1101

Bell Laboratories **Cover Sheet for Technical Memorandum**

The information contained herein is for the use of employees of Bell Laboratories and is not for publication. (See GEI 13.9-3)

Title- **A Regular Expression Matcher for the 'C' language**

Date- **June 2, 1976**

TM- **76-9141-5**

Other Keywords- **UNIX**
Language C
String Processing

Author	Location	Extension	Charging Case- 39380-000
A. L. Sabsevitz	**PY 1A-139**	**7311**	Filing Case- 39373-99

ABSTRACT

As the Programmer's Workbench (PWB) community has grown, so has the requirement for string processing facilities. This memo describes a primitive regular expression matching facility written for language C. Routines have been provided to compile regular expressions and to apply such compiled expressions to string subjects. Substrings may be returned, and pointers delimiting the matched portion of the subject string are available. These routines, structured after analogous routines internal to the UNIX editor, ed(I), should greatly facilitate string processing operations in programs written in language C. These routines have been used successfully in numerous 'C' programs including versions of grep(I) and ed(I).

Pages Text 1	Other 3	Total 4
No. Figures 0	No. Tables 0	No. Refs. 0

E-1932-C (6-73) SEE REVERSE SIDE FOR DISTRIBUTION

TM-76-9141-5
WALKER,ELIZABETH A CM
MH2C548 06/10/76
SUBJECT MATCH UNSU

BELL TELEPHONE LABORATORIES, INC.

DISTRIBUTION
(REFER SEI 13.9-3)

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES

OFFICIAL FILE COPY
PLUS ONE COPY FOR
EACH ADDITIONAL FILING
CASE REFERENCED

DATE FILE COPY
(FORM E-1328)

10 REFERENCE COPIES

ALBERTS,BARBARA A
ALLEN,R C
ARNOLD,S L
ARTHURS,E
ARTIS,H P
AVIL,H J
BACCASH,JEANNE M
BAYER,D L
BEL BRUNO,KATHLEEN A
BERGLAND,G D
BERTH,R P
BIANCHI,M H
BILLINGTON,MARJORIE J
BIRCHALL,R H
BIREN,IRMA B
BLUM,MARION
BLY,JOE
BOCKUS,ROBERT J
BOURNE,STEPHEN R
BOWRA,JAMES W
BOWYER,L RAY
BOYCE,W M
BOYLE,GERALD C
BOYLE,J T,JR
BROWN,C W
BROWN,W R
BROWN,W STANLEY
BRUNO,BEVERLY A
BUCHANAN,D N E
BURROWS,THOMAS A
BZOWY,D
CABLE,GORDON G,JR
CALLAHAN,R L
CANADAY,RUDD H
CARDOZA,WAYNE M
CARLSON,HELEN V
CARPENTIER,A M
CARRAN,JOHN H
CARR,DAVID C
CATO,H E
CHAFFEE,N F
CHENOWETH,G R
CHEN,STEPHEN
CHODROW,M M
CHRIST,C W,JR
CLAYTON,D P
COLLINS,J P
COLLUM,D JOHN

COMPLETE MEMORANDUM TO

COMMRADE,GERALDINE
COOK,T J
COREY,D A
COSTELLO,PETER E
COSTON,W P
CRANE,R P
CROWE,MARGARET M
CRUME,L L
CRUM,T A
CUNNINGHAM,ELIZABETH T
CUNNINGHAM,S J,JR
DAVIDSON,CHARLES LEWIS
DAVIS,D R
DAVIS,R DREW
DE GRAAF,D A
DE JAGER,D S
DE PAOLA,THELMA T
DE SIMONE,LESLIE I
DEVLIN,N V
DOLOTTA,T A
DOWDEN,IRIS S
DOWD,P G
DRUMMOND,R E
DUDLEY,ELIZABETH H
DUNN,J C
DWYER,T J
ERRICHIELLO,PHILIP M
ESTES,VIRGINIA DANIELLE
ESTOCK,RICHARD G
EVENSON,E K
FASCIANO,V
FEDER,J
FELDMAN,STUART I
FELS,ALLEN M
FIORE,RHODA J
FLANDRENA,R
FLEISCHER,H I
FLYNN,JOHN F
FLYNN,MARY L
FOLEY,G
FONTENOT,SHARON M
FORTNEY,V J
FOX,A
FOY,ANGELENE
FRANK,AMALIE J
FRANK,H GREGORY
FRASER,A G
FREEMAN,K G
FUCHS,EDWARD
GANNON,T F
GEERS,T J,JR
GERARD,A
GERE,E A
GERGOWITZ,E B
GEYLING,F T
GILBERT,G W
GITHENS,J A
GLASER,W A
GLASSER,ALAN L
GODWIN,R E

