```
//---------- 1 File ----------

/*@ open(path, flag)  =

REQ TRUE

ENS  (ex fd: fd=-1 ; emp ; (!_(fd))^* ; fd)

\/ (ex fd  : !(fd=-1) /\  flag=0 ; open(fd) ; (!write(fd))^* /\ (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)

\/ (ex fd : !(fd=-1) /\  flag=1 ; open(fd) ; (!read(fd))^* /\ (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)

\/ (ex fd : !(fd=-1) /\  flag=2 ; open(fd) ; (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)  @*/


/*@ fopen(path, flag)  =

REQ TRUE

ENS (ex fd: fd=-1 ; emp ; (!_(fd))^* ; fd)

\/ (ex fd  : !(fd=-1) /\  flag=0 ; open(fd) ; (!write(fd))^* /\ (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)

\/ (ex fd : !(fd=-1) /\  flag=1 ; open(fd) ; (!read(fd))^* /\ (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)

\/ (ex fd : !(fd=-1) /\  flag=2 ; open(fd) ; (!close(fd))^*  .  close(fd)  .  (_)^* ; fd)  @*/


/*@ close(fd) =

   REQ  TRUE

   ENS (ex r : r=unit ; close(fd) ; (_)^* ; r) @*/

/*@ fclose(fd) =

   REQ  TRUE

   ENS (ex r : r=unit ; close(fd) ; (_)^* ; r) @*/

/*@ write(fd, buf, size) =

   REQ  TRUE

   ENS (ex r : TRUE ; write(fd) ; (_)^* ; r) @*/
```

```
/*@ read(fd, buf, size) =
   REQ  TRUE
   ENS (ex r : TRUE ; read(fd) ; (_)^* ; r) @*/
/*@ exit(code) =
   REQ  TRUE
   ENS (: TRUE ; exit() ; (_)^* ; code; -2 ) @*/
/*@ return(t) =
   REQ  TRUE
   ENS (: TRUE ; emp ; (_)^* ; t; -1) @*/

//---------- 2 Thread ----------
/*@ pthread_create(a, b, c, d) =
   REQ TRUE
   ENS (ex  l : TRUE ; pthread_create(a)  ; (!pthread_join(a))^*  .  pthread_join(a)  .  (_)^* ;
l)  @*/
/*@ pthread_join(a, b) =
   REQ  TRUE
   ENS (ex r : TRUE ; pthread_join(a) ; (_)^* ; r) @*/
/*@ pthread_mutex_lock(a) =
   REQ TRUE
   ENS (ex  l : TRUE ; pthread_mutex_lock(a)  ; (!pthread_mutex_unlock(a))^*  .
pthread_mutex_unlock(a)  .  (_)^* ; l)  @*/
/*@ pthread_mutex_unlock(a) =
   REQ  TRUE
   ENS (ex r : TRUE ; pthread_mutex_unlock(a) ; (_)^* ; r) @*/

//---------- 3 Memory ----------
/*@ free(ptr)  =
   REQ  TRUE
```

```
   ENS (ex  r : r=unit ; free(ptr) ; (!_(ptr))^* ; r) @*/
/*@ malloc(size) =
   REQ TRUE
   ENS (ex  l : TRUE ; malloc(l)  ; (!free(l))^*  .  free(l)  .  (_)^* ; l)  @*/
/*@ realloc(ptr, size) =
   REQ TRUE
   ENS (ex  l : TRUE ; realloc(l)  ; (!_(ptr))^* /\ (!free(l))^*  .  free(l)  .  (_)^* ; l)  @*/
/*@ return(t) =
   REQ  TRUE
   ENS (: TRUE ; emp ; (_)^* ; t; -1) @*/
/*@ printf(t, p) =
   REQ  TRUE
   ENS (: TRUE ; printf(p) ; (_)^* ; unit) @*/
/*@ strncpy(a, b, c) =
   REQ  TRUE
   ENS (: TRUE ; strncpy(a) ; (_)^* ; unit) @*/

//---------- 4 Socket ----------
/*@ socket(a, b, c) =
   REQ TRUE
   ENS (ex  l : l>=0 ; socket()  ; (!close(l))^*  .  close(l)  .  (_)^* ; l)  @*/
/*@ close(fd) =
   REQ  TRUE
   ENS (ex r : r=unit ; close(fd) ; (_)^* ; r) @*/

//---------- 5 Database ----------
/*@ sqlite3_open(a, b) =
   REQ TRUE
```

```
   ENS (ex  l : TRUE ; sqlite3_open(b)  ; (!sqlite3_close(b))^* .  sqlite3_close(b)  .  (_)^* ; l)
@*/

/*@ sqlite3_close(a) =

   REQ  TRUE

   ENS (ex r : TRUE ; sqlite3_close(a) ; (_)^* ; r) @*/


//---------- 6 Protocol ----------

/*@ fgets(a, b, c)  =

   REQ  TRUE

   ENS (ex  r : r=NULL ; fgets(a) ; return() ; r) @*/

/*@ returnChunkSize(a)  =

   REQ  TRUE

   ENS (ex  r : r=-1 ; returnChunkSize(a) ; return() ; r) @*/

/*@ malloc(a)  =

   REQ  TRUE

   ENS (ex  r : r=NULL ; malloc(a) ; return() ; r) @*/

/*@ gethostbyaddr(a, b, c)  =

   REQ  TRUE

   ENS (ex  r : r=NULL ; gethostbyaddr(a) ; (!_(r))^* ; r) @*/

/*@ pthread_mutex_lock(a)  =

   REQ  TRUE

   ENS (ex  r : !(r=0) ; pthread_mutex_lock(a) ; (!_(a))^* ; r) @*/

/*@ pthread_mutex_unlock(a)  =

   REQ  TRUE

   ENS (ex  r : TRUE ; pthread_mutex_unlock(a) ; (_)^* ; r) @*/

/*@ strncpy(a, b, c)  =

   REQ  TRUE

   ENS (ex  r : TRUE ; strncpy(b) ; (_)^* ; r) @*/
```

```
/*@ strcpy(a, b)  =
   REQ  TRUE
   ENS (ex  r : TRUE ; strcpy(b) ; (_)^* ; r) @*/
/*@ memcpy(a, b, C)  =
   REQ  TRUE
   ENS (ex  r : TRUE ; memcpy(a, b) ; (_)^* ; r) @*/
/*@ return(t) =
   REQ  TRUE
   ENS (: TRUE ; return() ; (_)^* ; t; -1) @*/
```