Software Tools Communications

NUMBER 5 APRIL 1981

--- NEXT SOFTWARE TOOLS MEETING ---

The next Software Tools Users Group Meeting will be held on June 23, 1981 at the Joe C. Thompson Conference Center at the University of Texas in Austin. The registration fees will be:

	Regular	Student
Pre-registration	$25	$10
On-site registration	$35	$15

Rooms have been reserved at the Villa Capri Motor Hotel and the Austin Hilton. The local arrangements chairman is:

David Phillips
Computation Center
UT Austin
Austin TX 78712

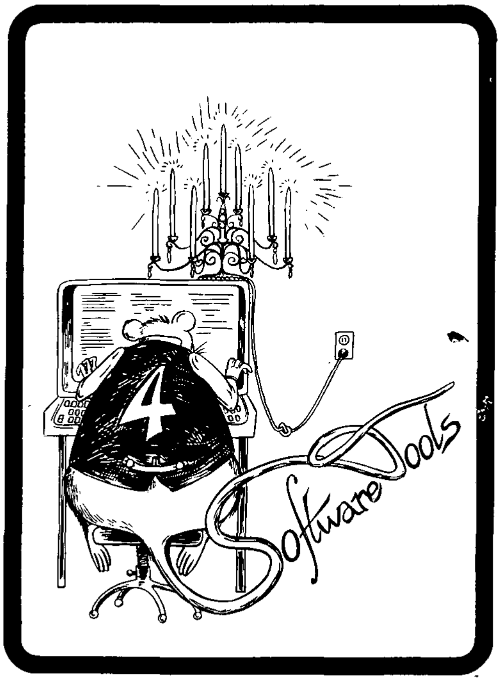
512-471-3241
DAVID@UTEXAS-11

100 word abstracts for presentations should be submitted by June 1, 1981 to:

Joseph S. D. Yao
Science Applications, Inc.
1710 Goodridge Drive
McLean, VA 22102

This work was partially supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research U.S. Department of Energy under contract W-7405-ENG-48.

PUB-402 4-81/1500



PUB 402

--- NEW ADDRESS FOR SOFTWARE TOOLS GROUP ---

We now have an address. Please send correspondence such as change-of-address and newsletter contributions to:

Software Tools Users Group
242-1259 El Camino Real
Menlo Park, CA 94025

Please mark correspondence clearly on the outside indicating its contents with words such as

"Attention: Change of Address".

--- SOFTWARE TOOLS BASIC TAPE DISTRIBUTION ---

The new basic distribution tape is now available. Much work has gone into producing it. Many thanks to Allen Akin, Debbie Scherrer, and David Hanson who have made the tape possible. The tape is available for $35.00. An order form is included in this newsletter.

--- NON-STANDARD TAPE DISTRIBUTIONS ---

An assortment of tapes representing several versions of the software tools are available. Several sources other than the Software Tools Users Group distribute machine-specific tools implementations, including:

DEC-10 TOPS10; CDC Cyber; Cray:
Professor David Hanson
Department of Computer Science
University of Arizona
Tucson, AZ 85721
602-626-3617

PRIME:
Allen Akin
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332
404-894-4465

Z80 CP/M:
Philip Scherrer
Unicorn Systems
30261 Palomares Road
Castro Valley, CA 94546
415-881-4490

The original Kernighan-Plauger tape:
Addison-Wesley Publishing Company
Reading, MA 01867
617-944-3700

For other systems, consult the List of Machines and Operating Systems available from the users group liaison. (You may write to our address given on page one, attention "Users Group Liason".)

--- SAN FRANCISCO SOFTWARE TOOLS USERS GROUP MEETING MINUTES ---

The Software Tools Users Group Conference was held on January 20, 1981. The meeting was attended by about 400 people. Like previous meetings, the meeting was held preceding the USENIX Conference. The summary below was taken from notes supplied by Jon Powell of Lawrence Berkeley Laboratory and Jim Woods of NASA/AMES. (Thanks for sending in such helpful meeting notes.)

INTRODUCTION

The Next Generation of Software Tools - Anthony I. Wasserman, UC San Francisco and UC Berkeley

The keynote presentation made the point that there is a need to develop a new generation of software tools. Present generation tools do not capture information on the software development process, do not support a systematic development methodology, and do not support the entire development life cycle. Current generation tools do not give the programmer enough automated support. Properly designed software tools can improve programmer productivity and help to reduce the high cost of software. Properly designed tools have singularity of purpose, ease of use, are self-documenting, consistent, adaptable, intelligent, support the entire software life cycle, and support software management. Specific features that next generation tools should have include provision of:

1) Databases for tools to customize usage (have both novice and wizard modes of operation).

2) Well-designed human interfaces using other than keyboard devices.

3) Smalltalk-like simultaneous window display features.