COMPLETE MEMORANDUM TO

GOGUEN,N H
GRAHAM,R L
GRANDNER,W J
GRANT,GEORGE E
GRAVEMAN,R F
GRAVES,W G,2ND
GREENBAUM,HOWARD J
GREENWALSH,H WAIN
GRESHAM,J P
GROEGER,JANET R
GROSS,ARTHUR G
GRUCHOWSKI,M P
GUIDI,PIER V
HAIGHT,R C
HALPIN,T
HAMILTON,LINDA L
<HAMILTON,PATRICIA A
HARKNESS,CAROL J
HELD,R W
HESS,B E
HOALST,B C
HOLTMAN,JAMES P
HOOKER,J W
HOPKIN,JAMES W,JR
HOWARD,L A,JR
HO,DON T
HUPKA,FLORENCE
HYMAN,BRUCE
IFFLAND,FREDERICK C
IPPOLITI,O D
IPPOLITO,N P
IRVINE,M M
IVIE,EVAN L
JABLONSKI,GRAZYNA C
JACKOWSKI,D J
JENSEN,PAUL D
JOHNSON,STEPHEN C
JONES,B R
JONES,M B
JONES,S E
JONES,VIRGINIA L
KAISER,J F
KAMINSKY,M L
KAPLAN,J J
KAPLAN,MICHAEL M
KARL,J W,3RD
KAUFELD,J C,JR
KAUFMAN,LARRY S
KAYEL,R G
KEATING,NANCY A
KEESE,W M
KEMP,CHARLES W
KESSELL,D L
<KETCHLEDGE,B A
KETTERER,BARBARA A
KEVORKIAN,D E
KLAPMAN,RICHARD N
KNOX,BEVERLY S
KOETHER,MARY E
KOLB,J

COMPLETE MEMORANDUM TO

KOZAREK,J F
LACY,J V
LANDOLINA,W C,JR
LAUTENBACH,DEBORAH A
LAWSON,D A
LEHRMAN,WILLIAM
LESSER,P V
LEVENBERG,R A
LIND,R O
LINK,G L
LIST,W H S
LOIKITS,E A
LOMUTO,N
LONG,P F
LORENC,A
LUDERER,GOTTFRIED W R
LUKACS,M E
MACHOL,R E,JR
MADOR,H P
MAJERNIK,JOHN F
MALLON,J M
MANOCHIO,L J
MARANZANO,J F
MAROWITZ,JAY E
MASHEY,J R
MC GILL,R
MC GONEGAL,CAROL A
MC MILLAN,B D
MC MILLAN,L G
MC MILLAN,W F
MCILROY,M DOUGLAS
MEISER,BLAIR R
MENNINGER,R E
METZLER,HELEN M
MIGLIORE,SANDRA A
MIKOLAJEK,J M
MILLSPAUGH,JOHN A,JR
MILLS,ARLINE D
MORGAN,SAMUEL P
MORRIS,ROBERT
MUHA,RALPH
MUI,LILY C
NAU,CARL DAVID
NAZIF,Z
NEFF,R B,3RD
NEY,LESLIE
NIMTZ,ROBERT J
NOLL,LAURA W
NOWITZ,D A
O SHEA,W T
O'BRYAN,H M,JR
O'NEILL,D M
OSOLINIK,F J,3RD
OSSANNA,J F,JR
PASE,D F
PALUCK,MARY BETH
PAN,J W
PARA,PAUL S
PATEL,C K N
PATTON,G C

