Observational Study (Sample Bug from "find" Utility)

Please make sure that you
1) have downloaded and installed the VM
2) have followed the tutorial on the desktop, and
3) note down the starting and end time for each task (bug diagnosis, fixing)
4) take no more than 45 minutes for each bug.

Please read the bug report below and get familiar with the source code.

* Required

Bug Report

find segfaults when using -regex

For instance, finding every file/directory using -regex
$ ./find -regex '.*'
Segmentation Fault

For instance, find every file/directory that contains an 'a'
$ ./find -regex '.a'
Segmentation Fault

1. **ID (Ignore)** *

....

Note down your STARTING TIME and determine the root-cause of the error.

You can inspect and execute
* the failing test case: <testcase.sh>
* the source code and executable: <source-folder>

2. **How familiar are you with the source code?** *

Mark only one oval.

○ Not at all familiar
○ Slightly familiar
○ Moderately familiar
○ Very familiar
○ Extremely familiar

3. **How difficult was it to understand the runtime actions leading to the error?** *

Mark only one oval.

○ Not at all difficult
○ Slightly difficult
○ Moderately difficult
○ Very difficult
○ Extremely difficult

4. **Which *techniques* did you use to understand the runtime actions leading to the error?** *

Check all that apply.

□ Trace-based Debugging (using printing; e.g., println, log4c)
□ Interactive or Online Debugging (using breakpoints; e.g., gdb, jdb)
□ Post-Mortem or Offline Debugging (using core dumps and stack traces)
□ Regression Debugging to identify failure-inducing changes (e.g., git bisect)
□ Statistical or Spectrum-based Debugging to find suspicious statements (e.g., Tarantula)
□ Program Slicing (e.g., Frama-C, CodeSurfer)
□ Time Travel or Reversible Debugging (e.g., UndoDB)
□ Algorithmic or Declarative Debugging (e.g., Java DD)

5. **Which *tools* did you use specifically to understand the runtime actions leading to the error?** *

Comma-separated list of tools, e.g., gdb, log4c, code surfer

....

6. **How much *time* did you spend understanding the runtime actions leading to the error?** *

Mark only one oval.

○ 1 minute or less
○ 2 - 5 minutes
○ 5 - 10 minutes
○ 10 - 20 minutes
○ 20 - 30 minutes
○ 30 - 40 minutes
○ 40 - 50 minutes
○ 50 - 60 minutes
○ 60 minutes or more

Manual Bug Diagnosis

7. **IMPORTANT: Enter 3 to 5 regions in the source code needed to explain the root cause of the error.** *

e.g., find/find.c:184-187 (format= "subdirectory/program: line numbers")

8. *

9. *

10.

11.

12. **IMPORTANT: In your own words, what is the root cause of the error? How does it come about?** *

Please use the program regions specified above as references and be as specific as possible.

13. **How confident are you about the correctness of your explanation.** *

If you cannot explain the error, select "Not at all confident"
Mark only one oval.

○ Not at all confident

○ Slightly confident

○ Moderately confident

○ Very confident

○ Extremely confident

14. **If you could *not* explain the error, what prevented you from doing so?**

15. **Which concrete steps did you take to understand the runtime actions leading to the error?** *

e.g., First, ... After that .. Finally,...

....

....

....

....

....

Automated Bug Diagnosis

16. **In a few words, "If only I was told that ..., I would have *earlier* understood the runtime actions leading to the error".**

....

17. **Do you believe that the root cause of the error can be explained intuitively by the push of a button?** *

Mark only one oval.

○ Yes, in principle a tool might be able to explain this error.

○ No, there will never be a tool that can explain this error.

18. **Why do you (not) believe so?**

....

....

....

....

....

Note down your STARTING TIME and fix the error.

19. **How difficult was it to fix the error?** *

Mark only one oval.

○ Not at all difficult

○ Slightly difficult

○ Moderately difficult

○ Very difficult

○ Extremely difficult

20. **How much *time* did you spend fixing the error?** *

Mark only one oval.

○ 1 minute or less

○ 2 - 5 minutes

○ 5 - 10 minutes

○ 10 - 20 minutes

○ 20 - 30 minutes

○ 30 - 40 minutes

○ 40 - 50 minutes

○ 50 - 60 minutes

○ 60 minutes or more

Manual Bug Fixing

21. **IMPORTANT: Copy & paste the generated patch here.** *

To derive the patch, execute "diff.sh" in the folder corresponding to that error.

....

....

....

....

....

22. **In a few words and on a high level, what did you change to fix for the error?** *

....

....

....

....

....

23. **How confident are you about the correctness of your fix.** *

If you cannot fix the error, select "Not at all confident"

Mark only one oval.

○ Not at all confident

○ Slightly confident

○ Moderately confident

○ Very confident

○ Extremely confident

24. **In a few words, how did you make sure this is a good fix?** *

....

....

....

....

....

25. **If you could not fix the bug, what prevented you from doing so?**

....

....

....

....

....

Automated Bug Fixing

26. **Do you believe that this error can be fixed reliably by the push of a button?** *

Mark only one oval.

- ○ Yes, in principle a tool might be able to fix this error reliably.
- ○ No, there will never be a tool that can fix this error reliably.

27. **Why do you (not) believe so?**

....

....

....

....

....

Feedback

28. **How much time did you spend filling this questionnaire?**

Mark only one oval.

- ○ less than 5 minutes
- ○ 5 to 10 minutes
- ○ 10 to 20 minutes
- ○ 20 to 30 minutes
- ○ 30 to 40 minutes
- ○ more than 40 minutes