4) Database tools for source code, documentation, control files, and test data for the entire software life cycle.

5) Ability to capture information about design decisions and the software development process in databases for later study.

Wasserman concluded with an appeal for a powerful personal development system utilizing state-of-the-art 32 bit microprocessors, Winchester disks, Smalltalk style windowing, and high resolution graphics.

SESSION 1 - **The Virtual Operating Systems Approach - Dennis Hall, Lawrence Berkeley Laboratory, Chair**

Virtual Operating Systems - Dennis Hall, Lawrence Berkeley Laboratory

Dennis Hall presented the Virtual Operating System approach to the problem of the proliferation of operating systems. Dennis characterized the design philosophy behind the Virtual Operating System approach by describing the following design decisions.

1) Choose a language (C and Ratfor) and stick with it.

2) Design and develop utilities from the top down.

3) Design and develop a virtual machine from the bottom up.

4) Copy success (notably Kernighan and Plauger, the UNIX system, and other Software Tools developers).

5) Innovate with caution.

Dennis then went on to describe some of the tools available in the public domain using the Virtual Operating System approach. These tools included syntax checkers, cross referencers, translators for EFL (Bell Labs Extended Fortran), ICON (Snobol replacement), Ratfor to various high-level language translators, and a variety of scientific and engineering applications programs.

Virtual Aethers - Joe Sventek, Lawrence Berkeley Laboratory

The concept of a "virtual aether" was described. A "virtual aether" is basically a software implementation of a hardware Ethernet, and is generally embedded in an operating system kernel to process message traffic both within the system and outside of the system. Operations on "virtual aethers" include creating, accessing, de-accessing, depositing a message, receiving the next message, and fetching message sub-addresses. Virtual Aethers are used as a testbed for network protocols, and for system-independent implementations of servers, daemons, and pipes. Joe also outlined a "virtual aether" network mailer. Versions of this software are planned first for RSX-11M (Spring DECUS Tape), to be followed by VMS and VAX UNIX. Send a stamped self-addressed envelope to Joe for a paper preprint:

Joseph Sventek
Computer Science & Applied Mathematics
Lawrence Berkeley Laboratory
Berkeley, CA 94720

Portable Layers of Graphics Software - Mike O'Dell, Lawrence Berkeley Laboratory

Mike dealt with the question of integrating computer graphics into Software Tools. He made the point that since a computer graphics application must deal not only with the host on which it runs, but also with the graphic peripheral, portability in computer graphics is very hard to achieve. Mike saw a need for four types of computer graphics.

1) Business graphics (e.g., pie-charts, histograms)

2) Data presentation and analysis (e.g., census, scientific data)

3) Add-ons to existing programs (e.g., plotting filters)

4) High performance color graphics.

Mike went on to warn users to be wary of the ANSI X3H3 standards effort, and mentioned a specific need for illustration editors.

IEEE Micro Operating Systems Interface Standards - Sam Kirk, TRW/VIDAR

Sam described the IEEE efforts to develop a standard for a micro operating system interface. He emphasized the need to evaluate existing interfaces such as UNIX and the Virtual Operating System Primitives. He specifically mentioned that the operating system standards activity is critical for the more powerful and functional 16 bit microcomputers.

SESSION 2 - **New and Enhanced Utilities - David Martin, Hughes Aircraft, Chair**

Toward a More Interactive Shell - David Martin, Hughes Aircraft

David has extended the Lawrence Berkeley VAX shell to have a TENEX flavor. Specifically, he has added features to reduce keystrokes, reduce filename spelling errors, do command editing and resubmittal, and to provide more user feedback. Future plans are to integrate the shell and VI-style screen editing, add MESA-like file mapping primitives, and incorporate LISP features. David is also working on a portable ADA environment to be based on the Software Tools.

A Virtual Terminal Handler - Allen Akin, Georgia Institute of Technology

Allen described the "Virtual Terminal Handler" approach to solving the problem of interfacing a set of terminal dependent tools (such as screen editors) to a large set of terminals. To solve this problem, a virtual terminal handler was designed to :

1) Create and isolate a terminal screen buffer for handler-only read / write access.

2) Interpret a terminal description file.

3) Build a finite-state machine for character mode input that is general enough to handle terminal function keys.

4) Track changes to each row of the terminal screen buffer to minimize extra cursor movement.

5) Pass exceptions back to the user program.

The virtual terminal handler has been used for indexing ACM student chapter publications, multi-user Star Trek, and various other applications. The code is 2000 lines of Georgia Tech extended Ratfor.

Product Development and the Software Tools - Edward G. Happ, Interactive Data Corporation

Using Ratfor, Ed has built a 20000 line stock portfolio package based on Barr Rosenberg's model. The package is used to quantify investment risk to subscribers of the service. The software was written to run under CMS on twin Amdahl 470-V8s supporting 600 users. Since the system was written to be easy to use by non-programmers, it employs menus and wordy error reporting. A walk-through of sample output showed an investor recently gaining a 25% return on investment in three months, after selecting high technology and banking stocks. Ed felt that Ratfor was a definite aid to productivity in completing the project.

