

# Examining the Strength of the diff Command for Patching in a Command Line Interface

Brian Hazzard

Advisors – Professors Aaron Cass and Chris Fernandes

## Abstract

Frequently used in a command line interface, the diff command has been used for nearly 40 years as a file comparison tool [1]. In this study, we test the effectiveness of the diff command line tool as a way to preview changes for different actions, or as a patch. We specifically look at the effectiveness in regards to selective undo: a manner of undoing previous actions in a non-linear way. Results found that users who were simply shown what the outcome of their patch would be were able to navigate the selective undovisualizations faster than those who were shown diff statements.

[1] James Wayne Hunt and MD MacIlroy. An algorithm for differential file comparison. Bell Laboratories, 1976.

## Results

The non diff approach takes both less time and less clicks than diff does. In all cases, the average time to complete a scenario is less for non diff subjects. Only in scenario 3 does diff beat out non diff: the average number of clicks is less than a click greater in the non diff case (27.25 for non diff and 26.5 for diff). Strictly on a speed basis, it is safe to say that the non diff approach leads to faster time using the visualization.

All users who answered questions about the scenarios well (either 3 out of 4 or 4 out of 4) indicated confidence levels of 3.75 or more which is surprisingly lower than the average confidence level for everyone of 3.9. This means that there must be several situations where a user thought they were correct, but were actually misled either by their visualization or understanding of selective undo.

Type of Visualization	Average Confidence Level	Questions Answered Correctly
diff	3.75	4
diff	4	4
diff	4	4
diff	3.75	3
diff	4.5	3
diff	3	2
diff	3.5	2
diff	2.75	1
non-diff	4	4
non-diff	4.25	4
non-diff	5	4
non-diff	3.75	3
non-diff	4.5	2
non-diff	3.5	2
non-diff	4.25	1
non-diff	4.5	1

Figure 1 : Table showing the confidence level of the participants vs the amount of questions they answered correctly.

## Methods

Participants were randomly divided into two groups ahead of time, either the diff or non diff group. The two groups were read the same instructions, put in the same scenarios, and asked the same questions. The only difference between the two groups was how the changes in a file were displayed. Participants were walked through a tutorial of how to navigate the program, and as instructed, they would type in the commands that were listed there. After they input the given commands, the users are asked to selectively undo an action using a provided visualization tool. Then, they were asked several questions regarding comprehension of their tasks.

## Conclusions

Based on our results, it appears that our users are able to navigate visualizations more quickly when showed data that is not in a diff format. Therefore, our work can be extended to other diff implementations: if a program views how quickly a user can interact with it as a strength, then it would be better to use a non diff approach. It's important to realize that diff is primarily used as a comparison tool between two files, and we attempted to extend that to comparisons between actions. Since participants were only tested on final results, it makes sense that they could better recollect data that was plainly displayed to them (the non diff format) rather than having to remember what was added or subtracted to a file.

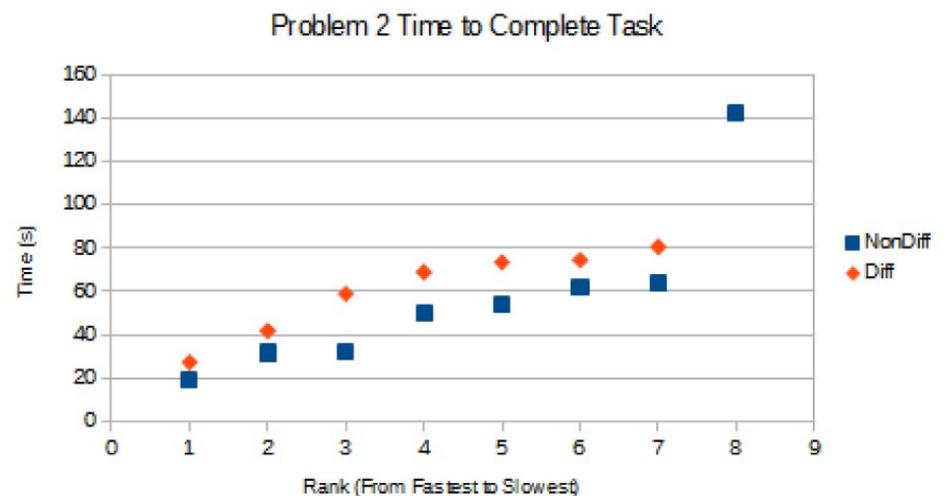


Figure 2 : Graph showing the completion time for one of the scenarios.