

Test Cases

1. HD

The HD file represents the hard disk. The file system structure has been initialized properly on this hard disk. You do not need to do any modifications on HD.

The file system structure is shown in Figure 1.

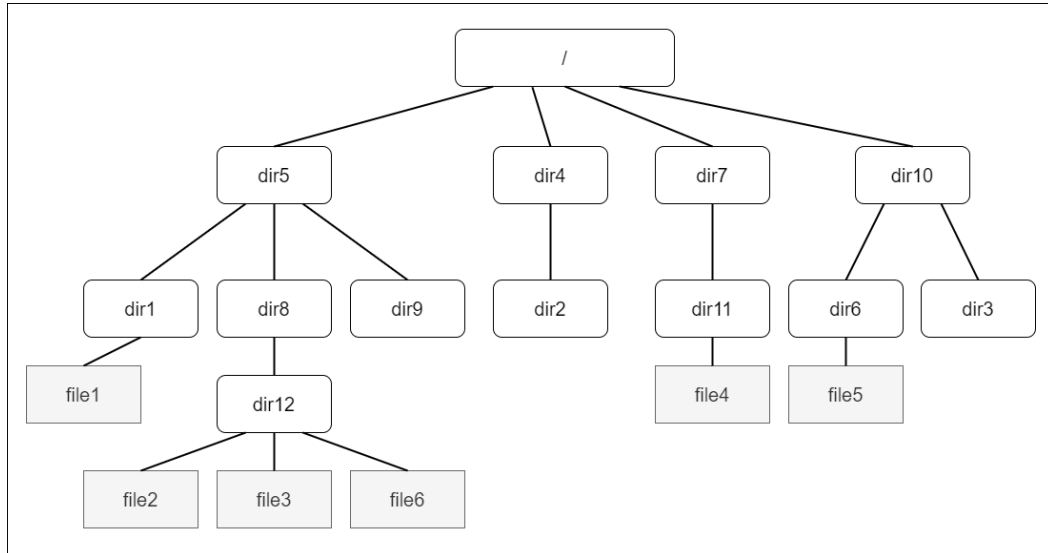


Fig.1 Test file system

2. Recommend test cases:

(1) open_test()

You can call open_t() to get the inode of a file, for example :

```
char pathname[MAX_COMMAND_LENGTH] = "/dir5/dir1/file1";  
int i_number = open_t(pathname);
```

Compile and run:

```
gcc -o open_test open_test.c call.c
```

```
./open_test
```

The test cases are shown below:

filepath	note
/	The root directory
/dir5	One of the 1 st level directory
/dir5/dir1	One of the 2 nd level directory
/dir5/dir1/file1	A file under the 2 nd level directory
/dir5/dir8/dir12/file2	A file under the 3 rd level directory

(2) read_test()

Then use read_t() reading some content of file1 to buffer and display read size.

Compile and run:

```
gcc -o read_test read_test.c call.c
./read_test
```

Or

`./read_test >> test_result.txt`, to save the result.

Here are several suggested cases:

```
read_t (i_number, offset, buf, count)
```

(Only the offset and count will be changed for each case.)

offset	count	note	
0	100	Begin at direct block, end in the same block	Access only one block
4100	1000	Begin at direct block, end in the same block	
8500	300	Begin at indirect block, end in the same block	
40965	800	Begin at indirect block, end in the same block	
15	5000	Begin at direct block, end in another direct block	Access multiple blocks
100	50000	Begin at direct block, end in indirect block	
9000	60000	Begin at indirect block, end in indirect block	
File_size - 50	10000		Overflowing original file test
File_size + 50	10		Offset out of range
10	MAX_FILE_SIZE-100		Access the maximum number of blocks

Noted:

- **Your programs must be compiled under XUbuntu! Other VM, Windows or MAC may incur incompatible issues.**
- **The grading scheme is different from the test case! Test cases are used for self-checking only.**
- **The test data can be found in /test-cases/Plato.txt. Because it's quite long, you can use: `./read_test >> test_result.txt`, to save the result.**