SESSION 3 - **Implementation Issues - Joe Sventek - Lawrence Berkeley Laboratory, Chair**

LTSS, CTSS on CDC7600, CRAY-1 - **Margaret Hug,** Los Alamos Scientific Laboratory

Margaret described some horrendous compatability problems in using the tools and Ratfor on Cray and CDC machines. Apparently the character handling on these machines is very awkward. She also described an environment with Fortran packages involving several thousand lines of code that must not break when a new version of Ratfor is installed.

CP/M on Z80 - **Philip Scherrer, Unicorn Systems**

The seemingly-impossible was reported in this talk. This brave soul shoehorned Ratfor and the tools into a Z80 system with floppy disks running CP/M One of the main problems was getting the source code from 9-track tape to floppy through the use of a 300 baud modem. CP/M was described as being closer to a program loader than an operating system.

MPX on SEL - **Walt Donovan, NASA/Ames Research Center**

Walt described various software problems that he ran into in bringing up the tools under MPX on a 32-bit SEL. Some of the software problems mentioned include fixed-size files, FORTRAN data statement handling, a bug in the FORTRAN compiler where lowercase "w"s anywhere in the program text destroy the code, and a problem where the system logs you off if you have not typed anything in 30 seconds. He also described the "Bermuda Triangle Problem" where every so often, the CPU stops with a triangle light pattern on the CPU display. Walt is not going to implement a shell since MPX does not support the notion of a "process".

RSX-11M on PDP11, VMS on VAX-11 - **Joe Sventek, Lawrence Berkeley Laboratory**

Joe reported briefly on directory and raw mode I/O primitive extensions. Implementation difficulties included multiple file types and a primitive process structure.

SESSION 4 - **Future Directions - Panel Discussion**

Allen Akin, Georgia Institute of Technology
Skip Egdorf, Los Alamos Scientific Laboratory
Mike O'Dell, Lawrence Berkeley Laboratory
Debbie Scherrer, Lawrence Berkeley Laboratory

This session involved a discussion of a diverse set of topics. Skip wanted a users group as active as the FORTH users group. Debbie said that the purpose of the users group is to coordinate distribution of a uniform program development environment to users of a diverse set of vendor-supplied systems, and to act as a clearing house for improvements and additions to this standard environment. The users group would accomplish its mission by satisfying this goal. Mike stated that bad operating systems are still being designed, and suggested that programmers not work for employers with poor machines and software. B. Upshaw (LBL) suggested that user portability is actually more important than program portability. Allen claimed that it is too early for standards. His statement elicited an audience comment that at least the ASCII character set and RS232 interface standards are useful, even if terminal installers only use pins 2, 3, and 7.

SESSION 5 - **Special Interest Group Meetings**

The final session was a discussion among people interested in implementing the tools on various machines and operating systems, including DEC, IBM, CDC, DataGeneral, Hewlett-Packard, Modcomp, Xerox, Honeywell, Univac, Micros, and others.

--- LIST OF REGISTRANTS FROM SAN FRANCISCO MEETING ---

By popular request, a list of registrants at the San Francisco Software Tools/Usenix meetings is available to attendees who did not pick theirs up. Send a self-addressed, stamped ($0.36), 8.5 X 11 manila envelope to:

Tom Ferrin
School of Pharmacy
Room 926-S
University of California
San Francisco, CA 94143

--- SOFTWARE TOOLS USERS GROUP ADMINISTRATIVE MEETING ---

An administrative meeting was held among principals of the Software Tools Users Group after the main San Francisco Meeting. In order to accomplish an objective that the users group become self-supporting, several decisions were made:

1) An executive board was appointed.

2) Membership dues will have to be charged to cover the cost of the newsletter (details on membership to be provided in a future issue of the **Communications**).

3) A minimal fee will be charged for copies of the basic tape.

More information on these decisions will be available in the future newsletters.

The New Basic Tools Tape

Allen Akin
Georgia Institute of Technology

The Users Group Basic Tools Tape is now available. This note outlines its contents and highlights some differences. between it and the original Software Tools code. An order form is included in this newsletter.

Overview

The Software Tools Bootstrap

The first eleven files on the tape are designed to assist the person implementing Software Tools on a machine with only Fortran and basic operating system support. These files include portable primitives using Fortran I/O, a bootstrap version of the Ratfor preprocessor in Fortran, programs for testing several important implementation features, four useful major tools, and Debbie Scherrer's famous Cookbook.

The Cookbook fully describes the tape's contents, details the functions and interfaces of the primitives the implementor must develop, and offers good advice on the entire bootstrapping process.

