Software Tools COMMUNICATIONS

NUMBER 7 NOVEMBER 1981

EDITORIAL

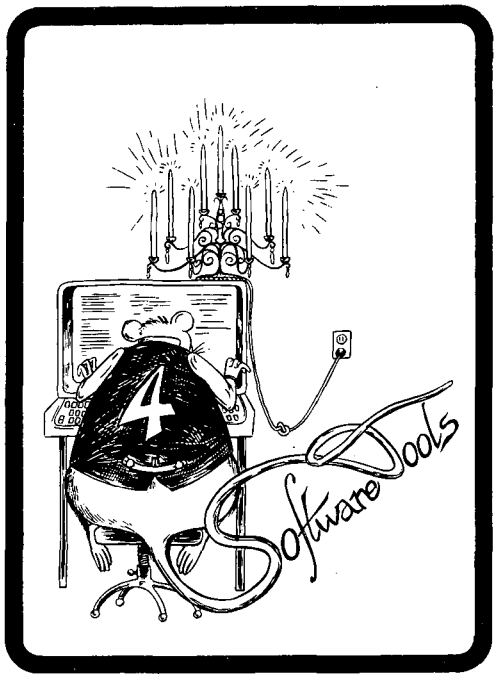
Starting in 1978, the Lawrence Berkeley Laboratory began providing organizational support for the Software Tools Users Group. This was part of our Department of Energy funded research program on uniform user interfaces in heterogeneous environments.

Originally, we wanted to answer two questions:

1) Would providing source code to end users reduce or eliminate the traditional friction between software users and suppliers?

2) Could the resulting proliferation of variants be controlled without a strong central organization?

Providing source code to users was criticized as a cure worse than the disease. Many people felt the result would be chaos. But in our opinion, restricting source access to a small group of experts was guaranteed to produce friction. We wanted to find out what would really happen when source code was freely distributed to a large community of users. We hoped a loosely knit users group would provide the controls necessary to deal with variations.



This work was 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 11-81/1700

The activities of the group were to be:

1) Collecting and evaluating extensions and variants.

2) Publishing standards for primitives and accumulating standard sets of utilities.

3) Collecting and distributing information on current development activities, to help avoid duplication of effort.

4) Operating a software distribution center.

Over the past three years the group grew to 1700 members, held well-attended meetings, published a newsletter and software catalog, established standards, and distributed a tape. The tape contained a collection of utilities developed at different sites. These were evaluated and edited for consistency. The cookbook instruction manual included specifications for primitives, creating a de facto standard. The software catalog provided a wealth of information about tool developers around the world -- information that took time and effort to compile. The calls and letters suggested the users were happy, and it appeared, then, that we had met both goals. Or had we?

Nearly all the above activities were engineered by LBL. Important exceptions included the software catalog compiled by Rick Kiessig, Allen Akin's and Dave Hanson's notable help with the basic tape, and Vinton Goff's tape distribution service. On the other hand, the working groups, set up at the 1979 Toronto meeting, failed to produce standards or collections of utilities. Moreover, LBL is still producing the newsletter, maintaining the mailing list, performing all the managerial functions, and serving as the primary software collection and information clearing house (with help from Michael Bourke). Thus, although the group was successful, it did not evolve to self-sufficiency. We have not been able to shift control of the software over to the users. Why not?

One problem has been the need to rely on volunteer labor. We have had no shortage of volunteers, all of whom have been well-meaning, enthusiastic, and capable. These people are bright and energetic, and have been able to see the advantages of having a uniform environment available. However, their management typically has been unwilling to allow them to do volunteer work. In the view of most managers, letting your best people work on software development tools means your product schedule will slip. While this view may be short sighted, it is certainly prevalent.

The volunteer work has been relegated to after-hours and short-term efforts, with severly limited resources. We were left with a large body of volunteers who were, individually, only able to donate small efforts. This type of organization works well in political campaigns, collections for saving whales, and the like. But it does not appear to work well in situations that require major commitments (editing newsletters, collecting and evaluating software, serving as an information clearing house, and so forth).

Coordinating an army of volunteers is a challenging task. When responsibility is distributed, it is difficult to adhere to strict schedules such as those needed for newsletter production. It was common for us to spend more effort in coordination

DISCLAIMER

than to do the job ourselves.

Another problem is the lack of continuity and constantly shifting control. It is unreasonable to expect a volunteer to continue a task forever. Therefore, volunteer help seems subject to constant turnover. The group coordinator is forever finding and retraining replacements. When personnel changes occur, it is difficult to inform the group of the changed procedures. The members, in turn, find it frustrating to have to keep learning new procedures. A strong central clearing house can alleviate many of these difficulties, but requires a long-term commitment. It is now clear few installations are willing to make such a commitment.

