

# Memoria condivisa in sistemi UNIX

```
int shmget(key_t key, int size, int flag)
```

**Descrizione** invoca la creazione di un'area di memoria condivisibile

**Parametri**

- 1) key: chiave per identificare la memoria condivisibile in maniera univoca nel sistema (IPC\_PRIVATE e' un caso speciale)
- 2) size: taglia in byte della memoria condivisibile
- 3) flag: specifica della modalita' di creazione (IPC\_CREAT, IPC\_EXCL, definiti negli header file sys/ipc.h e sys/shm.h) e dei permessi di accesso

**Restituzione** identificatore numerico per la memoria conivisa in caso di successo (descrittore), -1 in caso di fallimento

Abbiamo una richiesta verso il sistema operativo, utilizzando sempre il meccanismo delle system call, di istanziazione della memoria condivisibile.

Questa System Call si chiama "Shared Memory Get" in cui noi passiamo una chiave come primo parametro, che non è altro che un codice numerico che identifica in maniera univoca lo storage effettivo dove eventualmente le informazioni da utilizzare sono materializzate, poi come secondo parametro chiediamo una certa taglia (numero di byte) per lo storage da condividere, e infine come terzo ed ultimo parametro abbiamo da specificare i soliti flags per la creazione di questo oggetto.

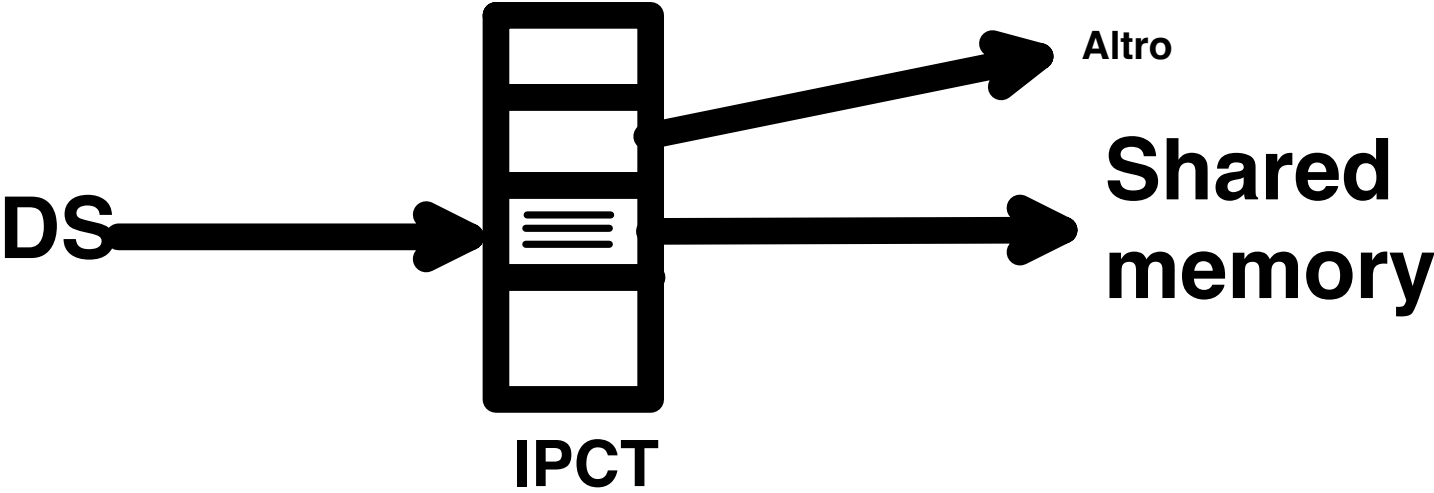
Noi possiamo creare l'oggetto con IPC\_CREAT, se questo esiste con questa stessa chiave KEY possiamo semplicemente utilizzare quello già esistente, oppure possiamo richiedere l'operazione in maniera esclusiva IPC\_EXECL.

Il valore di ritorno è l'identificatore numerico, denominato anche "descrittore" della *shared memory*.

Questo codice numerico che ci viene ritornato viene utilizzato per poi chiamare altre system call. Quando noi dovremo realmente andare ad utilizzare questa memoria che è stata creata e resa condivisibile.

Questo descrittore indicizza una STRUTTURA UNICA valida per qualsiasi processo, la Inter Process Communication Table (IPCT).

All'interno di un sistema operativo unix esiste questa IPCTable e questa tabella ha un elemento che ci permette di arrivare su un oggetto di inter process communication e chiaramente uno shared memory è un oggetto di inter process communication, perché noi potremmo eseguire tramite questa shared memory la comunicazione fra processi anche non relazionati fra di loro.



