# Test Case: get\_block

## Function Prototype

extern int get\_block(int blknum, byte \*buf);

|  |  |
| --- | --- |
| ID | 1 |
| Description | Get a block from the simulated disk; verify the retrieved contents in memory are the same as the actual contents of the block. If the block specified is outside the disk, an error should occur. |
| Related Requirements | 1. The simdisk must be null initialized. 2. One block of test data in a random block location must be defined. 3. The block of test data must also be saved outside the simdisk so it can be verified against. 4. The endian-ness of the block must be known and consistent with the test data. 5. An array of byte type must be initialized (with a length of one block) to store the test output. |
| Test Category | Block Level i/o |
| Author | dsmullen |
| Is Automatable | No |
| Is Automated | No |
| Required Fields | None |
| Expected Result |  |
| Actual Result |  |
| Pass/Fail |  |
| Remarks |  |
| Test Summary | 1. Edit a block at a random block location and insert the block of test data. 2. Call the get\_block function:    1. Specify blknum as the random block location.    2. Specify \*buf as the predefined array of byte type. 3. View the contents of the specified byte array. 4. Compare the contents to the externally saved block contents. 5. Call the get\_block function a second time:    1. Specify blknum as a location larger than the amount of blocks on the simdisk.    2. Specify \*buf as the predefined array of byte type. 6. Observe function’s return value. |

# Test Case: put\_block

## Function Prototype

extern int put\_block(int blknum, byte \*buf);

|  |  |
| --- | --- |
| ID | 2 |
| Description | Write a block to the simulated disk; verify the retrieved contents on disk are the same as the contents of the block in memory. If the block specified is outside the disk, an error should occur. If the block length is larger than one block, only one block should be written. |
| Related Requirements | 1. The simdisk must be null initialized. 2. Two blocks of test data in memory must be defined:    1. Inside a byte array with the length of one block, a block of random test data must be defined.    2. Inside a byte array with length greater than one block, a block of random test data must be defined. 3. The block of test data must also be saved outside the simdisk so it can be verified against. 4. The endian-ness of the block must be known and consistent with the test data. |
| Test Category | Block Level i/o |
| Author | dsmullen |
| Is Automatable | No |
| Is Automated | No |
| Required Fields | None |
| Expected Result |  |
| Actual Result |  |
| Pass/Fail |  |
| Remarks |  |
| Test Summary | 1. Call the put\_block function:    1. Specify blknum as a random block location.    2. Specify \*buf as the predefined block contents with length of one block. 2. View the contents of the simdisk at the specified block location. 3. Compare the contents of the simdisk to the externally saved block contents. 4. Call the put\_block function a second time:    1. Specify blknum as a location larger than the amount of blocks on the simdisk.    2. Specify \*buf as the predefined array of byte type. 5. Observe function’s return value. 6. Call the put\_block function a third time:    1. Specify blknum as a random block location.    2. Specify \*buf as the predefined block contents with length greater than one block. 7. View the contents of the simdisk at the specified block location. 8. Compare the contents of the simdisk to the externally saved block contents. 9. Verify the length of the content written is equal to one block. |