Newcastle University

Celebrating 50 Years of Campus-wide Computing

The IBM System/360-67 and the Michigan Terminal System

13:30–17:00 Thursday 13 June 2019
Event Space, Urban Sciences Building, Newcastle Helix

Jointly organised by the School of Computing and the IT Service (NUIT)

School of Computing and IT Service

The History of Computing at Newcastle University

Thursday 13th June 2019, 13:30
Event Space, Urban Sciences Building, Newcastle Helix, NE4 5TG

Souvenir Collection

Newcastle University

Celebrating 50 Years of Campus-wide Computing

The IBM System/360-67 and the Michigan Terminal System

13:30–17:00 Thursday 13 June 2019
Event Space, Urban Sciences Building, Newcastle Helix
Jointly organised by the School of Computing, and the IT Service

Two of the most significant developments in the 65 year history of computing at Newcastle University were the acquisition of a giant IBM System/360-67 mainframe computer in 1967, and subsequently the adoption of the Michigan Terminal System. MTS was the operating system which enabled the full potential of the Model 67 – a variant member of the System/360 Series, the first to be equipped with the sort of memory paging facilities that years later were incorporated in all IBM's computers – to be realized.

The pioneering achievements enabled by the Model 67 under MTS include not only a time-sharing service, and interactive usage, but also extensive computer networking facilities, enabling the provision of remote computing services, in particular to Newcastle's partner institutions, Durham University and Newcastle Polytechnic (now Northumbria University). All of these developments were barely imagined by computer users elsewhere in the 60s.

A new permanent display has been created of iconic artefacts, documents, videos and photographs to celebrate these developments. This is an initial contribution to a planned large and ambitious exhibition *The History of Computing at Newcastle University*, housed in the Atrium of the Urban Sciences Building, the new home of the School of Computing. This exciting new project will mark Newcastle University's commitment to chronicling and celebrating its computing history.

The unveiling of the display is to be marked by a public celebration, featuring the following invited speakers:

Michael T. Alexander – Senior Systems Research Programmer, and Assistant Research Scientist at the University of Michigan Computing Center (1965-1996), and principal developer of MTS
Jason Bain – Assistant Director Infrastructure, University IT Service (1991-1993 & 1999-)
Elizabeth Barraclough – NUMAC Computer Manager (1957-1993)
Ewan Page – Director of the Computing Laboratory (1957-1976)
Brian Randell – Professor of Computing Science (1969-)

Programme:

13:30 – 13:45 Introduction (John Fitzgerald)
13:45 – 14:00 Ewan Page
14:00 – 14:45 Michael T. Alexander
14:45 – 15:15 Break
15:15 – 15:30 Elizabeth Barraclough
15:30 – 16:00 Brian Randell
16:00 – 16:15 Jason Bain
16:15 – 17:00 Unveiling ceremony & reception

This celebration is planned for the afternoon of Thursday 13th June, in the Event Space and the Atrium of the Urban Sciences Building, Newcastle Helix. During that morning the Newcastle University IT Service will have a small ceremony marking the University Council's decision that their new building is to be named "The Elizabeth Barraclough Building".

The celebration will be a free public event, but space limitations are such that would-be attendees will have to request a place via the online form provided.

Background Note

The Flowers Report, and computing in the North East

The acquisition in 1967 of the IBM System/360-67 was a consequence of a groundbreaking Government initiative to improve the computing provisions at a number of British universities. This had been prompted by the 1966 Flowers Committee Report on the computer requirements of Universities and Research Councils.

Alone among these universities, Newcastle and Durham decided to join together and pool their financial resources, and to insist on the importance of interactive services and multi-user timesharing as an effective way of serving a large, disparate user population.

Despite the Government's policy of favouring UK computer manufacturers, Newcastle and Durham succeeded in obtaining permission to acquire a computer from IBM, which had become the globally dominant computer company. Furthermore, they were allowed to order a very special computer, the System/360 Model 67. In fact, when installed at Newcastle, the System/360-67 was the largest IBM computer in any British university, and was used to provide one of Europe's first time-sharing services.

What was System/360?

IBM's System/360 computer series, introduced in 1964, was revolutionary in its scope and extent, and was greeted rapturously by computing experts. But like the entire computer industry at the time, it was intended for conventional "batch processing": jobs were submitted by users to a queue, the results to be collected later. This suited the established business market very well, and for their new range of System/360 mainframe computers IBM provided the very successful OS/360 operating system.

What was the System/360-67?

However interest was starting to arise, especially in some leading American and British universities, in the possibility of providing *interactive* computing facilities to their user populations which, typically, were made up of many very different types of users, each with different aspirations.

The University of Michigan persuaded IBM – with difficulty – to produce a modified version of the largest S/360-65, the S/360-67, which incorporated special features (in essence: "Virtual Memory") aimed at facilitating timesharing, based on work at Michigan by Professors Bernie Galler and Bruce Arden and their colleagues, and on earlier developments at MIT.

IBM built comparatively few Model 67s, but such was the success of the new architecture that from their next computer series (System 370) onwards, they incorporated and developed the ideas introduced in the Model 67. Where IBM led, the general computer industry followed! Indeed, these developments from over fifty years ago are one of the main technical origins of the present-day cloud computing world.

The Roger Broughton Collection

Thanks to the efforts of the late Roger Broughton in creating and curating Newcastle's extensive Collection of Historic Computer Artefacts some of the most significant components of Newcastle's System/360-67 have been preserved. These include the very impressive operators' console, and the display panel from the DAT (Dynamic Address Translation) Box; it was the "DAT Box" which enabled the magic of virtual memory.

These will be displayed together with other significant artefacts alongside a 3D-printed model of the entire System/360-67 computer room, which has been created here in the USB by the School of Computing's Open Lab Research Group.

MTS – The Michigan Terminal System

The provision of an effective time-sharing service depended on software as well as hardware. IBM began to create an interactive time-shared operating system (TSS/360) to take advantage of the System/360-67's abilities. But meanwhile Michigan – instrumental in the creation of this machine – were developing their own time-shared operating system, the Michigan Terminal System (MTS).

TSS/360 was found not to be working satisfactorily, so in 1968 Newcastle and Durham took the technically and politically brave decision to switch to MTS, rather than fall back to the use of the non-interactive batch system, OS/360.

By 1969 MTS had become one of the most successful time-sharing operating systems available. It was adopted by a consortium of eight universities in the United States, Canada, and the UK, and used, under constant development, for 33 years (1967 to 1999)!

MTS was popular with its users for its simple, powerful command language, and with the institutions for its reliability, and for the intimate connections that their own programmers had with the system itself. The bonds of friendship and mutual respect forged between the staff at the eight universities remain to this day.

IBM later produced the very successful VM/370 operating system in order to take full advantage of virtual memory facilities. At Newcastle, however, MTS continued as the basis of university computing on a succession of very large central computers right up until 1992. Usage of monolithic mainframes was by then being phased out in favour of smaller, dedicated Unix "minicomputers", and personal workstations.

And 1992 was the year in which Newcastle's Computing Service rolled out its first "Open Cluster" of 30 uniform PCs. To quote from a 1997 publication announcing the refurbishment of a Cluster Room: "*In 1992, we retired our last mainframe computer, an Amdahl 5860 with 48MB of main memory. The machines installed in this room today have a combined total of approximately 3,500MB of memory and offer about 500 times the computing power of that machine.*"

[end]

School of Computing and IT Service

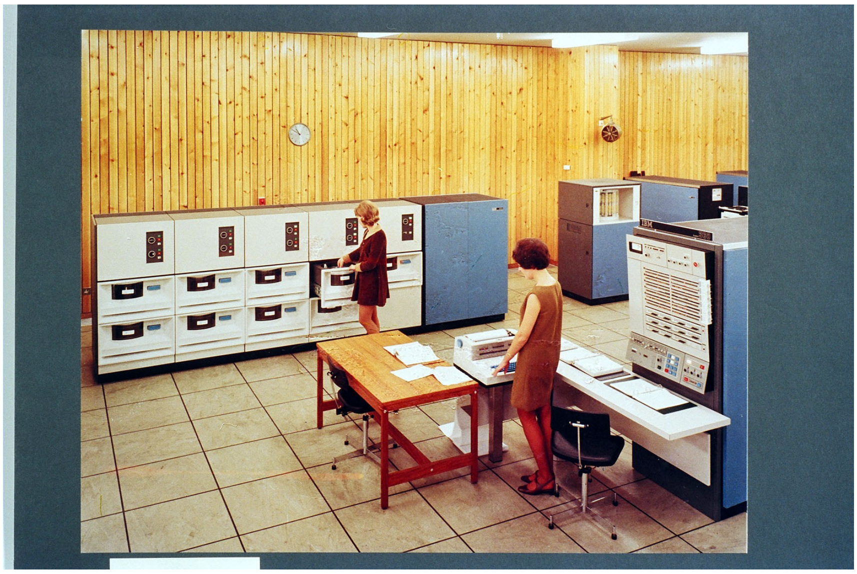
The History of Computing at
Newcastle University

NUMAC and the IBM System/360-67

> “The Northumbrian Universities Multiple Access Computer (NUMAC) is the name given to the system installed to serve the computing needs of the Universities of Durham and Newcastle upon Tyne. NUMAC is the first computing system in the United Kingdom to be jointly owned and operated by two universities. . . The computer chosen is the IBM System 360, Model 67.” [*NUMAC Inauguration – 18 March 1968*]

The background to this choice was a Government initiative (the 1967 Flowers Report) to improve the computing provisions at a number of British universities.

Alone among these universities, Newcastle and Durham decided to join together and pool their financial resources, and to insist on the importance of interactive services and time-sharing as an effective way of serving a large user population. They succeeded in obtaining permission to acquire a special kind of computer from IBM. When installed in Newcastle, it was the largest IBM computer in any British university, and provided one of Europe’s first time-sharing services.



The IBM System/360-67 in Claremont Tower

The IBM System/360 computer series was aimed at conventional batch processing. However, the University of Michigan persuaded IBM to produce a modified version of the S/360-65, which would incorporate features to facilitate time-sharing, based on work at Michigan by Professor Bernie Galler and his colleagues, and earlier developments at MIT.

Other institutions expressed interest, and IBM added the modified System/360-65 to their product range as the **System/360-67**, and started to develop TSS/360, an interactive time-sharing operating system, for it. Meanwhile Michigan were developing their own operating system, the Michigan Terminal System (**MTS**).

At Newcastle, it was found that TSS/360 did not work satisfactorily, so the technically and politically brave decision was made to switch to MTS; in fact, MTS became one of the earliest successful time-sharing operating systems. (Ewan Page had visited Michigan some years earlier, and been shown their operating system developments by Galler.)