Another factor is simply that what is good for the group is costly to the individual. It takes effort to write an article for the newsletter, prepare a presentation for the meeting, or even prepare software for distribution. Although there is some (small) amount of glory associated with having one's name in print, the primary reward for the efforts is a feeling of self-satisfaction at having done something worthwhile. Those in academic environments are usually willing to accept these rewards, and often consider technology transfer as part of their professional responsibility. However, those in commercial environments (most of our membership), do not consider this part of their job. In many commercial installations, technology transfer outside the organization is discouraged. LBL attempted to alleviate the burdens on the individual by summarizing information for the newsletter and editing software for distribution. However, as the group grew, the demands of these activities went way beyond what we were capable of handling.

Another unanticipated problem has been the overwhelming volume of inquiries. Originally, information about the group was spread by word-of-mouth. However, as more and more publications are learning of our existence, more and more people need information about specific machine implementations, tool extensions, and the like. It is common to receive thirty such requests in one day. The software catalog, containing information about tool developers and specific machine implementations, was created to fill this need. However, few people have obtained the catalog, perhaps because a phone call is more convenient than writing a check and ordering the catalog by mail. A centralized information clearing house, especially one that was automated, would have helped.

The final problem is the difficulty of collecting and evaluating software variants. There are few utilities available (on any system) that assist in maintaining versions of code, merging or deleting enhancements, or comparing and evaluating alternatives. The original tape was produced by the traditional "brute-force" method. (Two of us sat down and read, evaluated, and rewrote utilities by hand. This took eight months...) We had no mechanical way of extracting features from one version and putting them into another, nor of removing unwanted features and system dependencies, nor of merging several variants into one. Until research comes up with solutions to these problems, it will continue to be difficult for the users group to deal with a large body of variants. (As a note of hope, our current research in this area is promising.)

In summary, although the activities of the group were successful, the group failed to become a self-supporting entity. The problem of dealing with volunteer labor, lack of incentives for individuals to participate, high demands for overhead and coordination support, and a shrinking DOE budget have placed too great a burden

on LBL. We conclude that while providing source code to end users makes them happy, it is not currently practical to exercise control without a strong central organization. It remains to decide how best to end the experiment.

We at LBL are simply no longer able to provide the support necessary to maintain the Software Tools Users Group. Activities will continue on a scaled down basis until the Santa Monica meeting. At that point we will relinquish all control and responsibility. Time at the Santa Monica meeting will be set aside for discussion of these issues. We suggest that anyone with interest in the group's continued existence come prepared with concrete proposals. We are willing to turn over the mailing list and other paraphernalia to any group capable of contributing something to the cause.

Alternatives

For those of you still dedicated to the group and the principles behind it, there are several alternatives other than disbanding. They include:

1 - Reform as a group with a monetary basis.
It is possible that with an elected board of governors, the group could either charge a membership fee or charge individually for each service (newsletters, tapes, etc.). It is unlikely that reasonable fees could cover much more than editorial, clerical, production, and mailing costs. So this option would still rely heavily on voluntary leadership efforts. This is the approach taken by Usenix and many personal computing groups.

2 - Turn the group over to a vendor to run.
The tools have been extended enough now to become a viable commercial product and several vendors are already marketing specific machine implementations. It is possible that one of them would be willing to take on the group, charging appropriate fees for the services but perhaps donating some of the organizational effort in exchange for the visibility brought them. The disadvantage with this approach is that, by leaving the decisions in the hands of a single vendor, the users would risk losing control over the development of standards and extensions. This type of organization is similar to vendor users groups such as DECUS, SHARE, and VIM. It would take careful planning to avoid the "them vs. us" attitude often surfacing in vendor supported groups.

3 - Form a commercially-run group
This is a combination of approaches 1 and 2, and similar to the way Uni-ops and /usr/group run. The group is formed by a set of people who have something (monetary) to gain from the movement -- vendors especially, but also entrepreneurs who would do newsletter production, tape collection and preparation, etc. This approach could work well, with the right set of people.

4 - Merge with an already-existing group with similar interests
There are several groups that immediately come to mind. Usenix has been supportive of the tools group and we have been working in close association with them. Usenix is a technically-oriented organization that was set up with goals similar to those of the tools group. However, Usenix remains an organization of Western licensees, so tools members would not be able to join in a voting status.

