1291
UNSD

Bell Laboratories

Subject: Recommendations for A Company-Wide UNIX* Education Program

date: June 6, 1978

from: R. M. Coben
E. H. Dudley
R. A. Faulkner
R. F. Graveman
B. W. Kernighan
R. D. Reese
R. F. Rosin
D. S. Sand
G. C. Vogel
G. A. Wilson
B. C. Wonsiewicz

ABSTRACT

Bell Laboratories has become heavily committed to the UNIX time-sharing system, its associated utilities, and the programming language C. Although current efforts in UNIX education are extensive, they tend to be disorganized, redundant, and incomplete. The Ad Hoc UNIX Education Committee was chartered to recommend a solution to this problem, addressing specifically the needs of the technical population.

Our committee recommends that a new supervisory group be formed whose sole concern will be the many aspects of UNIX education. This supervisory group should be aligned with a central UNIX system support organization.

This report presents an overview of the use of UNIX at Bell Laboratories, our view of the challenge for UNIX education, and our specific recommendations. Appendices to this report cover these topics in detail and include tentative recommendations for course modules and outlines.

*UNIX is a Trademark of Bell Laboratories

Bell Laboratories

Subject: Recommendations for A Company-Wide UNIX* Education Program

date: June 6, 1978

from: R. M. Coben
E. H. Dudley
R. A. Faulkner
R. F. Graveman
B. W. Kernighan
R. D. Reese
R. F. Rosin
D. S. Sand
G. C. Vogel
G. A. Wilson
B. C. Wonsiewicz

MEMORANDUM FOR FILE

In late 1977, in response to perceived deficiencies in UNIX education for technical staff, the INCEP Computer Science Curriculum Committee formed the Ad Hoc Subcommittee on UNIX Education, whose members have written this report.

The first section surveys the growth of computing, including UNIX and C, in Bell Laboratories and describes the UNIX education facilities available. Subsequent sections describe the challenge for UNIX education and our recommendations for meeting that challenge.

1. Introduction

Bell Laboratories is changing rapidly. Both our products and the technology we use to design them are following the revolution toward stored program control and dependence upon the computer. As a result, many new employees skilled in computer science are being hired, and many digital logic designers are being retrained in programming techniques, languages, and systems.

1.1. Growth in Computing; Introduction of UNIX and C

In the period from 1971 through 1977, the number of minicomputers at Bell Laboratories grew from about 180 to almost 800. The investment in minicomputers in this period grew from $10 million to over $50 million. In 1971, a new minicomputer-based operating system called UNIX became available to a limited number of Bell Labs users for the first time. UNIX, along with the programming language C, is certainly the fastest growing computing system in our company and probably in the Bell System today. In 1973, there were fewer than 10 UNIX systems in Bell Laboratories; five years later the number of systems has grown to over 100.

Appendix A discusses the growing BTL dependence on interactive computing, the dependence of BTL word processing on UNIX, and the heavy involvement of UNIX in project development. Clearly UNIX is a system which touches thousands of Bell Laboratories employees. For many, the UNIX contact may be casual - perhaps through usage of some subset of the text editing utilities. For many others, UNIX will become a principal part of their jobs. As one example, Department 8234 estimates the establishment of 3 to 4 new UNIX-dependent OSS projects per year, employing a minimum of 6 programmers each.

*UNIX is a Trademark of Bell Laboratories

1.2. UNIX Education Available Today

The 1978-79 INCEP Catalog offers only three UNIX-related courses for in-hours students: one (8L107) covering the C language, and two (8C243, 8C244) covering UNIX. In addition a recently proposed course, "Design of Software Products Based on UNIX", is being surveyed at Holmdel, but the course has not been designed and no resources have been committed to this task.

Last year's INCEP program offered the C course and a course devoted to the internals of the UNIX operating system. It was not surprising that 50 per cent of the students that signed up for the internals course, 8C244, at Whippany in September, 1977, did so wanting to learn *how to use* UNIX rather than details of its construction, but no such INCEP course was available. These students had to be turned away at the Whippany pre-course meeting because the catalog description for the course precluded including the material these students needed.