Professor Ewan Page
Director of the Computing Laboratory,
Newcastle University 1954-1979

Professor Bernie Galler
Computer Sciences,
University of Michigan 1955-1994

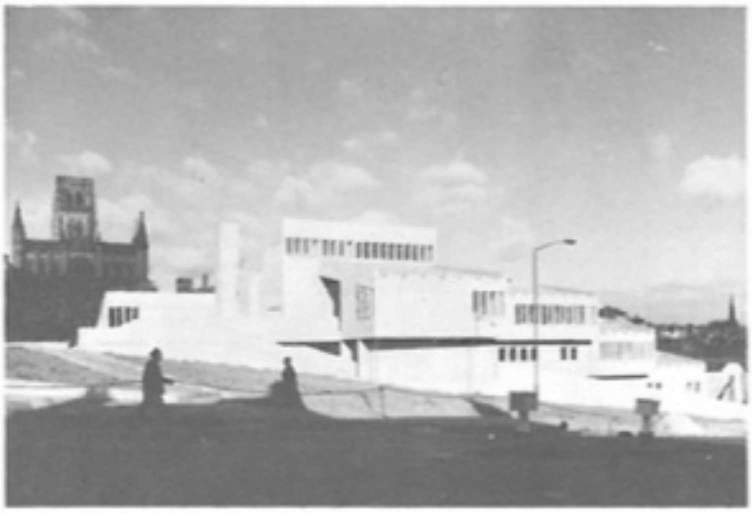
MTS was adopted and developed by a consortium of eight universities in the United States, Canada, and the UK for thirty-three years (1967 to 1999)! At Newcastle, it was used on a series of large IBM and Amdahl mainframe computers until 1992, when usage of single, massive mainframes was being phased out in favour of sets of smaller, dedicated Unix workstations, plus PCs and Macs.

Elizabeth Barraclough, *NUMAC Director (1967-1993)*

In 1972 IBM added virtual memory features to the entire successor System/370 series. The success of this and subsequent series and their VM operating system, and indeed of virtualization technology across the entire computer industry, undoubtedly owes much to the System/360-67.

Northumbrian Universities Multiple Access Computer

Universities of Durham and Newcastle upon Tyne

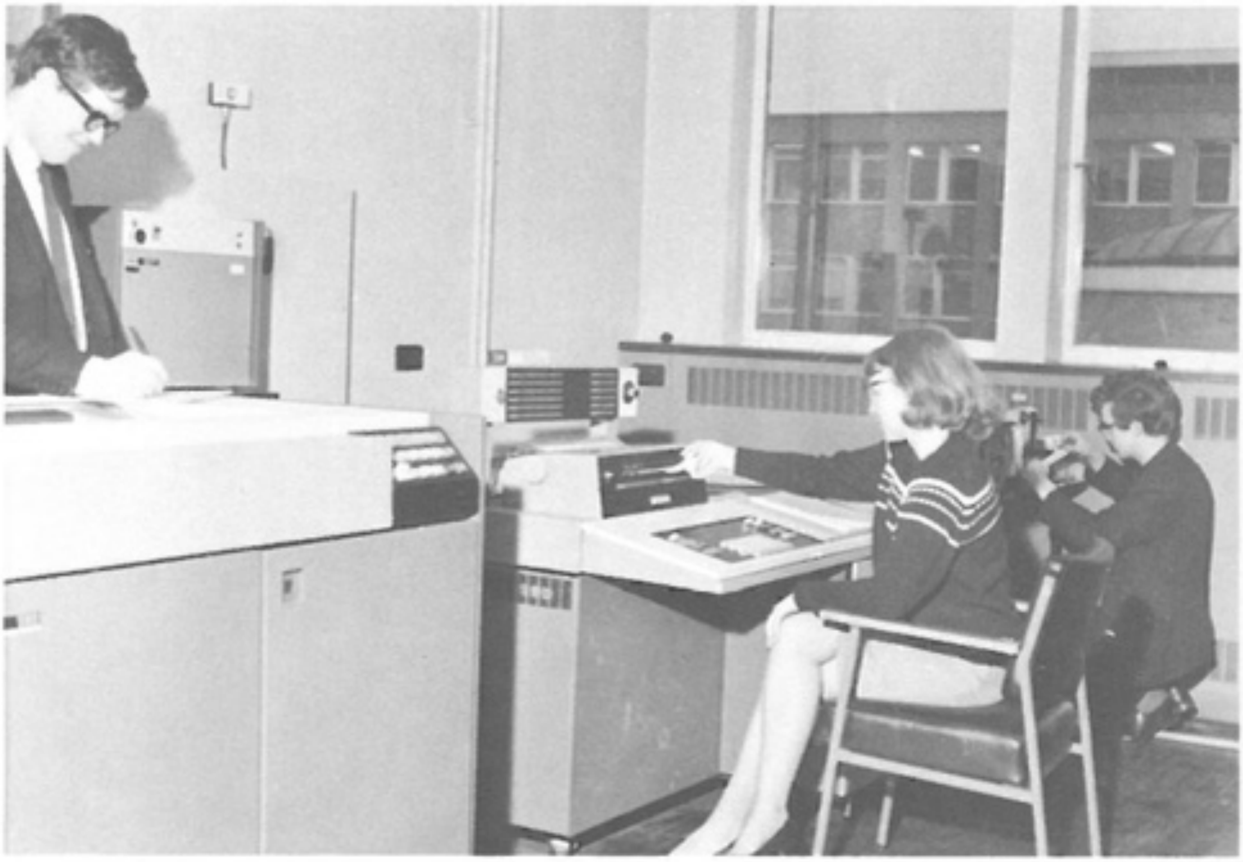


Dunelm House *Durham*

The Claremont Tower *Newcastle*

N.U.M.A.C. INAUGURATION - 18 March 1968

The Northumbrian Universities Multiple Access Computer (N.U.M.A.C.) is the name given to the system installed to serve the computing needs of the Universities of Durham and Newcastle upon Tyne. N.U.M.A.C. is the first computing system in the United Kingdom to be jointly owned and operated by two universities; such co-operation has enabled a much more powerful system to be available for use by research workers and students than could have been bought by either University acting alone. The total cost of about £1 million has been provided by pooling grants of £400,000 and £175,000 to the two Universities from the Computer Board, by a substantial allocation from the Universities themselves and by a generous contribution under their Educational Scheme by IBM (United Kingdom) Limited.



The 1130 Computer Terminal at Durham

The computer chosen is the IBM system 360, Model 67; situated in the University of Newcastle upon Tyne Computing Laboratory are the central processor unit, the core store of 512 K bytes (K = 1,024, byte = 8 bit character), a drum of 4 million bytes, a multiple disc unit capable of holding 233 million bytes available for access on eight replaceable discs, magnetic tape drives, appropriate selector and multiplexor channels controlling the flow of information and peripheral devices including printing, card and paper tape equipment and graph plotters. A small on-line satellite computer, the IBM 1130, also with printer, plotter, card and paper tape equipment, has been placed in Durham. Typewriter terminals have been installed in both Durham and Newcastle. A wide range of data preparation equipment for both cards and tape is available in both Universities.

The Model 67 can operate either as a fast multiprogramming batch processing system or as an experimental time-sharing system. In the former mode, the 1130 in Durham is connected to the central system by a private G.P.O. line and programs and data may be entered from the peripheral units in Newcastle and Durham, results transmitted to them and another program executed at the same time under the control of a systems program (HASP). Programmers will be able to refer to programs or data previously recorded on magnetic disc or tape. At times an 1130 in the Newcastle Computing Laboratory will be connected in the same way as the Durham 1130. When either 1130 is not connected to the main system, it can be used separately as an efficient computer for small research applications or student examples.

When the Model 67 is operated in time-sharing mode, several users in Newcastle and Durham will be working simultaneously at typewriter terminals under the experimental time-sharing system TSS/360. The users will be able to employ a conversational approach, entering modifications to programs or data through the Keyboard and receiving information from the typed output. They will be able to call upon their own programs and data previously stored on the multiple disc unit and upon common information also available there. Initially, six terminals in the Newcastle laboratory and two in Durham will be connected, followed shortly by five more at different places in the Newcastle campus and others at Durham. Other programs will be processed as a background task, depending on the activity at the terminals.

The arrangements for running the system are aimed to ensure that, after initial agreement on the general principles of operation, each University is free to adopt its own policy for the allocation of time and facilities among its members within its share (70%) for Newcastle and 30% for Durham). The design of the 360, Model 67 and the programming systems to be employed permit this independent control of its use. The running costs are being provided by an annual grant from the Computer Board for early years of the system's operation. Systems programming, advisory and operating staff have been appointed in both Universities and will work with the programming and engineering staff from IBM (United Kingdom) Limited to provide the services that the users require.

Data preparation in the Newcastle University Computing Laboratory

The central processer multiple disc and drum storage units of the IBM 360/67 computer.

Peripheral units of the 360/67 computer.

The users view of a modern computer a terminal for the time sharing System 360 Model 67 computer.

The majority of the demand on the computing system will arise from research workers in science, engineering and medicine, although other workers, especially in the bibliographic and social science fields, will make substantial demands. Although the needs for computer time for each student example are slight, the numbers involved produce an appreciable demand from this source also.

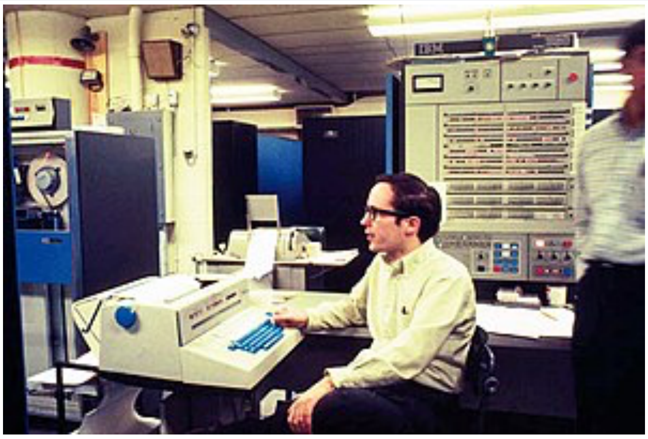
Printed by U.N.P.S.

WIKIPEDIA

IBM System/360 Model 67

The **IBM System/360 Model 67** (**S/360-67**) was an important IBM mainframe model in the late 1960s.[1] Unlike the rest of the S/360 series, it included features to facilitate time-sharing applications, notably a DAT box to support virtual memory and 32-bit addressing. The S/360-67 was otherwise compatible with the rest of the S/360 series.