Usenix has also encountered the same problems with volunteer labor that we have Another possibility is joining with the Uni-ops group. This organization was set up to be "the action group for users of C, Unix, Unix equivalents and Software Tools in business and personal computing". The founder is proposing to several organizations (including this group) that they join forces to publish one combined magazine monthly. A merger like this would allow each individual group to maintain its own autonomous structure, while sharing the editorial (and perhaps some of the managerial?) functions. The one difficulty we see with merging with this group is that their orientation toward business and personal computing may not be appropriate for the many university and non-commercial members we have.

Merger with another group is an attractive alternative. The software tools approach nicely parallels the UNIX approach, with Unix becoming a stand-alone portable operating system and the tools performing the same function only on top of a local system. We would like to see the two movements working more closely together, especially in the development of standards and extensions. However, the tools portion of the group would have to retain a strong leadership and organization to avoid being lost in the commercial rush to jump on the Unix bandwagon.

LBL sincerely appreciates the efforts and enthusiasm of those who have assisted with the group. We would especially like to thank Jim Pool, previously of DOE, currently of ONR, for his constant support of the project; Brian Kernighan for his good-natured encouragement; our technicians Tonia Cantrell and Shirley Cassinelli for their help above and beyond the call of duty; George Kapus for his outstanding logo; Allen Akin and his colleagues at Georgia Tech, as well as David Hanson at the University of Arizona, for their help with the preparation of the basic tape; Vinton Goff for his unswerving efforts to distribute the tape in the face of unbelievable problems; Michael Bourke for his patience in dealing with an overwhelming assortment of phone calls; Rick Kiessig for all his work in preparing the software catalog; Gary Trujillo and Tom Clarkson for their gallant efforts on the newsletter; Walt Brown, for his help in standardizing the ratfor preprocessor; Dennis Hall for his constant support and patience in the face of hysteria; Claude Finn for staying up all night to make copies of the tape for the San Francisco meeting; Dave Martin, Allen Akin, Joe Yao, David Phillips, George Pajari, Wally Wedel, Ed Szurkowski, Tom Ferrin, and ISC for their help in arranging the meetings; Dave Stoffel for his support, enthusiasm, and reassuring words in times of crisis; Skip Egdorf and Neil Groundwater for their continuing interest and organizational support; Todd Kushner, Dale Wolfe, Jim Woods, John Powell, Bob Munn, John Bass, Steve Jones, Burton Leathers, Dan Forsyth, Perry Flinn, Mike O'Dell, Joe Sventek, Lesta Nadel, and the many others who helped in a myriad of ways; and the directors and members of the Usenix group, who have supported our movement and shared efforts in arranging the meetings.

Debbie Scherrer

SOFTWARE TOOLS USERS GROUP
MEETING NOTICE AND CALL FOR PAPERS

T I M E A N D P L A C E

Tuesday 26-Jan-1982
Miramar Sheraton Hotel
Santa Monica, California

This meeting will again precede the USENIX meeting, scheduled for January 27-29 at the same hotel.

R E G I S T R A T I O N

Preregistration:	$20	($10 for students)
On-site registration:	$40	($20 for students)

C A L L F O R P A P E R S

If you are interested in making a technical presentation, please submit a short (100-300 word) abstract by January 8 to:

Dave Martin
Hughes Aircraft Company
Building R1, Mail Stop C320
P.O. Box 92426
Los Angeles, Ca. 90009

dpm@lbl-unix (ARPAnet)

Include your audio/visual equipment requirements and an estimate of the time required for your presentation.

Some suggested topics:

- o Standardization efforts
- o New VOS implementations and portability experiences
- o Software engineering methodology
- o Experience with the PASCAL tools
- o Extensions (new utilities and routines) for:
  - * Networking and Communications
  - * Terminal-independent Screen Management
  - * Database Management
  - * Program Verification and Validation
  - * Text Processing
  - * Office Automation

Advance registration is strongly suggested, due to the large turnout expected for this and the USENIX meetings. To receive a pre-registration packet, contact:

USENIX Association
Box 8
Rockefeller University
1230 York Avenue
New York, NY 10021

212-570-8934

NOMINATIONS

In response to the call for nominations in the last issue, we have received responses from a number of candidates willing to serve on a Board of Directors. However, due to the necessary change in organization of the group, we felt elections (if there are to be any) should be postponed until discussion at the Santa Monica meeting. However, we have printed the candidates's statements of qualifications. Anyone interested in presenting reorganization proposals might like to work with or through these people:

David P. Martin
Hughes Aircraft Co.
MS 367/C320
Bldg. C1
P. O. Box 92426
Los Angeles, CA 90009
213-648-9927

B.S. University of Louisville, 1975
M.S.E.E. University of Southern California, 1978

Staff Engineer, Hughes Aircraft Company

Interested in interactive programming environments, particularly command processors and text editors.

