REPORT ON
REAL-TIME/MINICOMPUTER LABORATORY

by

Michael A. Malcolm
Gary R. Sager

Research Report CS-75-23

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

September 1975

REPORT ON REAL-TIME/MINICOMPUTER LABORATORY

Michael A. Malcolm

Gary R. Sager

1. Purpose of the laboratory

The Real-Time/Minicomputer Laboratory in the University of Waterloo's Computer Science Department serves a purpose similar to that of a chemistry or physics laboratory by giving access to a diverse array of computer mainframes and peripheral equipment. It is used primarily as a teaching facility for introducing undergraduate students to some practical aspects of computer science. Students are exposed to "hands-on experience" using inexpensive minicomputer equipment to solve various systems programming and real-time applications problems.

During the past two terms, students of Math 479a (Real-Time Applications Using Minicomputers) have made extensive use of the lab. The lab has also been used for the research of one graduate student and two faculty members (see [1], [2] and section 5). We anticipate that a number of Math 499 undergraduate student projects and Master's projects will be generated by the lab this comming year. Hopefully, in another year, facilities of this student laboratory can be extended to serve Math 471 (Computer Architecture) and Math 132b (Introduction to Computer Architecture), as well as provide research facilities for a limited number of faculty and graduate students.

The lab has been in use since Winter 1974. We started with a borrowed Honeywell 316 which has since been returned to Honeywell. In May 1974 we took delivery of a Data General NOVA/2. By January 1975 the NOVA software was working sufficiently well to support a class of 20 Math 479a students. Since then, we have greatly increased the NOVA software support; this is discussed in Section 3. In May 1975 we acquired a Microdata 1600/30; we are currently developing software for it.

Perhaps the most interesting addition to the lab came this summer when an electric train was interfaced to the NOVA. It is used as a source of real-time control problems for Math 479a.

A detailed list of machine configurations is given in the following section.

2. Current lab hardware

i. NOVA

NOVA/2-10 CPU with power-fail auto restart

8K 16-bit words core memory, 800 ns cycle time

3-drive cassette unit

Digital I/O: 16-bits in + external interrupt
16-bits out + strobe out

Analog interface: 8-channel 10-bit A/D inputs
2-channel 12-bit D/A outputs
Z-axis scope control

Real-time clock

2 asynchronous interfaces (110-9600 bits per second (bps))

Synchronous line adapter (2400-19200 bps)

2 synchronous modems (used to connect the NOVA with the Honeywell HIS 6050 at 19200 bps)

Volker-Craig CRT terminal (running at 4800 bps)

Teletype with paper tape reader and punch

Custom made: Digital I/O general-purpose adapter
External interrupt generator
Electric train controller (attaches to digital I/O)
Interface to unlock computer cabinet drawer under program control

Approximate new value: $18,000

ii. Microdata

Micro 1600 CPU

1600/30 macro machine in 1536 words of ROM

1K 16-bit words of alterable control memory

16K bytes of core memory (1 micro sec/byte)

High-speed paper tape reader and punch

Teletype interface

4 asynchronous ports

Real-time clock

Approximate new value: $18,000

3. Cross software

Both the NOVA and Microdata come with vendor-supplied software such as assemblers, editors, operating systems, etc. This software provides tools for further software development; however, most of it assumes a single-user hands-on environment during editing, assembling, loading, debugging, etc. Thus, it is difficult to support more than 5 or 6 students developing non-trivial programs using only vendor-supplied software, not to mention the inconvenience of using paper tapes and cassettes for storing files and loading programs. We have been able to support 30 students on one machine (the NOVA) using cross software running on the Honeywell

HIS 6Ø5Ø timesharing system (TSS).

Most of the cross software is written in portable Fortran, including the assembler, relocatable loader, library file editor and emulator for the NOVA, and a micro-assembler and micro-processor simulator for the Microdata. The loader, library file editor and emulator for the NOVA were written locally during the past 6 months. The assemblers and the micro-processor simulator were supplied by the vendors; however, the NOVA assembler was debugged and substantially extended locally. We have also developed a NOVA program called the <u>NOVA/TSS Communication System</u> (NTCS) which allows a NOVA user to sign onto TSS and transfer files between the two computers. Normally, this is used to transfer a core-image file from the HIS 6Ø5Ø to the NOVA for execution.

The basic idea of the cross software is to take advantage of the strong points of the machines involved. We are taking advantage of the interactive multiprogramming system of the HIS 6Ø5Ø to allow many students to be simultaneously active editing, assembling and loading programs as well as using the emulator to run some preliminary debugging tests. The HIS 6Ø5Ø provides a flexible file system with good protection and backup, thereby obviating the necessity of this level of support on the NOVA itself. The offline support of program storage and development has been useful for systems work as well as student assignments; for example, NTCS is maintained on the Honeywell. The HIS 6050 also provides a high speed line printer for assembly listings, load maps, and core dumps, plus extensive performance reports from the emulator.