IBM System/360 Model 67



IBM System/360 Model 67-2 (duplex) at the University of Michigan, c. 1969

Manufacturer	International Business Machines Corporation (IBM)
Product family	System/360
Release date	August 16, 1965
Memory	512 KB–1 MB Core

Contents

Origins

The S/360-67 was intended to satisfy the needs of key time-sharing customers, notably MIT (where Project MAC had become a notorious IBM sales failure), the University of Michigan, General Motors, Bell Labs, Princeton University, and the Carnegie Institute of Technology (later Carnegie Mellon University).[2]

Left side, 2167 configuration console for the IBM/System 360 Model 67-2 (duplex) at the University of Michigan, c. 1969

In the mid-1960s a number of organizations were interested in offering interactive computing services using time-sharing.[3] At that time the work that computers could perform was limited by their lack of real memory storage capacity. When IBM introduced its System/360 family of computers in the mid-1960s, it did not provide a solution for this limitation and within IBM there were conflicting views about the importance of time-sharing and the need to support it.

A paper titled *Program and Addressing Structure in a Time-Sharing Environment* by Bruce Arden, Bernard Galler, Frank Westervelt (all associate directors at the University of Michigan's academic Computing Center), and Tom O'Brian building upon some basic ideas developed at the Massachusetts Institute of Technology (MIT) was published in January 1966.[4] The paper outlined a virtual memory architecture using dynamic address translation (DAT) that could be used to implement time-sharing.

After a year of negotiations and design studies, IBM agreed to make a one-of-a-kind version of its S/360-65 mainframe computer for the University of Michigan. The S/360-65M[3] would include dynamic address translation (DAT) features that would support virtual memory and allow support for time-sharing. Initially IBM decided not to supply a time-sharing operating system for the new machine.

As other organizations heard about the project they were intrigued by the time-sharing idea and expressed interest in ordering the modified IBM S/360 series machines. With this demonstrated interest IBM changed the computer's model number to S/360-67 and made it a supported product. When IBM realized there was a market for time-sharing, it agreed to develop a new time-sharing operating system called TSS/360 (TSS stood for Time-sharing System) for delivery at roughly the same time as the first model S/360-67.

The first S/360-67 was shipped in May 1966. The S/360-67 was withdrawn on March 15, 1977.[5]

Before the announcement of the Model 67, IBM had announced models 64 and 66, DAT versions of its 60 and 62 models, but they were almost immediately replaced by the 67 at the same time that the 60 and 62 were replaced by the 65.[6]

Announcement

IBM announced the S/360-67 in its August 16, 1965 "blue letters" (a standard mechanism used by IBM to make product announcements). IBM stated that:[7]

- "Special bid restrictions have been removed from the System/360 Model 67" (i.e., it was now generally available)
- It included "multiprocessor configurations, with a high degree of system availability", with up to four processing units [while configurations with up to four processors were announced, only one and two processors configurations were actually built][1]
- It had "its own powerful operating system...[the] Time Sharing System monitor (TSS)" offering "virtually instantaneous access to and response from the computer" to "take advantage of the unique capabilities of a multiprocessor system"
- It offered "dynamic relocation of problem programs using the dynamic address translation facilities of the 2067 Processing Unit, permitting response, within seconds, to many simultaneous users"

Virtual memory

The S/360-67 design included a radical new component for implementing virtual memory, the "DAT box" (Dynamic Address Translation box). DAT on the 360/67 was based on the architecture outlined in a 1966 *JACM* paper by Arden, Galler, Westervelt, and O'Brien[4] and included both segment and page tables. The Model 67's virtual memory support was very similar to the virtual memory support that eventually became standard on the entire System/370 line.

The S/360-67 provided a 24- or 32-bit address space[1] – unlike the strictly 24-bit address space of other S/360 and early S/370 systems, and the 31-bit address space of S/370-XA available on later S/370s. The S/360-67 virtual address space was divided into *pages* (of 4096 bytes)[1] grouped into *segments* (of 1 million bytes); pages were dynamically mapped onto the processor's real memory. These S/360-67 features plus reference and change bits as part of the storage key enabled operating systems to implement demand paging: referencing a page that was not in memory caused a page fault, which in turn could be intercepted and processed by an operating system interrupt handler.

The S/360-67's virtual memory system was capable of meeting three distinct goals:

- **Large address space.** It mapped physical memory onto a larger pool of virtual memory, which could be dynamically swapped in and out of real memory as needed from random-access storage (typically: disk or drum storage).
- **Isolated OS components.** It made it possible to remove most of the operating system's memory footprint from the user's environment, thereby increasing the memory available for application use, and reducing the risk of applications intruding into or corrupting operating system data and programs.
- **Multiple address spaces.** By implementing multiple virtual address spaces, each for a different user, each user could potentially have a private virtual machine.

The first goal removed (for decades, at least) a crushing limitation of earlier machines: running out of physical storage. The second enabled substantial improvements in security and reliability. The third enabled the implementation of true virtual machines. Contemporary documents make it clear that full hardware virtualization and virtual machines were *not* original design goals for the S/360-67.

Features

The S/360-67 included the following extensions in addition to the standard and optional features available on all S/360 systems:[1]

- Dynamic Address Translation (DAT) with support for 24 or 32-bit virtual addresses using segment and page tables (up to 16 segments each containing up to 256 4096 byte pages)
- Extended PSW Mode that enables additional interrupt masking and additional control registers
- High Resolution Interval Timer with a resolution of approximately 13 microseconds
- Reference and change bits as part of storage protection keys
- Extended Direct Control allowing the processors in a duplex configuration to present an external interrupt to the other processor
- Partitioning of the processors, processor storage, and I/O channels in a duplex configuration into two separate subsystems
- Floating Addressing to allow processor storage in a partitioned duplex configuration to be assigned consecutive real memory addresses
- An IBM 2846 Channel Controller that allows both processors in a duplex configuration to access all of the I/O channels and that allows I/O interrupts to be presented to either processor independent of what processor initiated the I/O operation
- Simplex configurations can include 7 I/O channels, while duplex configurations can include 14 I/O channels
- Three new supervisor-state instructions: Load Multiple Control (LMC), Store Multiple Control (SMC), Load Real Address (LRA)
- Two new problem-state instructions: Branch and Store Register (BASR), and Branch and Store (BAS)
- Two new program interruptions: Segment translation exception (16) and page translation exception (17)

The S/360-67 operated with a basic internal cycle time of 200 nanoseconds and a basic 750 nanosecond magnetic core storage cycle, the same as the S/360-65.[1] The 200 ns cycle time put the S/360-67 in the middle of the S/360 line, between the Model 30 at the low end and the Model 195 at the high end. From 1 to 8 bytes (8 data bits and 1 parity bit per byte) could be read or written to processor storage in a single cycle. A 60-bit parallel adder facilitated handling of long fractions in floating-point operations. An 8-bit serial adder enabled simultaneous execution of floating point exponent arithmetic, and also handled decimal arithmetic and variable field length (VFL) instructions.

New components

Four new components were part of the S/360-67:

- 2067 Processing Unit Models 1 and 2,
- 2365 Processor Storage Model 12,
- 2846 Channel Controller, and
- 2167 Configuration Unit.

These components, together with the 2365 Processor Storage Model 2, 2860 Selector Channel, 2870 Multiplexer Channel, and other System/360 control units and devices were available for use with the S/360-67.

Note that while Carnegie Tech had a 360/67 with an IBM 2361 LCS, that option was not listed in the price book and may not have worked in a duplex configuration.

Basic configurations

Three basic configurations were available for the IBM System/360 model 67:

- Simplex—one IBM 2067-1 processor, two to four IBM 2365-2 Processor Storage components (512K to 1M bytes), up to seven data channels, and other peripherals. This system was called the IBM System/360 model 67-1.
- Half-duplex—one IBM 2067-2 processor, two to four IBM 2365-12 Processor Storage components (512K to 1M bytes), one IBM 2167 Configuration Unit, one or two IBM 2846 Channel Controllers, up to fourteen data channels, and other peripherals.
- Duplex—two IBM 2067-2 processors, three to eight IBM 2365-12 Processor Storage components (768K to 2M bytes), one IBM 2167 Configuration Unit, one or two IBM 2846 Channel Controllers, up to fourteen data channels, and other peripherals.

A half-duplex system could be upgraded in the field to a duplex system by adding one IBM 2067-2 processor and the third IBM 2365-12 Processor Storage, unless the half-duplex system already had three or more. The half-duplex and duplex configurations were called the IBM System/360 model 67-2.

Operating systems

When the S/360-67 was announced in August 1965, IBM also announced TSS/360, a time-sharing operating system project that was canceled in 1971 (having also been canceled in 1968, but reprieved in 1969).

IBM's failure to deliver TSS/360 as promised opened the door for others to develop operating systems that would use the unique features of the S/360-67:

- MTS, the Michigan Terminal System, was the time-sharing operating system developed at the University of Michigan and first used on the Model 67 in January 1967. Virtual memory support was added to MTS in October

1967. Multi-processor support for a duplex S/360-67 was added in October 1968.[8]
- CP/CMS was the first virtual machine operating system. Developed at IBM's Cambridge Scientific Center (CSC) near MIT. CP/CMS was essentially an unsupported research system, built away from IBM's mainstream product organizations, with active involvement of outside researchers. Over time it evolved into a fully supported IBM operating system (VM/370 and today's z/VM).
- VP/CSS was developed by National CSS to provide commercial time-sharing services. It was based upon CP/CMS.

Legacy

The S/360-67 had an important legacy. After the failure of TSS/360, IBM was surprised by the blossoming of a time-sharing community on the S/360-67 platform (CP/CMS, MTS, MUSIC). A large number of commercial, academic, and service bureau sites installed the system. By taking advantage of IBM's lukewarm support for time-sharing, and by sharing information and resources (including source code modifications), they built and supported a generation of time-sharing centers.

The unique features of the S/360-67 were initially *not* carried into IBM's next product series, the System/370, although the 370/145 had an associative memory that appeared more useful for paging than for its ostensible purpose.[9] This was largely fallout from a bitter and highly visible political battle within IBM over the merits of time-sharing versus batch processing. Initially at least, time-sharing lost.

However, IBM faced increasing customer demand for time-sharing and virtual memory capabilities. IBM also could not ignore the large number of S/360-67 time-sharing installations – including the new industry of time-sharing vendors, such as National CSS[10][11] and Interactive Data Corporation (IDC),[12] that were quickly achieving commercial success.