The bootstrap preprocessor handles a subset of the Ratfor language, but a large enough subset to compile the full preprocessor found later on the tape.

Library routines on the fourth file of the tape provide utility functions like pattern matching, string handling, and input/output. The I/O routines make use of Fortran I/O statements, and have been shown to work with only minor changes on a number of systems.

Test programs are provided to verify proper character code translation, command line parsing, and file handling.

A full Ratfor preprocessor, text formatter, archival storage manager, and text editor round out the bootstrapping section of the tape. Together with the primitives and library routines, these programs provide the core of a fully operational Software Tools installation.

Ratfor Preprocessor

The Ratfor preprocessor has been reorganized internally, resulting in slightly faster operation and (hopefully) improved modifiability. The quality of the output code has also been improved.

The standard Fortran-style DO statement is now accepted (although 'break' and 'next' statements may not be used within such DO-loops).

The 'return' statement may have a parenthesized expression as a parameter. (In FUNCTION subprograms, this construct assigns the value of the expression to the function name and causes an immediate return.)

The 'break' and 'next' statements may be followed by an integer specifying the number of levels of nested loops to affect. For example, "break 1" (which is identical to plain "break") terminates the innermost loop containing the 'break' statement; "next 2" terminates the innermost loop containing the 'next' statement and forces the next iteration of the smallest loop containing that innermost loop.

A 'switch' statement (with syntax similar to the 'switch' statement of C) has been added to provide a multi-way branch capability. Unlike the C 'switch', control does not fall through from case to case. A warning: the 'switch' statement causes an undeclared variable with a name of the form "I[0-9]*" to be used. Thus, code containing 'switch' statements will not work if the subsequent Fortran compilation includes a check for undeclared variables or if the program contains an IMPLICIT declaration that changes the default typing for variables beginning with the letter "I".

Variable names that are longer than 6 characters and those that contain underscores are "squashed" into legal 6-character Fortran variable names, with the first character left unchanged so as not to disturb implicit typing. Variable names may be up to 100 characters long; all characters are significant.

A file containing the general-purpose Software Tools macro definitions is automatically opened and processed before each user program.

Ratfor now handles the full range of macro preprocessor capabilities, including macros with arguments, special builtin functions, arithmetic expression evaluation, conditional preprocessing, and immediate and deferred replacement text scanning.

The 'string' declaration has been added to permit simple initialization of unpacked, EOS-terminated character strings.

Integer constants in radices other than 10 may be specified with the syntax "<base>%<constant_in_base>".

Special text can be passed through the preprocessor untouched by beginning each line of such text with a percent sign (%).

Quoted filenames may be used in 'include' statements, to prevent difficulties caused by filenames containing special characters.

Text Formatter

Several significant additions have been made to the text formatter.

The 'bd' request may be used to boldface output text. It works exactly like the 'ul' (underline) and 'cu' (continuous underline) requests.

The 'cc' request allows the user to select the "control character" that identifies formatting requests.

The 'cu' request continuously underlines output text. It works like 'ul' and 'bd' (mentioned above).

User-defined commands (macros) can be declared with the 'de' and 'en' requests. For example, the "paragraph" command 'PP' used in this note was defined with the sequence

```
.de PP
.sp
.ne 2
.ti +5
.en PP
```

Macros are invoked in exactly the same way as ordinary formatting requests are made. They may also have parameters, which are referenced in the macro text by strings of the form "$n" (where n is a digit from 0 through 9), meaning the n'th parameter supplied on the macro invocation.

Headers and footers may have three parts, which are respectively left-justified, centered, and right-justified.

Odd-page headers and footers may be specified with the 'oh' and 'of' requests, while even-page headers and footers may be specified with the 'eh' and 'ef' requests.

The 'ju' and 'nj' requests may be used to turn justification on and off.

Margins from the top of the page to the header, from the header to the text, from the text to the footer, and from the footer to the bottom of the page may be set with the 'm1', 'm2', 'm3', and 'm4' requests, respectively.

Number registers are available to provide accumulators for saving integer data and for doing simple arithmetic operations like counting. The 'nr' request sets or modifies any of the 26 number registers; a string of the form "@nx" where x is a lower case letter causes the current value of the designated number register to be inserted into the input stream.

A "page offset" independent of the current margin and indentation settings may be specified with the 'po' request as well as with a command line flag.

The 'so' request can be used to read the contents of a named file, thus causing the contents to replace the 'so' request in the input stream. (This construct is similar to the 'include' statement of Ratfor.)

The 'st' request allows the user to begin output at a particular line on the current page.

Text Editor

Two versions of the text editor reside in file 11 of the tape: an in-core version of limited power, and a more complete version that uses random-access disk files for scratch storage. The in-core editor is provided for those people whose systems cannot support random access files; it is essentially the original editor from Software Tools.