The listing for "Typical Text" for 8C244 in the new INCEP catalog shows: Lions, *A Commentary on the UNIX Operating System*. It is ironic that material prepared by a professor for his students at the University of New South Wales in Australia was selected by Bell Laboratories for presenting the internals of a system developed here.

The foregoing should not be interpreted to mean that substantial effort has not be made in UNIX education in BTL. There are a number of excellent tutorial documents, videotape courses, course outlines, and computer-aided tools built for this purpose. In spite of this activity, UNIX education is in urgent need of substantial direction, consolidation, and improvement.

At present there is no single organization responsible for UNIX education. In the next few years, students and teachers will devote many thousands of EMTS-days to learning UNIX. It is in Bell Laboratories' interest to redirect and focus its UNIX education effort and to commit the necessary resources *now* so that those who need to use UNIX may learn to do so as efficiently as possible.

2. The Challenge For UNIX Education

Recent courses on almost any aspect of UNIX have been consistently oversubscribed, regardless of location. A variety of factors explain why this is so. First, UNIX and C are relatively new and rapidly evolving subjects. This means that many persons are just now being introduced to UNIX facilities.

Second, both UNIX and C have wide applicability throughout Bell Labs. The facilities include text editing and formatting, C program development, host (e.g., IBM) program development and remote submittal (RJE), source code control (SCCS) for programming projects, microprocessor support tools (for MAC-8), modification request control (MRCS) for automatic trouble reporting and other modification coordination of development software, etc. This wide range of facilities means that everyone from typists and secretaries through engineers and computer scientists will have need for some of the available features.

Third, new employees always place heavy requirements upon educational programs, and particularly on courses dealing with subjects like UNIX. On top of the record high influx of new employees in 1977, the company has set for 1978 its highest technical recruiting goals in history: at least 1000 MTS/AMTS and 600 associate technical staff members. According to employment personnel, "as many as possible" of these people will have backgrounds in computer science or computer engineering. Assuming a minimum objective of 50 per cent, we can project at least 750 and up to perhaps 1000 new employees with some computing background this year, many of whom will need UNIX quickly to get onboard in their projects. Of the total, many will need education as soon as they arrive, and, unlike students entering a university, they will not appear at one location simultaneously.

Fourth, the recent company emphasis on computer-aided design and stored-program-controlled systems means retraining many digital logic designers in programming techniques, languages and systems. This will become increasingly significant as the microprocessor

continues to make its impact. With virtually all MAC-8 applications being developed on UNIX systems, the need for training such people on UNIX is critical.

One could cite other important reasons why the demand for UNIX education has accelerated and will assuredly persist. It is clear that people must learn how to use these new techniques and systems, be it in a classroom, by reading and self-study, or by computer-aided instruction. In addition there must be a mechanism established to provide easily accessible personal consultation for people learning and using UNIX. Each individual has different needs related to his or her current job, educational background, and even individual style of learning. This large, diverse need will be met somehow no matter what is done about UNIX education. It seems clear that Bell Laboratories should provide systematic, high-quality UNIX education as quickly as possible so that the need is not met inadequately, inefficiently, or too late.

3. Recommendations

An organization should be given the resources and responsibility to develop and execute a comprehensive UNIX education program. The duties should include: creating, testing, distributing, and maintaining tutorial materials, some of which may be appropriate for external distribution; planning, teaching, and documenting both short and conventional length courses; and establishing a program for post-classroom education. The latter must include ready access to consultation services to be provided by others.

Such a group should be staffed partly by permanent members, with demonstrated skills in both computer science and classroom teaching, and partly by temporary members serving an internship who might bring special UNIX skills to the group. The group should consist of at least six technical staff members with additional clerical support as required. In addition, there is a need for physical resources: properly equipped classrooms, and sufficient access to UNIX systems. (See Appendix B.)

The need is important. The resources should be committed, and the new organization should be held accountable for the results.