In 1972, IBM added virtual memory features to the S/370 series, a move seen by many as a vindication of work done on the S/360-67 project. The survival and success of IBM's VM family, and of virtualization technology in general, also owe much to the S/360-67.

In 2010, in the technical description of its latest mainframe, the z196, IBM stated that its software virtualization started with the System/360 model 67.[13]

References

- E.W. Pugh, L.R. Johnson, and John H. Palmer, *IBM's 360 and early 370 systems,* MIT Press, Cambridge MA and London, ISBN 0-262-16123-0, includes extensive (819 *pp.*) treatment of IBM's offerings during this period
- Melinda Varian, *VM and the VM community, past present, and future* (http://www.princeton.edu/~melinda/25paper.pdf), SHARE 89 Sessions 9059-9061, 1997

1. *IBM System/360 Model 67 Functional Characteristics* (http://www.bitsavers.org/pdf/ibm/360/funcChar/GA27-2719-2_360-67_funcChar.pdf), Third Edition (February 1972), IBM publication GA27-2719-2
2. *The IBM 360/67 and CP/CMS* (http://www.multicians.org/thvv/360-67.html), Tom Van Vleck, 1995, 1997, 2005, 2009
3. Susan Topol (May 13, 1996). "A History of MTS—30 Years of Computing Service" (https://www.msu.edu/~mrr/mycomp/mts/others/feat02.htm). *Information Technology Digest*. University of Michigan. **5** (5).
4. B. W. Arden; B. A. Galler; T. C. O'Brien; F. H. Westervelt (January 1966). "Program and Addressing Structure in a Time-Sharing Environment" (http://portal.acm.org/citation.cfm?doid=321312.321313). *Journal of the ACM*. **13** (1): 1–16. doi:10.1145/321312.321313 (https://doi.org/10.1145%2F321312.321313).

1–16. doi:10.1145/321312.321313 (https://doi.org/10.1145%2F321312.321313).

5. "System/360 Dates and characteristics" (http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_FS360.html) at IBM Archives > Exhibits > IBM Mainframes > Mainframes reference room > Mainframes basic information sources
6. DIGITAL COMPUTER NEWSLETTER (http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=AD0694645), Office of Naval Research, Mathematical Sciences Division, July 1965--pages 5-6: IBM System/360 time-sharing computers
7. Varian, *op. cit., p.* 17 (Note 54) – S/360-67 announcement
8. Pugh, *op. cit., p.* 364 – MTS on dual processor S/360-67 in 1968
9. IBM. *IBM Maintenance Library 3145 Processing Unit Theory - Maintenance*. SY24-3581-2.:CPU 117–129
10. "A technical history of National CSS" (http://www.computerhistory.org/corphist/documents/doc-42ae226a5a4a1.pdf), Harold Feinleib, Computer History Museum (March 2005)
11. "From the very beginning… from my vantage point — early history of National CSS" (http://www.computerhistory.org/corphist/documents/doc-422fd82791f26.pdf), Dick Orenstein, Computer History Museum (January 2005)
12. Varian, op. cit., pp. 24, Note 76 – IDC systems (quoting Dick Bayles)
13. SG24-7832-00: IBM zEnterprise System Technical Introduction (http://publibfp.dhe.ibm.com/epubs/pdf/sg247832.pdf), page 57: “Starting in 1967, IBM has continuously provided software virtualization in its mainframe servers.”

External links

- A. Padegs, "System/360 and Beyond" (http://www.research.ibm.com/journal/rd/255/ibmrd2505D.pdf), *IBM Journal of Research & Development, vol. 25 no. 5, pp.* 377-390, September 1981
- *IBM System/360 System Summary* (http://bitsavers.informatik.uni-stuttgart.de/pdf/ibm/360/systemSummary/GA22-6810-12_360sysSumJan74.pdf), thirteenth edition, January 1974, IBM publication GA22-6810-12, pages 6–13 to 6-15 describe the model 67
- IBM System/360 Model 67 Reference Data (Blue card) (http://archive.michigan-terminal-system.org/documentation/documents/IBM360-67RefCard.pdf)
- Several photos (http://www.eecis.udel.edu/~mills/gallery/gallery8.html) of a dual processor IBM 360/67 at the University of Michigan's academic Computing Center in the late 1960s or early 1970s are included in Dave Mills' article describing the Michigan Terminal System (MTS)
- Pictures of an IBM S/360-67 at Newcastle (UK) University (http://history.cs.ncl.ac.uk/anniversaries/40th/images/ibm360_672/index.html)
- TSS/360 Concepts and Facilities (http://www.mirrorservice.org/sites/www.bitsavers.org/pdf/ibm/360/tss/Z20-1788-0_TSS360_concepts.pdf)
- Time-sharing in the IBM System/360 model 67 (https://www.computer.org/csdl/proceedings/afips/1966/5067/00/50670061.pdf)

Retrieved from "https://en.wikipedia.org/w/index.php?title=IBM_System/360_Model_67&oldid=884553326"

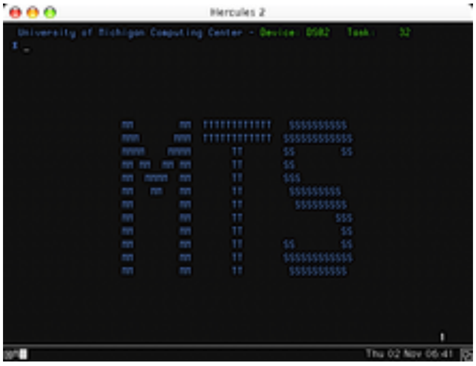
This page was last edited on 22 February 2019, at 11:39 (UTC).

WIKIPEDIA

Michigan Terminal System

The **Michigan Terminal System** (**MTS**) is one of the first time-sharing computer operating systems.[1] Developed in 1967 at the University of Michigan for use on IBM S/360-67, S/370 and compatible mainframe computers, it was developed and used by a consortium of eight universities in the United States, Canada, and the United Kingdom over a period of 33 years (1967 to 1999).[2]

Michigan Terminal System (MTS)



```
University of Michigan Computing Center - Device: DS82   Task:    32
#_

           mm        mm  TTTTTTTTTTTT   SSSSSSSSSS
           mmm      mmm  TTTTTTTTTTTT  SSSSSSSSSSSS
           mmmm    mmmm       TT       SS        SS
           mm mm  mm mm       TT       SS
           mm  mmmm  mm       TT       SSS
           mm   mm   mm       TT        SSSSSSSSS
           mm        mm       TT         SSSSSSSSS
           mm        mm       TT               SSS
           mm        mm       TT                SS
           mm        mm       TT       SS       SS
           mm        mm       TT       SSSSSSSSSSSS
           mm        mm       TT        SSSSSSSSSS
```

The MTS welcome screen as seen through a 3270 terminal emulator.

Developer	University of Michigan and 7 other universities in the US, Canada, and the UK
Written in	various languages, mostly 360/370 Assembler
Working state	Historic
Initial release	1967
Latest release	6.0/1988 (final)
Available in	English
Platforms	IBM S/360-67, IBM S/370 and successors
Default user interface	Command line interface
License	Free (CC BY 3.0 (https://creativecommons.org/licenses/by/3.0/))

Contents

Overview

The University of Michigan Multiprogramming Supervisor (UMMPS) was developed by the staff of the academic computing center at the University of Michigan for operation of the IBM S/360-67, S/370 and compatible computers. The software may be described as a multiprogramming, multiprocessing, virtual memory, time-sharing supervisor that runs multiple resident, reentrant programs. Among these programs is the Michigan Terminal System (MTS) for command interpretation, execution control, file management, and accounting. End-users interact with the computing resources through MTS using terminal, batch, and server oriented facilities.[2]

Official website	archive.michigan-terminal-system.org (http://archive.michigan-terminal-system.org/)

The name MTS refers to:

- The UMMPS Job Program with which most end-users interact;
- The software system, including UMMPS, the MTS and other Job Programs, Command Language Subsystems (CLSs), public files (programs), and documentation; and
- The time-sharing service offered at a particular site, including the MTS software system, the hardware used to run MTS, the staff that supported MTS and assisted end-users, and the associated administrative policies and procedures.

MTS was used on a production basis at about 13 sites in the United States, Canada, the United Kingdom, Brazil, and possibly in Yugoslavia and at several more sites on a trial or benchmarking basis. MTS was developed and maintained by a core group of eight universities included in the MTS Consortium.

The University of Michigan announced in 1988 that "Reliable MTS service will be provided as long as there are users requiring it ... MTS may be phased out after alternatives are able to meet users' computing requirements".[3] It ceased operating MTS for end-users on June 30, 1996.[4] By that time, most services had moved to client/server-based computing systems, typically Unix for servers and various Mac, PC, and Unix flavors for clients. The University of Michigan shut down its MTS system for the last time on May 30, 1997.[5]

Rensselaer Polytechnic Institute (RPI) is believed to be the last site to use MTS in a production environment. RPI retired MTS in June 1999.[6]

Today, MTS still runs using IBM S/370 emulators such as Hercules, Sim390,[7] and FLEX-ES.[8]

Origins

In the mid-1960s, the University of Michigan was providing batch processing services on IBM 7090 hardware under the control of the University of Michigan Executive System (UMES), but was interested in offering interactive services using time-sharing.[9] At that time the work that computers could perform was limited by their small real memory capacity. When IBM introduced its System/360 family of computers in the mid-1960s, it did not provide a solution for this limitation and within IBM there were conflicting views about the importance of and need to support time-sharing.

A paper titled *Program and Addressing Structure in a Time-Sharing Environment* by Bruce Arden, Bernard Galler, Frank Westervelt (all associate directors at UM's academic Computing Center), and Tom O'Brian building upon some basic ideas developed at the Massachusetts Institute of Technology (MIT) was published in January 1966.[10] The paper outlined a virtual memory architecture using dynamic address translation (DAT) that could be used to implement time-sharing.

After a year of negotiations and design studies, IBM agreed to make a one-of-a-kind version of its S/360-65 mainframe computer with dynamic address translation (DAT) features that would support virtual memory and accommodate UM's desire to support time-sharing. The computer was dubbed the Model S/360-65M.[9] The "M" stood for Michigan. But IBM initially decided not to supply a time-sharing operating system for the machine. Meanwhile, a number of other institutions heard about the project, including General Motors, the Massachusetts Institute of Technology's (MIT) Lincoln Laboratory, Princeton University, and Carnegie Institute of Technology (later Carnegie Mellon University). They were all intrigued by the time-sharing idea and expressed interest in ordering the modified IBM S/360 series machines. With this demonstrated interest IBM changed the computer's model number to S/360-67 and made it a supported product.[1] With requests for over 100 new model S/360-67s IBM realized there was a market for time-sharing, and agreed to develop a new time-sharing operating system called TSS/360 (TSS stood for Time-sharing System) for delivery at roughly the same time as the first model S/360-67.

