A UNIX MICRO REQUIREMENT FOR COMPUTER SCIENCE MAJORS

Edward A. Fox
Sandra E. Birch

May 1986

TR-86-11

A UNIX† MICRO REQUIREMENT FOR COMPUTER SCIENCE MAJORS

Edward A. Fox
Sandra E. Birch

Department of Computer Science
Virginia Tech
Blacksburg VA 24061
Telephone (703) 961-6931

Abstract

In the fall of 1985, over 230 freshmen majoring in computer science at Virginia Tech purchased microcomputers running the UNIX operating system. This event had been planned since the spring of 1983, at which time the faculty of the Department of Computer Science voted to institute a requirement of personal computers for computer science majors. During the summer of 1984, a commitment to requiring UNIX was integrated into the plans. A Request for Proposal for such systems was distributed to vendors in the fall of 1984, and selection took place early in 1985.

This paper reviews the planning and decision–making process, and explains why UNIX was chosen as the most suitable system to use in the training of computer science students. Implementation of the microcomputer program is discussed, including revisions to the freshman curriculum and the distribution of required hardware and software.

CR Categories and Subject Descriptors: K.3.2 **[Computers and Education]:** Computer and Information Science Education; K.6.2 **[Management of Computers and Information Systems]:** Installation Management - *computer selection;* K.8 **[Personal Computing]**

General Terms: Economics, Management

Additional Keywords and Phrases: computer science majors, microcomputer selection, personal computer requirement, undergraduate computer instruction, UNIX

† UNIX is a Trademark of AT&T Bell Laboratories, Inc.

Introduction

In the fall of 1985, more than 230 freshmen beginning their studies in the Department of Computer Science at Virginia Tech purchased personal computers running the UNIX[1] operating system. This event was the result of extensive planning by the Department of Computer Science, which decided in the spring of 1983 to require the purchase of microcomputers.

Virginia Tech is the first major U.S. public university where students have been required to buy microcomputer systems running UNIX System V. This paper traces the history relating to the requirement, explains key features of UNIX, explores the rationale for the decision, and describes the implementation of the program.

History

There are two aspects to the decision to require students to buy computers running UNIX: first, the microcomputer requirement, and second the choice of the UNIX operating system. Since these two matters were considered and decided consecutively, an historical account is in order.

The Computer Science faculty voted in the spring of 1983 to require students to buy personal computers. This would assure students of better access to computing facilities as the University's public facilities were fast becoming saturated. Students would have maximal opportunity to learn to use computers as tools for problem solving and to also use computers for other coursework, eg., word processing in English classes. Determination was made that almost all of the courses in the Computer Science curriculum could be taught relying on reasonably powerful microcomputer systems. Reports of successful efforts at other universities encouraged the planners further.

At that time, UNIX systems were too expensive to be purchased by students. A standard Request for Proposal (RFP) was prepared in the fall of 1983, but was not distributed since the University had just completed a similar process for the College of Engineering, with the IBM PC family of systems being selected. It had been stated that the PC Junior would be adequate for engineering students. However, the PC Junior was considered to be too limited and too slow for computer science students. Thus, the department chose to wait a year before preparing another RFP.

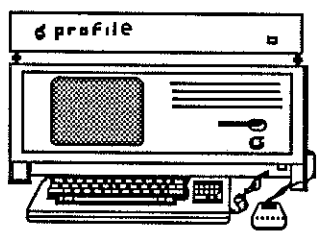
---

[1] See one of many UNIX books (eg., [Bourne 1982], [Kernighan & Pike 1984]) or "UNIX History and Features" below.

In the fall of 1984 an initial draft of the RFP was prepared, calling for personal computers running UNIX. Based on trends in the industry, it was projected that by 1985 such a system might become more reasonably priced for students, given suitable quantity discounts for educational institutions. The RFP was refined and distributed late in 1984. Certain hardware features were required, such as a main memory size (RAM) of 512K bytes. However, the most stringent stipulations were in the area of software and price. The RFP stated that all hardware and software costs must total no more than $2500, and that a package price for two years of maintenance be $500 or less. These figures were determined by the Virginia Tech administration, and reflected the philosophy that the cost of an education at a state university should be kept as low as possible. Software requirements stipulated full microcomputer versions of UNIX, either Berkeley's Standard Distribution 4.1 or AT&T's System III, along with compilers for C, Fortran, and Pascal. The three languages required were important since C is commonly used with UNIX, Fortran is needed in numerical analysis courses, and Pascal has been and will continue to be the primary high–level language used by Virginia Tech students during the first two years of their undergraduate training. (See [Feuer & Gehani 1982]) for a comparison between C and Pascal, to better understand this choice.)