A variety of materials and educational aids need to be developed, either by this new group or by others in cooperation with this group's overall plan. We believe that short, well-defined, well-executed, and well-tested modules and materials provide an excellent means for learning UNIX skills. Such modules can be assembled into more traditional, longer courses when desired or appropriate. Ready access should be provided to these modules and courses by regular formal presentations and through the ILCs (Individualized Learning Centers). A more detailed account of our observations and recommendations concerning educational materials, course format, physical resources, continuing education, and commitment may be found in Appendix B. Tentative recommendations for course lists, course content, and course modules are contained in Appendix C.

We believe that "learning by doing" must be the theme, because few people learn computing by listening or reading alone. Problems, exercises, and examples of high quality are essential and should be treated as a valuable resource. Everything - from materials, course format and presentation, to the student's learning - should be tested.

Staffing the UNIX education group not only with full-time professionals but also with experts borrowed on internships from diverse organizations will serve to spread a new level of expertise in UNIX education throughout the company, as well as to enrich the group with the central education responsibility.

4. Conclusions

The need for improved UNIX education in Bell Laboratories is urgent and widespread. Our present efforts are too diffuse and inadequately supported to meet this need. The responsibility and resources to develop and present appropriate materials and courses should be assigned to a single group as soon as possible.

Specifically we recommend:

- That a supervisory group of at least six full-time members of technical staff be given responsibility for developing and executing a coherent UNIX education program.
- That classrooms be upgraded to support adequate display of interactive computer usage.
- That sufficient computer time, connection ports, and memory space be allocated to UNIX education so students can carry on their educational activities both inside and outside of the classroom.

While six EMTS is a substantial additional committment to UNIX education, this represents less than .5 percent of the estimated 10,000 hours per day spent by BTL employees using UNIX.

R. M. Cohen
E. H. Dudley
R. A. Faulkner
R. F. Graveman
B. W. Kernighan
R. D. Reese
R. F. Rosin
D. S. Sand
G. C. Vogel
G. A. Wilson
B. C. Wonsiewicz

8223 RMC
7792 EHD
*8261 RAF
HO 9413 RFG
MH 1273 BWK
DR-3241-RDR-pcb
IH 3233 RFR
WH 5512 DSS
8234 GCV
8225 GAW
1523 BCW

Atts.
Appendix A,B,C

Copy to (with atts.)
W. C. Ridgway, III and members of the INCEP Computer Science Curriculum Committee
R. R. Conners
T. A. Dolotta
M. P. Fabisch
R. C. Fremon
J. R. Harris
R. G. Hinderliter
J. R. McEowen
S. P. Morgan
B. A. Tague
P. H. Thayer, Jr.
S. H. Washburn
C. R. Wischmeyer

APPENDIX A

BELL LABORATORIES INTERACTIVE COMPUTING ENVIRONMENT

1. Dependence Upon Interactive Computing

In 1971 there were 300 computer terminals serving all Bell Laboratories locations. By the end of 1977, the number was almost 5000. The growth in conversational computing at Bell Laboratories represents a major new trend in product development.

Within the last year the general purpose computation centers at Holmdel and Indian Hill have begun to offer UNIX time-sharing service. The offering has been so enthusiastically received that these two locations have already had to install six PDP 11/70 systems to keep pace with the demand. At the time of this writing, two additional systems are about to be installed. Area 90 alone now has eleven PWB systems supporting time-sharing, with somewhere between 1500 and 2000 userids. Current estimates from the Raritan River computation center indicate a 140 per cent growth rate in UNIX connect hours per year and a 70 per cent annual growth rate in UNIX ports.

2. BTL Word Processing Moving to UNIX

In 1976, a task force was chartered under R. C. Fremon* to determine an appropriate computer system for use in supporting word processing at Bell Laboratories. Among the recommendations of that group was the specification for computing systems with "a single software system, at least approximately equivalent to UNIX." In fact, since that task force report, one PDP 11/70 at Indian Hill and two PDP 11/70 systems at Murray Hill have been installed for the express purpose of supporting Word Process Center typists on UNIX. Similar service is being provided on PWB/UNIX at Holmdel and Piscataway, although not on dedicated systems. Three more dedicated word processing machines, PDP 11/70s running PWB/UNIX, will be installed in August: two at Whippany, one at Murray Hill.

