Waterloo Microcomputer Systems for the 1980's

D.D. Cowan and J.W. Graham
University of Waterloo
Waterloo, Ontario, Canada

Introduction

Over the years the University of Waterloo has developed academic programs which have been rich in traditional material, but which have also had a pragmatic flavour. As a growing industrial nation, Canada has an insatiable requirement for people with state-of-the-art technical and managerial skills, and our government has encouraged us to enter into "mission-oriented" projects. The University of Waterloo was founded as a technologically-oriented university and as such has developed a broad requirement for computing.

Today we have a very large Computer Science program, and computer techniques are used in all faculties, wherever appropriate. In fact, most of the students regardless of discipline take at least one practical computing course during their undergraduate years. During the term, at least 25,000 accounted student "jobs" are run each day, but many more which run on the numerous computers not under the supervision of the Department of Computing Services, are not counted. This volume of computer use has created the following challenges.

(i) There are never enough facilities to provide for every requirement. It is, therefore, important to find the most economical means of providing computer services.

(ii) The services offered must be versatile, with many computer languages and flexible hardware facilities such as a selection of standard or special-purpose input-output devices.

(iii) The facilities must be easy to use to enable those with a non-computer background to be productive. In other words, the economies strived for in (i) must not be achieved at the expense of the student or researcher.

These challenges have been with us for 20 years; time has changed only the volume and applications. To meet the challenges, many systems have been developed at universities and colleges over the years: At the University of Waterloo they include such systems as WATFOR, WATFIV(6), WATBOL(4), cafeteria-style computing and WIDJET(5,8). To a certain extent these systems have become dated, and there is a constant search for better solutions. Currently, most universities are exploring the microprocessor and its application in the personal workstation with local area networks. This paper describes the evolution of various educational computing systems at the University of Waterloo and the development of microcomputer-based educational systems which are currently being used and created.

A Model for an Educational Computer System

Students in many classes which use computers are constantly learning new concepts or tools. Many of these concepts are tried on the computer and once mastered are used as building blocks on which to construct the next concept. In other words, students are constantly learning new ideas and rarely have an opportunity to consolidate their learning by constant repetition and application. These remarks actually apply to most subjects and are not just representative of computer-related courses.

Such a learning environment tends to impose certain requirements on a computer laboratory which has been established for creating and running programs and data. The next few paragraphs describe many of the characteristics of such an environment.

An educational computer system should have friendly software and hardware. In other words, the software should be simple to use initially, new ideas should be added in easy incremental stages and error messages should attempt to pinpoint errors both during creation and operation of the program.

The computer system should have fast response consistent with the complexity of the task being performed. A student needs immediate re-inforcement to know that the work is proceeding correctly. If fast response is not present then the student forgets the material being learned and student learning efficiency is impaired.

The computer system should be economical to operate since more and more students are requiring access to computing facilities.

The system needs to be reliable, we all know how frustrated we become when the tools we use do not operate properly.

Ideally, an educational computer system should be load independent; performance should not seriously degrade from light to heavy loading of the system.

An educational computer system should supply extensive hardware and software facilities to support a comprehensive set of educational activities.

Finally, the system should allow data sharing but at the same time should be secure. In other words the teacher can allow sharing of information between teacher and student or between students, but there should be controls so that unauthorized sharing is not permitted. Providing this type of security on a large-scale time-sharing system usually leads to a complex operating system.

This list of properties describes an educational computing environment, it is not meant to be exhaustive but we believe many of the major issues are covered.

A History of Educational Computer Systems at Waterloo

To meet the challenge presented by the model many educational computer systems have been developed over the years at the University of Waterloo as well as other universities and colleges.

Compilers and interpreters such as WATFOR, WATFIV, WATBOL and Waterloo Pascal(1) were designed to be simple to use, provide comprehensive diagnostics and be responsive.

Once these software systems were developed, it was necessary to provide a delivery system for computing services that satisfied some of the other parameters of the model.

