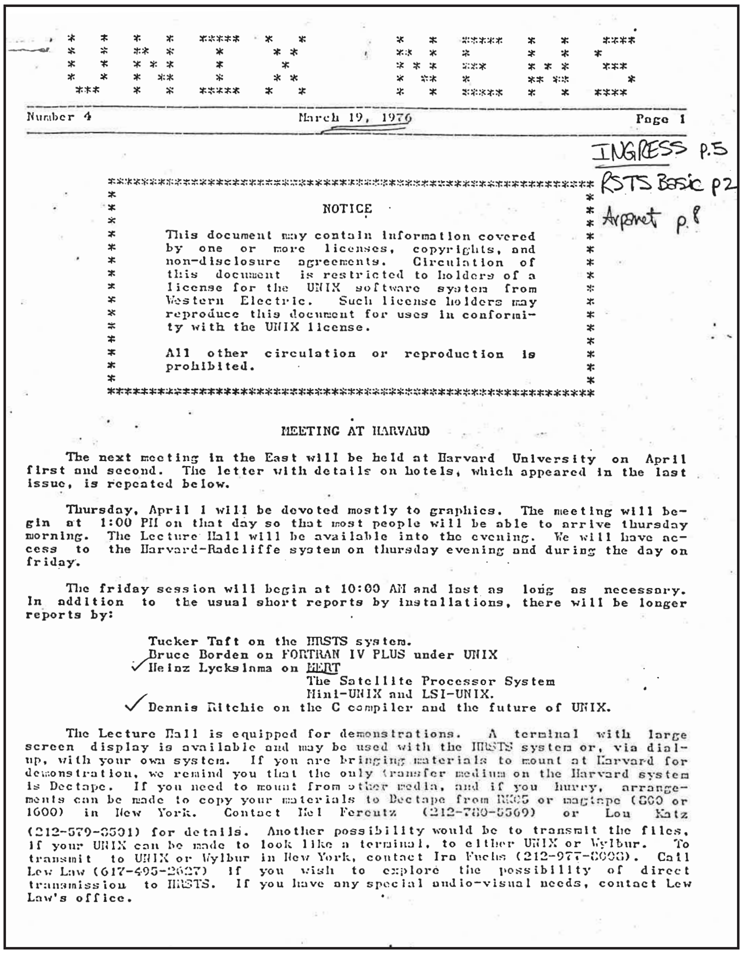
UNIX News

Number 4, March 19, 1976



```
*   *   *   *   *****  *   *        *   *  *****  *   *   ****
*   *   **  *     *     * *         **  *  *      *   *  *
*   *   * * *     *      *          * * *  ***    * * *   ***
*   *   *  **     *     * *         *  **  *      ** **      *
 ***    *   *   *****  *   *        *   *  *****  *   *  ****
```

Number 4 March 19, 1976 Page 1

INGRESS p.5
RSTS Basic p2
Arpanet p.8

```
*************************************************************
*                                                           *
*                         NOTICE                            *
*                                                           *
*     This document may contain information covered        *
*     by one or more licenses, copyrights, and             *
*     non-disclosure agreements. Circulation of            *
*     this document is restricted to holders of a          *
*     license for the UNIX software system from            *
*     Western Electric. Such license holders may           *
*     reproduce this document for uses in conformi-        *
*     ty with the UNIX license.                            *
*                                                           *
*     All other circulation or reproduction is             *
*     prohibited.                                           *
*                                                           *
*************************************************************
```

MEETING AT HARVARD

The next meeting in the East will be held at Harvard University on April first and second. The letter with details on hotels, which appeared in the last issue, is repeated below.

Thursday, April 1 will be devoted mostly to graphics. The meeting will begin at 1:00 PM on that day so that most people will be able to arrive thursday morning. The Lecture Hall will be available into the evening. We will have access to the Harvard-Radcliffe system on thursday evening and during the day on friday.

The friday session will begin at 10:00 AM and last as long as necessary. In addition to the usual short reports by installations, there will be longer reports by:

Tucker Taft on the HRSTS system.
✓Bruce Borden on FORTRAN IV PLUS under UNIX
✓Heinz Lycksinma on MERT
The Satellite Processor System
Mini-UNIX and LSI-UNIX.
✓Dennis Ritchie on the C compiler and the future of UNIX.