The scratch-file editor boasts a number of enhancements.

An optional "-" argument causes line counts printed by the 'e', 'r', and 'w' commands to be suppressed. (This form of the command is useful when running the editor from a script file.)

Tagged subpatterns have been added to the pattern matching routines, allowing portions of a matched string to be re-used in the replacement string of an 's' (substitute) command.

A "browse" command 'b' has been added to print CRT-screen-sized chunks of the edit buffer.

The 'k' command may be used to copy a block of lines from one place to another. Its syntax is identical to that of the 'm' (move) command.

Commentary information preceded by a sharp ('#') will be ignored by the editor. (This feature is particularly useful for documentation of editing scripts.)

If the underlying operating system supports process spawning, the '@' command can be used to submit a line of text for execution by a command interpreter like the LBL shell.

The user now has at his disposal a number of scratch files named by the strings "$1", "$2", "$3", etc. (These scratch files may be used in the editor's 'e', 'f', 'r', and 'w' commands or they may be passed to the system command interpreter through the '@' command.)

Multiple commands may be executed by a single global prefix.

Improvements in the internal line-buffering scheme have increased the editor's speed and capacity.

The editor now issues a reminder to the user if a "quit" command is entered before the edit buffer has been saved.

Archive File Manager

Two versions of the archiver from Software Tools have been provided. The first indicates both the beginning and the end of archive members with special markers, while the second depends on the archive member size information stored in a header record.

The first archiver is relatively independent of file storage techniques used by the underlying operating system, so the archives it produces should port without much difficulty. (Unfortunately, non-text files cannot be handled portably.) Furthermore, these archives may be edited without destroying their integrity. All the information on the tape has been supplied in a format suitable for use with this archiver.

The second archiver is a more straightforward extension of the original version in the book. It can easily be modified to handle non-text files, since its operation is independent of file contents, but it may be sensitive to file storage formats or to trailing blanks in transported archives.

The Basic Tool Set

File number 12 contains a large number of additional programs that comprise the basic tool set. These programs do not require primitives over and above the set developed during the bootstrapping operation, and have been ported to a variety of computer systems.

The basic tool set is comprised of the following programs:

ch	make substitutions in lines of text
comm	print lines common to two files
cpress	compress input files
crt	display files on CRT terminal
crypt	exclusive-or encryption/decryption
date	print current date and time
dc	desk calculator
detab	convert tabs to spaces
diff	isolate differences between files
entab	convert spaces to tabs and spaces
expand	decompress input files
fb	search blocks of lines for text patterns
field	manipulate fields in text data
find	search lines for text patterns
kwic	make key-word-in-context index
lam	laminate contents of files

ll	print longest and shortest line lengths
macro	general-purpose macro processor
mcol	format output into multiple columns
mv	move (rename) a file
os	convert backspaces into overstrikes
pl	print specified lines/pages in a file
pr	paginate files
rev	reverse lines
rm	remove (delete) files
sedit	stream-oriented text editor
show	show unprintable characters in a file
sort	sort and/or merge text files
spell	find spelling errors
split	split a large file into pieces
tail	print last few lines of a file
tee	copy input to output and to named files
tr	transliterate characters
tsort	topologically sort symbols
uniq	strip adjacent repeated lines from a file
unrot	un-rotate lines processed by kwic
wc	count lines, words, characters
xref	cross-reference generator

The Lawrence Berkeley Laboratory Shell

For those installations fortunate enough to have an implementation of a "process spawn" primitive, the File 13 provides a copy of the shell (command line interpreter) written by the folks at Lawrence Berkeley Laboratory. The LBL shell is comparable in power to the Sixth Edition Unix (trademark of Bell Laboratories) shell. All in all, it is a highly desirable addition to an installation, since it allows easy interconnection of programs and provides "personnel portability" with Unix and similarly-equipped Software Tools systems.

Documentation

File 14 of the Basic Tools Tape contains the Software Tools Programmer's Manual, a compendium of reference information for all the basic tools and tutorials for some of the major ones.

Spelling Dictionary

For those installations with word-processing applications, a dictionary of approximately 42,000 words has been included on the tape. The dictionary can be used for spelling checks, games, and any number of other interesting tasks.

Experimental Tools

The 15th file of the tape should be of interest to anyone looking for new or unusual tools and library routines. The programs in this file are candidates for inclusion in the basic tools set and are provided for the evaluation of the Software Tools community as a whole. In all cases, they have been included on the tape exactly as submitted; they have not been tested for portability, good design, or even correctness. Nevertheless, they have proven useful at some Software Tools installation, so they have been provided for general review.

