CS4218 Hackathon Report

Team_15

CAO SHENGZE
KIM HYUNG JON
YAP HAN CHIANG

1.Introduction

For duplicates:

Under the lab7, the definition of duplicates is given as:

a duplicate - behavior that exercises the same problem or an identical way of reproducing another bug (e.g., if a function f(String s){} does not handle special symbols in the input, then failures reported for cases f("\"); f("&"); f("$"); are duplicates! Only one of them should be reported)

After checking with our lab TA, we know what matters is not the function, but the functionality. If a function contains multiple functionalities, then it may have multiple bugs. For the bugs 1-6, they are all about sort but as they are totally different cases that the sortApplication cannot handle properly, therefore we think they are 6 bugs instead of 1.

After checking carefully, we are quite sure all the following bugs are related to different functionalities.

For Test Case:

On the bug report table, the Test Case would be a reference to our implementation inside a Java Test Class "ErrorRevealingTest" if the test case contains a certain file. Otherwise, the test case would just be the argument which would generate the errors. Both have relative test inside the Java Test Class.

Both the ErrorRevealingTest and all the test files are zipped with this report. The test files should be put at the same directory as the src folder. The ErrorRevalingTest should be put at the package sg.edu.nus.comp.cs4218test.integration(Together with the Team I's integration test files.

For System Related Erros:

The Team I's project would fail 13 test cases under our Windows machine due to OS problem. as the OS problem should not be regarded as BUGs, we put these errors under our proper bug report just for Team I's information.

For TeamI's assumptions:

Team I make some assumptions which we think violate the specifications of the project description. For example, wc should accept multiple files; the pipeline should be allowed to use together with semicolon.

We ignore these invalid assumptions and still think the flaws as bugs.

General method used for testing:

1. First step is to look through all the test cases that Team I has and check whether they all pass. We also try to match their integration test cases to a pairwise table to see what is missing.
2. We go through the applications one by one then. For each application, we first look at its own performance (unit test), then we try to use it together with other applications (integration test) to see whether we can find errors. We focus more on the pairs that not appear in their own test cases.
3. During step 2, instead of only generating new test cases, we also copy our tests into the other team's test folder and run the tests, after making adjustments to remove build errors. We observed the result, and first determined whether the failures were happening because of different implementation (e.g. different way of counting the chars and lines, different way of output) or actually due to bugs.
4. We do system test by running their application directly, we randomly choose shell functions and applications and combine them to see whether a bug happens.
5. During the process 1-4, we list down all the test cases that would generate errors, then we analyze the reason behind the bugs and remove the duplicates.

2. Bug Report

Bug Number	Description	Test Case	Comments
1	Sort: if the "-n" option is chosen, numbers come behind alphabets	See the testBugRevealing1()	
2	Sort: if the "-n" option is chosen and at least one line in the file contains only whitespace(s), sort is unable to process the file and throws exception	See the testBugRevealing2()	
3	Sort: if multiple files are used, the last line of nth file and the first line of n+1th file get merged to one	See the testBugRevealing3()	
4	Sort: some special characters get come behind numbers or alphabets	See the testBugRevealing4()	
5	Sort: numeric sort does not work on stdin (produces same result as non-numeric sort)	See the testBugRevealing5()	

6	Sort: sometimes reads less empty lines than the file actually contains. Confirmed happening if file only contains empty lines or the only non-empty line in the file contains only whitespaces.	See the testBugRevealing6()	
7	Wc: does not work with more than one files. Team commented that their implementation is to consider only the last arg as filename, but this violates the project description that wc should accept “given files”.	See the testBugRevealing7()	
8	Doulbe quote: wrapping arguments containing whitespaces in double quotes to indicate that the string is a single argument and that the whitespaces should not be considered as argument separators does not work. The relative single quote error is at Bug No.21	sort "fileBugRevealing8 with space.txt”	
9	Double quote: Process of the double quote should be executed before the command, then echo a”b” should just print ab instead of regrading it as an invalid command.	echo a”b”	
10	Single quote: Process of the single quote should be executed before the command, then echo a’b’ should just print ab instead of regarding it as an invalid command.	echo a’b’	
11	Redirection input would not work if the redirection is not at the last	echo a > 1.txt; cat < 1.txt 1.txt	
12	Pipeline and semicolon cannot appear together (Already see the assumption in the report, but still think the pipeline and semicolon should be able to	echo a | cat; echo b | cat	

	appear together)		
13	**Globbing is not implemented at all**	cat testDir/*	
14	Command substitution: does not work with more than 2 command substitutions	sort `cat cmdSubFile.txt` `head cmdSubFile4.txt`	
15	SedApplication: does not support regex, it just treat the regex as charSequence	echo a | sed s/[a]/b	
16	SedApplication: cannot use symbol “s” as separator symbol. The specification is: However, this separation symbol should not be used inside the regexp and the replacement string. But it does not say the s cannot be used.	echo a | sed ssasbs	
17	Cmd substitution with results of a pipe does not work	echo `echo a | cat`	
18	Pipe: cannot handle large amount of bytes e.g. result of cal 2017	cal 2017 | cat	
19	Cal: if 3 or more arguments are passed and the arguments are invalid, cal outputs nothing instead of throwing an exception	cal 1 1 1	
20	Grep: cannot handle input from stdin with certain contents.	see the testBugRevealing20 ()	
21	The single quote cannot transfer the space to a space character. The double quote actually has the same problem,but it seems like the same bug as in bug No.8. Therefore, not put it here.	echo “a b”|grep ‘ ‘	

Some other system related errors:

Here are some other problems that we find, these problems may occur due to OS, therefore, we do not put it into the bug reports. It would be better if these problems can be fixed so that all the test cases would pass in different OS as well.

1. The test case testWcWithStdinWithOptionMForNonAscii() in WcApplicationTest may fail if the test machine does not contain the relative code for test. My computer, it would get 8 instead of 3. As I think it a System problem, I did not put it into the bug report.
2. The test case testPipeCatAndSedAndWc() may fail under windows as in windows a newline character is \r\n, which with length of 2 instead of 1. Lots of other integration test cases with Wc all have this problem as well.
3. The test case testGValidEchoDate() may fail if system language is not English. Some other test cases have this problem as well.