The task force report in September, 1976, said that text production at Bell Laboratories "is done by several categories of people, including 203 typists, 125 stenographers, 675 secretaries, and an unknown number of others including technical and administrative staff members." At the time of the task force report (9/76), the word processing centers were producing "1.5 million pages of text per year." Yet an estimate of the total text processing picture could be derived from the fact that "each year BTL buys nearly five million sheets of the kinds of paper that are used for typed originals." It would appear that the demand for UNIX-based word processing support is immense.

3. Heavy Involvment of UNIX in Project Development

An especially important role for the UNIX-based systems (USG/UNIX, MERT, and PWB/UNIX) is in the support of Operations Support System (OSS) projects. Many of these systems utilize either UNIX-based or UNIX-derived operating systems. Just a few of the important UNIX project applications include:

- AEMIS - a system for managing automatic call distribution,
- RBCS - Record Based Coordination System, a CORDS OSS,
- RCMAS - Recent Change Management System, an OSS for ESSs,
- DIR/ECT UPGRADE, white pages composition.

---

*"Task Force on Computer for Word Processing Report", Sept. 15, 1976

- DIR/ECT RTLR - Real-Time Listing Retrieval from white pages.
- CCIS/STP - CCIS Signal Transfer Point.

Important UNIX applications include development laboratory control systems for #1/ESS, #1A/ESS, VSS, TSPS, ACS, and HICAP/MATS. In addition, UNIX is providing significant support in the development of the 3B Processor projects and the MAC-8 microprocessor. SCCS - Switching Control Center System - providing centralized maintenance for electronic switching machines and EADAS - Engineering and Administration Data Acquisition System - for gathering and analyzing traffic data are Columbus-based projects utilizing an adaptation of USG/UNIX called "CB-UNIX", which contains many local features appropriate to the Columbus applications.

APPENDIX B

OBSERVATIONS AND DETAILED RECOMMENDATIONS

The following material represents ideas and opinions expressed by committee members during discussion and in early drafts of this report. It does not necessarily represent the concensus of our thinking, but various items may direct and stimulate the thinking of a UNIX education group, if one is formed.

1. Educational Materials

Although there is some educational material available, it is fragmentary and deficient in a number of ways.

Some important parts of UNIX, such as the basic user interface and the text editor, are adequately described in written tutorials. In contrast, some topics have no tutorial material at all; in fact, there is often little more than the cryptic synopses in the "Programmer's Manual" (which itself is frequently out of date).

The Programmer's Manual badly needs an introduction or separate tutorial which tells the uninitiated how to read it. Examples would greatly ease the problem of using the Manual, which for most beginners (and many others) is nearly impossible at present. Examples should include the simplest uses of the command, so that a user could type exactly what is written and get the intended response from the system. Other examples should illustrate the use of the more common options.

A variety of tutorials is needed. Typical subjects would include effective use of the commands on the UNIX system, the UNIX Shell, and various topics under "Programming in a UNIX Environment," such as archiving, Make, etc.

More specialized topics like Lex, Yacc, and System Administration could also use good tutorial documentation and seem particularly suitable for seminar or workshop courses.

Because UNIX is an evolving system, material goes out of date rapidly. In addition, there are several major UNIX variants in wide use, each with its own rules, conventions, and need for documents and information. Different systems have different hardware facilities. There is an urgent need to provide appropriate materials and to keep them current.

Some video and audio taped courses exist, but they are sometimes technically deficient (recording a CRT terminal display so that the characters are legible isn't easy) or inappropriate for local needs. Typically, we recommend well-designed, carefully-produced, and self-contained tapes on specific subjects. These will generally be more appropriate and easier to use, both in the classroom and in ILCs, than series of interrelated tapes spanning entire 15-week courses, especially for material beyond the introductory level.

Many prefer computer-aided instruction, as exemplified by the heavy use of the *learn* program. Demand for *learn* at some locations has been great in spite of the fact that the few available *learn* scripts are in need of revision. Because the overhead associated with *learn* is high, many installations limit its use during working hours. In view of the heavy demand, *learn* should be upgraded to cover more topics, offer more alternative tracks within a topic, and make more efficient use of computer resources.