I feel that the group should become more involved with the cataloging and distribution of the various implementations which are available. There seems to be a general lack of information on exactly what tools have been developed for what systems and how they are being distributed. One first step, which I would like to pursue, is the formation of an implementor's group chartered to address the standardization, cataloging and distribution of the tools.

Dave Stoffel
SCION, Inc.
Reston, VA
703-476-6100

Academic Background:

I received my B.S. in psychology and computer science from the University of Maryland, College Park in December of '79. I am currently a graduate student in UOM's Computer Science Department. (at this time, I am on the "five year plan"; I hope to pass the comps at the Phd level in another 3.5 years). My interests at the university are centered in the AI area.

Professional Background:

While an undergraduate, I worked as a student systems programmer for the UOM Computer Science Center. For the past two years, I have worked for the Computer Systems Laboratory, Division of Computer Research and Technology, National Institutes of Health. My work at NIH is probably best described as applied research in the area of man-machine interaction. This work involved the knowledge engineering of 'intelligent assistant' software systems, and interfacing unusual communication devices to these systems. (Some of these devices were voice output synthesisers, voice input recognizers, and touch-sensitive panels).

I am currently employed at Scion Inc., Reston, Va. I will be continuing my work in man-machine interaction.

H. W. (Skip) Egdorf
OS-4, MS 679
Los Alamos National Laboratory
Los Alamos, NM 87544
505-667-4844

Public interest in the software tools has grown considerably since I began working with them about 5 years ago. The formation of the software tools user's group at the Toronto meeting came as something of a surprise, as I had felt that I was one of the very few with an interest in use of the tools to provide program and programmer portability. It was with great pleasure, therefore, that I watched the growth of the user's group and the increase in use of the tools. I feel that the group has two main areas to address now, or in the near future.

I feel that the user's group must encourage development of additional software tools by the incorporation of such tools into our distribution as they become available. While the new PASCAL tools are now available, and will be updated into our distribution, they are still the same tools which (very nicely) support the PROGRAMMING aspect of software development, and give little support to the other phases of the software development cycle. The main thing the group has to offer is public-domain software which demonstrably increases programmer productivity. The state of the art in software engineering has advanced. The tools have not. If this offering of the group becomes outdated, then the reason for the group's

existence will go away.

The user's group has attracted more interest within the academic community than within the commercial sector. I do not wish to discard our academic members; however, the user's group misses a large segment of tools users in the many companies which use RATFOR and the tools for development of commercially valuable packages. These users of the tools would benefit by association with the user's group, as the user's group would benefit from their association. These 'hidden' users of the tools should be discovered, and be made aware of our existence.

Neil Groundwater
Analytic Disciplines
8320 Old Courthouse Road, Suite 300
Vienna, VA 22180
703-893-6140

B.S. in Computer Science, Penn State University, 1971.
Currently working on Masters at Virginia Tech.
Holder of CDP and CCP (Operating Systems) certificates.

In 1972 I joined New York Telephone and for five years applied the UNIX operating system to telephone network data analysis applications in cooperation with Bell Labs. Since then I have been with Analytic Discplines Inc., in Vienna, Virginia, where we have been consultants to the Navy in software development and testing. Our office operates both a UNIX system and a VAX/VMS system which has had the Tools running on it since its first day of operation. I have most recently developed a Text Control System for the Tools environment and expect it to be contributed to the User Group for the next distribution.

The focus of our Tools participation thus far has been to adapt the Tools environment to the Navy software development life-stream. Given the notion that a portable development environment is a key to software productivity, we want to see the User Group continue to distribute Tools for various host operating systems in order that software developers can limit the number of re-inventions of the wheel.

Steve Jones
General Electric Co.
Special-Purpose Computer Center
Mail Stop 31EE
1285 Boston Avenue
Bridgeport, CT 06602
203-382-3652

I have been working with the tools for over 3 years at General Electric, porting ratfor to a number of systems including HP and DEC. I've been distributing the tools within GE and am now involved with efforts by our software engineering program to develop a visual screen front-end to the editor, initially aimed at Perkin-Elmer machines but designed to be ported to various systems.

I am particularly interested in working with the software tools movement to assure standardization.