File 15 contains the following programs:

banner	convert text to banner size
cmp	line-by-line file comparison
darken	darken printing by overstriking
dprint	bidirectional printing for Diablo 1610/1620
join	join successive lines of text
label	produce mailing labels
mail	Software Tools mail facility
man	run off section of user's manual
os	convert backspaces to overstrikes
pad	pad lines to fixed length
pg	listing program for fast CRTs
postmn	Software Tools mail checker
pp	general purpose define/include preprocessor
rsa	toy RSA public-key cryptosystem
stats	gather and print statistical measures
ta	tape archiver

File 15 also contains the following subprograms:

ctod	convert string to double-precision floating
ctop	convert string to packed Hollerith format
ctor	convert string to floating point
ctov	convert string to PL/I character-varying format
decode	perform formatted conversion from character
dtoc	convert double precision floating to string
encode	perform formatted conversion to character
gcd	find greatest common divisor
input	easy-to-use formatted input routine
invmod	find inverse in modular arithmetic
page	display file on terminal a page at a time
parscl	parse command line arguments
prime	retrieve the i'th prime number from a file
print	easy-to-use formatted output routine
ptoc	convert packed Hollerith string to unpacked format
pwrmod	calculate exponential in modular arithmetic
rtoc	convert floating point to character string
set_copy	make a copy of one set in another
set_create	generate a new, initially empty set

set_delete	remove given element from a set
set_element	see if a given element is in a set
set_equal	return TRUE if two sets are equal
set_init	cause a set to be empty
set_insert	place given element in a set
set_intersect	place intersection of two sets in a third
set_remove	remove a set that is no longer needed
set_subset	see if one set is a subset of another
set_subtract	place difference of two sets in a third
set_union	place union of two sets in a third
ssolve	solve systems of simultaneous linear equations
vtoc	convert PL/I character varying array to string
dtoc	convert double precision number to character string
ctod	convert character string to double precision number
putdbl	write double precision number to file
rtoc	convert single precision number to string
ctor	convert string to single precision number
putreal	write floating point number to file
printf	formatted print routine

A Simple Text Formatter Macro Package

Allen Akin
Georgia Institute of Technology

In response to requests at the January '81 meeting I've prepared a simple set of macro commands for the Software Tools text formatter. Since there is some interest in a "standard" newsletter submission format, you might want to consider use of these macros when you prepare articles for the newsletter.

Descriptions of Macros

Formatting Environment

At Georgia Tech we've found it useful to maintain one copy of a document, while retaining the ability to produce copies of that document that are suited for printing on a line printer, a daisy-wheel printer, or an offset printing press. To do this, we've adopted the convention that each document begins with an invocation of a "formatting environment" macro named EV. The file containing the text of EV for a particular output device is then named on the command line that invokes the formatter. I have included two environment definitions: one suited to most line printers and pica typewriters, and the other suited to elite typewriters.

Title Page

The title page of a manuscript can be generated with the BT, AU, AD, PI, and ET macros.

BT begins the title page. The line following the invocation of BT must contain the title of the document.

AU begins the list of authors. The lines immediately following AU must contain the names of the authors of the document.

AD precedes the addresses of the authors.

PI indicates that the document's publication information follows on the next few lines. Typically this information will include publication date, document ID number, etc.

ET marks the end of the title page information.

Section Headings

There are four levels of section headings: chapter (the CH macro), major (MH), sub (SH), and paragraph (PH). Each of these macros must be followed by a single line containing the heading information.

Paragraphing

Since the operation occurs so frequently, a special macro for introducing paragraphs has been defined. This macro (PP) requires no parameters.

Long Quotations

Multiline quotations are often set off from surrounding text by indentation and a reduced size type font. The font size reduction is beyond the capabilities of the text formatter, but the macros BQ (begin quote) and EQ (end quote) can accomplish the indentation.