Whenever possible, impersonal learning facilities, such as audio and video tape and *learn*, should be offered only in conjunction with a "source" of further information or personalized counseling. Otherwise, the saving in human instructor time which these tools offer will be more than offset by time lost by students who cannot get answers quickly for straightforward questions.

In summary, there is a need for a variety of educational materials. All should have clearly stated educational objectives; all should be tested and reviewed in real-life contexts to assure

that they perform as intended. Documents which meet the test should be treated as full technical publications.

Staff members perceive little credit for writing tutorials, teaching classes, or consulting. Excellence in these areas deserves reward and this should be made explicit in performance reviews.

2. Course Format

The INCEP format poses special problems for learning UNIX. Most INCEP courses meet once a week for fifteen weeks, and are given once per term at most. Such courses are frequently too diffuse to let the student get the material he or she needs quickly and are not generally suited to skill-oriented topics like UNIX use. Preparing a fifteen week course is a formidable job, as witnessed by the difficulty in finding willing INCEP instructors. As a result, courses are offered infrequently.

Short courses can have more precisely stated objectives, so the student is likely to get what he or she wants. Because short courses are easier to teach, they can be offered more often, and because short courses are easy to take, students are more likely to attend all the lectures and complete all the homework than in a long course where travel and job demands often preempt course attendance.

The growing number of individualized learning centers (ILCs) is beginning to offer Bell Labs employees the opportunity to take both traditional and short courses at a pace consistent with the time they have avilable. Ideally, educational "modules" should be designed for use in both conventional long and short courses, as well as in ILCs.

Prerequisites should be clearly defined, and advanced courses should not be watered down to accommodate students who have not complied. Prerequisites are easier to enforce if they are available in short courses which are offered frequently.

In teaching skills, it is important to include subtler issues like background, philosophy, and style. The Why should not get lost in the How.

3. Physical Resources

There is often a shortage of the physical resources needed for teaching a hands-on discipline like computing: systems are overloaded enough that classes have difficulty obtaining logon names, phone lines and terminals, CPU time, and file space. There are few classrooms properly equipped with the video monitors, terminals and phone facilities required so that everyone can see how UNIX is used.

4. Continuing Education

Education extends past the classroom. Users need help in solving day-to-day problems and in keeping abreast of new developments.

At present, UNIX counseling is hard to find even though it is an efficient way for people to get answers to specialized, sophisticated problems. The precise manner in which UNIX counseling may be most effectively offered should be determined by the UNIX education group recommended above.

It is difficult to keep abreast of the new developments in UNIX. Traditionally, information has been passed by word of mouth or a slow diffusion process which is unsuited to a large widely-dispersed user community. A newsletter supplemented by tutorial or at least educational seminars seems in order. Both the Mini-System Newsletter and the UNIX Users Meetings deal with topics too advanced and too specialized to meet this need for many ordinary users.

5. Commitment by All Concerned

An explicit commitment to complete the course objectives should be obtained from every student and from every student's manager. In turn, course objectives should be clearly defined and adhered to.

APPENDIX C

CURRICULUM RECOMMENDATIONS

1. Background

The following material is representative of the thinking of the Ad Hoc UNIX Education Committee in areas of curriculum. Much of it is tentative, and the reader may even find redundancy and inconsistency. These matters require considerably more time to study than this committee has had available. While the incompleteness of our results offers evidence that an independent UNIX education group is needed to do this work well, we hope these ideas will serve to stimulate such a group to include unusual educational methods and curriculum structures in their deliberations.

The body of this paper refers to "modules" and "short courses" as well as "traditional courses". Indeed the distinctions among these elements was a central theme in many of our discussions. Just as carefully-designed, easily managed software modules can be assembled into a broad spectrum of systems meeting a variety of needs, so can similarly designed and conceived learning modules be assembled into educational vehicles meeting a variety of individual needs.

Modular construction also implies the ability to change one module in the system without affecting the internals of any other module, and this characteristic could be of great advantage in maintaining educational material for a subject matter which changes almost daily. It is important, however, to maintain consistency and coherency within a module so that texts, video tapes, exercises, etc. for one module are up to date and not mutually contradictory.