While waiting for the Model 65M to arrive, UM Computing Center personnel were able to perform early time-sharing experiments using an IBM System/360 Model 50 that was funded by the ARPA CONCOMP (Conversational Use of Computers) Project.[11] The time-sharing experiment began as a "half-page of code written out on a kitchen table" combined with a small multi-programming system, LLMPS from MIT's Lincoln Laboratory,[1] which was modified and became the UM Multi-Programming Supervisor (UMMPS) which in turn ran the MTS job program. This earliest incarnation of MTS was intended as a throw-away system used to gain experience with the new IBM S/360 hardware and which would be discarded when IBM's TSS/360 operating system became available.

Development of TSS took longer than anticipated, its delivery date was delayed, and it was not yet available when the S/360-67 (serial number 2) arrived at the Computing Center in January 1967.[12] At this time UM had to decide whether to return the Model 67 and select another mainframe or to develop MTS as an interim system for use until TSS was ready. The decision was to continue development of MTS and the staff moved their initial development work from the Model 50 to the Model 67. TSS development was eventually canceled by IBM, then reinstated, and then canceled again. But by this time UM liked the system they had developed, it was no longer considered interim, and MTS would be used at UM and other sites for 33 years.

MTS Consortium

MTS was developed, maintained, and used by a consortium of eight universities in the US, Canada, and the United Kingdom:[2][13]

- University of Michigan (UM), 1967 to 1997,[14] US
- University of British Columbia (UBC), 1968 to 1998, Canada
- NUMAC (University of Newcastle upon Tyne, University of Durham, and Newcastle Polytechnic),[15] 1969 to 1992, United Kingdom
- University of Alberta (UQV), 1971 to 1994,[16] Canada
- Wayne State University (WSU), 1971 to 1998, US
- Rensselaer Polytechnic Institute (RPI), 1976 to 1999, US
- Simon Fraser University (SFU), 1977 to 1992,[17] Canada
- University of Durham (NUMAC),[15] 1982 to 1992,[18] United Kingdom

Several sites ran more than one MTS system: NUMAC ran two (first at Newcastle and later at Durham), Michigan ran three in the mid-1980s (UM for Maize, UB for Blue, and HG at Human Genetics), UBC ran three or four at different times (MTS-G, MTS-L, MTS-A, and MTS-I for general, library, administration, and instruction).

Each of the MTS sites made contributions to the development of MTS, sometimes by taking the lead in the design and implementation of a new feature and at other times by refining, enhancing, and critiquing work done elsewhere. Many MTS components are the work of multiple people at multiple sites.[19]

In the early days collaboration between the MTS sites was accomplished through a combination of face-to-face site visits, phone calls, the exchange of documents and magnetic tapes by snail mail, and informal get-togethers at SHARE or other meetings. Later, e-mail, computer conferencing using CONFER and *Forum, network file transfer, and e-mail attachments supplemented and eventually largely replaced the earlier methods.

The members of the MTS Consortium produced a series of 82 *MTS Newsletters* between 1971 and 1982 to help coordinate MTS development.[20]

Starting at UBC in 1974[21] the MTS Consortium held annual *MTS Workshops* at one of the member sites. The workshops were informal, but included papers submitted in advance and *Proceedings* published after-the-fact that included session summaries.[22] In the mid-1980s several *Western Workshops* were held with participation by a subset of the MTS sites (UBC, SFU, UQV, UM, and possibly RPI).

The annual workshops continued even after MTS development work began to taper off. Called simply the "community workshop", they continued until the mid-1990s to share expertise and common experiences in providing computing services, even though MTS was no longer the primary source for computing on their campuses and some had stopped running MTS entirely.

Mugs from MTS Workshop VIII, Ann Arbor, July 1982

MTS sites

In addition to the eight MTS Consortium sites that were involved in its development, MTS was run at a number of other sites, including:[13]

- Centro Brasileiro de Pesquisas Fisicas (CBPF)[23] within the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq),[24] Brazil
- Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA),[25] Brazil
- Hewlett-Packard (HP), US
- Michigan State University (MSU), US
- Goddard Space Flight Center, National Aeronautics and Space Administration (NASA), US

A copy of MTS was also sent to the University of Sarajevo, Yugoslavia, though whether or not it was ever installed is not known.

INRIA, the French national institute for research in computer science and control in Grenoble, France ran MTS on a trial basis, as did the University of Waterloo in Ontario, Canada, Southern Illinois University, the Naval Postgraduate School, Amdahl Corporation, ST Systems for McGill University Hospitals, Stanford University, and University of Illinois in the United States, and a few other sites.

Hardware

In theory MTS will run on the IBM S/360-67, any of the IBM S/370 series, and its successors. MTS has been run on the following computers in production, benchmarking, or trial configurations:[2]

- IBM: S/360-67, S/370-148, S/370-168, 3033U, 4341, 4361, 4381, 3081D, 3081GX, 3083B, 3090-200, 3090-400, 3090-600, and ES/9000-720
- Amdahl: 470V/6, 470V/7, 470V/8, 5860, 5870, 5990
- Hitachi: NAS 9060
- Various S/370 emulators

Computing Center staff member Mike Alexander sitting at the console of the IBM System 360 Model 67 Duplex at the University of Michigan, 1969

The University of Michigan installed and ran MTS on the first IBM S/360-67 outside of IBM (serial number 2) in 1967, the second Amdahl 470V/6 (serial number 2) in 1975,[26][27] the first Amdahl 5860 (serial number 1) in 1982, and the first factory shipped IBM 3090-400 in 1986.[28] NUMAC ran MTS on the first S/360-67 in the UK and very likely the first in Europe.[29] The University of British Columbia (UBC) took the lead in converting MTS to run on the IBM S/370 series (an IBM S/370-168) in 1974. The University of Alberta installed the first Amdahl 470V/6 in Canada (serial number P5) in 1975.[16] By 1978 NUMAC (at University of Newcastle upon Tyne and University of Durham) had moved main MTS activity on to its IBM S/370 series (an IBM S/370-168).

Amdahl 470V/6 P2 at the University of Michigan, 1975

MTS was designed to support up to four processors on the IBM S/360-67, although IBM only produced one (simplex and half-duplex) and two (duplex) processor configurations of the Model 67. In 1984 RPI updated MTS to support up to 32 processors in the IBM S/370-XA (Extended Addressing) hardware series, although 6 processors is likely the largest configuration actually used.[30] MTS supports the IBM Vector Facility,[31] available as an option on the IBM 3090 and ES/9000 systems.

In early 1967 running on the single processor IBM S/360-67 at UM without virtual memory support, MTS was typically supporting 5 simultaneous terminal sessions and one batch job.[2] In November 1967 after virtual memory support was added, MTS running on the same IBM S/360-67 was simultaneously supporting 50 terminal sessions and up to 5 batch jobs.[2] In August 1968 a dual processor IBM S/360-67 replaced the single processor system, supporting roughly 70 terminal and up to 8 batch jobs.[32] By late 1991 MTS at UM was running on an IBM ES/9000-720 supporting over 600 simultaneous terminal sessions and from 3 to 8 batch jobs.[2]

MTS can be IPL-ed under VM/370, and some MTS sites did so, but most ran MTS on native hardware without using a virtual machine.

Features

Some of the notable features of MTS include:[33]

- The use of Virtual memory and Dynamic Address Translation (DAT) on the IBM S/360-67 in 1967.[34]
- The use of multiprocessing on an IBM S/360-67 with two CPUs in 1968.
- Programs with access to (for the time) very large virtual address spaces.
- A straightforward command language that is the same for both terminal and batch jobs.
- A strong device independent input/output model that allows the same commands and programs to access terminals, disk files, printers, magnetic and paper tapes, card readers and punches, floppy disks, network hosts, and an audio response unit (ARU).
- A file system with support for "line files" where the line numbers and length of individual lines are stored as metadata separate from the data contents of the line and the ability to read, insert, replace, and delete individual lines anywhere in the file without the need to read or write the entire file.[35]
- A file editor ($EDIT) with both command line and "visual" interfaces and pattern matching based on SNOBOL4 patterns.[36]
- The ability to share files in controlled ways (read, write-change, write-expand, destroy, permit).[37]
- The ability to permit files, not just to other user IDs and projects (aka groups), but to specific commands or programs and combinations of user IDs, projects, commands and programs.[37]
- The ability for multiple users to manage simultaneous access to files with the ability to implicitly and explicitly lock and unlock files and to detect deadlocks.[35]
- Network host to host access from commands and programs as well as access to or from remote network printers, card readers and punches.[38]
- An e-mail system ($MESSAGESYSTEM) that supports local and network mail with the ability to send to groups, to recall messages that haven't already been read, to add recipients to messages after they have been sent, and to display a history of messages in an e-mail chain without the need to include the text from older messages in each new message.[39]
- The ability to access tapes remotely, and to handle data sets that extend across multiple tapes efficiently.[40]
- The availability of a rich collection of well-documented subroutine libraries.[20][41][42]
- The ability for multiple users to quickly load and use a collection of common reentrant subroutines, which are available in shared virtual memory.
- The availability of compilers, assemblers, and a Symbolic Debugging System (SDS) that allow users to debug programs written in high-level languages such as FORTRAN, Pascal, PL/I, ... as well as in assembly language.
- A strong protection model that uses the virtual memory hardware and the S/360 and S/370 hardware's supervisor and problem states and via software divides problem state execution into system (privileged or unprotected) and user (protected or unprivileged) modes. Relatively little code runs in supervisor state. For example, Device Support Routines (DSRs, aka device drivers) are not part of the supervisor and run in system mode in problem state rather than in supervisor state.[37][43][44]
- A simulated Branch on Program Interrupt (BPI) instruction.[45]

Programs developed for MTS

Program and Addressing Structure in a Time-Sharing Environment

B. W. Arden, B. A. Galler, T. C. O'Brien and F. H. Westervelt

The University of Michigan, Ann Arbor, Michigan*

Abstract. The problem studied is the effect of a time-sharing environment on the structure of programs and on the addressing strategies which may be employed in the hardware. An account is given of some very recent developments toward reduction in the system overhead needed to facilitate time-sharing. One hardware-software scheme designed to implement this reduction is described in some detail.