Cafeteria-style computing was one of the first delivery systems installed at Waterloo to satisfy the requirements of economy, responsiveness and reliability. Such a system consisted of card-reader-printer stations which were operated by the students.

As the cost of cards and paper increased and the cost of terminals decreased, it became more economical to provide a different form of delivery system. The WIDJET system consisted of a real-time editor operating on a minicomputer and supporting about 30 terminals. Students created programs using the editor and submitted them to a batch stream where they were processed by the appropriate language processor. The batch stream was usually on a large-scale computer connected to the mini by a communications line. The WIDJET system offered the advantages of cafeteria-style computing, but was lower in cost. A data-sharing capability was provided since programs and data could be stored on either computer and designated files could be made accessible to both students and teachers. Because computing power was being shared among several users system performance degraded as peak load was reached.

Microcomputer-Based Systems

A brief study conducted in 1979 confirmed that many of the jobs run on computers at Waterloo could be done using the computational capacity possessed by microcomputers or personal workstations. The major problem was the lack of proper software and the lack of appropriate input/output facilities. It seemed reasonable to develop the necessary software as we had been doing for years for the larger machines. However, the hardware presented new challenges. None of the inexpensive, mass-produced microcomputers had the appropriate hardware to operate our planned software, mainly because the memory was not large enough, and because there was insufficient flexibility, particularly with respect to input/ouput.

It was noted that on campus there were more than 1000 "dumb" ASCII terminals, mostly with CRTs and keyboards, and a study was initiated to consider the problems of their conversion to personal workstations meeting our specifications. The study led to the design of the microWAT(7,9), a prototype of which became operational in December 1980. Dozens were produced in 1981, and are in use at Waterloo and other locations.

The microWAT is a computer system of one or more circuit boards mounted on a rather simple bus. A typical sytem consists of 4 cards, namely the CPU card, 48K RAM card, 64K bank-switched RAM or 60K bank-switched ROM card and the IEEE-488 bus interface card. This system can be mounted inside most of our ASCII terminals, and uses their power supply. It converts the terminal into a personal workstation which has considerable versatility, with access to diskette drives, printers, plotters and extensive software systems. If desired or necessary, the microWAT can be mounted in its own chassis with its own power supply. In fact, it can be placed at a location remote from the terminal in case security is a problem.

At the same time as the devlopment of the microWAT, we investigated the possibility of expanding existing microcomputers by providing them with a large memory so that they could incorporate our planned software. We modified a PET microcomputer by adding 64K of bank-switched RAM, a 6809 microprocessor and an RS232 interface. This design eventually led to the Commodore SuperPET which is a personal workstation similar to the microWAT.

Waterloo micro Software Systems

Once the microcomputer design was chosen, it gave considerable latitude to the software developers. In most microcomputing systems, the language interpreters have to occupy the same 64K address space as the monitor, library and user programs and work area. This seriously restricts the size of the interpreter, with a corresponding negative impact on the scope of implementation of the software. With the bank-switching architecture, the software system can occupy whatever space it requires; the cost is simply another bank-switched RAM or ROM card when necessary.

This flexibility permitted us to develop extensive implementations of the following languages:

APL
BASIC
COBOL
FORTRAN
Pascal

These implementations are all interpreters, which provide a "friendly" environment for debugging. Of course, APL is always implemented as an interpreter, and BASIC usually has been. However, it is not usual to find COBOL, FORTRAN and Pascal implemented as interpreters, and their users have reported a level of satisfaction usually experienced only by APL and BASIC users.

These language systems operate using a common library of routines, many of them related to input/output and other system specific functions. This common library permits a simple interface to whatever operating system is being used. This will be further discussed in a subsequent section.

The system also contains an editor which can be used either in a full-screen or line-oriented mode, depending upon the ASCII terminal being used.

A 6809 microprocessor Assembler Development system was also prepared, and runs on both machines. It has a number of important features:

(i) It includes a special linker which permits the assembly language programmer to use the bank-switched hardware with literally no effort.

(ii) It uses an extensive assembler including structured programming facilities which permit the user to write "branchless" assembler code for ease of readability and subsequent maintenance.