Several companies met the requirements of the RFP. A committee of Computer Science faculty, staff, and students thoroughly tested a number of hardware and software packages. A best and final offer was allowed, and the selection of a special system from Apple Computer, Inc., and UniPress Software, Inc., was finally approved by the Computer Science faculty. Two years of AppleCare maintenance, supported by University personnel in Communications Network Services (CNS), was offered. The system included a Mac-L (a Mac-XL configuration with an external Apple Profile 10M disk instead of an internal drive) with 1M of RAM (user memory with UNIX running is over 750K bytes), one 400KB diskette drive, and a mouse. Figure 1 indicates how the system appears.

Figure 1: Mac-L System



Apple software included MacWorks, MacWrite, and MacPaint. UNIX System V, Assembler, and compilers for C, Fortran, and Pascal were provided by UniPress Software Inc., as the distributor for several other companies. A separate purchase of UNIX System V (source code) for Motorola 68000 processors was made from AT&T by the Department of Computer Science. Access to this software allows use of text processing programs and the addition or fixing of other UNIX routines. This follows a long tradition of computer science departments having the capability of adapting UNIX to special needs and requirements.

Apple Computer, Inc. and UniPress Software, Inc. donated a number of systems to the University. Most of those are operating in the University's public laboratories. Non-computer science majors taking computer science courses, as well as upper-classmen and graduate students, utilize these facilities. In addition, all students have access to printers in these 24-hour laboratories and can use the microcomputers there for small editing tasks. More than a dozen of the donated systems were assigned to the Department of Computer Science so that faculty, staff, and graduate assistants could learn UNIX or become familiar with this particular version, and prepare for their teaching responsibilities.

UNIX History and Features

UNIX is an operating system which has a large number of associated software tools aimed at supporting program development, text processing, and other activities. Before giving specific reasons as to why UNIX was chosen as the proper software environment for training computer science majors, it is of value to review the history and features of UNIX. Those familiar with this information may wish to skip to the next section.

Since the first version of UNIX was developed in 1969 by Ken Thompson of Bell Laboratories Research Group, many enhancements have been made, leading to a host of products. Dennis M. Ritchie and many others joined the original effort. The name UNIX is a pun on MULTICS, the system developed over more than a five year period through joint efforts at MIT, GE, and Bell Laboratories [Organick, 1972]. UNIX was strongly influenced by MULTICS, but was originally implemented by a much smaller group in a much shorter time frame.

The C programming language [Kernighan & Ritchie, 1978] was developed at Bell Laboratories to support UNIX, and was used almost exclusively in the third version of that system, completed in 1973. Most system development work undertaken in the UNIX environment has been in C, and C is now used for similar efforts under other operating systems such as MS-DOS and VMS.

UNIX Version 7 was distributed in 1978, for PDP-11 and Interdata 8/32 systems; the first author of this paper began using UNIX at that time. AT&T later released UNIX System III and V, and the University of California at Berkeley prepared other versions: 4.1 and 4.2BSD for VAX, and 3BSD for PDP-11 series. Many other names have been given to licensed "ports" of UNIX to a variety of computers, including: XENIX, VENIX, and ULTRIX.

UNIX systems have a kernel, the central core of the operating system, which supports a set of system calls so that user programs can have desired tasks carried out. UNIX kernels manage memory, processes, and I/O; recent efforts have aimed at standardizing UNIX at the kernel interface level. Interactive users, however, typically communicate with a "shell" program that surrounds the kernel; they use the Bourne shell (sh), developed at Bell Labs, or the C shell (csh), developed at Berkeley. The shell understands commands and can be programmed [Bourne, 1978].

