System Programming Assignment 1

**Announced: Wednesday October 2, 2013.**

**Due: Tuesday October 15, 2013 23:59.**

**[NAME]**

file\_merger - merge two different new version files and show conflict

**[SYNOPSIS]**

./file\_merger [old\_path] [new1\_path] [new2\_path] [output\_path]

**[DESCRIPTION]**

file\_merger can merge two new versions which they originate from a common file.

Ex1

|  |  |  |  |
| --- | --- | --- | --- |
| origin | version1 | version2 | output |
| 1 | 1  2 | 1  3 | 1  >>>>>>>>>> ex1\_2  2  ========== ex1\_3  3  <<<<<<<<<< |

The difference between origin and version1 is adding ‘2’, while the difference between origin and version2 is adding ‘3’. Therefore, they are conflicted on origin line 2, and show conflict region in output.

Ex2

|  |  |  |  |
| --- | --- | --- | --- |
| origin | version1 | version2 | output |
| 1  2  3 | 1  2 | 1  3 | 1  >>>>>>>>>> ex2\_2  2  ========== ex2\_3  3  <<<<<<<<<< |

The difference between origin and version1 is deleting ‘3’, while the difference between origin and version2 is deleting ‘2’. Therefore, they are conflicted on origin line 2 and line 3, and show conflict region in output.

Ex3

|  |  |  |  |
| --- | --- | --- | --- |
| origin | version1 | version2 | output |
| 1  2  3 | 2  2  3 | 3  2  3 | >>>>>>>>>> ex3\_2  2  ========== ex3\_3  3  <<<<<<<<<<  2  3 |

Ex4

|  |  |  |  |
| --- | --- | --- | --- |
| origin | version1 | version2 | output |
| 1  2  3  4  5 | 1  3  4  5 | 1  2  3  5 | 1  >>>>>>>>>> ex4\_2  3  4  ========== ex4\_3  2  3  <<<<<<<<<<  5 |

Ex5

|  |  |  |  |
| --- | --- | --- | --- |
| origin | version1 | version2 | output |
| 1  2  3 | 1  2 | 1  2 | 1  2  >>>>>>>>>> ex5\_2  ========== ex5\_3  <<<<<<<<<< |

**[ALGORITHM]**

1. apply suitable lock on each file to prevent data changed by another process.
2. calculate diff from origin for each new versions.
   1. build lcs table with taking a line as a unit.
   2. calculate diff alone lcs path

ex. Compare Ex2(origin) and Ex2(version1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0(common head) | 1 | 2 | 3 |
| 0(common head) | (XY, 1) | (X, 1) | (X, 1) | (X, 1) |
| 1 | (Y, 1) | (XY, 2) | (X, 2) | (X, 2) |
| 2 | (Y, 1) | (Y, 2) | (XY, 3) | (X, 3) |

You need a dummy common head to detect some change on first line. Therefore, you can find diff by go from (3, 2) to (0, 0) alone lcs path.

Going left means the line is deleted in new file, while going up means the line is added in new file.

1. merge two diff
   1. If there are two different operations on a line, adding different data or one adding one deleting, conflict happens. You should highlight conflict areas as example by “>>>>>>>>>> FILEPATH”, “========== FILEPATH” and “<<<<<<<<<<” as closure.
   2. In addition, to prevent excessively small fragment, we define diff block margin **3 lines**. Ex4 is for this case. Version1 in Ex4 has difference at line 2 of origin, while version2 has difference at line 4 of origin. 4 - 2 < 3, so we merge them.

Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| origin | version1 | version2 | output | final output |
| 1  2  3  4  5 | 1  3  4  5 | 1  2  3  5 | 1  >>>>>>>>>> ex4\_2  ========== ex4\_3  2  <<<<<<<<<<  3  >>>>>>>>>> ex4\_2  4  ========== ex4\_3  <<<<<<<<<<  5 | 1  >>>>>>>>>> ex4\_2  3  4  ========== ex4\_3  2  3  <<<<<<<<<<  5 |

**[SUBMISSION]**

Please follow the directory hierarchy TA provide to you. Just change the root directory name and pack with tar command.

**$ mv sp\_hw1b02XXXXXX sp\_hw1$(YOUR\_STU\_ID)**

**$ tar -zcvf sp\_hw1$(YOUR\_STU\_ID).tgz sp\_hw1$(YOUR\_STU\_ID)**

Then, submit to **ceiba** before deadline.

**[JUDGE]**

TA would run such command bellow to judge.

**$ tar -zxvf sp\_hw1$(YOUR\_STU\_ID).tgz**

**$ make -C ./sp\_hw1$(YOUR\_STU\_ID)/src clean && make -C ./sp\_hw1$(YOUR\_STU\_ID)/src**

**$ ./judge ./sp\_hw1$(YOUR\_STU\_ID)/bin**

**[GRADING]**

There are 8% in this homework. 1% for using file lock correctly, 4% for example test cases, other 3% for extra test cases.

Notice: Extra test cases may be **extremely big** such that you can’t hold all file in memory. Please handle this case.

**Late assignment**: Only the assignments submitted before the deadline will receive the full credit. 5% of your credits will be deducted for every single day delay.

**Plagiarism**: There is absolutely **NO** tolerance for plagiarism. Your codes will be checked by MOSS (<http://www.cs.berkeley.edu/~aiken/moss.html>).

**[CONTACT]**

email: [r01922084@ntu.edu.tw](mailto:r01922084@ntu.edu.tw)

or

p2: wmin0

Please use prefix [SP\_HW1] at the title. Thanks.