1. *The Time-Sharing Problem*

The discussion which follows is concerned with a *time-shared* digital computer system. By this is meant a system from which many people (or machines) may demand access and expect to receive responses after short enough delays to satisfy them. Thus, time-sharing is an external characteristic of the system, and it may be implemented internally in various ways. For example, one may postulate a computer which is fast enough to complete every conceivable task given it in a short enough time that every task can be immediately run to completion without having any "user" be aware that his task was somewhat delayed. The more common strategy is that of *multiprogramming.* Here several programs are maintained in an active state (with others probably waiting in a queue), and at various times each is given control of some part of the computer, until one or another of them is finished, or until a new task is brought in to replace an older one, according to some scheduling algorithm. Actually, multiprogramming does not owe its existence to time-sharing. When it became possible to do several processes simultaneously (usually, but not necessarily, computation and input-output) attempts were made to keep the hardware (in particular, the central processor) as busy as possible. This was done by switching control to another program whenever one task was forced to wait for the completion of a subtask by another part of the computer, e.g., an input-output processor.

Time sharing imposes the additional burden of keeping the "user" busy, as well as the hardware. Fast response by the computer to many users (e.g., 150 to 200 or more) requires that each task be given a "time slice", and if the task can not be completed during its "time slice", that it must be interrupted to allow another task its turn. Thus, the following very stringent demands and requirements are placed on a multiprogramming system that attempts to provide a time-sharing environment. (1) At any moment in time one may expect to find a great many partially completed programs, each waiting for a turn at the central processor, an input-output processor or some other part of the computer. (2) Very effective use must be made of high speed storage, since many programs must have access to it, but only a small fraction of these programs can reside there at any one time (and most programs will automatically find a way to fill all available storage). (3) The overhead incurred in keeping track of the programs which are partially completed or not yet begun and

* Computing Center.

the overhead incurred in switching control among them (while protecting each from the others), must be reduced to a minimum; otherwise, it will quickly become intolerable.

A way to provide the service implied by the requirements of time-sharing might be in the use of several central processors attached to the same high speed storage to share the load (and, incidentally, to provide better system reliability). Such a system is referred to as a *multi-processing system*. Of course, one encounters an additional problem if two or more processors attempt to execute the same task simultaneously, when this is logically prohibited.[1] However, the basic problem is still that which is characteristic of the multiprogram environment aggravated by time-sharing, in which most programs are in the status of partial completion.

The effects of each of the three requirements listed above are observable in the discussion which follows. In particular, to make the most effective use of high speed storage it is expected that most routines[2] and data areas to be shared by several users should not have to appear more than once in storage. As soon as one of these common routines or data areas appears in answer to a call from some user, it should be accessible to any others who may need it, even if the first user is not yet finished with it. Moreover, we expect to load into high speed storage only those sections of code and data actually referenced. This *dynamic loading* facility can reduce drastically the storage requirements of many programs. An extension of this philosophy leads one to expect data arrays that may be assigned little or no storage at the beginning of the execution of a task, but which may grow and contract quite unpredictably during execution. Such arrays are encountered, for example, in list processing programs and in the tables maintained by translators.

2. *Relocatable Programs*

There is a general principle in programming (and in many other areas as well) that one should "bind" variables as late as possible; i.e., the later one manages to specify (and therefore fix) the values of any variables in a system, the more flexibility and generality the system has. Thus, while it is very difficult to move a program written as absolute numerical instructions to execute from a different place in storage, an absolute program written in a symbolic assembler code may be moved by a re-assembly, and a standard relocatable binary version of program may be moved by merely reloading it. The loader for the IBJOB system on the IBM 7090/7094 system allows symbols to remain unassigned (even relative to each other) until the program is loaded; although this allows some changes to be made more easily while loading, it still requires a reloading to move the program to another place in storage. In each case the addresses in the program are eventually "bound" or fixed, but the later the binding occurs, the easier it is to accomplish necessary changes.

If one attempts to use conventional storage-addressing hardware in a time-shared multiprogramming environment, one must be able to move (or relocate) sections of code or data to make the occupied areas of high speed storage more compact and thus make larger areas available into which incoming code or data may

[1] Examples of such tasks are the programs which allocate storage, input-output processors and arithmetic processors to tasks.

[2] Especially standard system routines, such as translators, input-output transmission and conversion routines, and library routines.

be loaded.[3] The necessary relocations are unpredictable, however, and the system must be able to initiate such moves at any time. In most systems commonly used today such *dynamic relocation* is not attempted. Most often a program and all of the subprograms it may call upon must be loaded in advance and prelinked. While "ping-ponging" or "chaining" may sometimes be used to make *addressable storage* appear to be larger than *physical storage*, the fact remains that most often all of the relocation and linking must be accomplished before execution is begun, thus binding the program's addresses very early. It is then very difficult, if not impossible, to move a program. The information as to which parts of which words in the program depend on the present location of the program is generally not available during execution, and even if it were, the cost of modifying most of the words in a section of code to be moved is prohibitive. This leads to severe penalties should one try to handle several arrays which must grow unpredictably, since there are a limited number of areas to grow into (probably two, up from the top of the program and down from the top of storage, for example). Reloading the program into different parts of storage has sometimes been attempted, but this is time-consuming and can only be done with programs satisfying rather strict conventions.

It appears then that unless a solution to the dynamic relocation problem can be found, the overhead incurred in a time-shared system is likely to be very severe. The direction in which the solution appears to be is that of reducing this overhead by (1) reducing the amount of information which needs to be moved into, out of, and around in high speed storage, and (2) eliminating the need for changing parts of words when they are moved.

For the first of these goals, one may adopt such strategies as loading code and data only on demand—at the time it is referenced (dynamic loading), and only if a usable copy of it is not already in high speed storage. This implies a heavy use of *common routines*. These routines will take the form of an invariant part (instructions and constants which are not changed during execution), which may be shared by several tasks, and a variable part (temporary storage and data). Each task that calls upon a common routine must provide its own storage for the variable part, and some mechanism is necessary to allow the common code to refer easily to the appropriate variable part of storage provided by each user. In addition to reducing the amount of information to be loaded, it is possible to take advantage of the invariance in saving transfer time, i.e., one need not transfer out for later reloading any information which has not been changed from its originally loaded form; it may simply be read out of secondary storage again.

To achieve the second goal it is necessary to remove the dependence of instructions and data on the current location of the program in storage. As a general rule, the program should not be allowed to store or contain within itself any information which is related to the physical location of the program. This strategy is greatly facilitated by the use of *base registers*.

3. *Base Registers*

A base register is typically a special register whose value represents the amount of relocation needed by a program or data area. It is therefore a pointer to the *base* of

[3] Some strategies have been proposed which require code or data to be reloaded each time into the same area of physical storage to avoid relocation procedures. This changes the criterion

the area. By arranging to have its contents added to each address just before that address is used to access storage (but after modifications due to index registers, etc.), one obtains a physical address. This physical address may be changed (i.e., the program may be relocated), by merely moving the code and data unchanged and resetting the base register to point to the new base location. This eliminates the need for modifying parts of words, but it still requires that the program and its data areas move as a unit; i.e., they must be in one contiguous part of storage. Unfortunately, this is incompatible with the need to have each task that calls upon a common routine provide its own storage area for the variable part of the routine's computation. This suggests that there be at least two base registers, one pointing to the base of the code to be executed, and one pointing to the base of the data region.

If one looks at the types of use that are likely to be made of a time-sharing system, however, it is found that provision must be made for the sharing of data regions as well as the sharing of subroutines (i.e., common data as well as common routines). Any group activity in which several independently written programs must access common data areas makes it very unlikely that these data areas will be contiguous. This implies that there should be several base registers, rather than just two. Another type of program which shows the need for more than two base registers is the list-processing program, where any one of many lists may suddenly grow or contract. It is difficult to handle this situation if all data areas must have preassigned contiguous areas. Again, a larger number of base registers is indicated. Yet another example is the compiler or assembler, which must build many tables of information about the program being translated. It is difficult to predict in advance which table will become large, and normally one finds the maximum area set aside for each table in case it gets large. If efficient use of storage is to be made, these tables must grow individually on demand, which implies the use of several base registers to allow independent growth.

Of course, a program would not be allowed to store the contents of any of these registers into its own storage, since then it would contain information directly related to where its code or data is currently located, and it would probably work incorrectly if relocated. A device which is now becoming quite popular is the *privileged,* or *master* mode of operation, as opposed to *user,* or *slave* mode. In the privileged mode, the system monitor may load, store or otherwise manipulate such special registers as base registers, trap control registers, etc., but in user mode a trap occurs if one attempts to manipulate these special address-dependent registers or other special registers, which have to do with relocation and the protection of one user from another.

If one assumes then that many base registers may be suitably protected by a privileged mode of operation, the system monitor may be expected to load all of these registers in advance whenever a user program is started (or restarted after an interruption). However, whenever any program or data area is relocated, each base register must be examined in turn to see if the area to which it points was one which moved, so that that base register may be updated.[4] One alternative to loading all

from having enough space in storage to having enough space in just the right places, or else one is committed to searching through the queue of programs waiting to execute until one finds a program that fits the exact pattern of available locations. It is easy to construct examples of programs blocking others from being loaded.

[4] If the base registers are used to hold quantities other than addresses, as is the case in some machines, there is the additional problem of knowing when the contents of a base register is actually an address which needs modification.

registers in advance is to trap to the system monitor each time a base register should be loaded. In this way only those needed are loaded, and always with the most recently assigned address. This still does not keep all the other base registers up to date (the monitor would need to keep track of which registers contain addresses), nor does it solve an even more difficult question; viz., how many base registers should one provide in the first place?

4. *Symbolic Base Addresses*

One way out of the difficulties mentioned in the preceding section (i.e., determining the proper number of base registers and keeping all base registers updated), is to assign symbolic addresses to sections of code or data. These symbols may then be mapped into physical addresses on each reference, so they are always correct, even if the referenced area has been moved. If (1) the mapping to physical addresses is accomplished by the hardware on each reference, (2) the symbol is assigned by a trap to the monitor on the first reference, and (3) the symbol is not reassigned during the execution of the program, then one need not have more than one trap to the monitor for each section of code or data. Subsequent references will already have been "linked" directly to the appropriate section by the assigned symbolic address. (Of course, most symbols will appear as displacements from symbolic addresses linked in this way; such symbols will not need any linking at all.) It should be noted that the mapping of each symbol into a physical address amounts to an automatic application of indirect addressing; this will become clearer below. The idea of making symbolic references to blocks of storage was proposed by J. Dennis [1] in a paper on segmentation (see J. Dennis [4]).