COMPLETE MEMORANDUM TO

PEARLMAN,BERTRAM
PECK,W DOUGLAS
PERDUE,ROBERT J
PEREZ,CATHERINE D
PERNESKI,A J
PETERSON,RALPH W
PETERSON,T G
PETSCHENIK,N H
PHILLIPS,S J
PILLA,M A
PINSON,ELLIOT N
PISKORIK,ELAINE M
PLODKOWSKI,R J
POLICH,K K
PRIESTER,R W
PRINS,G C
PRYOR,ROGER W
PUERLING,BRUCE W
QUINN,MARGARET E
RAACK,G A
RATTA,G A
REGO,R A
REMS,S B
RHODES,STEVE
RIDDLE,GUY G
RITACCO,J E
ROBBINS,MARILYN F
ROBERTS,CHARLES S
ROCHKIND,M M
ROCHKIND,MARC J
RODRIGUEZ,ERNESTO J
ROJEK,RICHARD C
ROMITO,LETITIA J
ROM,DOUGLAS B
ROSENBLUM,M J
ROSLER,LAWRENCE
ROVEGNO,HELEN D
RUGGIERO,G N
RYAN,T R
RYBA,DENNIS A
SABSEVITZ,A L
SALMON,RUTH L
SATZ,LAWRENCE R
SCHNITZER,M D
SCOTT,FLOYD E
SEEBACH,R J
SEMAN,JOHN A
SEMMELMAN,C L
SHORTER,J W
SIDOR,DONALD J
SIX,F B
SMITH,B L
SMITH,DALE W
SMITH,KIME H,JR
SMITH,LESLIE T
SOUTHERN,M J
SPIRES,R J
STARRICK,KATHLEEN K
STETTER,V G
STEVENSON,D E
...
347 TOTAL

+ NAMED BY AUTHOR > CITED AS REFERENCE < REQUESTED BY READER (NAMES WITHOUT PREFIX WERE SELECTED USING THE AUTHOR'S SUBJECT OR ORGANIZATIONAL SPECIFICATION AS GIVEN BELOW)

MERCURY SPECIFICATION.....

COMPLETE MEMO TO:
COPLTP = COMPUTING/PROGRAMMING LANGUAGES/TEXT PROCESSING, EDITING
UNED# = UNIX/EDITING AND DOCUMENT PREPARATION
UNPL# = UNIX/PROGRAMMING LANGUAGES
UNSU# = UNIX/SERVICE,UTILITY PROGRAMS

COVER SHEET TO:

HO CORRESPONDENCE FILES
HO 5C101

TM-76-9141-5
TOTAL PAGES

PLEASE SEND A COMPLETE COPY TO THE ADDRESS SHOWN ON THE OTHER SIDE.

TO GET A COMPLETE COPY:

1. BE SURE YOUR CORRECT ADDRESS IS GIVEN ON THE OTHER SIDE.
2. FOLD THIS SHEET IN HALF WITH THIS SIDE OUT AND STAPLE.
3. CIRCLE THE ADDRESS AT RIGHT. USE NO ENVELOPE.

Bell Laboratories

subject: **A Regular Expression Matcher for the 'C' language**
Case: 39380-000
File: 39373-99

date: **June 2, 1976**

from: **A. L. Sabsevitz**
PY 9141
1A-139 x7311

TM: **76-9141-5**

MEMORANDUM FOR FILE

Two 'C' routines have been written which give the 'C' language some primitive regular expression matching capabilities. They will compile an arbitrary regular expression and apply it to a designated string. The structure for these routines was extracted from the UNIX editor, ed(I). They were then extensively modified and expanded with several new features. These routines have been used successfully in numerous 'C' programs including versions of grep(I) and ed(I).

REGCMP() takes a regular expression and generates a compiled form of it. REGEX() applies this compiled vector against a string. Information such as where the match started and where it completed is available to the caller. Substrings of the matched string may be returned in user specified variables.

A command, REGCMP, has been written for those applications which use invariant regular expressions. REGCMP will compile regular expressions into files which may later be included into a 'C' compilation. This precludes the need to compile the expressions at execution time.

Attached are two manual pages which describe the 'C' system command and routines.

A. L. Sabsevitz

PY-9141-ALS-als

Atts.
REGCMP (PWB I)
REGEX (PWB III)

NAME

regcmp – regular expression compile

SYNOPSIS

regcmp [–] file1 file2 ...

DESCRIPTION

Regcmp, in most cases, precludes the need for calling *Regcmp(PWB III)* from *'C'* programs. This saves on both execution time and program size. The command *regcmp* compiles the regular expressions in *file* and places the output in *file.i*. If the – option is used, the output will be placed in *file.c*. The format of entries in *file* is a name (*C* variable) followed by one or more blanks followed by a regular expression enclosed in double quotes. The output of *regcmp* is *C* source. Compiled regular expressions are represented as extern char vectors. *File.i* files may thus be *included* into *C* programs, or *file.c* files may be compiled and later loaded. In the *C* program which uses the *regcmp* output, *regex(abc,line)* will apply the regular expression named *abc* to *line*. Diagnostics are self-explanatory.

