Simulated Unix File System Project Evaluation

ENGR 3950U / CSCI 3020U
November 26 & December 3, 2012
Instructor D. Kamran Sartipi

1. Test Suite for Simulated Unix File System

Write “four test cases” for each category of tests mentioned below and test your project accordingly. You should submit this test suite with your source code files in a zipped file.

1. **Primitive level**: the test cases in this set will check the different one-level operations, such as: successfully creating/deleting/opening/closing files and receiving corresponding file descriptors.

2. **Advanced level**: the test cases in this set will check cases involving a combination of different operations of the file system including: creating several directories and files; opening and closing files; writing and reading from files; and directory listing.

Test No.	Test Level	Test Description	Expected Output	Observed Output
1	Primitive			
...	Primitive			
…	Primitive			
...	….			
…	Advanced			
...	Advanced			

2. Simulated Unix File System Project Demonstration

1. Project demonstration will take place in the regular Lab Locations. **All group** members should participate in a 15-minute demonstration period according to a schedule that will be posted later.
2. The members of each group should be present in the Lab at least 20 minutes before the scheduled demonstration time to prepare their demonstration.
3. The source codes “should be downloaded” from the Blackboard System and compiled in front of the TAs (Groups cannot use codes other than the ones already submitted to the Blackboard). The students can use their own laptop which will be connected to the data projector.

4. At the beginning of demonstration, TAs will give each group 1-2 minutes time to download and “compile” their code and make it ready for evaluation. If a group cannot compile and run their code in this period of time, group members will lose the code evaluation mark and only the source code will be evaluated.
5. Once the code is compiled and run, TAs will ask you to test its functionality.
6. TAs will ask questions regarding the source code from **each** member of the group.
7. **Simdisk.data**: students should be able to explain the data patterns created in the simdisk.data as a result of running the test cases.
8. There are more than 20 test cases from simple tests to complex tests. Two samples are provided below:
   a. Test #2 (simple): sequence of the operations and the expected result are as follows:
   i: 1
   m: /foo/bar, 0
   **** There should be error as /foo does not exist.

   b. Test #15 (complex): sequence of operations and the expected result are as follows:
   i: 1
   m: /foo, 1
   m: /foo/bar, 1
   m: /foo/zam, 1
   o: /foo
   R: fd
   R: fd
   **** There should be "zam" returned

3. Marking Scheme [100 Marks]

1. **[70 Marks]** *Successful passing the test cases*. The TAs will ask you to run approximately 20 test cases ranging from simple to complex test cases.
2. **[5 Marks]** *Task breakdown*. Submit to the TA a task breakdown showing the percentage of contribution of each group member to the design and implementation parts of the project.
3. **[10 Marks]** *Explanation of important operations*. These include:
   a. How to open a file multiple times, and how to create multiple files.
   b. Implementation of I-node table and File Control Block.
   c. Implementation of index-block and accessing to the file blocks.
   d. Structure of the simdisk.data.
4. **[15 Marks]** *Source code walkthrough*. The TAs will select some files and will ask individual group members to briefly explain different parts to of its source code.