The Lecture Hall is equipped for demonstrations. A terminal with large screen display is available and may be used with the HRSTS system or, via dial-up, with your own system. If you are bringing materials to mount at Harvard for demonstration, we remind you that the only transfer medium on the Harvard system is Dectape. If you need to mount from other media, and if you hurry, arrangements can be made to copy your materials to Dectape from RK05 or magtape (800 or 1600) in New York. Contact Mel Ferentz (212-780-5569) or Lou Katz (212-579-3591) for details. Another possibility would be to transmit the files, if your UNIX can be made to look like a terminal, to either UNIX or Wylbur. To transmit to UNIX or Wylbur in New York, contact Ira Fuchs (212-977-8000). Call Lew Law (617-495-2627) if you wish to explore the possibility of direct transmission to HRSTS. If you have any special audio-visual needs, contact Lew Law's office.

UNIX News, Number 4, was published March 19, 1976 by Professor Melvin Ferentz of Brooklyn College of CUNY. We have included excerpts from that issue and have reproduced the text as it appeared in the original, including any typographic errors. Note: We have not included the mailing list and other addresses and telephone numbers that appeared in the original issue, since they are out of date.

Meeting at Harvard

The next meeting in the East will be held at Harvard University on April first and second. The letter with details on hotels, which appeared in the last issue, is repeated below.

Thursday, April 1 will be devoted mostly to graphics. The meeting will begin at 1:00 PM on that day so that most people will be able to arrive thursday morning. The Lecture Hall will be available into the evening. We will have access to the Harvard-Radcliffe system on thursday evening and during the day on friday.

The friday session will begin at 10:00 AM and last as long as necessary. In addition to the usual short reports by installations, there will be longer reports by:

Tucker Taft on the HRSTS system.
Bruce Borden on FORTRAN IV PLUS under UNIX
Heinz Lyckslnma on MERT
The Satellite Processor System
Mini-UNIX and LSI-UNIX.
Dennis Ritchie on the C compiler and the future of UNIX.

The Lecture Hall is equipped for demonstrations. A terminal with large screen display is available and may be used with the HRSTS system or, via dial-up, with your own system. If you are bringing materials to mount at Harvard for demonstration, we remind you that the only transfer medium on the Harvard system is Dectape. If you need to mount from other media, and if you hurry, arrangements can be made to copy your materials to Dectape from RK05 or magtape (800 or 1600) in New York. Contact: Mel Ferentz or Lou Katz for details. Another possibility would be to transmit the files, if your UNIX can be made to look like a terminal, to either UNIX or Wylbur. To transmit to UNIX or Wylbur in New York, contact Ira Fuchs. Call Lew Law if you wish to explore the possibility of direct transmission to HRSTS. If you have any special audio-visual needs, contact Lew Law's office.

Johns Hopkins Software

The Unix Hotline has on it notices from Johns Hopkins University on the availability of several pieces of software.

A version of BASIC-PLUS (version 5B-24) adapted for Unix is available from Johns Hopkins University, Department of Electrical Engineering. It is fully compatible with RSTS/E BASIC-PLUS; nearly any program run on RSTS/E BASIC will work on this one. The only exceptions are:

1) programs are limited to 12K
2) None of the privileged "sys" calls are available; most of the unprivileged ones are.

usenix 40TH ANNIVERSARY 1975–2015

All of the I/O facilities of BASIC-PLUS (record I/O, virtual arrays, etc.) work on the Unix version: full pathnames can be specified in OPEN, RUN, NAME-AS statements. A "public library" feature allows one to keep frequently used BASIC programs in a directory which is referenced whenever a "$" is prefixed to a pathname.

Binary copies are available to those with a RSTS binary license; source is available to those with a source license. Distribution is available on Dectape or RK disk.

A UNIXed version of MACRO (PDP-11 assembler) is available from JHU Electrical Engineering. It has "macro library" and "cross reference" capability, and is reasonably efficient. To get binary, you need a DEC MACRO binary license for RT11, RSX11D, or DOS; to get source, a DEC source license. Distribution on RK disks or DECtape.

