Published: 11/15/68

Identification

QED text editor
Ken Thompson

Introduction

QED is an interactive text editing facility. It is designed to operate on MULTICS. It has facilities that allow it to be used as a text editor and as a text manipulating language.

General Discussion

Regular Expressions

Regular expressions are defined rigorously as follows:

A character is a regular expression and matches that character.
"%" is a regular expression and matches the null regular expression (lambda).
"<name>" is a regular expression and matches the externally defined regular expression with the name "name". (See the enter request)
"^" is a regular expression and matches the null character at the beginning of a line.
"$" is a regular expression and matches the null character before the character <cr>. (Usually at the end of a line.)
"." is a regular expression and matches any character except <cr>.
"[abc]" is a regular expression and matches the character "a" or "b" or "c".
"[^abc]" is a regular expression and matches all graphic characters except "a", or "b" or "c".
A regular expression followed by "*" is a regular expression and matches any number of adjacent occurrences of the regular expression.
The expression consisting of two adjacent regular expressions is also a regular expression and matches adjacent occurrences of the regular expressions.
The expression consisting of two regular expressions separated by "¦" is also a regular expression and matches either of the regular expressions.
A regular expression in parentheses is also a regular expression and matches the regular expression. Parentheses are used to alter the above implied order or evaluation. ("ab¦cd" will match "ab" or "cd" while "a(b¦c)d" will match "abd" or "acd".)
Nothing else is a regular expression.

In subsequent discussion, "<regexp>" will indicate any legal regular expression.

Buffer Names

Buffer names are sequences of characters not including the character <cr>. A buffer name is represented by "(" followed by the characters composing the name followed by ")". If the buffer name is exactly one character in length, it need not be placed in parenthesis. Examples of buffer names follow: "()", "(xyz)", "x" (which is the same as "(x)"), "(a b c d e)".

In subsequent discussion, "<bufnam>" will indicate any legal buffer name.

Text Addressing

Lines in the current buffer may be addressed in the following ways:

1) By current line number.
   A decimal number not beginning with "0" or an octal number beginning with "0" is interpreted as current line number. The first line is numbered 1, the second 2, the tenth 10 or 012, etc. The current number of a line is its current position in the text buffer. This number may change during editing.

2) By absolute line number.
   The character "'" followed by a decimal number not beginning with "0" or followed by an octal number beginning with "0" is interpreted as an absolute line number. Absolute line numbers are assigned to the current buffer after a successful read request. These line numbers never change except after read requests. New lines created during editing have undefined absolute line numbers. The character "'" not followed by a digit causes a search for the first undefined absolute line number after the current line and cycling to the current line. If there is no line with the given absolute line number, an error will be noted.

3) By ".".
   The value of "." is the current line. This value is changed by most editor requests.

4) By "$".
   The value of "$" is the last line in the text buffer. This number may change during editing.

5) By context.
   The structure "/<regexp>/" causes a search for a text pattern that matches the <regexp>. The search begins at the line after the current line and cycles to the current line. If the search is successful, the value of "/<regexp>/" is the first line found containing the text pattern.

6) By additive combinations of 1-5.
   An address followed by "+" or "-" followed by another address is also an address. The value is obvious. Evaluation is done left to right. At no time during evaluation may an address

exceed the bounds of the number of lines in the text buffer. In all unambiguous cases, the "+" may be omitted. (".+4" is the same as ".4", but "5+2" is not "52". Note that "059" is the same as "05+9" is the same as "14".)

In subsequent discussion, "<addr>" will indicate any legal address.

Editor Input

Text Input

There are a number of QED requests that expect to be followed by literal text input. This text must be preceeded by a space or a carriage return. The text itself consists of an arbitrary string of characters that terminates in the character "\f". The "\f" character is not part of the literal text, but only serves to show the end of the text.

In subsequent discussion, "<text>" will indicate literal text input.

Editor Requests

In describing the editor requests, it is convienent to break up the requests into three categories.

1) Requests that require no addresses. If such a request is given with an address, an error will be noted. If the request is "X", this type of request will be represented by "X".