At the shell level, UNIX provides many capabilities. A hierarchical file system, where all hardware elements are included as "special files" in the "/dev" directory, contains the system and user data. Device independent I/O simplifies many programming tasks. Managing processes is straightforward, leading to an elegant style of computer use whereby tasks are broken into subtasks which function concurrently. Pipes are easily set up so one process can transform input to an intermediate form which is then handled by a sequence of one or more "filters;" if each process performs a single function then the resulting pipeline effects a composition of those functions. Processes can be submitted for background execution in parallel with foreground (interactive) commands so users have the ability to learn about and control simultaneous job processing.

There are hundreds of UNIX routines available as commands which can be executed interactively or through files of commands, called "shell scripts." System commands display and manipulate files or directories. Some do sorting, comparing, or looking for patterns. There are editors, compilers, and text formatters. Since most UNIX systems have storage space for different users, there are also facilities for mail handling.

By 1983, the two most complete versions of UNIX were System III developed at Bell Labs, and 4.1 BSD from Berkeley. Both provided very good environments for program development, text processing, and for other computing activities. System III was emerging as the most common form for commercial applications, while 4.1BSD was the typical UNIX system used in computer science research. Either was considered fully adequate for our students. While these versions have been improved with releases of System V and 4.2 BSD, respectively, they were not widely available on personal computers until late 1985. Choosing one or the other may not be very critical since efforts of AT&T and Sun Microsystems Inc. indicate that System V will eventually be enlarged and improved to have all important features now in 4.2 BSD [Chandler, 1985].

Rationale for a UNIX Requirement

UNIX, an operating system that has become well known in recent years, has been hailed by some and criticized by others. This section summarizes many of the reasons the department selected this system.

A. Familiarity with UNIX is likely to directly benefit students entering the job market

1) UNIX is becoming a standard operating system for micros, minis, and large systems

While UNIX first became popular as an operating system for Digital Equipment Corporation minicomputers such as the PDP-11 and VAX series, it now operates on personal computers and supercomputers as well. AT&T has made a firm commitment to support UNIX System V, and Digital Equipment Corporation supports UNIX 4.2BSD under the name of ULTRIX. Because of this support by leading computer manufacturers, the availability of versions of UNIX on most types of computers, and the almost fanatical enthusiasm of many UNIX users, UNIX is becoming the closest approximation to a standard operating system that has yet emerged. As a result, students studying UNIX and/or C will have an obvious advantage for future employment.

2) There is a tremendous demand for people experienced in UNIX and C

In the spring of 1985, a petition was circulated by the department's graduate students requesting that a graduate level course on UNIX and C be offered. Nearly 30 are now enrolled in the course; some having postponed taking other required courses so as not to miss this opportunity. Clearly, computer science graduate students feel UNIX and C are important topics to master. Many have explained this interest as stemming in part from the awareness that these skills will help them obtain desirable jobs.

Programmers experienced in UNIX are indeed in short supply and this trend will continue during the next several years as the number of UNIX systems steadily grows. The C language is being used with other operating systems as well, as a highly effective implementation language, so the need for good C programmers is also rapidly growing.

B. Learning C and UNIX helps one learn computer science methods, and vice versa

1) Regular expressions are utilized in the editors, in filters (eg., grep), and in the shells, so students learn to actually use ideas that are often taught only in the abstract in formal language theory classes.

2) Work with the file system helps develop understanding of tree structures and hierarchies.

3) Structured programming of the shell helps teach good software development practices; many other command languages lack such elegance [Bourne, 1978].

4) The C language is a good vehicle for systems programming because of its handling of data types (including casts, structs, typedefs), its use of pointers, the ability to manipulate bits, and the storage flexibility of unions [Gehani, 1985].

5) Extensions like C++ can be taught as well, once they are more fully developed [Stroustrup, 1985].

6) UNIX tools such as LEX and YACC are useful in compiler courses and motivate students to use what they have learned about language theory to perform useful tasks.

7) UNIX text processing software increases student exposure to handling textual data in elegant ways - pattern matching capabilities, relational algebra operations (eg., join), report preparation (eg., AWK), and macro processing.

8) The style of UNIX development, where prototypes are quickly constructed by composing tools, is important to learn as an example of good software engineering. Tools like MAKE [Feldman, 1979] help with medium size development efforts that are possible in university settings.

9) UNIX use requires understanding of the notion of process, which helps students comprehend the functions of an operating system.

10) Students must become system administrators, and thereby learn even more about such subjects as operating systems, backup procedures, and handling different devices.