A UNIXed version of Per Brinch-Hansen's "Sequential Pascal" compiler and interpreter are available from JHU-Electrical Engineering The compiler is written in Pascal, consists of 7 passes, and is rather slow. Output from the compiler is run under a separate Pascal interpreter. Compiled files can call the interpreter automatically; hence Pascal programs can be invoked as commands. The lexical conventions have been improved over Brinch-Hansen's original specification; lowercase and tabs are accepted, and [] are used for array specifiers instead of (. .). Most of the UNIX system entries can be invoked from Pascal; users can easily add their own following the design of those already installed.

Prerequisite software: MACRO assembler (availabe from JHU for licensees) and Jeff Rottman's Linker (available from Princeton).

Distribution on RK disk or Dectape. A service charge may be involved.

Mailing List

The attached mailing list is a major revision of the old list. It is based upon a list of licensees dated February 1976 and is ordered on state and by zip code within the state. It is likely that errors have crept in during the editing. Please check your listing and send in corrections. An area of difficulty is multiple installations under a single license. If you know of facilities other than those listed, please let us know.

Software Exchange

As of a few days ago, there were no user submissions to the software exchange. The exchange does have a new C compiler and a new ar.c. We hope the lack of submissions indicates that everybody is busy putting things in shape to send in.

In setting up the exchange, we are hoping that people will send in "trivial" things as well as significant things. The trivial is often the most duplicated in effort.

Manuals

Informal discussions indicate that permission might be granted to the Users' Group to allow a single printer to reproduce the manuals and sell them to us in individual or quantity lots. In order to determine whether this is reasonable or desireable, we need an estimate of the number of copies of each of the manuals you might order. If you are interested in this collective venture please contact Lou Katz (see mailing list).

Notes on West Coast UNIX User's Meeting

Berkeley, California, February 27–28, 1976

From: John Lowry, Carl Sunshine, Steve Zucker

The meeting was attended by about 35 people (half from Berkeley), and hosted by Bob Fabry (Berkeley). (See us for a complete list of participants and addresses.) First there were four major presentations on the UCLA Security Kernel, The INGRES data base system, The Harvard Radcliffe Student Timesharing System (HRSTS), and the Berkeley PDP 11/70 system (Ken Thompson). Then each attendee presented a brief summary of activity at his site. Friday evening and Saturday were devoted to discussing several general topics of interest including interprocess communication, the ARPANET software, multiple-machine UNIX systems, and UNIX development and standards. This note summarizes each of the three areas.

UCLA Security Kernel

Gerry Popek

The object of the UCLA work is to produce a verifiably secure operating system. UCLA has implemented a security kernel in PASCAL (a PASCAL to C translator is available from UCLA) and hope to apply automatic verification to the PASCAL code. The kernel has been designed to be the minimum amount of software that can guarantee data security. The code consists of 4-5K words, including a few drivers, and was initially designed as the basis for a Virtual Machine Monitor (VMM). For the sake of efficiency they have decided to try instead to interface UNIX directly to the security kernel by system calls in the same fashion as MERT, the Bell Labs Multi-Environment Real Time System. They plan to run a separate stripped down version of UNIX in supervisor mode as a part of each UNIX "process" with the various UNIXs communicating by means of shared segments and a kernel supplied message facility. No attempt is made to guarantee confinement. The kernel is to provide protection for processes, cevices, whole file systems, and segments, so the size of the protected objects is large ("the grain of protection is coarse").

In addition to the UNIX processes, there will be a scheduler process and an initiator, the former providing data to the kernel assigning execution priorities to the various processes, and the latter replacing the UNIX logon process with code that establishes the protection environment for the user that signs on.

The main problems anticipated are:

(1) The coordination of use of the shared segments between UNIXs. One shared segment will be used for the storage of inodes, file table entries, etc. for each file system. Semaphores may be required in addition to the kernel message facility to coordinate access to these segments.

(2) Constructing an ARPANET NCP or TCP. It is likely that some of this will have to be built into the kernel.