(iii) It gives the user of the assembler access to the common system library mentioned earlier.

All of the Waterloo micro software sytems were written using a special systems development language called the Waterloo Systems Language(2,3). It is usually referred to as WSL (pronounced "whistle"), and this system also runs on many different maxi, mini and microcomputers including the microWAT, the IBM Series/1, the IBM 370, the DEC PDP/11, the DEC VAX and the IBM Personal Computer. In fact, the system library we have mentioned, is more appropriately called the WSL library, since it is available to WSL and all software which has been prepared using WSL. Of course, the library is written in WSL, and it follows that the WSL compiler is itself written using WSL. More details about WSL can be found in the article listed in the reference.

A Workstation Configuration

Both the microWAT and SuperPET are used in teaching laboratories at the University of Waterloo. About thirty of these machines are connected to a minicomputer (an IBM Series/1). The minicomputer has a large disk file and a printer and is connected to each microcomputer by a 9600 baud line. There is no local storage on the microcomputers, since the minicomputer acts as a disk server for the microcomputers.

Each microcomputer in the configuration is capable of running the various components of the micro Software System so that the student has extensive software facilities available.

This type of configuration provides all the facilities suggested by the model. Performance does not vary with load since all microcomputers are independent and can use the 9600 baud line simultaneously at full capacity without affecting any part of the system. Security between users is localized to the file system since each user has a computer and is only sharing a common storage facility.

Software Portability

When the 6809 was chosen as the microprocessor for the microWAT and SuperPET, we realized that microcomputers which incorporated other microprocessors such as the 68000, 8086, 8088 and Z8000 were about to be announced, and some were even available. It would, therefore, be unwise to prepare extensive software systems specifically for the 6809. The systems language WSL is meant to produce code which is portable. By using WSL to write the language interpreters, the editor, the library, the assembler development system, and in fact the WSL compiler itself, it should be a relatively easy job to move or "port" the system to other machines and interface it with their operating systems.

The Next Generation of Microcomputers

A number of new microcomputers which are based on the 8086, 8088 or the 68000 microprocessors have become available within the last year. We are currently working with a number of these microcomputers although in the last few months we have concentrated on the IBM Personal Computer. All the software described in the section on "Waterloo micro Software Systems" is operational on the IBM Personal Computer(10); of course the assembler is for the 8086/8088 microprocessor. This software was ported to this computer in about four months and we are quite pleased with its performance. The language processors and editor have also been augmented to take advantage of the powerful graphics and color processing capability of the Personal. In the Fall of 1982 we will have 32 IBM Personal Computers connected to a Series/1 minicomputer in a configuration similar to the microWAT and SuperPET systems mentioned in a previous section. An additional 32 IBM Personal Computers and a Series/1 are to be installed the first of next year.

We have also started to port the software to a number of 68000-based machines although the work is not yet complete.

Since the software is portable it has also been implemented on the IBM 370 under VM/CMS and on the DEC VAX.

Microcomputers as Systems Component

The previous sections generally describe the various hardware and software components of some of the microcomputer systems at Waterloo. This section is meant to outline some of the ways in which it can be integrated into our computing environment.

(i) A stand-alone personal computer: If you require a printer or diskettes for storage of data and programs, you can purchase off-the-shelf hardware which is supported by the WSL library.

(ii) A combination ASCII terminal and personal computer: Since the microcomputers contain an RS232 interface, it is possible to connect the microcomputer to a medium-or large-scale host computer. One application of this is to permit the user to employ the microcomputer in "pass through" mode. In this way the microcomputer becomes a "dumb" terminal to the host computer.