2) Requests that require one address. If such a request is given with no addresses, a default address is assumed. If such a request is given with one or more addresses, only the last address is used. If the request is "X" and the default address is "$", this type of request is represented by "($)X".

3) Requests that require two addresses. If such a request is given with no addresses, two default addresses are assumed. If such a request is given with one address, this address is used for both required addresses. If two or more addresses are given, only the last two address are used. If the request is "X" and the default addresses are "1,$", this request is represented by "(1,$)X".

The following is a list of QED's request repertoire.

A) ($)A<text>. Append
Text input is inserted after the addressed line. "." is set to the last line input. If no lines were input, "." is set to the addressed line.

B) B<bufnam>. Buffer
The buffer <bufnam> becomes the current buffer. If this buffer does not exist, it will be created. The previous value of "." for this buffer will become the value of ".". Initially, the current buffer is "0".

C) (.,.)C<text>. Change
The addressed lines are deleted and replaced by the text input. "." is set to the last line input. If no lines were input, "." is set to the line before the first line deleted. The addressed lines cannot cross line zero (circularly).

D) (.,.)D. Delete
The addressed lines are deleted. "." is set to the line after the last line deleted. The addressed lines cannot cross line zero (circularly).

E) E/<regexp>/<name>/. Enter
The <regexp> will be cataloged under the name. Any previously <regexp> with the same name will be deleted.

G) (1,$)G<request>/<regexp>/. Global
The QED request will be executed on all lines containing the regular expression. "." is not set by the global request, but may be set by the executed request. The following are the only legal constructions of the global request.

(1,$)GA<text>/<regexp>/	Append
(1,$)GC<text>/<regexp>/	Change
(1,$)GD/<regexp>/	Delete
(1,$)GI<text>/<regexp>/	Insert
(1,$)GM<bufnam>/<regexp>/	Move (appended)
(1,$)GP/<regexp>/	Print
(1,$)GS/<regexp>/<string>/<regexp>/	Substitute
(1,$)GY/<string>/<string>/	Transform
(1,$)G:/<regexp>/	Absolute line
(1,$)G=/<regexp>/	Current line

I) (.)I<text>. Insert
Text input is inserted before the line addressed. "." is set to the addressed line.

K) (1,$)K<cr>. Sort
The addressed lines are sorted by ascending ascii collating sequence. "." is carried with the sorting. ("." may change value, but will point at the same line.)

L) L<type><segnam><cr>. List
The segment will be read and printed. The value of "." is unchanged. Conversion of type <type> will be performed. The following types are recognized.

<sp> ascii

M) (.,.)M<bufnam>. Move
The addressed lines are removed from the current buffer. "." of the current buffer is set to the line after the last line removed. The entire contents of the named buffer are deleted. The removed lines are then placed in the named buffer and "." in the named buffer is set to the last of these lines. The named buffer can be the current buffer.

P) (.,.)P or <addr><cr>. Print
The addressed lines are printed. "." is set to the last line printed.

Q) Q<cr>. Quit
QED will return to its caller.

R) ($)R<type><segnam><cr>. Read
The segment will be read and appended after the addressed line. "." is set to the last line read. Conversion of <type> will be performed. The following types are recognized.

<sp> ascii

S) (.,.)S/<regexp>/<string>/. Substitute
Occurrences of the <regexp> in the addressed lines will be replaced by the string. "." is set to the last line substituted.

V) (1,$)V<request>/<regexp>/. Exclude
The QED request will be executed on all lines not containing the regular expression. "." is not set by the exclude request, but may be set by the executed request. The following are the only legal constructions of the exclude request.

(1,$)VA<text>/<regexp>/	Append
(1,$)VC<text>/<regexp>/	Change
(1,$)VD/<regexp>/	Delete
(1,$)VI<text>/<regexp>/	Insert
(1,$)VM<bufnam>/<regexp>/	Move (appended)
(1,$)VP/<regexp>/	Print
(1,$)VS/<regexp>/<string>/<regexp>/	Substitute
(1,$)VY/<string>/<string>/	Transform
(1,$)V:/<regexp>/	Absolute line
(1,$)V=/<regexp>/	Current line

W) (1,$)W<type><segnam><cr>. Write
The addressed lines will be written into the segmant. Conversion of <type> will be performed. The following types are recognized.

