```
  1: #ifndef _PROC_IPC_IPC_H    /* wrapper symbol for kernel use */
  2: #define _PROC_IPC_IPC_H    /* subject to change without notice */
  3:
 ...
 64: #if defined(_STYPES)
 65:
 66: /* SVR3 binary compatibility ipc_perm structure */
 67: typedef struct ipc_perm {
 68:     o_uid_t    uid;        /* owner's user id */
 69:     o_gid_t    gid;        /* owner's group id */
 70:     o_uid_t    cuid;        /* creator's user id */
 71:     o_gid_t    cgid;        /* creator's group id */
 72:     o_mode_t mode;        /* access modes */
 73:     ushort_t seq;        /* slot usage sequence number */
 74:     key_t    key;        /* key */
 75: } ipc_perm_t;
 76:
 77: /* SVR3 binary compatibility Control Commands. */
 78: #define    IPC_RMID    0    /* remove identifier */
 79: #define    IPC_SET        1    /* set options */
 80: #define    IPC_STAT    2    /* get options */ [#r]
 81:
 82: #else    /* !_STYPES */
 83:
 84: /* SVR4 ipc_perm structure */
 85: typedef struct ipc_perm {
 86:     uid_t    uid;        /* owner's user id */
 87:     gid_t    gid;        /* owner's group id */
 88:     uid_t    cuid;        /* creator's user id */
 89:     gid_t    cgid;        /* creator's group id */
 90:     mode_t    mode;        /* access modes */
 91:     ulong_t    seq;        /* slot usage sequence number */
 92:     key_t    key;        /* key */
 93:     struct ipc_sec *ipc_secp;    /* security structure ptr */
 94:     long    pad[IPC_PERM_PAD];    /* reserve area */
 95: } ipc_perm_t;
 96:
 97: /* Control Commands. */
 ...
174:
175: /* Common IPC Definitions. */
176: /* Mode bits. */
177: #define    IPC_ALLOC    0100000        /* entry currently allocated */
178: #define    IPC_CREAT    0001000        /* create entry if key doesn't exist */
179: #define    IPC_EXCL    0002000        /* fail if key exists */
180: #define    IPC_NOWAIT    0004000        /* error if request must wait */
181:
182: /* Keys. */
183: #define    IPC_PRIVATE    (key_t)0    /* private key */
184:
185:
186:
187: #if defined(_KERNEL)
 ...
221:
222: #endif /* _PROC_IPC_IPC_H */
```



```
  1: #ifndef _LINUX_IPC_H
  2: #define _LINUX_IPC_H
  3:
  4: #include <linux/types.h>
  5:
  6: #define IPC_PRIVATE ((__kernel_key_t) 0)
  7:
  8: /* Obsolete, used only for backwards compatibility and libc5 compiles */
  9: struct ipc_perm
 10: {
 11:     __kernel_key_t    key;
 12:     __kernel_uid_t    uid;
 13:     __kernel_gid_t    gid;
 14:     __kernel_uid_t    cuid;
 15:     __kernel_gid_t    cgid;
 16:     __kernel_mode_t    mode;
 17:     unsigned short    seq;
 18: };
 19:
 20: /* Include the definition of ipc64_perm */
 21: #include <asm/ipcbuf.h>
 22:
 23: /* resource get request flags */
 24: #define IPC_CREAT  00001000   /* create if key is nonexistent */
 25: #define IPC_EXCL   00002000   /* fail if key exists */
 26: #define IPC_NOWAIT 00004000   /* return error on wait */
 27:
 28: /* these fields are used by the DIPC package so the kernel as standard
 29:    should avoid using them if possible */
 30:
 31: #define IPC_DIPC 00010000  /* make it distributed */
 32: #define IPC_OWN  00020000  /* this machine is the DIPC owner */
 33:
 34: /*
 35:  * Control commands used with semctl, msgctl and shmctl
 36:  * see also specific commands in sem.h, msg.h and shm.h
 37:  */
 38: #define IPC_RMID 0     /* remove resource */
 39: #define IPC_SET  1     /* set ipc_perm options */
 40: #define IPC_STAT 2     /* get ipc_perm options */
 41: #define IPC_INFO 3     /* see ipcs */
 42:
 43: /*
 44:  * Version flags for semctl, msgctl, and shmctl commands
 45:  * These are passed as bitflags or-ed with the actual command
 46:  */
 47: #define IPC_OLD 0    /* Old version (no 32-bit UID support on many
 48:                 architectures) */
 49: #define IPC_64  0x0100  /* New version (support 32-bit UIDs, bigger
 50:                 message sizes, etc. */
 51:
 52: #ifdef __KERNEL__
 53:
 54: #define IPCMNI 32768  /* <= MAX_INT limit for ipc arrays (including sysctl changes) */
 55:
 56: /* used by in-kernel data structures */
 57: struct kern_ipc_perm
 58: {
 59:     key_t        key;
 60:     uid_t        uid;
 61:     gid_t        gid;
 62:     uid_t        cuid;
 63:     gid_t        cgid;
 64:     mode_t        mode;
 65:     unsigned long    seq;
 66: };
 67:
 68: #endif /* __KERNEL__ */
 69:
 70: #endif /* _LINUX_IPC_H */
 71:
 72:
```


(For readability, not all matches are indicated.)