The Text of the Macros

```
.######################################
.# Formatting environment for Elite   #
.#  Elite type (eg Diablo 1620 @12cpi)#
.#    1 inch page offset for binding  #
.#    78 space (6.5 inch) line        #
.#                                    #
.######################################
.de EV
.bp
.po 12
.in 0
.rm 78
.pl 66
.m1 3
.m2 2
.m3 2
.m4 3
.ls 1
.he
.fo
.ef
.eh
.of
.oh
.fi
.ju
.en
```

```
.######################################
.# Formatting environment for pica #
.# (eg. line printer or Diablo     #
.# 1620 @10cpi)                    #
.# 1 inch page offset for binding  #
.# 65 space line (6.5 inches)      #
.######################################
.de EV
.bp
.po 10
.in 0
.rm 65
.pl 66
.m1 3
.m2 2
.m3 2
.m4 3
.ls 1
.he
.fo
.ef
.eh
.of
.oh
.fi
.ju
.en
```

```
.################################
.#  Basic formatting macros for  #
.#     Software Tools Manual     #
.################################
.#
.#
.#
.# BT --- begin title page
.#
.de BT
.EV
.st 12
.ce 100
.bd 100
.en BT
.#
.# AU --- authors' names on title page
.#
.de AU
.bd 0
.sp 12
.en AU
```

```
.#
.# SH --- sub-heading
.#
.de SH
.ti
.sp 2
.ne 4
.bd
.en SH
.#
.# PH --- paragraph heading
.de PH
.PP
.ul
.en PH
.#
.# PP --- begin paragraph
.#
```

```
.#
.# AD --- authors' addresses on title pg
.#
.de AD
.sp 12
.en AD
.#
.# PI --- enter publication information
.#
.de PI
.sp 2
.en PI
.#
.# ET --- end title page
.#
.de ET
.ce 0
.bp
.en ET
.#
.# CH --- chapter heading
.#
.de CH
.bp
.ce
.bd
.cu
.en CH
.#
.# MH --- major heading
.#
.de MH
.sp 3
.ne 8
.ce
.bd
.en MH
.#
.#
.#
```

```
.sp
.ne 2
.ti +5
.en PP
.#
.# BQ --- begin indented quotation
.#
.de BQ [<length_of_quotation>]
.sp
.ne 2
.ne $1
.in +5
.rm -5
.en BQ
.#
.# EQ --- end indented quotation
.#
.de EQ
.in -5
.rm +5
.sp
.en EQ
.#
.# BE - begin example text
.#
.de BE [<length_of_example>]
.sp
.ne 2
.ne $1
.nf
.in +10
.en BE
.#
.# EE --- end example text
.#
.de EE
.sp
.fi
.in -10
.en EE
```

Some Examples of Usage

```
.EV
.BT
Simple Text Formatter Macro Package
.AU
T. Allen Akin
.AD
School of Information and Computer Science
```

```
Georgia Institute of Technology
Atlanta, GA   30332
404-894-4465
.PI
January 29, 1981
.ET
.CH
A "Chapter Heading"
.PP
Chapter headings force a new page to be started.
If desired, the text under a chapter heading can be indented with
the PP macro (as has been done with this paragraph).
.MH
A "Major Heading"
.PP
Major headings do not cause a page break, but they do skip a few lines.
Text can appear after a major heading;
as with the chapter heading, the PP macro can be used to control
indentation.
.SH
A "Subheading"
.PP
Subheadings are unexceptional, following the pattern set by
chapter headings and major headings.
By convention, text after subheadings is indented with PP.
.PH
Paragraph Headings
form the base level of the heading structure.
Of course, paragraphs need not be headed, as shown by most samples
in this paper.
.PP
This is an example of a new paragraph introduced by the PP macro.
The following chunk of text is intended to represent a long direct
quote delimited by the BQ and EQ macros:
.BQ
.bd
Software Tools
is ideal for use in a "software engineering" course, for a second
course in programming, or as a supplement in any programming course.
All programmers, professional and student, will find the book invaluable
as a source of proven, useful programs for reading and study.
Numerous exercises are provided to test comprehension and to extend
the concepts presented in the text.
.EQ
Furthermore, the next chunk of text shows how an "example"
(like a program segment) can be outlined with the BE and EE macros:
.BE
# len - compute length of string
integer function len (str)
integer str (ARB)

for (len = 0; str (len + 1) ~= EOS; len = len + 1)
```

```
   ;

return
end
 .EE
```

This concludes the demonstration of the simple macro package for the Software Tools text formatter.

--- STANDARD TAPE PROBLEM REPORT FORM ---

At the end of the newsletter, there is a problem report form. This form is intended to capture any bugs or problems that are found in the tools on the basic distribution tape. These forms should be sent in so that bugs can be reported in future newsletters.

--- CALL FOR HELP WITH STANDARD TAPE BUG FIXES ---

While we have the mechanism just described for collecting reports of bugs and problems with the basic distribution tape, we still need someone willing to serve as the focus for dealing with whatever problems are turned up. If anyone out there is willing to help investigate such bugs and fixes as are reported and integrate fixes into the basic distribution for future releases, please come forward by writing to the STUG P.O. box, attn. "Bug Fix Volunteer".

--- SOFTWARE TOOLS OPEN FORUM AND CATALOGUE ---

Richard Kiessig is putting together a software catalogue containing information about who is doing what with the Software Tools. He hopes to have some copies of the catalogue available at the Users Group meeting in June. The catalogue will include information that is collected from a "Software Tools Open Forum and Catalogue" form, found elsewhere in this newsletter. The form will be used to help spread information about who is working on what, who has just put the tools up on a MINSK 19 (with 27 bit words), or whatever. If you have anything to share with the group, even if it is still in the idea stage, please take a moment to jot it down, fold and seal the form (avoid staples if possible--the Postal Service says it jams their machines), and mail it back. The information will be made available to other users through the newsletter and the software catalogue. We are especially interested in hearing about:

1) Machines and operating systems on which you've implemented some or all of the tools.