Dorothy Jo Schmeling
Group C10
Mail Stop 296
Los Alamos Scientific Laboratory
Los Alamos, NM 87545
505-667-8228

Education: BSES in Computer Science, 1980 from University of South Florida
Currently working on a Master's in applied statistics at the University of New Mexico

Ms. Schmeling is a staff member at LANL in the Computer Users Services Group. Her duties in that group include a considerable amount of development and enhancement of the software tools packages which they have implemented on their Cray and CDC 7600 machines.

Bruce Dawson
Digital Equipment Corporation
(ZK1-3/B21)
110 Spit Brook Road
Nashua, New Hampshire 03061
Phone: (603) 884-8058

My job at Digital Equipment Corporation requires me to be familiar with a large variety of software tools on a diverse set of computer architectures. I have been successful at gaining and expanding on this familiarity. I have introduced many tools to the Software Engineers at Digital and am developing ways to integrate them into the work-a-day habits of all Software Engineers.

Among my formal commitments at Digital, I found time to port LBL's Software Tools to the TOPS-20 Operating System. Doing this required me to draw on experience with porting other tools across significantly different machine architectures. I believe that this experience would be of benefit to the Software Tools User Group.

The current set of tools under the User Group's umbrella are very good. They have demonstrated their general port-ability, roubustness, and suitability. However, because they are good, users are getting used to them and are pushing at the limits of the tools' capabilities. If I serve as an officer of the Software Tools User Group, I will work toward expanding the tool's capabilities in an orderly and desirable manner.

MAKING MIDGETS MIGHTY
Tools on CP/M

By Philip H. Scherrer

The software tools have now been completely ported to a micro-computer environment. The CP/M (trademark of Digital Research) operating system which runs on 8080, Z80 (trademark of Zilog), and 8085 processors was chosen because of its wide availability on systems with (barely) sufficient hardware. The goal was to provide the friendly, programmer efficient tools environment on inexpensive systems. A secondary goal was to allow a user to work almost interchangeably on a CP/M system and on larger systems. The work is available as a product of Unicorn Systems. The purpose of this note is to describe some of the problems encountered and solutions adopted.

First, we should describe the typical target system. CP/M was designed to run on a minimum size disk-based micro-computer. The system on which we did our development consisted of a Z80 processor with 64K bytes of program memory, a CRT terminal, an office-quality printer (Diablo 630), and two single-sided, double density 8-inch floppy disk drives (total of 1.1M-bytes). Our system supports interrupt driven serial ports for the terminal and printer, which allows type-ahead and full speed printing. Interrupts are not handled by CP/M, but by the small basic I/O routines that CP/M uses. Most such micro systems have no interrupt controllers at all! This means that concurrent processing is essentially absent. The disk space and memory available on our system is 50 percent larger than on many CP/M systems.

All the tools from the STUG distribution tape, as well as many of the extensions specified in the CACM article, have been brought up and run quite well. The only problem with the tools as presently implemented is their size. The part of the library that is always loaded (primitives, etc.) is greater than 16K-bytes. Since this includes almost the entire set of I/O primitives and many of the support library routines, many programs add only a little to this minimum. The problem with the size is felt in the load time (3 to 5 seconds with 8-inch disks) and the disk space taken by the absolute load images CP/M requires. The complete tools package with all binarys, sources, manual, and dictionary takes 15 single density (241k-bytes each) diskettes. A working set of tools does fit on one 564k-byte disk, with not much to spare. Even with these space problems, we have found the environment to be very workable.

Since program load size and disk access time is so important in the micro environment, the I/O primitives were designed to be as memory and disk access efficient as possible. The primitives use the dynamic memory allocation scheme, first implemented by the Georgia Tech. group, to supply file buffers and file descriptor blocks. The buffers allocated to each open file are kept in a linked list when inactive so they are available if needed again. When more memory is needed for more buffers or other purposes, buffers are freed from this cache (after any pending writes are completed). The scheme for picking which buffers to free gives preference to files doing random I/O. Since CP/M recognizes only 128-byte blocks, while the physical blocking on many double-density floppies and hard disks is 512-bytes or more, there is a read-ahead scheme to try to read

each physical block only once. The combination of a cache with read-ahead using dynamically allocated buffers has served to both reduce disk access time and program load size.

While the hardware limitations made the implementation of the LBL shell impractical, a strategy has been found to make a rewritten shell work reasonably efficiently. Some of the ideas on which the Unicorn shell is based will be described here. A more complete description will be submitted to a future newsletter.

