è stato creato - e noi possiamo veramente utilizzarlo? L'utilizzazione avviene tramite il valore di ritorno di questa API. Il valore di ritorno è il cosiddetto descrittore della coda, ossia il CODICE OPERATIVO che ci serve per andare a spedire o ricevere messaggi. Nel momento in cui creiamo questa message queues ci viene ritornato il descrittore D, ossia il codice che ci permette di arrivare a chiamare dei servizi verso questa message queues. Mentre il codice C che ci permette di arrivare sui file ci permette di indicizzare un elemento di una tabella, che ci porta sulla sessione, che poi ci porta sul file, il codice D che noi otteniamo non va assolutamente a lavorare su quella tabella di descrittori di file, LAVORA SU UN'ALTRA TABELLA, non necessariamente la tabella è implementata come un oggetto contiguo in memoria, e questo codice D ci porta su una entry di questa TABELLA, che è la entry che ci porta poi ad identificare la struttura dati su cui eventualmente noi potremmo voler poi lavorare chiamando delle operazioni di spedizione/ricezione.



Quindi noi utilizziamo D per chiamare una spedizione, tramite D il sistema va all'interno della tabella qui sopra, ed identifica in memoria tramite pointers, dove sono presenti le informazioni associate a questo deposito. Questa tabella ora prenderà un nome, e si chiamerà da oggi in poi tabella IPC(Inter process communication), ed è una tabella UNICA su sistemi unix, non è per processo: quando creiamo una message queues stiamo cambiando lo stato di questa tabella, quindi quando andiamo in una certa posizione di questa tabella ad occupare/riservare una entry e otteniamo il descrittore associato a quella entry, questo codice noi lo possiamo comunicare anche ad altri processi/thread, scrivendolo su un file e quando questi processi o thread leggeranno da quel file il codice numerico, possono tranquillamente (tramite quel valore) arrivare sulla entry della tabella IPC, e quindi arrivare anche su questa message queues.

NOTA

Il descrittore indicizza questa volta una struttura unica valida per qualsiasi processo

Per poter lavorare su una message queues che esiste già possiamo avere due possibilità:

1. La message queues esiste e quindi è registrata l'informazione che ci porta su questa queues all'interno della tabella IPC; Ma l'abbiamo creata noi. E ci viene ritornato D che noi utilizziamo per puntare - quando chiameremo i servizi - a quella specifica entry per indicizzare in particolare quella specifica entry della tabella IPC.
2. La message queues esiste già, e quindi noi cerchiamo di crearla con la stessa chiave, però viene aperta una entry già esistente e ci viene ritornato esattamente lo stesso codice D per accedere ai servizi, però ora siamo un altro processo P', mentre magari nella creazione del punto 1 eravamo P.
3. P' o P scrivano D su un file, e un ulteriore processo P'' legga questo D da file, lo riconosca e quindi possa utilizzare quel codice per andare a chiamare i servizi che ci portano a lavorare sulla message queue.