11) Textbooks such as [Comer, 1984] are about systems very similar to UNIX, and are widely used in operating systems courses. Clearly the popularity of UNIX was one of the motivating factors of many authors' efforts to build UNIX-like systems, and to use them to teach students about the theory and practice of operating systems. A number of textbooks, e.g. [Peterson & Silberschatz, 1983], have chapters about UNIX to illustrate some of the important concepts of modern operating system design.

C. Using UNIX and C allows interchange of software developed in research projects

1) All of the major software research and development efforts of the Computer Science Department at Virginia Tech are being done with C and/or UNIX. Students will be given software resulting from some of those efforts. For example, the first author of this paper is supervising students developing: an adaptation of LOGO for the Mac-L, a microcomputer version of part of the SMART information retrieval system, and a front-end to the Virginia Tech Library System.

Another faculty member is supervising development of a Prolog compiler that will also be ported. Data base software will be developed and the second graduate course in that area, CS5662, has for the spring of 1986, made knowledge of the C language a prerequisite.

2) UNIX software is relatively complete so students need not buy a number of expensive packages. UNIX itself is expensive, but provides a large number of services. While special arrangements had to be made to obtain compilers for Pascal and Fortran, a typical UNIX software package provides languages such as C and Snobol. There are games, text processing routines, source code maintenance facilities, and macro processors among the available packages.

3) UNIX software from other computer science departments can be utilized. UNIX is available in a great many computer departments around the world, so the base for development is very large.

Some negative factors considered in the decision–making process were also addressed.

1) While running UNIX requires expensive hardware and software, the price/performance ratio is actually lower than for less expensive systems, and having a 10M hard disk and 1M of RAM ensures rapid response even on large problems. Students can also use their system for almost all their computer science work as well as work in other disciplines.

2) UNIX systems are not able to run the many commercially available software packages, such as are available under MS-DOS, but these computers also support the Macintosh operating system with MacWrite and MacPaint included in the package. Students also have a monetary incentive to do their own application development for simple tasks, or else will learn how to combine UNIX tools to accomplish their goal.

3) Administering a UNIX system is difficult for new computer scientists. However, learning these techniques helps in the training of students and will instill confidence - the only problem is during the initial learning period when this burden weighs most heavily. The time to become reasonably proficient in UNIX may well be as much as six months, but this is offset by the fact that students will own their systems for four years.

4) Only computer scientists are likely to have a UNIX requirement in the near future. However, this can be viewed as a way that students learning UNIX now will get a head start on their colleagues as UNIX continues to rapidly proliferate.

5) Students transferring out of computer science may have difficulties selling their systems. While this is a possibility, there are many more students trying to transfer into the curriculum, so students should be able to sell their systems to those incoming students.

Implementation of the Microcomputer Requirement

Selection of a microcomputer and software package was only the beginning. No less than fifteen departments in addition to the Department of Computer Science have been involved in the

implementation of the micro requirement. The Personal Computer Auxiliary, a unit of the University's Communications Network Services, is playing one of the most active roles.

First, financial arrangements had to be determined. Since Virginia Tech is the state's land-grant university, the cost of an education should be within the reach of most residents. An arrangement with the Virginia Tech Education Foundation, a non–profit organization, allowed students the option of purchasing their microcomputers over a two year period through monthly payments. The Admissions Office, anticipating that the cost of the computer system would be a deterrent to prospective incoming freshmen, issued more offers of admission than in previous years. A greater percentage of students actually accepted and the 1985-6 freshman class is 60 students larger than previous freshman classes. The requirement appears to have encouraged more students than usual to accept admission to the department.

All entering freshmen were sent a letter from the department head in April explaining the program. This preliminary contact was followed in May by detailed commitment and order forms mailed by the Personal Computer Auxiliary of CNS. By mid-July, orders for 230 systems were mailed to Apple Computer, Inc. and UniPress Software, Inc. with delivery expected in August. Three staff members from the Department and CNS spent three days "burning-in" all the hardware and loading the UNIX software and compilers on each of the hard disks. To serialize individual software packages, three code numbers were retrieved from each boot diskette and sent to Unipress, Inc. for authorization numbers. This process saved a great deal of time and trouble as students did not have to contact Unipress individually to serialize their own systems.