Il descrittore (DS) che ci viene riportato non è altro che l'indice di questa entry all'interno della tabella IPCT.

Tramite questo descrittore noi indicizziamo questa tabella unica all'interno del sistema per arrivare a potere lavorare su questa shared memory. Lo stesso concetto lo avevamo visto per quanto riguardava il message queue. Questo descrittore ha una validità universale all'interno del sistema: per cui se noi ci facciamo restituire un descrittore, lo scriviamo poi su un file F, e un altro processo legge da questo file F un altro descrittore, arriveremo entrambi all'interno della stessa shared memory.

Nel momento in cui abbiamo creato questa shared memory essa esiste, ma non è ancora raggiungibile, con questa system call la stiamo solo istanziando.

Chiaramente quando non ci serve più vogliamo rimuovere questa shared memory per permettere al S.O di fare Garbage Collection e di fare riuso della memoria.

```
int shmctl(int ds_shm, int cmd, struct shmid_ds *buff)
```

**Descrizione** invoca l'esecuzione di un comando su una shared memory

**Parametri**

- 1) ds\_shm: descrittore della memoria condivisa 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

**Restituzione** -1 in caso di fallimento

Il secondo parametro è il comando da eseguire. E tra i comandi abbiamo proprio quello di rimozione che è una MACRO IPC\_RMID.

Abbiamo anche la possibilità di richiedere informazioni sullo stato corrente con IPC\_STAT di questa shared memory (permessi di accesso correnti), oppure settare alcuni parametri sullo stato di gestione come ad esempio i permessi di accesso stessi.

Il terzo parametro è un pointer ad una struttura dati, e questa struttura è così composta:

```
struct shmid_ds {
    struct ipc_perm shm_perm; /* operation perms */
    ...
}
```

```
int shm_segsz; /* size of segment (bytes) */
time_t shm_atime; /* last attach time */
time_t shm_dtime; /* last detach time */
time_t shm_ctime; /* last change time */
unsigned short shm_cpid; /* pid of creator */
unsigned short shm_lpid; /* pid of last operator */
short shm_nattch; /* no. of current attaches */
};

struct ipc_perm {
    key_t key;
    ushort uid; /* owner euid and egid */
    ushort gid;
    ushort cuid; /* creator euid and egid */
    ushort cgid;
    ushort mode; /* lower 9 bits of shmflg */
    ushort seq; /* sequence number */
};
```

16:26

Una cosa interessante che abbiamo su questo shared memory descriptor è esattamente il campo "short shared memory number of attaches". Questo numero ci può essere ritornato dal kernel del sistema operativo nel momento in cui noi chiamiamo questa system call, chiediamo una stat, informazioni correnti sullo stato della shared memory. Questo contatore rappresenta il numero degli attach correnti. Una shared memory ha un attach se esiste una page table che ci porta alcune pagine logiche esattamente ad essere rappresentate su questa shared memory. Il numero degli attaches rappresenta il numero delle istanze di mappatura tramite la page table di pagine logiche, sulle pagine fisiche.

```
void *shmat(int ds_shm, void *addr, int flag)
```

**Descrizione** invoca il collegamento di una memoria condivisa allo spazio di indirizzamento del processo

**Parametri** 1) ds\_shm: descrittore della memoria condivisa da collegare  
2) \*addr: indirizzo preferenziale per il collegamento (NULL come default)  
3) flag: modalita' di accesso (SHM\_R, SHM\_W, SHM\_RW)

**Restituzione** indirizzo valido per l'accesso alla memoria in caso di successo, -1 in caso di fallimento

```
int shmdet(const void *addr)
```

**Descrizione** invoca lo scollegamento di una memoria condivisa dallo spazio di indirizzamento del processo

**Parametri** \*addr: indirizzo della memoria da scollegare

**Descrizione** -1 in caso di fallimento

Allora con queste possiamo aumentare il numero degli attach correnti su questa shared memory su qualche address space. La prima è una shared memeory attach, in cui il primo parametro è il descrittore della shared memory, il secondo parametro è l'indicazione di qual è l'indirizzo logico preferenziale dove noi vorremmo andare ad accedere alle informazioni associate a quella shared memory. Infine possiamo differenziare la modalità di accesso a questa shared memory utilizzando pagine che sono a partire da \*addr (che è un indirizzo logico che forniamo). Se come secondo parametro utilizziamo NULL, la scelta di dove andare a collocare all'interno dell'AB la relazione tra le pagine logiche e lo storage condivisibile, viene scelta dal sistema operativo, e ci viene ritornato l'indirizzo logico per poter accedere.