INTRODUCTION TO XENIX

A COMPUTER AIDED INSTRUCTION PACKAGE

GRADUATE PROJECT

BY

JULIA MCCLAUCHLIN
August, 1989

COMMITTEE MEMBERS

Ms. Charlotte Busch,
Chairperson

Mr. Allen Bush,
Member

Dr. Roy Ellzey,
Member
Abstract

This project is a Computer Aided Instruction (CAI) package to aid in teaching introductory Xenix concepts. The package includes tutorials and tests. Topics which are covered are the history of the Unix/Xenix operating system, overview of the Xenix environment, logins and passwords, some useful commands with which to get started, file and directory overview, directory manipulation, file manipulation, redirecting input/output, and pipelines. The package is designed so that the lessons are set up as separate files and are self-paced.
Background

A large responsibility of the Information Center at the Corpus Christi Army Depot (CCAD) is that of training the employees to use computers and various software packages. In the past, training has been offered in Lyrix, a word processor for a multi-user Unix based system; Symphony, an integrated package (word processor, data base management system, spread sheet) for the PC; Informix, a database management system for the Unix environment; and Xenix, the multi-user operating system based on Unix.

The format of the classes has been that of traditional classroom instruction. The Learning Resource Center (LRC) has been equipped with PCs and with terminals which access the Local Area Network, so that hands-on experience is possible during any class. No CAI packages for any of the aforementioned software have been available in the past.
Introduction

Since the computing environment at Corpus Christi Army Depot is primarily a Xenix environment, most users of computers at the Depot have a real need to know Xenix. Classroom instruction in the use of Xenix is periodically available. Because of the large number of employees and the relatively small staff of instructors, however, it is difficult to keep pace with training needs using only traditional classroom instruction.

The Introduction to Xenix software package is a tutorial approach to CAI. According to Rockart and Morton (85), a tutorial approach is designed in large part to replace a teacher in presenting facts. Part of a tutorial's intention, they state, is to relieve the teacher of much of the burden and to provide practice work.

The CAI package on introductory Xenix can be used individually by employees and thus will help the effort to keep up with the demand for training. It can be used at the employee's convenience at his own work area in order to strengthen his background or to reinforce material presented in the classroom.

A user can progress through the lessons at his own pace. Within the tutorials, he can scroll backwards to review earlier panels and can easily repeat earlier lessons whenever he chooses.
Even though these tutorials presently have Xenix as their subject matter, the general framework of programs could be used to present other subjects. Test questions on other subjects could be entered using a modified version of the maintenance program, and tutorial panels could be easily constructed using a text editor such as vi. Thus, the package can help to relieve some of the pressure in the effort to train users in a variety of other subjects.
General Overview - Student Tutorial Programs

The student tutorial program is written in C and is designed to present menus, receive user responses, and read and manipulate the appropriate tutorial and test files. The first program presents the student with a main menu of eight lessons from which to choose. The main menu program then calls the submenu program with the appropriate tutorial and quiz file as arguments. The student chooses whether to view the tutorial on the lesson or to take a quiz over the lesson material.

If the student chooses to view the tutorial, the tutor function in the program presents the tutorial panels on the screen. The function reads the tutorial file panels into the screen array. The show_screen function is then called upon to present the first screen. Afterwards, the show_screen function calls itself, moving forward or backward through the array as the student chooses to scroll forward or backward.

If the student chooses to take the quiz on the lesson material, the test function in the program reads the questions from the question file into an array of structures, and then presents the test questions upon the screen randomly, making use of the system time as a seed to the random function. After the last question is presented, the program prints the number of correct answers out of the total number of questions.
Lessons have been developed so as to present introductory Xenix information in a clear and concise manner. There are eight lessons, each of which consists of its own tutorial and test, and is designed to be self-paced. When the student chooses the test option, a set of test questions is selected at random from the bank of questions for that lesson. The lesson topics are as follows:

Lesson 1. History of Unix/Xenix
Lesson 2. Overview of the Xenix environment.
Lesson 3. Logins and Passwords
Lesson 4. Useful commands with which to get started
Lesson 5. Overview of files and directories
Lesson 6. Working with directories
Lesson 7. Working with data files
Lesson 8. Redirecting input/output

Pipelines
The test maintenance program is designed to provide easy maintenance of the test files. With this program, it is possible to add questions, update questions, delete questions, query the test bank, or print a test. The program itself displays the main menu with a choice of lessons. When the user chooses a lesson, that test file is opened and the questions are read into an array of structures. The options menu is then presented for adding, updating, deleting, or printing questions. All adding, deleting, and updating is done to the questions in the array of structures in main memory. Upon exit, the set of questions is written back to the file only if changes have been made.