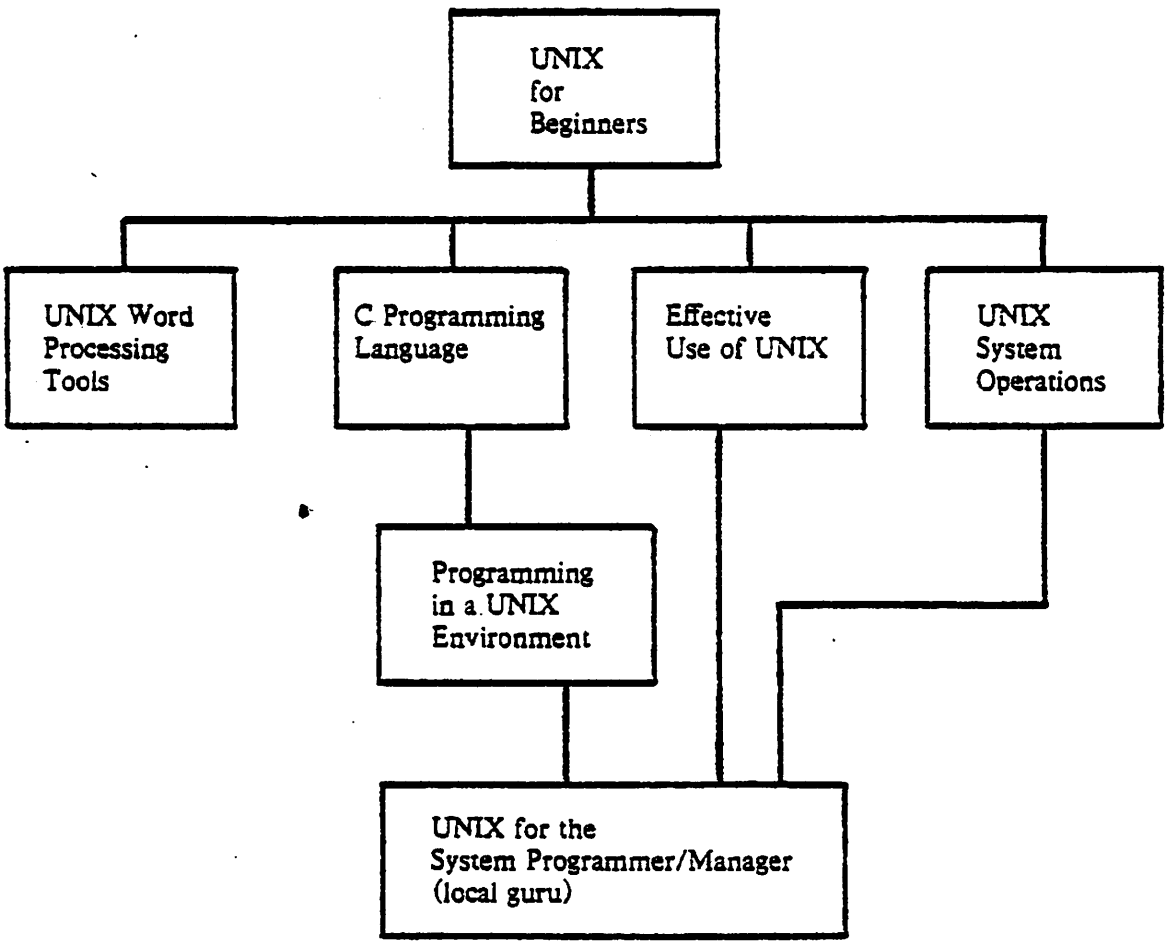
For instance, modules on the editor could probably be assembled using various combinations of tutorial text, reference text, examples, exercises, live demonstration, video taped demonstration, *learn* script, etc. Such modules should be designed to be used for independent study, as part of short courses, and as part of more traditional long courses.