(iii) A personal workstation using host computer files: Since an RS232 interface connection is available between the micro and the host it is possible for software on the micro to read and write files on the host. This is accomplished by using a special simple program called HOSTCM (host communications module) which is a job running under the operating system. HOSTCM runs on several different computer systems including the IBM 370, the IBM Series/1, the DEC PDP/11 and the DEC VAX. The WSL library has a system facility which permits it to recognize file names with the prefix "HOST." When input/output is executed involving such a file, the WSL library communicates with HOSTCM using the RS232 interface, and the appropriate information is passed between the host and the microcomputer. Thus HOSTCM either causes the data to be written into a host computer file, or it reads the required data from that file and transfers it to the microcomputer. This feature, while limited by the speed of the communications line, permits the user to access host files when the application requires it. For example,

  (a) the microcomputer can have periodic access to a large data base stored on the host,

(b) the microcomputer can write a file to special peripherals such as a high-quality printer on the host, or

(c) the microcomputer can use the host file system for archiving and dearchiving of programs.

(iv) A component in a distributed processing system: The host computer and the microcomputer can carry on a conversation along the RS232 line. Thus a system can be developed which divides processing between the host and the microcomputer. Of course, two microcomputers can be used with each other in the same way.

(v) A host program-preparation system: Since all the language processors operate on the IBM 370 exactly as they do on the microcomputer, programs can be prepared and debugged on the smaller machine, and can eventually be run in production on the host machine which has larger memory and is faster. This feature is particularly useful for students who may run out of capacity on the microcomputer before their assignment has reached completion. They can transfer their programs and data to the IBM 370, and use the larger system to complete the project.

Future Plans

The microcomputer has proven to be a valuable contribution toward meeting the challenges described at the beginning of the paper. However, as has always been our experience, it has introduced new challenges. For example, it is clear that all such workstations should have access to a high-speed local area network (LAN) so they can share data and peripheral devices without having to use a large-scale computer as a host. This would not only be more economical, but faster as well. Experiments are now being conducted with early versions of off-the-shelf networks, and some effort in the latter category was the one described earlier using the IBM Series/1 as a file server and printer for about 30 microcomputers. Since September 1981 we have used such a system, and have installed a second one in January 1982. These installations are considered to be interim solutions until an effective and economical LAN is put into operation.

The microcomputers in this paper are only our first attempts at designing and building a personal workstation and software. We plan to port the micro software to other personal workstations, including ones designed at Waterloo and many which are now available on the open market.

References

(1) F.D. Boswell, The Waterloo Pascal Compiler, WATNEWS, University of Waterloo, November-December 1979.

(2) F.D. Boswell, Waterloo Systems Language, WATNEWS, University of Waterloo, March 1982.

(3) F.D. Boswell, Waterloo Systems Language, Tutorial and Language Reference Manual. WATFAC Publications 1982. ISBN 0-919884-00-8.

(4) Donald D. Cowan and J.W. Graham, Teaching Data Processing Using Cobol and WATPAK/C, WATNEWS, University of Waterloo, February 1977.

(5) Donald D. Cowan, The WIDJET System for the IBM Series/1, WATNEWS, University of Waterloo, July-August 1978.

(6) Paul Dirksen and Ian McPhee, Structured WATFIV and WATFOR-11, WATNEWS, University of Waterloo, January 1977.

(7) Carl M. Durance, The microWAT, WATNEWS, University of Waterloo, April 1982.

(8) J.W. Graham, The WIDJET System for the IBM Series/1, IBM Hardware Selection, Online Conferences Ltd., Uxbridge.

(9) J.W. Graham, and T.A. Wilkinson, Personal Workstations for Program Development, Proceedings CIPS Session 82, May 1982.

(10) Terry Stepien, The IBM Personal Computer, WATNEWS, University of Waterloo, July-August 1982.

(11) J.W. Welch, Waterloo microCOBOL, WATNEWS, University of Waterloo, May-June 1982.

(12) T.A. Wilkinson, Waterloo microSystems, WATNEWS, University of Waterloo, October 1981.

(13) T.A. Wilkinson, The Host Connection, WATNEWS, University of Waterloo, November-December 1981.

(14) T.A. Wilkinson, The Portable Languages, WATNEWS, University of Waterloo, January-February 1982.

WATNEWS references may be obtained by writing:

WATNEWS
Computer Systems Group
University of Waterloo
Waterloo, Ontario,
CANADA N2L 3G1.