The NOVA, on the other hand, is used mainly for students to gain hands on experience with computers, especially in the areas of machine-level debugging and real-time applications. In these environments, the multiuser and file system capabilities would be difficult to provide (especially at any level approaching the quality of the HIS 6050). With the cross software, it is possible to support up to 30 to 40 advanced students while still providing each student with ample time as sole user of the NOVA.

An additional advantage is gained by the loader since it is now feasible to assemble and load NOVA programs on the HIS 6050 which cannot be loaded on the NOVA due to the core required for the loader to run on the NOVA: 1325 (base 8) words are used by the second pass of the NOVA relocatable loader.

4. Communications

In order to obtain the full advantage of a multi-computer installation, extensive communication facilities must be available. The usefulness of the cross software discussed in section 3, for example, would be severly limited without the ability to transmit the programs from the HIS 6050 to the target machine. The impact of all research and development work is greatly heightened by the ability to share the results freely throughout the computer installation.

The connection between the HIS 6050 and the NOVA 2 is done by the NTCS program which makes the NOVA appear as a remote terminal to the HIS 6050; for a detailed discussion of these communication programs and protocols, refer to [1]. A special program on TSS, together with NTCS, transfers source, object or core image files between the HIS 6050 file system and the NOVA cassettes in either direction. The bidirectional transfer capability has proved quite useful in detecting and isolating bugs in the cross software (as well as assuring its compatibility with the Data General NOVA versions) by every combination of editing, assembling and loading between corresponding versions of processors on the two machines; this was especially valuable when determing whether an error was occurring in the load code generation or interpretation.

Communication to the Microdata is done by a program which makes the NOVA emulate a teletype. This rather unlikely sounding arrangement was necessitated by the fact that the Microdata required a TWX version teletype (which was not immediately available) in order to boot the system and run diagnostics. The arrangement has turned out to be more useful than originally planned, since the emulated teletype paper tape reader and punch is in actuality the NOVA cassettes, and the line speed has been increased to 2400 bps, so that booting and back up can be done without paper tape and at a relatively high transfer rate. It is also possible to transfer files from the HIS 6050 to the NOVA cassettes and then to the Microdata as though they are paper tape. The disadvantage of this technique is that it ties up both the NOVA and Microdata, but it will be adequate until sufficient software and hardware support exists for the Laboratory (see section 6).

5. Developments in progress

Three interrelated projects are underway in the Laboratory.

A. System Implementation Languages: Work has begun on development of one or more high level languages for future systems in the Laboratory. This will provide a unifying theme for the Laboratory by allowing systems developed on one machine to be executed on the others despite architecture differences. Current efforts center on the language "B", which is already implemented on the HIS 6050 and PDP 11; we hope to have a version of B for the Microdata before January, 1976.

We are also considering the design and implementation of a systems language based on some "state of the art" ideas in programming language design.

B. "Total Conversion" Systems: The portability of software systems has received considerable attention due to the convenience and economy of having the same piece of software execute on any machine, regardless of hardware architecture. An ultimate goal is to develop a complete operating system which can be ported to another machine with approximately the same amount of effort required to port an existing and properly designed compiler to that machine. While this removes many of the inherent problems of portability (for example, that of programs which are dependent upon operating system characteristics), there are many very basic problems which have never been confronted, such as the development of abstractions of a linking loader, input/output control and file system which are machine and device independent yet primitive enough to allow ease of porting.

A number of these problems have been solved since we began working on implementing B on the Microdata. Reinaldo Braga, a graduate student, has designed and written a machine-independent linking loader. This will be a key tool for porting B to new computers. The problem is now reduced to that of recoding the B compiler code generation section and the basic I/O primatives. We are also considering the problems of bootstrapping the system from a machine whose architecture and/or peripheral devices differ greatly from those of the target machine.

This study will involve many different machines, but initial work is being done on the Honeywell HIS 6050 using the Microdata 1600/30 as the target machine.

C. System Instrumentation: The instrumentation of a system involves insertion of "probes" to measure performance, verify the intermediate steps of the execution and detect potential error situations. This was done using the NOVA emulator mentioned in section 3. It is intended to make the instrumentation an integral part of the architecture of a machine using microcode in order to make information concerning the behaviour of programs available during all stages of implementation and use. We plan to use the instrumentation to aid the development of the Total Conversion System.

This will be a continuation of the study described in [2].

6. Future plans

As a result of experience with the Microdata (cf. section 4), we propose that the minilab should be reorganized somewhat along the lines of a distributed machine as discussed in C. Foster [3]. Specifically, one minicomputer, we shall call it the "HUB", will serve as a "file machine", connected to the HIS 6050, the PDP-11/45 UNIX system, and to each of the other minicomputers in the lab. The file machine would have a small disk and perhaps a slow printer. This mini would probably be a NOVA, since the NTCS program for communication between the HIS 6050 and the NOVA has been written. The reason for a central mini file system is based mainly on the ease of connecting one simple system to a large number of different types of minis, and on the ready availability of a good disk operating system for performing the file transfers. Adding a new mini to the Lab, i.e., attaching it to the file machine, will be no more difficult than programming a simple device handler for that mini. Were we to attempt to interconnect every mini to all others, the number of communication programs and protocols would soon become unmanageable. The HIS 6050 is unsuitable for this HUB application because we want to attach inexpensive stripped-down minis to the HUB; they will have to load core images for execution over the communication line to the HUB. The protocol and data restrictions on HIS 6050 communication lines prevent this.

