

```

1: #ifndef _PROC_IPC_SHM_H /* wrapper symbol for kernel use */
2: #define _PROC_IPC_SHM_H /* subject to change without notice */
...
56:
57: /*
58: * Permission Definitions.
59: */
60:
61: #define SHM_R IPC_R /* read permission */
62: #define SHM_W IPC_W /* write permission */
63:
64: /*
65: * Message Operation Flags.
66: */
67:
68: #define SHM_RDONLY 010000 /* attach read-only (else read-write) */
69: #define SHM_RND 020000 /* round attach address to SHMLBA */
70:
71:
72: typedef ulong_t shmatt_t;
73:
74:
75: /*
76: * Structure Definitions.
77: */
...
195:
196: /*
197: * Shared memory control operations
198: */
199: #define SHM_LOCK 3 /* Lock segment in core */
200: #define SHM_UNLOCK 4 /* Unlock segment */
201:
202:
203: #ifdef _KERNEL
204:
205: /*
206: * Macros to lock and unlock a kshmid_ds cell.
207: * idp is a pointer to the kshmid_ds cell to be locked/unlocked.
208: */
209: #define SHMID_LOCK(kshmp) LOCK_PLMIN(&(kshmp)->kshmlck)
210: #define SHMID_UNLOCK(kshmp, pl) UNLOCK_PLMIN(&(kshmp)->kshmlck, (pl))
211:
212: struct proc;
213: extern int shmconv(int, struct kshmid_ds **);
214: extern void shminit(void);
215: extern void shmfork(struct proc *, struct proc *);
216: extern void shmexit(struct proc **pp);
217: extern void shmexec(struct proc **pp);
218:
219: #else
220:
221: #ifdef __STDC__
222:
223: void *shmat(int, const void *, int);
224: int shmctl(int, int, struct shmids *);
225: int shmdt(const void *);
226: int shmget(key_t, size_t, int);
227: #else
228:
229: int shmctl();
230: int shmget();
231: void *shmat();
232: int shmdt();
233:
234: #endif /* __STDC__ */
235:
236: #endif /* _KERNEL */
237:
238: #if defined(__cplusplus)
239: }
240: #endif
241:
242: #endif /* _PROC_IPC_SHM_H */

```

```

1: /* Copyright (C) 1995, 1996, 1997, 2000, 2002, 2004
2: Free Software Foundation, Inc.
3: This file is part of the GNU C Library.
4:
5: The GNU C Library is free software; you can redistribute it and/or
6: modify it under the terms of the GNU Lesser General Public
7: License as published by the Free Software Foundation; either
8: version 2.1 of the License, or (at your option) any later version.
9:
10: The GNU C Library is distributed in the hope that it will be useful,
11: but WITHOUT ANY WARRANTY; without even the implied warranty of
12: MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13: Lesser General Public License for more details.
14:
15: You should have received a copy of the GNU Lesser General Public
16: License along with the GNU C Library; if not, write to the Free
17: Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
18: 02111-1307 USA. */
19:
20: #ifndef _SYS_SHM_H
21: # error "Never include <bits/shm.h> directly; use <sys/shm.h> instead."
22: #endif
23:
24: #include <bits/types.h>
25:
26: /* Flags for `shmatt'. */
27: #define SHM_RDONLY 010000 /* attach read-only else read-write */
28: #define SHM_RND 020000 /* round attach address to SHMLBA */
29: #define SHM_REMAP 040000 /* take-over region on attach */
30:
31: /* Commands for `shmctl'. */
32: #define SHM_LOCK 11 /* lock segment (root only) */
33: #define SHM_UNLOCK 12 /* unlock segment (root only) */
34:
35: __BEGIN_DECLS
36:
37: /* Segment low boundary address multiple. */
38: #define SHMLBA (__getpagesize ())
39: extern int __getpagesize (void) __THROW __attribute__((__const__));
40:
41:
42: /* Type to count number of attaches. */
43: typedef unsigned short int shmatt_t;
44:
45: /* Data structure describing a set of semaphores. */
46: struct shmids
47: {
48:     struct ipc_perm shm_perm; /* operation permission struct */
49:     int shm_segsz; /* size of segment in bytes */
50:     __time_t shm_atime; /* time of last shmatt() */
51:     __time_t shm_dtime; /* time of last shmdt() */
52:     __time_t shm_ctime; /* time of last change by shmctl() */
53:     __pid_t shm_cpid; /* pid of creator */
54:     __pid_t shm_lpid; /* pid of last shmop */
55:     shmatt_t shm_nattch; /* number of current attaches */
56: };
57:
58: __END_DECLS

```