Only one program can run at a time on CP/M. This means that either the shell or a tool is running: a tool cannot be spawned from a shell. So, the first problem is to find a way back to the shell at the end of a tool's execution. To implement this, we gave the routine "endst" the responsibility to start a successor program. Endst was modified to take an argument telling if the exit was normal, or occurred upon an error condition. If endst is entered as a result of a fatal error, CP/M is restarted, otherwise another program is loaded and started. Now, what program? A file called "sh.env" was defined to contain certain environment information such as the program to run next, as well as the date, search paths, disks to use for scratch files, etc. The environment file is read by "initst" at the start of each tool (and is updated by a few tools such as "date" and the shell). In this way, the shell (or any other program) can be automatically executed at the end of a successful tool.

In addition to the operating system and hardware problems, we encountered some language related obstacles. The FORTRAN compiler we chose is the FORTRAN-80 compiler by Micro-Soft. This compiler implements a standard FORTRAN with few extensions. However, some of the common extensions found in many compilers were used in the tools. In particular, standard array indices can be only a very limited set of expressions such as "integer+constant", "constant," "integer", etc. Constructs such as "integer+integer" are not allowed in the FORTRAN standard, nor in Micro-Soft's compiler. We had to eliminate these in the tools whenever they occurred.

Another restriction in the FORTRAN standard is the strict order of declarations that would severly restrict the advantages of "include" files if not fixed. A filter called "fsort" was developed to read the output of the ratfor pre-processor and sort declarations into an order acceptable to the compiler.

The most common problem, however, found in about half of the tools, was improper typing of variables and especially of function return values. On many systems, the ratfor "character" data type is implemented as an integer. Implementors on these systems tended to mix character and integer variables in the same arrays, or to incorrectly type a function as integer when it should have been character, and vice versa. On CP/M, however, we had defined the "character" type as "byte" (which is available as an 8-bit signed integer), so we had to significantly rewrite several tools which had previously mixed character and integer arrays. The mis-typing of functions also caused us considerable trouble. On machines where character and integers were returned from function calls in the same entities (usually a register), the errors were never apparent. However, the 8080 architecture makes it inefficient for the compiler to use the same registers to return byte and integer quantities, so any mis-typed function calls returned garbage. These bugs have now (hopefully) been removed.

The project is essentially complete and has resulted in a quite workable system. The I/O primitives have evolved through 3 generations with considerably more effort required than originally anticipated. (Does that sound familiar?) The benefits were wonderful. When the primitives and library were finally debugged, and the mistyping corrected, most of the tools came up with no surprises and little trouble. That is what bootstrapping is all about!

UNIVAC 1100 SOFTWARE TOOLS
"Taming of the Shrew"

By David Stoffel and Ben Cranston

We just couldn't resist the temptation of the implied challenge in Debbie's CACM paper; we decided that if the Software Tools could be implemented on a Univac 1100 series machine under the Exec-8 operating system, then the claims of Debbie and friends had passed the most difficult of tests.

Our shrew is a Univac 1100/42 running version 36 of the EXEC-8 operating system. Time-sharing on this operating system is performed by a demand job manifesting all the ills of a batch-oriented job control language. Terminal interaction with user programs has the life cooked out of it, not to mention any flexibility. User control over processes by means of executive requests is tedious and primitive. The gauntlet down, and the basic distribution tape in hand, we blindly, devotedly followed the reassuring instructions of the cookbook.

The first problem encountered was that of reading the basic Software Tools distribution tape. This was readily accomplished by writing a special tape reading program. Because this program called on executive requests (rather than reading the tape absolutely) and because the data was stored in standard data format files (Univac SDFF), the character counts for the disk files were not to match those of the tape files. Those who prepared the distribution tape cleverly anticipated this possibility and archived files with the appropriate archiver.

The COPY program (file2) was readily compiled by the FTN (c) ascii fortran compiler. After making a trivial change to the format statements of COPY, it worked correctly for logical units 5 and 6, 8 and 9. No hitches yet!

Ah, but would our luck hold with the BOOTSTRAP (file3) program? Yes, BOOTSTRAP compiled without modification to the tune of 22 warnings, 0 errors. Then, after modifying BOOTSTRAP to read from LUN 8 and write to LUN 9, and assigning (@use control statements) these logical units to the appropriate files, we were ready to attack the RATFOR sources.

We planned a three phase implementation of the Software Tools; 1) each of all the tools, excepting the SHELL, would have to be executable as an ordinary Univac program, 2) the SHELL would have to be executable as a Univac program and be able to spawn the other Software Tools programs, 3) the SHELL would have to be able to "spawn" any (or most) local Univac programs, i.e., those programs that had never been linked with the Software Tools library. Phase 1 was relatively simple; we had file access and terminal i/o working crudely within a couple weeks.