<sp> ascii

X) X. Status.

The name, value of ".", and length of the current buffer and all non-zero length auxiliary buffers are listed. The name of all named regular expressions are listed. (See the Enter request.) The value of "." is unchanged.

Y) (.,.)Y/<stringl>/<string2>/. Transform
Occurrences of characters in stringl in the addressed lines will be replaced by the corresponding character of string2. Stringl and string2 must be the same length. No character may appear twice in stringl. The value of "." is set to the last line transformed.

:) ($):. Absolute line
The absolute line number of the addressed line is printed. If the absolute line number is undefined, "?" will be typed. "." is set to the addressed line.

=) ($)=. Current line
The current line number of the addressed line is printed. "." is set to the addressed line.

<u>Usage</u>

While at shell level, type: "qed -pl- -p2-". The optional parameters are input and output segments to replace console input and console output. If an end-of-file is reached on the input segment, the input is switched to the console. If either of the parameters is "-", it means console.

<u>Comments</u>

All comments on (syntax¦semantics) of the (paper¦program) will be welcome.

Ken Thompson
Bell Telephone Labs. Inc.
Murray Hill, NJ
(201) 582-2394. (88-61-2394 from an MIT extension)
T269 8056 on CTSS

Requests

A)	(.) A<text>	Append
B)	B<bufnam>	Buffer
C)	(.,.)C<text>	Change
D)	(.,.)D	Delete
E)	E/<regexp>/<name>/	Enter
G)	(1,$)G<request>/<regexp>/	Global
	(1,$)GA<text>/<regexp>/	Append
	(1,$)GC<text>/<regexp>/	Change
	(1,$)GD/<regexp>/	Delete
	(1,$)GI<text>/<regexp>/	Insert
	(1,$)GM<bufnam>/<regexp>/	Move (appended)
	(1,$)GP/<regexp>/	Print
	(1,$)GS/<regexp>/<string>/<regexp>/	Substitute
	(1,$)GY/<string>/<string>/	Transform
	(1,$)G:/<regexp>/	Absolute line
	(1,$)G=/<regexp>/	Current line
I)	(.)I<text>	Insert
K)	(1,$)K<cr>	Sort
L)	L<segnam><cr>	List
M)	(.,.)M<bufnam>	Move
P)	(.,.)P or <addr><cr>	Print
Q)	Q<cr>	Quit
R)	($)R<type><segnam><cr>	Read
S)	(.,.)S/<regexp>/<string>	Substitute
V)	(.,.)V<request>/<regexp>/	Exclude
	(1,$)VA<text>/<regexp>/	Append
	(1,$)VC<text>/<regexp>/	Change
	(1,$)VD/<regexp>/	Delete
	(1,$)VI<text>/<regexp>/	Insert
	(1,$)VM<bufnam>/<regexp>/	Move (appended)
	(1,$)VP/<regexp>/	Print
	(1,$)VS/<regexp>/<string>/<regexp>/	Substitute
	(1,$)VY/<string>/<string>/	Transform
	(1,$)V:/<regexp>/	Absolute line
	(1,$)V=/<regexp>/	Current line
W)	(1,$)W<type><segnam><cr>	Write
X)	X	Status
Y)	(.,.)Y/<string>/<string>/	Transform
:)	($):	Absolute line
=)	($)=	Current line

Diagnostics

01 RE search failed.
02 Unrecognized command or address.
03 RE syntax error.
04 Address syntax error.
05 Address wrap around.
06 Address out of buffer.
07 Abs line search failed.
08 File system error.
09 Command syntax error.

16 Unknown RE type.
17 Out of memory.
18 Overflow on store.
19 Past EOF on store.
20 Free of block 0.

(16-20 are fatal.)