INGRES Data Base Management System

Eugene Wong (Berkeley)

Eugene Wong described the Interactive Graphics and Retrieval System (INGRES) that runs on a PDP 11/40 under UNIX. The current system uses four large processes that call each other sequentially. Some benefit could be obtained from better interprocess communication so that some of the processes could proceed in parallel. (See the ..CC '75 for details on INGRES.)

HRSTS—Harvard-Radcliffe Student Timesharing System

Chuck Prenner (formerly of Harvard, now at Berkeley)

An impressive amount of software has been developed for HRSTS. The following items were mentioned:

1. BASIC: two versions, one based on DEC RT11 BASIC with matrix extensions, another based on DEC RSTS BASIC (which users must first purchase from DEC).

2. MACRO
   LINKER—(Version 6 compatible)
   DDT—(Brought to RAND by Bob Greenberg)

3. TECO editor

4. FILCON—(like diff)

5. HARVARD SHELLS:
   A simplified shell for student use (hand-holding)
   Command completion as in TENEX

6. Terminal driver with several interesting features
   (a) Page mode for graphics terminals
   (b) Setable break characters
   (c) The ability to suspend and restart output

7. PPL and ECL i Extensible languages
   Manuals can be obtained from:
   Center for Research in Computing Techniques

Chuck also spoke of a very good (fast) Fortran that runs under UNIX. It is a modification of DEC Fortran and it is available (after paying DEC a $2500 fee).

The Berkeley 11/70 System

Ken Thompson

Berkeley runs a severely core-limited system, having only 64K words of memory to support an average load of 15 users (22–24 users peak). They use four "home-brew" multiplexers, each having 8 lines. Three of the lines are connected to 11/10 machines used for teaching assembly language programming and interrupt I/O. They have 2 RK05 disks, an RJS04 swapping disk, and 2 RP03 compatible DIVA disks (50 megabytes each).

As usual, Ken had a number of interesting things to say:

1) It appears that the group concept is about to disappear from UNIX, in favor of 16-bit user ids. At Berkeley, the 16-bit ids are partitioned as follows (with a major intent to segregate students from each other, and from the rest of the world)

UID=0:	Super-user (no change)
UID<0:	Student. No reference to other student's files, only "own" files and "public" files, those with owner UID between 1 and 255 (to which normal protections apply).
UID>0:	All others. No reference to students' files. Normal access protection applies to all files.

For class use, the upper 8 bits of the UID is the class number. The teacher has a UID with the low order 8 bits 0, and can access students' files.

2. Disk file quotas: If a directory contains a file named .Q (a quota), then, when the directory inode is in use, the .Q inode is associated with it in the incore inode table. The .Q inode contains the maximum number of blocks that may be used by files in the directory and its decendents, and a count of the number of used blocks. A subdirectory inherits the quota of its ancesters. A new system call was added to make directories with quotas. There were some problems associated with the link operation. It is allowed to exceed quotas temporarily so that the MV operation (rename) cannot cause quotas to be exceeded.

3 A limit was placed on the number of active processes that a user can own. (Enforced in the fork operation: while searching the processor table a count is made.)

4. Ken noted a circumstance under which locks on inodes were not honored within the kernel. He went through the kernel and located a number of other places where the same potential for failure exists. The fixes will appear in the next release.

5. An interesting technique for measuring disk activity was devised. It involves keeping a software busy bit for each of n devices (major or minor) in the low order bits of a word. The word is used as an index into a "time" table of 2**n 32 bit integers. Each time a clock ("uncorrelated" with the scheduling clock) interrupts, the selected "time" entry is incremented. As a result, the "time" entries record the busy time for each combination of devices. By also counting the number of words transferred and the number of transactions on each device, the seek times (including latency and transfer times) can be obtained.

6. Using the technique in 5, Ken determined that for the Berkeley system it was more efficient to use the RJS04 "swapping" disk for the "root" file system (/usr, /tmp, ...) and an RK05 for swapping.

7. A bottleneck was eliminated by allocating two swap buffer headers to avoid putting a process to sleep only to wake it up again to be swapped out.