Next, a team of upperclass students and graduate students were trained as installers and "trouble-shooters". Popularly known as the "Swat-team," the students were headquartered in the Computer Science Consulting Lab where a "hot-line" telephone was installed. Distribution of the systems was conducted during the three days preceding the start of classes.

Following a distribution schedule they had received in early September, students first attended a 2-hour seminar during which the "care and handling" of a Macintosh L was discussed by Computer Science and Apple Computer, Inc. staff members. Examples of the UNIX and MacWorks operating systems were briefly demonstrated. Special Apple T-shirts were distributed before the students moved on to the actual distribution site in the student union building. No system was issued unless the student had paid the proper fees and showed evidence of seminar attendance (a series of labels coded with the student's name which were used to identify his equipment boxes after pick-up).

Appointment times for equipment pick-up were determined by dormitory assignments. To facilitate delivery of students and their systems, all students in a single dormitory received their

equipment at approximately the same time and rode together in special vans to their dormitories. Members of the University's ROTC were available to carry equipment to the awaiting vans. Swat-team members were then waiting at the dormitories to move the equipment into the students' rooms and install the microcomputers.

In less than three days, 230 systems were delivered and installed. By the end of the distribution period, more than 230 students were enrolled in a special one-hour per week course on UNIX while also beginning their studies of Pascal in the first two introductory Computer Science courses. Whereas in previous years, students were using Pascall compilers on VAX 11/780 mainframes, they are now computing on their own Macintosh Ls. The Motorola 68000 assembler is being taught in the third course of the introductory sequence, instead of the VAX assembler. Each quarter the department plans to add at least one more course to the curriculum which will be completed on the Macintosh L with the UNIX System V operating system.

Upperclassmen interested in UNIX and C can take the senior seminar, which in winter quarter will cover C and in spring will cover UNIX. These are only one credit courses, but will convey the most important aspects of the subjects. Thus, starting this academic year, all graduates can become familiar with C and/or UNIX.

An advanced graduate course in fall 1985 is focusing on UNIX and C. Some twenty projects are underway to help with various UNIX activities. For example, several students are working with the Macintosh bitmap display and mouse so they can be better utilized through UNIX. These students are a demonstration of how many individual efforts are being made to help the overall UNIX effort in the Virginia Tech Department of Computer Science.

Summary and Conclusion

In the fall 1985, freshmen students majoring in the Virginia Tech Department of Computer Science are the first undergraduates in a public university required to buy their own personal computer running UNIX System V. It is expected that being trained as computer scientists with such a powerful tool will lead to a much better education in their major field.

Numerous individuals at Virginia Tech have worked together to make this happen, and we believe that the atmosphere of cooperation will spread to more and more of the students as they share their experiences in using a powerful microcomputer and well known operating system. Many further lessons will be learned from this effort at Virginia Tech, and undoubtedly, other universities will follow suite with similar requirements for UNIX, in future years.

References

Bourne, S. R. (1978). The UNIX Shell. *Bell System Technical Journal*, 57(6), 1971-1990.

Bourne, S. R. (1982). *The UNIX System*. Reading, MA: Addison-Wesley Pub. Company.

Chandler, D. (1985). The Monthly Report. *UNIX Review*, 3(10), 8-18.

Comer, D. (1984). *Operating System Design: The XINU Approach*. Englewood-Cliffs, NJ.: Prentice-Hall.

Feldman, S. (1979). Make - A Program for Maintaining Computer Programs. *Software - Practice and Experience*, V. 9, 255-65.

Feuer, A. and N. Gehani (1982). A Comparison of the Programming Languages C and Pascal. *ACM Computing Surveys*, V. 14, N. 1 (March), 73-92.

Gehani, N. (1985). *C: An Advanced Introduction*. Rockville, MD: Computer Science Press

Kernighan, B.W. and R. Pike (1984). *The UNIX Programming Environment*. Englewood-Cliffs, NJ: Prentice-Hall.

Kernighan, B.W. and D. M. Ritchie (1978). *The C Programming Language*. Englewood-Cliffs, NJ: Prentice-Hall.

Organick, E. I. (1972). *The Multics System: An Examination of its Structure*, Cambridge, MA: MIT Press.

Peterson, J. and A. Silberschatz (1983). *Operating System Concepts*. Reading, MA: Addison-Wesley.

Stroustrup, B. (1985). *The C++ Programming Language*. Reading, MA: Addison-Wesley.