Example:

```
name    "([A–Za–z][A–Za–z0–9_]*)$0"
telno   "\\({0,1}([2–9][01][1–9])$0\\){0,1} *"
        "([2–9][0–9]{2})$1[ –]{0,1}"
        "([0–9]{4})$2"
```

SEE ALSO

regex(PWB III)

AUTHOR

Arthur Sabsevitz

NAME

regcmp – regular expression compile
regex – regular expression execute

SYNOPSIS

char *regcmp(string1[,string2,...],0);
char *string1, *string2,...;

char *regex(re,subject[,ret0,...]);
char *re, *subject, *ret0 ,...;

DESCRIPTION

Regcmp compiles a regular expression and returns a pointer to the compiled form. *Alloc(I)* is used to create space for the vector. It is the user's responsibility to free unneeded space so allocated. A zero return from *regcmp* indicates an incorrect argument. *Regcmp(PWB I)* has been written to generally preclude the need for this routine at execution time.
Regex executes a compiled pattern against the subject string. Additional arguments are passed to receive values back. *Regex* returns zero on failure or a pointer to the next unmatched character on success. A global character pointer *__loc1* points to where the match began. *Regcmp* and *regex* were mostly borrowed from the editor, *ed(I)*; however, the syntax and semantics have been changed slightly.

symbols	*meaning*
[ ] * . ^	These symbols retain their current meaning.
$	Matches the end of the string, '\n' matches the newline.
–	Within brackets the minus means *through*. For example, [a–z] is equivalent to [abcd ... xyz]. The '–' can appear as itself only if used as the last or first character. For example, the character class expression [ ]–] matches the characters ']' and '–'.
+	A regular expression followed by '+' means *one or more times*. For example, [0–9]+ is equivalent to [0–9][0–9]*
{*m*} {*m*,} {*m*,*l*}	Integer values enclosed in { } indicate the number of times the preceeding regular expression is to be applied. *m* is the minimum number and *l* is a number, less than 256, which is the maximum. If only *m* is present,e.g. {m}, *m* indicates the exact number of times the regular expression is to be applied. {m,} is analogous to {m,infinity}. The plus ('+') and star ('*') operations are equivalent to {1,} and {0,} respectively.
(...)$*n*	The value of the enclosed regular expression is to be returned. The value will be stored in the *(n+1)th* argument following the subject argument. At present, at most ten enclosed regular expressions are allowed. *Regex* makes its assignments unconditionally.
(...)	Parentheses are used for grouping. An operator, e.g. *,+,{ }, can work on a single character or a regular expression enclosed in parenthesis. For example, (a*(cb+)*)$0.

Of necessity, all the above defined symbols are special. They must, therefore, be escaped to be used as themselves.

Example 1:

```
char *cursor, *newcursor, *ptr;
...
newcursor = regex((ptr=regcmp("^\n",0)),cursor);
free(ptr);
```

This example will match a leading newline in the subject string pointed at by cursor.

Example 2:

```
char ret0[9];
char *newcursor, *name;
...
name = regcmp("([A-Za-z][A-za-z0-9_]{0,7})$0",0);
newcursor = regex(name,"123Testing321",ret0);
```

This example will match through the string "Testing3" and will return the address of the character after the last matched character (cursor+11). The string "Testing3" will be copied to the character array *ret0.*

Example 3:

```
#include "file.i"
char *string, *newcursor;
...
newcursor = regex(name,string);
```

This example applies a precompiled regular expression in *file.i* (see *regcmp(PWB I)*) against *string.*

SEE ALSO

regcmp(PWB I),ed(I),alloc(III),free(III)

BUGS

The user program may run out of memory if regcmp() is called iteratively without freeing the vectors no longer required. The following user-supplied replacement for alloc(III) re-uses the same vector saving time and space.

```
/* user's program */
        ...
alloc(n) {
static int rebuf[256];
        return &rebuf;
}
free(ptr)
char *ptr;
{}
```

AUTHOR

Arthur Sabsevitz