8. Bell Labs is workng on a C-oriented microprocessor. Get in touch with Sandy Fraser at Bell for information.

9. We learned several things that may improve our response.

   a. Allocate more buffers. Berkeley uses 30, and we are certainly underallocated. A measurement technique for determining buffer utilization was suggested.

   b. The terminal output high and low water marks were set for the 300 baud terminals in use at Bell. With our 9600 baud terminals we should greatly increase these parameters, and the number of character queue elements.

Site Activity Summaries

Steve Holmgren, Center for Adv Comp.—Illinois

Runs an 11/45 with 128k, RP04, RK's, magtape.

Does text processing.

Developed NCP (currently has no server).

Also doing work on local networking.

Mike O'Malley—Berkeley

11/40 system

Runs the Illinois NCP.

Does real time speech processing.

Has added contiguous files.

John Bass—Cal Poly San Luis Obispo

Runs UNIX on an 11/35 (the OEM version of the 40)

Does graphics, data entry, 3-D display for architecture, and language processing.

Dave Farber—UCLA

Has a PASCAL to C translator.

has made changes to ed (e.g., warnings, writeout on hangup).

Jeff Schribman —UCB

Currently has RP03 type disks, DEC RJS04 and RK05's. The RK05 will leave shortly.

Machine connected to CDC 6400.

Runs a RJE station

Harvey Weinstein—Survey Research Center, UCB

Working on computer assisted telephone interviews.

Bill Bridge—Data Disk

Runs an 11/35 with LSI-11 hardware to control terminals.

Developing an automated newspaper production system to run on 11/70's with LSI-11's.

Oliver Roubine—SRI

Has developed a better C debugger.

Art Wilson—UCSD

Has a Fortran which uses 11/40 floating point from Illinois.
Has an 8080 cross assembler
They use the machine for various chemistry applications.

Mike Ubell—UCB

Has GT-40 debugger and Fortran editor.

John Lowry, Carl Sunshine, Steve Zucker—Rand

Runs UNIX on 11/45 with 128K, RP04, RK05's.

Doing intelligent terminal research.

Have developed a 2 dimensional editor, currently being recoded to use ed as a subroutine.

Interested in improved interprocess communication—passed out paper with ideas concerning how this might be accomplished.

Has remind, a Letter version of CROW.

Has file compression routines (25-40% storage savings on typical program and text files).

Runs the Rand NCP (including server).

Mark Kampo—UCLA

Is rewriting tty.o.

Is doing security research.

Has PSCAL, PASCAL, andEUCLID translators.

Has a TENEX compatible HROFF written in C.

Has send message, many new drivers, limited server ftp for the Illinois NCP.

Is connected with coordinated distribution of UNIX software.

TOVAR—Berkeley

Has worked on Dijkstra's primitives (semiphores).

Knows about the CT40.

Has implementd very distant host ARPANET protocol.

Has implemented contiguous files.

Has RK05 comp code for M45.s and dump-interrupting software.

Interested in network graphics program.

Oliver Whitby—SRI

Considering UNIX for scientific communication (Technical Journal Production).

Gary Raetz—Naval Postgraduate School

Working with a variety of display devices.

Has 8080 assembler.

Has a DDT and line editor.

Ira Fuchs—C.U.N.Y.

GT44 system

Runs as text editor front end for 370.

Reported that April 1 of the East Coast Meeting will emphasize graphics.

Discussion Topics

Interprocess Communication (IPC):

Steve Zucker and Carl Sunshine presented the shortcomings of current UNIX IPC facilities. Steve outlined his Port and mpipe implementation. (The two Rand papers had been passed out the day before, but few attendees had had a chance to read them.) There was agreement from several other sites that an ability to wait for multiple inputs would be very desireable, and that mpipes seemed a good way to do this. Bob Fabry was unhappy with the different interface mpipes provided to the reader (the presence of headers).