(We elected to implement logical file control in a quick and dirty fashion, just to get something on the air. Since then, we have rewritten, after redesigning, file access to be cleaner and to accommodate SEEK and NOTE for random file access.) Phase 2, consisting largely of implementing SPAWN, was certainly the most difficult aspect of our Software Tools implementation. Our SPAWN, using dynamic user banks, null at startup, expanded when needed, does not call on the operating system to load a "spawned" Software Tools program, but reads the absolute image into the dynamic banks. SPAWN then hands control to the loaded tool, and the tool runs to completion, whereupon it returns control to SPAWN. All "spawned" programs, along with the Shell, are one physical process. Phase 3 is yet under study; the crux of the problem is in getting control returned to the Shell on completion of the "local" Univac program. The os allows a mechanism akin to chaining, which implies that the Shell will have to save a state file, "chain" to the local program, and get control on completion of the local program as the result of a magic mechanism in the os which says "if the magic flag is set, force execution of the last running program or the program that caused it to run if it was chained". The Shell is that program: the program that chained to the last running program. Thus, on execution, the Shell has to check for a state file; if one exists, the Shell reinstates itself, otherwise, it starts as a new instance of itself. This phase 3 scenario seems to be workable, but design issues persist.

As of this writing, all programs from the supported basic distribution are running. We are still working on improving the SEEK and NOTE primitives, and debating the issues surrounding the spawning of local programs. We have little in the way of user response, as we have been leaking the news of the Software Tools implementation slowly, to "advanced" computer users, in fear of loosing nasty bugs on unsuspecting (naive) users. We are targeting release of our implementation to the general user community for the spring semester.

UNI-OPS MEETING

Uni-Ops, a new "action group for users of C, Unix, Unix equivalents, and Software Tools in business and personal computing", will he holding its first conference January 11-15, 1982 in San Francisco. The theme will be "Unix & Company - Unitory Operating Software for the Business of Computing". As well as the technical presentations, there will also be special classes for vendor marketing presentations. For more information contact:

Uni-Ops
P. O. Box 5182
Walnut Creek, CA 94596
415-933-8564

MACHINES

Below is a list of Tools developers who have implemented, or are implementing, the software tools on various machines., Many sites are working concurrently; this list represents those with whom we've had the most contact. If you know of additions or changes to this list, please let us know.

Machine	Contact	Affiliation
Burroughs B1700	C. R. Snow	Newcastle upon Tyne
CDC Cyber, 6000s	Bill Lee	U. of Texas
	David Hanson	U. of Arizona
CDC 7600 (LTSS)	Dotti Schmeling	Los Alamos Natl. Lab.
Cray (CTSS)	Dotti Schmeling	Los Alamos Natl. Lab.
	David Hanson	U. of Arizona
DataGeneral MV8000, Eclipse, and others	Claude Finn	Solvation (previously of DataGeneral)
DataGeneral (several)	Edward F. Miller	Software Research Associates
DataGeneral Eclipse	Jerome Silbert	V. A. Medical Center
DEC 11s RSX-11M	Joe Sventek	Lawrence Berkeley Lab.
	Rick Hambly	Harris Corp.
DEC 11s IAS	Joe Sventek	Lawrence Berkeley Lab.
DEC 11s RT-11	Rob Perry	Tektronix
DEC 10 TOPS-10	Rob Perry	Tektronix
	David Hanson	U. of Arizona
DEC 10 TENEX	Chris Petersen	ORINCON Corp.
DEC 20 TOPS-20	Bruce Dawson	Digital Equipment Corp.
DEC VAX/VMS	Joe Sventek	Lawrence Berkeley Lab.
	Dave Martin	Hughes Aircraft
Hewlett-Packard 1000	David L. Pederson	Rosemount Inc.
Hewlett-Packard 3000	Ken Poulton	Hewlett-Packard
Honeywell GCOS	Jerry J. Deroo	Garmaise & Assoc.
Honeywell (ACOS)	Toshiaki Saisho	Toshiba Corp.
IBM 370 CMS	Leo Noordhuizen	Philips ISA
	William J. Donovan	---
IBM 370 ES (revised CMS)	Ed Happ	Interactive Data Corp.
Modcomp	Bob Upshaw	Lawrence Berkeley Lab.
Multics	Jerry J. Deroo	Garmaise & Associates
Perkin-Elmer	Doug Porter	Porter, Carlin & Assoc.
Prime (400 or higher)	Jeanette Myers	Georgia Inst. of Technology
SEL	Walt Donovan	NASA/AMES Research Center
Univac	Dave Stoffel	SCION
UNIX V6	Walt Brown	Moravian College
UNIX V7	Debbie Scherrer	Lawrence Berkeley Lab.
	Tom Clarkson	Graphic Software Systems
Xerox Sigma 6	Norman Crowfoot	N. A. University
Z80/8080 CP/M	Philip Scherrer	Unicorn Systems