5. *Paging*

A great reduction in the system overhead incurred by physically moving code in, out, and around in high speed storage may be obtained by regarding the high speed storage as divided into fixed-size blocks, called *(physical) pages*. If a section of code or data is similarly divided into *(logical) pages* of the same size, then these logical pages may be scattered randomly into physical page areas, provided only that the hardware be able to associate an address in a logical page (i.e., an address referenced in the code) with a corresponding address in a physical page. This is illustrated in Figure 1, in which light arrows show page correspondences and the dark arrow

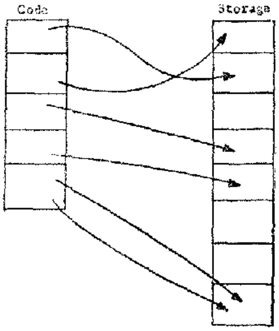


FIG. 1

shows one particular address correspondence. Note that paging (which was introduced on the Atlas computer) again amounts to an automatic application of indirect addressing.

6. *Symbolic Base Addresses with Paging*

It is natural to ask if base addresses, which identify logical blocks (referred to below as *segments*), and logical page addresses, which allow logical blocks to be stored in physically noncontiguous blocks of storage, can be combined into single logical addresses. Any logical address constructed in this way should behave, from a programming point of view, like a single physical address. In the next section a hardware organization is described that allows this, and in subsequent sections it is shown how quite reasonable software conventions can reduce the system overhead considerably. In combining the two types of address mapping, it will be seen that the storage address resulting from the first (symbolic) mapping will be used as the address of the beginning of a logical page. This, in turn, will be mapped into a physical address representing the beginning of a physical page, to which a displacement (or line number) will normally be added.

7. *A Hardware Proposal*

It is proposed that a symbolic address be viewed as divided into three parts (roughly equal in length[5]), as follows:

S_α	P_α	L_α
Segment Number	Page Number	Line Number

.

(Throughout this discussion the term *symbolic address* implies an effective address, in that all modification due to indexing, etc., has been done, and the address is that with which one normally makes a storage reference.) These parts of a symbolic address are referred to as S_α , P_α and L_α , respectively. A new register is postulated in the machine, called the Segment Table Register (STR). The contents of the STR will have two parts:

Segment Table Length	Segment Table Base

.

The Segment Table Base is the physical address of a table called the Segment Table, whose length is also in the STR. It is seen in Figure 2 that in the mapping of a symbolic address α, the number S_α serves as an index into the Segment Table by being added to the Segment Table Base. (The Segment Table Length is used to check that S_α is not too large.) It is intended that the STR will be loaded with the appropriate values whenever a new user is given control. This implies that each user has his own Segment Table, but since it is constructed automatically by the monitor, he will be unaware of it.

The word selected by S_α in the Segment Table will have the form

Page Table Length	Page Table Base

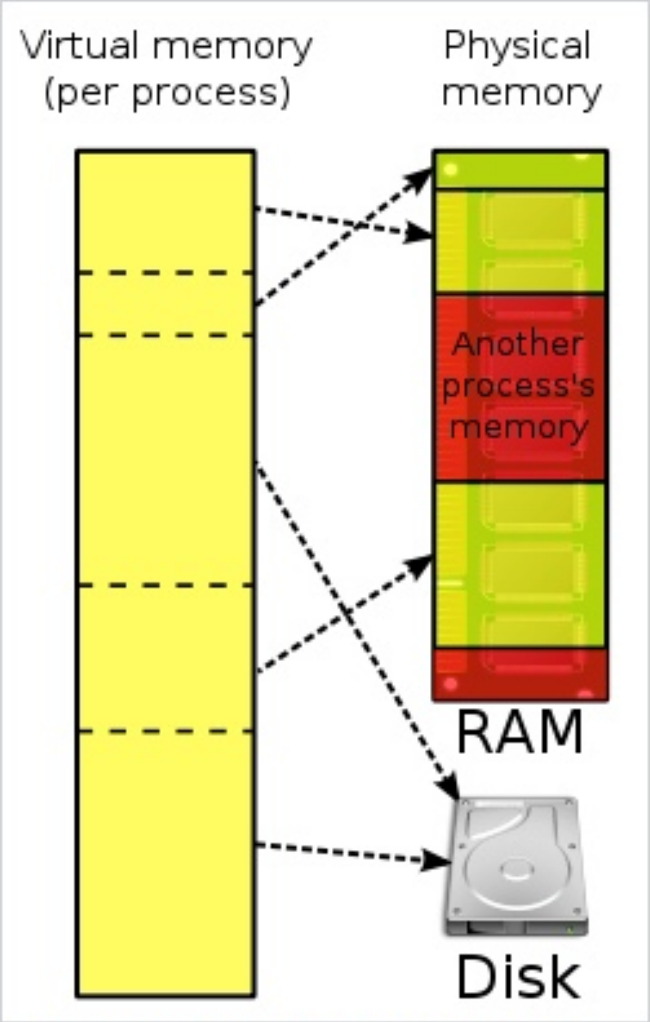
. (1)

[5] The number of digits in any part of any word described here is fixed, but unspecified, since it is very dependent on the particular computer involved.

Virtual memory

From Wikipedia, the free encyclopedia

This article is about the computational technique. For the TBN game show, see Virtual Memory (game show).



Virtual memory combines active RAM and inactive memory on DASD[NB 1] to form a large range of contiguous addresses.

In computing, **virtual memory** (also **virtual storage**) is a memory management technique that provides an "idealized abstraction of the storage resources that are actually available on a given machine"[1] which "creates the illusion to users of a very large (main) memory."[2]

The computer's operating system, using a combination of hardware and software, maps memory addresses used by a program, called *virtual addresses*, into *physical addresses* in computer memory. Main storage, as seen by a process or task, appears as a contiguous address space or collection of contiguous segments. The operating system manages virtual address spaces and the assignment of real memory to virtual memory. Address translation hardware in the CPU, often referred to as a memory management unit or *MMU*, automatically translates virtual addresses to physical addresses. Software within the operating system may extend these capabilities to provide a virtual address space that can exceed the capacity of real memory and thus reference more memory than is physically present in the computer.

The primary benefits of virtual memory include freeing applications from having to manage a shared memory space, increased security due to memory isolation, and being able to conceptually use more memory than might be physically available, using the technique of paging.

Extracts from the Directors' Reports 1965-1989

From 1957 to 1989 The Computing Laboratory at Newcastle University each year fully documented all the research, teaching and service activities within the Laboratory and also set out future plans for the forthcoming year. This information was compiled in an annual report entitled 'Report of the Director' – by Dr., later Professor, Ewan Page from 1957 to 1979, and then Professor Harry Whitfield from 1979 to 1989. Detailed listings of staff, publications, colloquia, and visitors were included in these reports. Here are provided extracts related to the provision of mainframe computing, starting with the System/360-67, and the use of the Michigan Terminal System (MTS).

1965-66
"We hope to meet the [Computer] Board soon and seek their approval for our proposals of co-operation with our colleagues in Durham for the early purchase and operation of a multiple access computing system to serve both Universities. Our plans are of a different kind from those in other Universities and research centres and if we are enabled to realise them promptly we believe that they may form a pattern for much University computing in the next few years."

1966-67
"Preparations for a new building, a new computer, and a new honours degree course have understandably occupied much of many persons' time during the year. It had been planned to move into the new accommodation on 1st September . . . staff were able to move when planned by gaining access by means that are no longer approved. Data preparation equipment followed a few days later [but] the KDF9 remains in the Kensington Terrace building and will do so until the New Year. [After protracted negotiations we received approval] to order an IBM SYSTEM 360, Model 67, for installation as soon as possible in order to gain benefits from the advanced software developments being made in the United States and to gain experience of time-shared computing. The system was ready for delivery on the 25th September, 1967, but the unreadiness of the computer room to receive it caused a postponement until 2nd October."

1967-68
"During this year the Computing Laboratory has occupied a new building, installed two new computers, moved a computer already in use, started teaching for a new Honours Degree and has begun a number of new research projects. . . Some time ago the Universities of Newcastle and Durham decided to co-operate in establishing powerful computing facilities for their use and this decision has led to the installation of NUMAC, the Northumbrian Universities Multiple Access Computer. This system, which is the most powerful in any British University at the present time, is the first to be owned jointly by two Universities. . . Since the System 360, Model 67 began its routine operation several systems have been operated and others studied. For most of the time there have been three sessions under the operating system OS/360 [and] two under the

time-sharing system TSS each day . . . We are now actively investigating the problems of implementing MTS [the Michigan Terminal System]."

1968-69
"From early in the academic year the Michigan Terminal System has been operated for much of the normal working day of University staff and students, replacing the TSS/360 sessions which had been mounted experimentally since delivery of the machine. . . Batch terminals in Durham and Edinburgh as well as in Newcastle are served at the same time as typewriter terminals . . . The demand for computing power continued to grow in Newcastle as elsewhere with the exponential increase which has become the norm for planners. As we strive to satisfy this demand by making reasoned applications for additional hardware and designing improvements to software system that we operate, other requirements become pressing. Fortunately, the need for space and for staff do not increase at the same rates as the need for computing power, but increase they do . . . Relief is urgently needed."

1969-70
"The IBM 360, Model 67, has continued to run satisfactorily for three shifts during the week, and is increasingly being operated by certain users at weekends. . . The total number of terminals that are attached in Newcastle and Durham is 31."

1970-71
"The various legal and administrative arrangements for admitting the Newcastle upon Tyne Polytechnic to participate in the NUMAC operation were completed with the approval of the Computer Board and at the end of the year the arrival of an IBM 1130 terminal in the Polytechnic allowed a start to be made on this extension of the computing service provided by the central equipment based in the Computing Laboratory. . . Near the end of the year a Marconi Elliott 905 computer, with an associated refreshed graphical display and a light pen, arrived in the laboratory. At present it may be used as a standalone system, while the communications software is developed for its attachment to the 360."

1971-72
"It is useful to review the consequences of those decisions taken sometime in 1966, which led to the formation of NUMAC and to the acquisition of the equipment we now have. Then the concept of joint ownership and management of a computing centre was novel; now such an operation is . . . commonly urged . . . Much more significant was the decision to buy a machine with hardware that permitted the use of a virtual memory organisation – this, too, at a time when such organisation was a rarity. The recent announcement by the biggest manufacturer guarantees that such organisation will become common over the next few years, if not predominant. . . Two years ago we foresaw the need for an early extension or replacement of our present computing equipment. We began to review which of the various alternatives then available would make it possible for our users to pursue their research work with least hindrance – an aim for a computing centre providing computing services which we have regarded as so obvious as to require no justification. . . Last year the Computer Board . . . arranged a temporary supplementation of our computer power by a service from Cambridge. Now the various alternatives which are open to us for a longer term solution are much clearer . . . Some

represent a path of development which is natural, easy and powerful, others are none of these. Which path we are to take and when we are to take it, are not yet known. Until they are, I fear that our users' powers of tolerance to a declining standard of service may be sorely tested, while ours of restraint under criticism, whose causes we are powerless to remove, may be even more severely examined."