2) New tools you've developed or ones you've enhanced.

3) Ideas you have on extensions to the tools environment, especially ideas for new primitives in other areas of interest (e.g., data base management, graphics, interprocess communication).

4) Ideas you have on how the users group could be made more effective.

5) Pointers to articles you've published relating to the tools.

To have information about your work included in the catalogue, to be distributed at the June meeting, send back your "Software Tools Open Forum and Catalogue" form at the end of this newsletter immediately.

--- THE PROGRAMMING LANGUAGE "Y" ---

The Y programming language by Dave Hanson at the University of Arizona is a structured programming language designed for simple systems programming applications. The syntax is modeled after Ratfor and the C programming language. The semantics of the language is very close to C except that not all C data types are supported. A paper on Y has been published in SIGPLAN Notices Vol. 16, Number 2, page 59.

--- SAN DIEGO DECUS MEETING ---

The DECUS meeting in November in San Diego was well attended, of course. The Software Tools philosophy was evident at several sessions, and was illuminated by Joe Sventek in his presentation on a "Virtual Operating System" (refer to the paper in the September, 1980, Communications of the ACM, Vol.23, No.9).

The presentation was met with enthusiasm, and the VAX and RSX versions of the Tools have been contributed to their respective SIGs for tape distributions.

--- UNI-OPS USERS GROUP ---

A new association for users of Unix, Software Tools, and the like for commercial applications calling itself "uni-ops" is now being formed. Membership is being extended to all interested persons at a cost of $24 per year, and includes a subscription to their newsletter, "Pipes & Filters", which will be issued monthly beginning in June, 1981.

Uni-Ops
P.O. Box 5182
Walnut Creek, CA 94596

--- PUBLICATIONS OF INTEREST ---

The following publications have appeared recently and may be of interest:

1) A Virtual Operating System - Dennis Hall, Deborah Scherrer, and Joseph Sventek; Communications ACM, September 1980.

2) NBS Software Tools Database - Raymond Houghton, and Karen Oakley; NTIS NBSTR 80-2159, October 1980, 104 pp., $9.

3) Conversion Products/Aid Survey - NTIS PB80-222029.

4) The original Software Tools Book by Kernighan and Plauger is to be published with all-Pascal examples. The book should be available sometime this summer.

"Software Tools Open Forum and Catalogue" Form
Please mail in as soon as possible.

NAME: ____________________

ADDRESS: ____________________

____

____

____

PHONE: ____________________

NETWORK ADDRESS: ____________________

Do you have most of the tools and primitives implemented? ____________________

On what operating system do your tools run? ____________________

What new tools have you implemented in addition to the basic tape?

____

____

____

What tools-related projects are you working on? ____________________

____

The following publications describe my work with the tools: ____________________

____

What ideas for new tools or projects have you had? ____________________

____

Would you be interested in getting a copy of the catalogue? ____________________

Additional Comments (use more paper if needed): ____________________

____

____

____

Richard Kiessig
Intelligent Decisions, Inc.
1235-C Henderson Ave.
Sunnyvale, CA 94086

STANDARD TAPE PROBLEM REPORT FORM

NAME: ____________________

ADDRESS: ____________________

____

____

____

PHONE: ____________________

NETWORK ADDRESS: ____________________

TOOL/ROUTINE/MANUAL: ____________________

PROBLEM: ____________________

____

____

____

____

____

SOLUTION: ____________________

____

____

____

____

____

____

____

____

____

____

Software Tools Users Group
Attention:Problem Report Form
242-1259 El Camino Real
Menlo Park, CA 94025

SOFTWARE TOOLS BASIC TAPE
Order Form

Name: ____________________

Address: ____________________

____

____

____

Phone: ____________________

Target computer(s) for Tools: ____________________

Please send the following tapes:

Portable 800 PBI, 2048 characters per block, ASCII	______
Portable 800 BPI, 80 characters per record, 40 records per block, ASCII	______
Other format: ____________________ (i.e. EBCDIC, different density or blocking)	ADD $10
VAX/VMS Tools (LBL) 1600 BPI, Files-11 Format	______
RSX-11M Tools (LBL) 800 BPI - DOS Format	______

Enclose $35 for each tape requested, plus $10 for any special formatting requests.

MAILING:
Tapes within the U.S. will be sent postage-paid. Tapes outside the US will be sent COD Air Freight.

Make checks payable, and mail to:

Deborah K. Scherrer
30261 Palomares Road
Castro Valley, CA 94546

Lawrence Berkeley Laboratory
Debbie Scherrer
CSAM – 50B/3209
University of California
Berkeley, CA 94720

Non-Profit Org.
U.S. Postage
PAID
Berkeley, CA
Permit No. 1123