Other suggstions to facilitate waiting for multiple inputs concentrated on the synchronization problem. Mark Kampo advocated a sleep/wakeup facility for user processes. Steve mentioned the problems of access control, assigning unique wakeup numbers, spurrious wakeups, and implementation with this. Others suggested a more specific facility to "wait" on a specified set of file descriptors and return when the first of them had data available to read. There is also the problem of testing for input (e.g., with EMPTY) and then doing the WAIT, with input arriving between the test and wait. This requires the standard "hyperawake" state kind of resolution when a wakeup to a running process leaves it "hyperawake" so a wait does not block it.

The major implementation problem in current UNIX for both of these suggestions is the difficulty of associating an inode with those processes (readers) "waiting" when a writer writes the inode.

The idea of asychronous, independent I/O channel type operation came up, but this was fairly generally denounced as a tremendous change and undesireable anyway.

Ken Thompson agreed that the current signal facility had a number of problems (but noted that the interrupts did not abort pending disk I/O since the process would be sleeping at a negative priority). Signals had been designed primarily for error or exception condition handling, and not for general interprocess synchronization which Ken was not convinced was a very significant need. By the end of the session he seemed more disposed toward accepting the improved IPC as a legitimate concern, and even willing to consider system changes or additions to provide it.

ARPANET:

Steve Holmgren outlined the ARPANET software design at the University of Illinois. The connection establishment (IPC) code for the NCP has been largely placed in a user process, requiring only 3–5K words (plus buffers) for the remainder of the NCP in the kernel. There is a separate minor device for each network host, and the open command to these network devices defaults to give a Telnet connection. By specifying different options, processes can also get particular sockets, LISTEN mode, or other special actions from the NCP. The server Telnet is still being designed, but will probably feed incoming characters through the TTY drivers.

Multiple-Machine UNIX:

Ken Thompson outlined the dual machine UNIX system he developed at Bell Labs last year using a DR11-C. The basic communication level provides 256 logical channels between two machines. Each data byte is prefixed with a channel number on transmission, and acknowledged by the channel number being received. For more efficient and reliable communication over low bandwidth or long delay lines, the characters on a channel may be buffered into a message with the channel number appearing only once in the header, and a checksum for error detection.

Connections are established between machines by sending control data over channel zero which essentially associates files with channel numbers in each machine. A daemon process which always has a read pending handles the initial connections.

At the user level, commands are interpreted by a Master Shell that looks for a special character (!) in the normal UNIX command syntax which indicates an operation to be performed on the remote machine. Thus !cat rfile > lfile would copy remote file rfile to local file lfile. The Master Shell creates a slave shell process in the remote machine, and all other local and remote processes needed, and sets up all the connections between the local and remote processes.

UNIX Development and Standards:

Because of the consent decree and other intricacies of the UNIX license, Bell is not able or willing to "support" UNIX in the traditional sense of standard releases, updates, fixes to discovered bugs, documentation, etc. There has been talk of ARPA support for a UNIX maintenance and standards project, but this is still uncertain.

There is a strong feeling that sites are going to continue doing their own development in different and sometimes even conflicting directions. There is also a strong frustration with efforts to use each others improvements which often seem to depend on other aspects of the system which are incompatible. A "standard" version of UNIX for reference purposes could make it easier to describe where changes had been made when sites trade software. The clearing house at Chicago Circle mentioned in the February UNIX News may provide this service by sticking a version number on some recent UNIX system and distributing it. But sites currently have widely differing versions of the system (version 5 to version 6 .1) so picking the initial standard is problematic.

Conditional compilation exists (undocumented) in version 6 (plus?) of C, which may help the situation if people "if out" changed standard code rather than deleting it, and "if in" their changes.

Ken Thompson described the way a new version of UNIX gets changed at Bell. The system is constantly undergoing changes, and when these become numerous enough, it is agreed that the manual must be rewritten to reflect the new facilities. In the process of rewriting the manual, everyone remembers all the other things they wanted to do to a particular area of code, and an escalating "frenzy" of activity ensues for a couple of weeks. Finally the manual gets closed and filed one day, after which activity tapers off (but doesn't cease) because people realize the new changes won't be known until the next manual rewrite.

Note to our readers
Do you have any back issues of *UNIX News*? Our archive is not complete, and some of the reproductions we have are difficult to read. We'd love to have copies of anything that you have. Contact production@usenix.org.