One function of the recommended UNIX Education Group will be to formulate a clear definition of the term module. A module is clearly much shorter than a full length course, and it should be coherent. Some modules should probably concern the use of only a single UNIX command, while others might address as broad a set of topics as covered in "UNIX for Beginners". One suggestion was that module length be defined in terms of the amount of time needed for a typical student to absorb a particular body of material, and that one-half day or even one-half hour be specified as an ideal module size. Certainly a module covering introductory material will accomodate more text material than one on more sophisticated concepts.

2. Course Outlines

The pages which follow are based on a set of course outlines prepared by Brandt, Maranzano, and Vogel (MF78-8234-3). The course sequencing immediately below is also adapted from that memo. R. F. Graveman's 1 July 1977 MF also influenced our thinking.

In each case, the set of course objectives and resources needed for presentation have been modified considerably. Suggestions for decomposition into modules and short courses were added, and, since this committee is addressing the needs of technical staff members only, comments on the assumed audience for each course have been omitted.



UNIX for Beginners

Prerequisites:

None

Goals:

Upon completion of this course students should be able to:

1. Login to UNIX.
2. Use the terminal correctly.
3. Use fundamental shell facilities.
4. Make use of basic UNIX commands such as: chmod, write, who, ln, mail, ls, cat, cp, rm, mv, cd, mkdir, rmdir, date, passwd, wc, cal, dc, and spell.
5. Understand the tree structure of the file system and the customary uses of /bin, /usr, and /etc as directory names.
6. Use the text editor for creating files and making simple changes to them.
7. Redirect I/O and use pipes.
8. Obtain access to currently available resources (people and documents) for reference, assistance and further study.
9. Appreciate what led to UNIX being developed and how it has been used within Bell Labs.
10. Use *learn* for further study.

Resources and Format:

Memos:

1. UNIX for Beginners - B. W. Kernighan
2. Tutorial Introduction to the UNIX Text Editor - B. W. Kernighan
3. PWB/UNIX R. F. Roadmap - J. R. Mashey and/or UNIX Newcomers Guide - R. F. Rosin.
4. UNIX Programmer's Manual augmented with an extensive roadmap for new users.

Presentations:

Approximately 3 sessions given in one week with time between sessions for hands-on experience.

Modules and Short Courses:

This topic could be encapsulated in one module or be broken down into three distinct units based on the units of presentation which were developed.

UNIX Word Processing Tools

Prerequisites:

1. UNIX for Beginners

Goals:

Upon completion of this course students should be able to:

1. Use advanced features of the editor.
2. Prepare documents using the -mm macro package
3. Generate tables with tbl.
4. Generate equations with eqn and neqn.
5. Understand the role of and interaction with a word processing center.
6. Use basic BTL document procedures (file numbers, standard formats, etc.).

(Those in need of more extensive word processing capability, such as clerical personnel or technical staff with special needs, should also be able to:)

7. Use the typographical error detection programs "spell" and "typo".
8. Observe good time-sharing practices.
9. Understand basic nroff and troff commands such as .br, .sp, .in, .ti, .ta and .fi.
10. Create macros and understand the pros and cons of doing so.
11. Use related shell concepts.
12. Cope with the limited diagnostic messages produced by nroff and troff.

Resources and Format:

Memos:

1. Troff Made Trivial - B. W. Kernighan
2. Advanced Editing on UNIX - B. W. Kernighan
3. Typesetting Mathematics - B. W. Kernighan, L. L. Cherry
4. TBL - A Program to Format Tables - M. E. Lesk
5. Nroff/Troff User's Manual - J. F. Ossanna
6. PWB/MM - Programmer's Workbench Memorandum Macros - D. W. Smith and J. R. Mashey
7. Typing Documents with PWB/MM - D. W. Smith, E. M. Psikorik
8. Appropriate tutorial material and examples are needed.

Videotape:

Presentations of various basic commands and options would probably be acceptable in this medium provided a qualified person were also available during presentation to handle questions.

Computer aided Instruction:

ed, spell, typo, file manipulation, good practices.

Presentation:

3 sessions/week, 5 weeks (1 week -mm, 1 week tbl and eqn, 1 week nroff/troff, 2 weeks macros and miscellaneous.)
Approximately 3/4 of each session would be practice and review.