The add_quest function allows for the addition of questions to the test bank. A maximum of twenty-five questions is allowed for each lesson. The function assigns the proper question number and then prompts for the question text, the four answer choices, and the correct answer. At the end, the number of questions added is displayed.

The chg_quest function allows for the updating of any questions. It prompts for the number of the question to be updated and then displays that question along with its four answer choices and the correct answer. The user is then given a chance to enter a change for any field or to press return to leave a field as it is.
The del_quest function allows for the deleting of any questions from the question bank. The function prompts for the number of the question to be deleted. That question is then displayed on the screen. The user then presses a 'd' to mark the question for delete or presses a 'u' to leave it in the bank. If deletion is chosen, an asterisk is put in the delete mark field and the question is not written back to the file upon exit.

The qry_quest function allows the user to view the test file on the screen. The questions are shown on the screen five at a time. The user may choose to view any screen of questions by entering the screen number or choosing 0 to exit.

The prt_test function allows for the printing of a test on a slaved printer or a shared printer. The user may choose to print all of the questions, a random group of questions, or a set of questions of his own choosing. In addition, the user may choose to print a key for the test. The questions are written to a file and then sent to the printer.
File Descriptions

Files used within the Xenix CAI package are eight tutorial files (one for each lesson topic) and eight test files. The tutorial files are text files created using the vi editor. The test files are created and updated from a C program which receives its input from the keyboard. The fields in the test files are as follows:

<table>
<thead>
<tr>
<th>field name</th>
<th>length in bytes</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qno</td>
<td>2</td>
<td>question number</td>
</tr>
<tr>
<td>quest</td>
<td>max 80</td>
<td>question</td>
</tr>
<tr>
<td>corr</td>
<td>1</td>
<td>correct answer</td>
</tr>
<tr>
<td>ch1</td>
<td>max 50</td>
<td>first answer choice</td>
</tr>
<tr>
<td>ch2</td>
<td>max 50</td>
<td>second answer choice</td>
</tr>
<tr>
<td>ch3</td>
<td>max 50</td>
<td>third answer choice</td>
</tr>
<tr>
<td>ch4</td>
<td>max 50</td>
<td>fourth answer choice</td>
</tr>
</tbody>
</table>
Environment of the project

The computing equipment supported by the Information Center at CCAD includes Vax, Sperry, and Intel systems. The Vax 11/750 at CCAD, running under Unix, is a large Phone-Net server which utilizes a Local Area Network to provide electronic mail service to sixty-six Intels running Xenix and five Sperrys running Unix.

The Sperry systems (5000/80 type) are mid-sized computers that serve from sixteen to sixty-four or more users. They run under the 5000 Series Operating System, which is based on the Unix Operating System V.

The Intel systems (286/310) run the Xenix operating system and are set up as LANs for up to twelve users. Users access the Intels using Wyse 75 or Wyse 50 terminals, Sperry terminals, Zenith PCs, IBM 3163s, or Televideo 950s terminals.

The Xenix CAI package has been written in the C programming language and has been loaded on the Vax at the Depot. It is accessible to any users who have a Vax login from any of the various terminals mentioned above. Due to memory requirements of the student tutorial program, it is not possible to run the Introduction to Xenix package on the Intel computers.
CCAD Computing Environment
Logical Layout

DDN

LAN

Intel
Intel
Intel
Intel
Sperry
DDN Host

VAX DDN HOST

Sperry
DDN Host

Sperry

8 Sperrys
CCAD Computing Environment
Physical Layout
Student Tutorial Program

MAIN MENU
xtutor.c

OPTIONS MENU
submenu.c

TUTORIALS
tutor()

TESTS
test()
Test Maintenance Program

MAIN MENU
maint.c

OPTIONS
MENU
optmen()

ADD
add_quest()

DELETE
del_quest()

CHANGE
cng_quest()

QUERY
qry_quest()

PRINT
prt_test()

EXIT

all_quest()
rand_quest()
sel_quest()