The `TIE` Processor

Klaus Guntermann
TH Darmstadt
Fachbereich Informatik
Institut für Theoretische Informatik

October 1989

1 Introduction

To install a program on a computer usually system dependent changes are needed. The systems of the `WEB` family allows to describe these changes in a so called change file. To incorporate the changes into the program `TANGLE` processes the original source file and inserts the changes into the proper places.

Sometimes it is good practice to develop a set of change files to be able to distinguish between different changes like program enhancements, bug fixes, system dependent changes and output device dependent modifications. Additionally this allows combinations of changes that can be used with a set of programs that share some common features.

The processors `TANGLE` and `WEAVE` can handle only one change file. The combination of change files is done best in another preprocessing step. `TIE` is able to create either a new master file or a single change file that comprises the effect of all change files.

Since `TIE` does not know anything about `WEB` programs but only about the structure of change files, it is possible to use this program for other programming languages as well. You just pretend that you want to create a new master file and `TIE` will go and insert all the changes from your change files.

2 Application

The current version of `TIE` accepts a master file and up to 9 change files. Because in general it is important to use the changes in their proper sequence the order of the files must be chosen carefully.

`TIE` must be called with at least 4 parameters:

1. either the option `-c` or `-m`. This option controls if a change file or a master file is created.

2. the name of the output file.
3. the name of the master file.
4. the name(s) of the change file(s). At least one must be present.

The multiple change files are used as if the single change files were merged into the master files step by step. Thus more than 9 change files might be incorporated by repeated application of `TIE`. Of course you may change the constant *max_file_index* in `TIE` which defines the maximum number of change files, too.

Please note that even when multiple change files are processed each of the files is read only once. For each of the change files and for the master file there is only one line in memory at a time (neglecting operating system buffering). Thus the storage requirement of `TIE` does not depend on the size of the change files or the size of a single changed section.

3 Example

To illustrate the actions `TIE` performs you may inspect the following example that exercises some of the border line cases.

Master	`CF1`	`CF2`	`CF3`
line 1	@x	@x	@x
line 2	line 2	line 1	changed line 4
line 3	line 3	changed line 2	@y
line 4	@y	changed line 3	final line 4
line 5	changed line 2	inserted line	@z
line 6	changed line 3	line 4	⟨EOF⟩
line 7	inserted line	@y	
line 8	@z	final line 2	
line 9		final line 3	
line 10	@x	changed line 4	
⟨EOF⟩	line 7	@z	
	@y		
	changed line 7	@x	
	@z	changed line 7	
	⟨EOF⟩	line 8	
		@y	
		final line 7	
		final line 8	
		@z	
		⟨EOF⟩	

Using these input files and running `TIE` twice to create a new master file and a new change file will result in the following output files:

MASTER	CHANGE
final line 2	@x
final line 3	line 1
final line 4	line 2
line 5	line 3
line 6	line 4
final line 7	@y
final line 8	final line 2
line 9	final line 3
line 10	final line 4
⟨EOF⟩	@z
	@x
	line 7
	line 8
	@y
	final line 7
	final line 8
	@z
	⟨EOF⟩