There are several advantages to the central file system, but the major savings would undoubtedly be in the reduced duplication of peripheral equipment and in the reduced faculty time needed to bring new mainframes and peripherals to a productive state. Certain peripherals would naturally attach to the file machine, in particular the disk to serve as a central file storage device for systems and diagnostic software; thereafter, peripherals on one machine could easily be used by another by transferring files to the central system for later transmission to the machine to which the peripheral is attached. In this way, we can provide a variety of mini mainframes and peripheral equipment for teaching and/or research while avoiding the necessity of duplicating peripherals, interfaces and software for each machine.

Since the intentionally simple protocol of the central machine will allow it to serve as a bootstrap device for any new machine, it will be possible to add very stripped mainframes to the Laboratory. This will be especially valuable if we decide to use small configurations for students to learn architecture (Math 132b). Thus, students could edit and assemble programs for the PDP 11 using UNIX or TSS, transmit the programs to the mini file system and thence to a stripped PDP 11 for a stand alone session. Likewise, editing, assembly and loading for stripped NOVA machines could be done on the HIS 6050 using the cross software already existing. Experience with the NOVA (cf. section 3) indicates that this arrangement can yield an order of magnitude increase in the hands on service rendered by a mini.

The anticipated cost of this arrangement is quite low when considered in context: it is already apparent that a file device of some sort is needed to replace the NOVA cassette drives, which are proving to be too unreliable for constant use by students. It would be wise to accept this lesson and build upon it; that is, rather than purchasing an expensive file device for each mainframe, we should make one such device accessible to the two minis we presently have, and to any we obtain in the future. Further savings accrue for each mainframe or peripheral added to the Lab. Undoubtedly, many unforseen savings will result from the ease of having all equipment in the Laboratory communicate.

We must also consider the availability and maintenance costs of the Laboratory as it is affected by reliability (or unreliability) of equipment. It is apparent that duplication of certain devices will increase availability, due partially to the interchangeability of parts; however, there is also an increase in maintenance requirements as the number of devices increase. It is possible that fewer, more reliable devices can provide the same availability while greatly decreasing the maintenance overhead. In the proposed centralized file system, failure of the HUB will seriously affect the operation of the Laboratory. Most of the stand-alone minis will rely on the availability of an interactive system (HIS 6050 TSS or PDP 11 UNIX) for editing, assembling and loading, and on the HUB for file transfers and bootstrapping. Should either of these systems be unavailable, the stand alone minis will not be available for course work. We are particularly concerned about the availability of the HUB. It would be possible to acquire inexpensive (and undoubtedly less reliable) bootstrap and file devices for each of the mainframes. Perhaps it would even be possible to interface the same type of device to all mainframes. We would then be faced not only with the maintenance of a large number of peripherals, but with an acute communication problem: without a central file device, every mainframe in the Laboratory would have to be connected to an interactive system. In our opinion, a highly reliable file device, such as a fixed head disk, would provide sufficient availability at the HUB and would require much less maintenance than a proliferation of less reliable devices. In addition, our choice of a NOVA/2 for the HUB will have the advantage of redundancy if we select the HUB mini to maximize interchangeability of parts with the existing NOVA/2.

The use of a star network of minicomputers for this application is not a new one. One is currently in use at Northwestern University (see [4]). Their experience is that the cost of adding a new minicomputer mainframe to their HUB computer (a PDP-8 with an 831K moving-head disk) is less than the cost of paper tape equipment, and performance is generally superior to paper tape speeds.

7. Acknowledgements

The lab has been received well by many students and faculty, many of whom have invested time and energy into making it a success. The following individuals have made noteworthy contributions to the lab: Laurie Rogers, Rick Madter, Gord Agnew, Nancy Kemp, Jim Morris, Gerhardt Roth, Gary Stafford, Michael Afheldt, Victor Barwell and Reinaldo Braga. In addition, nearly all of the students who have taken Math 479a have helped in debugging and designing the cross software.

8. References

[1] Malcolm, M.A., "A multi-task table-driven program design for half-duplex computer communications", U of W Computer Science Report, September 1975.

[2] Sager, G.R., "Emulation for system Measurement/Debugging" to be presented at IFIP Conference on Minicomputers, Lake Balaton, Hungary, Sept., 1975.

[3] Foster, C.C., "A view of computer architecture", Comm. A.C.M. vol 15, no 7, July 1972, pp. 557-565.

[4] Lennon, William J., "A minicomputer network for support of real-time research", Proc. ACM 1974 Annual Conference, San Diego, California, November 1974.