ADDRESSES OF IMPLEMENTORS

Thomas B. Clarkson III
Graphic Software Systems, Inc.
P.O. Box 673
Wilsonville, Oregon 97070
503-682-1606

Norman C. Crowfoot
Computer Services
NAU Box 15100
Flagstaff, AZ 86011
602-523-2971

Bruce Dawson
Digital Equipment Corporation
110 Spit Brook Road
K1-3/B21
Nashua, NH 03061
603-884-8058

Jerry J. Deroo
Garmaise & Associates Ltd.
130 Adelaide St. W., Suite 2014
Toronto, Ontario, Canada M5H 3P5

Walt Donovan
Nasa Ames Research Center
Moffett Field
Mountain View, CA 94035
Walt@bbnc or GAYDOS@BBNB

William J. Donovan Jr.
12815 S. W. 112 Terrace
Miami, Florida 33186
305-385-3764

Claude Finn
Solvation
400-1 Totten Pond Road
Waltham, MA 02154
617-890-8833

Richard M. Hambly
Harris Corporation
1680 University Avenue
Rochester, New York 14610
716-244-5830

David R. Hanson
Assistant Professor
Department of Computer Science
University of Arizona
Tucson, AZ 85721
602-626-3617

Edward G. Happ
Interactive Data Corporation
486 Totten Pond Road
Waltham, MA 02154
617-895-4225
617-890-6419
617-895-4189
617-890-1234

Bill Lee
University of Texas
Computation Center
Austin, TX 78712
512-471-3242
lee@utexas

Dave Martin
Hughes Aircraft Co.
MS C320
Bldg. R1
P. O. Box 92426
Los Angeles, CA 90009
213-648-9927
dpm@lbl-unix

Edward F. Miller, Jr.
Software Research Associates
P. O. Box 2432
San Francisco, CA 94126
415-957-1441

Jeanette Myers
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332
404-894-2746

Leo C. Noordhuizen
Philips ISA - TIS/CARD
Building SAQ 2524
5600 MD
Eindhoven, Netherlands
Netherlands 40-413941

David L. Pederson
Rosemount Inc.
Mail Station C11
12001 W. 78th Street
Eden Prairie, MN 55343
612-937-3679

Rob Perry
Tektronix
Information Display Division
PO Box 500
Delivery Station 60/772
Beaverton, OR 97077
503-682-3411 x3818

Chris Petersen
ORINCON Corp
3366 N. Torrey Pines Court, Suite 320
La Jolla, Ca 92037
714-455-5530
chrisp@lbl-unix

Doug Porter
Porter, Carlin and Associates
6030 Unity Drive
Suite M
Norcross, GA 30071
404-447-1341

Ken Poulton
HP Labs Bldg. 29A
1501 Page Mill Road
Palo Alto, CA 94304
415-857-8461
415-857-1742 (home)

Toshiaki Saisho
Toshiba Corp
Management Info and Systems Division
72 Horikawa-cho, Saiwai-ku
Kawasaki, 2IO Japan
044-522-2111

Debbie Scherrer
Computer Science & Math. Dept.
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720
415-486-5881
FTS 451-5881
scherrer@LBL-Unix

Dr. Philip Scherrer
Unicorn Systems
30261 Palomares Rd.
Castro Valley, CA 94546
415-881-4490 (eves)

Dotti Schmeling
Group C10
Mail Stop 296
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545
505-667-8228
FTS-843-8228

Dr. Jerome Silbert
Laboratory Service
Veterans Administration Medical Center
West Haven, CN 06516
203-932-5711 x466
FTS 641-7466

Dr. C. R. Snow
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Road
Newcastle upon Tyne NE1 7RU
England, Great Britain

David Stoffel
SCION, Inc.
Reston, Virginia
703-476-6100
comsat@usc-isie

Joe Sventek
Computer Science & Math. Dept.
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720
415-486-5205
Sventek@LBL-UNIX

Bob Upshaw
Lawrence Berkeley Laboratory
Bldg. 46A, Room 1140
Berkeley, CA 94720
415 486 6411

Lawrence Berkeley Laboratory
CSAM – 50B/3238
University of California
Berkeley, CA 94720