Modules and Short Courses:

Clearly short courses would be appropriate here in recognition of the differing needs of various kinds of people who might use this material. The goals outlined above offer the beginnings of a likely breakdown into both courses and modules.

C Programming Language

Prerequisites:

UNIX for Beginners or equivalent.

Goals:

Upon completion of this course students should be able to:

1. Write simple C programs using standard library functions in recommended programming style.
2. Use C language constructs such as:
   - declarations
   - structures
   - assignment
   - if, else, for, while, do, switch
   - pointers, members of structures
   - function calls
3. Understand and use UNIX C compiler and link editor options
4. Use basic C programming aids; such as, libraries, cref, ar, nm, lint.
5. Make elementary use of adb for C program debugging.
6. Use standard I/O.

Resources and Format:

Books and Memos:

1. The C Programming Language - B. W. Kernighan & D. M. Ritchie.
2. Elements of Programming Style - B. W. Kernighan
3. Software Tools - B. W. Kernighan
4. UNIX Programming (with significant updates) - B. W. Kernighan & D. M. Ritchie
5. ADB Tutorial - J. F. Maranzano, S. R. Bourne
6. Tutorial material on libraries and programming aids. (to be written under the sponsorship of a UNIX Education Group).

Video Tape:

Several units covering such topics as
introduction to software principles
introduction to C
language constructs organized as in 1 above,
compiler and linker
aids for program development and debugging
introduction to use of ADB

Presentations:

2 or 3 sessions per week with assigned problems and workshops to discuss solutions.

Modules and Short Courses:

This course decomposes into three shorter courses:

Introduction to programming - corresponding to Chapter 1 of Kernighan and Ritchie with UNIX for Beginners as a prerequisite. This would normally be taken by people with traditional engineering backgrounds.

Basic C - Corresponding roughly to Chapters 2 - 4 of K&R. This course would also prerequire UNIX for Beginners and previous exposure to some higher level programming language such as obtained in the preceding course. This would normally be taken by newly hired engineers with some exposure to software.

Advanced C - Corresponding to Chapters 5 - 8 of K&R and additional material referred to above under Goals and Memos. Prerequisite background would be knowledge of a modern higher level language, as might be obtained in the previous course, and UNIX for Beginners. This course would also serve as an entry vehicle for newly hired computer science graduates.

These courses could be taken in immediate succession or with intervening gaps. The first two courses might be thought of as corresponding to present INCEP offerings 8P100 and 8L107.

Further decomposition into to modules could follow along the lines of Kernighan and Ritchie.

UNIX System Operations

Prerequisites:

1. UNIX for Beginners
2. Permission of instructor (enrollment limited to those with need to know).
3. C Programming and Effective Use of UNIX might well be pre- or co-requisite.

Goals:

Upon completion of this course students should be able to:

1. Use console switch registers on PDP 11's
2. Manipulate tape drives
3. Mount and dismount disk packs
4. Perform system boot and set the date
5. Process arbitrary format tapes
6. Run accounting programs, add new users, remove users
7. Run and interpret file system checks; differentiate between perfect, acceptable, and catastrophic conditions.
8. Initiate shutdown and backup procedures
9. Retrieval of files from backup media
10. Mount/unmount filesystems; use major/minor device numbers; manage tty's (turn on and off, manage type and speed parameters).
11. Take a dump after a system crash, perform simple crash analysis, initiate restart.
12. Use and interpretation of error logging
13. How to interface with hardware service contractor
14. Manage inter-installation communication.
15. Use daemons.
16. Manage user privileges and security
17. Recognize problems with and manage physical resources; paper, disk space, etc.
18. Locate sources of system errors; harware/software, specific drivers, etc.
19. Refer users to sources of information specific to their needs.
20. Manage login messages and broadcast messages.

Resources and Format:

Memos:

1. PWB operations manual
2. UNIX Programmer's Manuals
3. Extensive additional documentation is needed in this area.

Presentation:

1. Hands-on training required (video tapes inappropriate) 3 sessions/week for 3 weeks.

Modules and Short Courses:

It is not clear whether a hands-on training program such as this can or should be presented as a series of short courses. Definition of appropriate modules will probably become clear as the course is developed.