This tutorial is to show how to use xSlice to help you locate the bug in a program.

Please read the following descriptions, and finish each step:

***Preparation***

1. Compile the correct version of the wordcount program without instrumentation  
   *> cc -o wordcount main.c wc.c*

*….. (ignore the warnings) …..*

1. Compile the faulty version of the wordcount program with instrumentation  
   *> atac cc -g -o wc\_err main\_err.c wc.c*

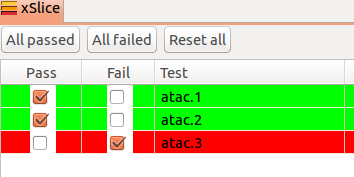
*….. (ignore the warnings) …..*

1. Clean the ATAC\_TRACE environmental variable  
   *> export ATAC\_TRACE=*

***Find out which test cases invokes the failure***

1. Run correct version and faulty version with the 1st test case  
   *> ./wordcount input1  
   ….. (some output) …..  
     
   > ./wc\_err input1  
   ….. (some output) …..*  
   The output generated by wordcount and wc\_err should be the same, which means this test case did not invoke the error
2. Run the 2nd test case  
   *> ./wordcount -w input1  
   ….. (some output) …..  
     
   > ./wc\_err -w input1  
   ….. (some output) …..*The output generated by wordcount and wc\_err should be the same, which means this test case did not invoke the error neither
3. Run the 3rd test case  
   *> ./wordcount -w < input1  
   ….. (some output) …..  
     
   > ./wc\_err -w < input1  
   ….. (some output) …..*The output of correct version and faulty version should be different. This means the 3rd test case invokes the failure

***Use xSlice to help you identify the location of the bug***

1. Run the tool using  
   *> xsuite wc\_err.atacp  
   then open the trace file “atac.trace” to load the coverage information to the tool*
2. Switch the tool to xSlice mode by clicking menu “Coverage” -> “xSlice”
3. Specify which tests are the correct test cases, and which are the failed test cases.  
   
4. Check the source code, the bug location should be highlighted by RED already.