

Aug 14, 03 15:01	lock.h	Page 1/2
<pre> /* Quimera, luglio 2003 * /* Versione 1.0.0 * /* * /* Libreria per la gestione della mutua * /* esclusione usando il "file locking". * /* * /* Dichiarazione dei tipi di variabili * /* e di procedure utili alla libreria. */ #ifndef _LOCK_H #define _LOCK_H #include <sys/types.h> #include <sys/stat.h> #include <unistd.h> #include <fcntl.h> /* ----- Dichiarazione delle funzioni e procedure ----- */ int create_mutex (const char *path_name); /* Requisiti : percorso del file valido * /* Ruolo: crea la mutua esclusione utilizzando il file * /* locking. Restituisce il descrittore del file o -1 se * /* l'operazione non e' riuscita e errno contiene l'errore */ int lock_mutex (int fd, short whence, off_t start, off_t len); /* Requisiti : fd valido * /* Ruolo: realizza il lock della mutua esclusione usando il * /* file locking. Restituisce 0 se l'operazione e' riuscita * /* o -1 nel caso contrario e errno contiene l'errore * /* Se il file e' libero il lock viene acquisito e la * /* funzione ritorna immediatamente; altrimenti la funzione * /* si blocchera' fino al rilascio del lock. * /* Valori possibili dei parametri: * /* whence : da dove calcolare l'offset * /* SEEK_SET dall'inizio del file * /* SEEK_CUR dalla posizione corrente * /* SEEK_END dalla fine del file * /* start : fornisce la posizione di partenzadel segmento * /* relativa a whence * /* len : e' la lunghezza del segmento in byte, 0 per * /* per tutta la lunghezza del file */ int unlock_mutex (int fd, short whence, off_t start, off_t len); /* Requisiti : fd valido * /* Ruolo: realizza l'unlock della mutua esclusione usando il * /* file locking. Restituisce 0 se l'operazione e' riuscita * /* o -1 nel caso contrario e errno contiene l'errore * /* Valori possibili dei parametri: * /* whence : da dove calcolare l'offset * /* SEEK_SET dall'inizio del file * /* SEEK_CUR dalla posizione corrente * /* SEEK_END dalla fine del file * /* start : fornisce la posizione di partenzadel segmento * /* relativa a whence * /* len : e' la lunghezza del segmento in byte, 0 per * /* per tutta la lunghezza del file */ int remove_mutex (const char *path_name); /* Requisiti : percorso del file valido * /* Ruolo: elimina la mutua esclusione utilizzando il file * /* locking. Restituisce 0 se l'operazione e' riuscita o -1 * /* nel caso contrario e errno contiene l'errore */ </pre>		

Aug 14, 03 15:01	lock.h	Page 2/2
<pre> #endif </pre>		

Aug 12, 03 23:41	lock.c	Page 1/1
<pre> /* Quimera, luglio 2003 */ /* Versione 1.0.0 */ /* /* Libreria per la gestione della mutua /* esclusione usando il "file locking". /* /* Definizione dei tipi di variabili /* e di procedure utili alla libreria. */ #include "lock.h" /* ----- create_mutex ----- */ int create_mutex (const char *path_name) { return open(path_name, O_EXCL O_CREAT); } /* ----- lock_mutex ----- */ int lock_mutex (int fd, short whence, off_t start, off_t len) { struct flock lock; /* inizializzazione della struttura */ lock.l_type = F_WRLCK; /* tipo: read or write */ lock.l_whence = whence; /* posizione d'inizio */ lock.l_start = start; /* inizio della regione dalla posizione specificata */ lock.l_len = len; /* lunghezza in bytes della regione */ return fcntl(fd, F_SETLKW, &lock); } /* ----- unlock_mutex ----- */ int unlock_mutex (int fd, short whence, off_t start, off_t len) { struct flock lock; /* inizializzazione della struttura */ lock.l_type = F_UNLCK; /* tipo: read or write */ lock.l_whence = whence; /* posizione d'inizio */ lock.l_start = start; /* inizio della regione dalla posizione specificata */ lock.l_len = len; /* lunghezza in bytes della regione */ return fcntl(fd, F_SETLK, &lock); } /* ----- remove_mutex ----- */ int remove_mutex (const char *path_name) { return unlink(path_name); } </pre>		