1972-73
"During the year we have started to take some computing resources from the IBM 370, Model 165, at Cambridge and hope that the coming year will see the remaining problems associated with the use of this link solved."

1973-74
"The last several Annual Reports of the work of the Computing Laboratory have all referred to the saturation of our main computing equipment . . . It is very pleasing to be able to report that the [Computer] Board has indeed agreed to provide funds and that an IBM 370/168 is to be delivered at the beginning of 1975 and linked to the existing 350/67. . . It had always been our intention that some part of the new computing system should be made available to those of other Universities who required its particular facilities. Accordingly, we had no hesitation in accepting the requirement of the Computer Board that we should make up to half of the power of the system available for outside users and we had prepared our plans [accordingly]. This we still hope to do, but the cuts in the current funds available to the Computer Board have reduced the number of staff that can be appointed for this service. . . On the one hand we and the whole community of users are fortunate that there is a group of academics interested in systems problems."

1974-75
"Builders and contractors have been with us throughout the year, taking over the main lecture room in the Laboratory, filling it with air conditioning equipment. . . Unfortunately, when the plant later began to run in earnest so that the new [IBM 370/168] computer could be operated, it was very soon found that an 'infection of unknown origin' beloved of our medical colleagues had in the mean time entered the water cooling system [causing] consternation, and for what seemed a long time, bewilderment. . . Various parts of the computing system itself were assembled in Newcastle from different parts of the world by the date planned. . . Tele-communications forms an increasingly important part of the activities of computing centres, especially ones such as NUMAC with a widely distributed set of users. For nearly eight months we have been exploring how the University telephone network could be used for data traffic but all our proposals have foundered on the rock of P.O. regulations . . . [However] the Post Office is hoping to introduce an Experimental Packet Switched System within the next year to which NUMAC will be connected."

1975-76
"Even quite casual readers of the technical press will have noticed that users from the Scottish Regional Computing Organisation were far from satisfied with the service they were receiving from NUMAC. . . If reductions in expenditure had not been necessary the [IBM 360/158] machine in Edinburgh need not have been withdrawn; the staff thought necessary for running our new [IBM 370/168] machine and serving a gradually

increasing load of 'IBM specific' work from other universities could have been provided; and smooth transition could have been made from an unsatisfactory machine configuration and operating system . . . I feel very proud of the way the NUMAC staff worked and I and all the users of the system have good reason to be grateful to them. . . The provision of a local communications network to provide better access for Newcastle users to the NUMAC facilities have been delayed much longer than we had hoped."

1976-77
"This year marks the twentieth anniversary of the founding of the Computing Laboratory, and of the appointment of its director, an occasion which was enthusiastically and memorably commemorated on the exact anniversary of this appointment, namely 1st April. However it has in fact been a year in which the Laboratory has had to manage without Professor Page's inspiring and demanding leadership, because of the responsibilities that have been imposed on him as Acting Vice-Chancellor. . . On the service side, the heroic exertions of the staff have met with much success, and considerable progress has been made in coping with the software problems which loomed so large in the Report of the Director for 1975/76. The patience and understanding that we received from the NUMAC community during this period are much appreciated."

1977-78
"The principal change on the main service machine, the IBM 370/168, has been the trebling of its main memory. This enhancement of the computing system has made it possible to perform much more computing for users, and to improve substantially the performance at terminals. . . We have also been pleased at the recognition by the Computer Board of NUMAC as one of the major National centres for computing services."

1978-79
"MVS was finally brought into service at Easter to replace the overnight MVT system [on the System 370/168]. Our plans to run MTS and MVS together under either the VM control program or our development of VM, the Hypervisor, have not yet been successful. Progress on the campus network has been achieved despite various setbacks partially caused by its success. The experience with the installation of the network has shown an insatiable demand for terminal access to the computer."

1979-80
"On the service side, the most significant change was due to a decision by the Computer Board at the end of 1979 that NUMAC should no longer be regarded as a National Research Centre for IBM-specific facilities [so that] the IBM 360/67 installation was not replaced when it ceased service on 31st July, 1980, and . . . the use of the IBM 370/168 installation by users outside of the three NUMAC institutions will gradually be reduced to 10% of its capacity."

1980-81
"On the service side the most significant development was the preparation and presentation of the report entitled 'Computing at the Universities of Durham and Newcastle in the 1980s'. . . the proposals have received wide support and form the basis for development within the Laboratory and within NUMAC. Perhaps the two most

important features to emerge from the proposals were the further development of NUNET and a desire on the part of the users to have the quality of the advisory services maintained. . . This year has seen the phasing out of the MVS service, and a significant reduction in the amount of time used by the Scottish universities."

1981-82
"On the service side a number of activities have come to fruition and others are in an advanced stage of preparation. The major event will be the installation in September 1982 of an IBM 4341 at the University of Durham. This will ease the workload of the IBM 370/168 and will improve the service to all our users. During the year our computer network, NUNET, has been developed and now provides access to a number of host machines. In addition to the main (IBM) computer there are now four PDP-11 based UNIX systems attached at Newcastle and two in Durham. Plans are in hand to connect the Durham IBM 4341, and two VAX computers and two more PDP-11 systems at Newcastle."

1982-83
"This year has seen the very smooth introduction of the IBM 4341 at Durham which increased the main computing resources of NUMAC by about a third. This, together with weekend unattended operation at Newcastle, has meant a very significant increase in the resources available to our users. . . The two Data Entry Systems, DES1 and DES2, have not always been able to cope with the demands placed on them . . . a third Data Entry System should be installed in time for next academic year . . . but it may well be necessary to move some students back onto MTS with a consequent reduction in service to research users. . . On a happier note, we have seen the introduction of the VAX 11/780 for Graphics applications. This new service is developing well and promises to be a success. A great deal of effort has been spent on the planning activities for the 1985 replacement of the main service machine at Newcastle."

1983-84
"Our most important activity this year has been the preparation of the case . . .for replacement of the IBM 370/168 in 1985. We are proposing to move towards a situation where most of the computing would take place on single-user workstations supported by large central server machines. Time-sharing, as we have know it for the past seventeen years, would be phased out over a period of some five years. . . To achieve an evolutionary development in this direction requires the ability to run our existing time-sharing operating system, MTS. The Computing Board, whilst approving our proposals for distributed computing, was unable to agree to the limitation to machines capable of running MTS. We therefore face the possibility that we may have to make a very disruptive change to a different time-sharing system."

1984-85
"Our most important activity on the Computing Service side of the Laboratory has continued to be the procurement of the replacement for the IBM 370/168. . . an Amdahl 5860 with 40 Mbytes of main storage is to be installed at Newcastle and the Durham machine is to be replaced by an Amdahl 470/V8 with 16 Mbytes of main storage. Both machines are expected to be in service in October 1985. In the event we still have IBM compatible machines on which we can continue to run our existing time-sharing system,

MTS, at least for an interim period while we investigate alternative manufacturer supported systems capable of supporting distributed workstations. . . Users now have direct access to the JANET network from NUNET terminals, without going through a NUMAC host. The year also saw the introduction of a standard file transfer service (Blue Book FTP)."

1985-86

"The installation of the Amdahl 5860 in late September 1985 and its introduction into service in early October must be regarded as the major event of the year. The whole process went so smoothly (and unannounced) that users 'merely' noticed that the system had suddenly become much more responsive and five times faster. Work is now proceeding to evaluate VM/CMS and UTS (Amdahl's UNIX offering) as possible alternative operating systems to MTS in accordance with undertakings given to the Computer Board for Universities and Research Council."

1986-87

"The Amdahl 5860 has provided a very powerful and reliable base for the main computing service and has enabled staff to devote more of their time to other matters. . . Investigation of alternative operating systems has proceeded well and decisions will soon be made about the form of the service for the next few years. A major element in relation to longer term planning is our involvement in the WINE (Workstation Integrated Network Environment) Project, which, though started by the MTS community, aims to provide a distributed system environment (or Intersystem) independent of any particular operating system. A great deal of work has gone into planning the campus Ethernet. . . Users will have noticed the introduction of "Fawn Boxes" which enable simple vdu terminals to have full-screen management across local (and remote) networks."

1987-88

"During the last year a great deal of stealthy progress has been made in the Computing Service, although little of this will be apparent to the users. We have just installed 13 kilometres of fibre optic cable around the campus . . . It is a significant step towards a high-speed integrated network. . . During the past year there has been a considerable increase in computing power available to the Universities of Newcastle and Durham with the installation of the second NUMAC 5860 at Durham."

1988-89

"On the Computing Service side of the Computing Laboratory the most important activities have been focussed on the development of networking services and the long term planning for the demise of MTS in 1992. At the beginning of the year we completed the installation of the fibre optic Ethernet backbone which connects most buildings on campus. The Newcastle ethernet is connected by a Megastream link to a similar Ethernet in Durham maintaining the uniformity of access to NUMAC and departmental facilities that we have always enjoyed with NUNET. The installation of the Gould NP/1, our first service machine to run the UNIX operating system, will enable our users to get some experience of UNIX, which will almost certainly be the operating system of choice on workstations and larger hosts in the 1990 and 1992 procurements."

School of Computing

The History of Computing at Newcastle University

The Personal Computer Revolution

The first true "personal computer", i.e. one that is designed for one person, is easy to use, and is cheap enough for an individual to buy, is generally accepted to be the Altair 8800, created in 1974 by Micro Instrumentation and Telemetry Systems (MITS), based on the 8-bit Intel 8080 microprocessor.

The first successfully mass-marketed personal computers were the Commodore PET, the Apple II, and the TRS-80 from Tandy Corporation, all introduced in 1977; at Newcastle such developments were first mentioned in the Computing Laboratory Director’s 1977/78 Report, but it was the 1978/79 Report which stated: “The other major development area has been on the microprocessor front . . . Several microprocessors funded from Computing Laboratory reserves have been installed in departments as part of an experiment to assess the growth of microprocessors on the computing service.”

In the UK, important developments included Sinclair Research’s **ZX80** (1980) and Acorn Computer Company’s **BBC Micro** (1981). The latter was adopted by most schools and was also successful as a home computer in the UK, despite its high cost (£330 in 1981, which is about £1,600 in 2018)..

Sinclair Research ZX80 (1980)

BBC Micro (1981)

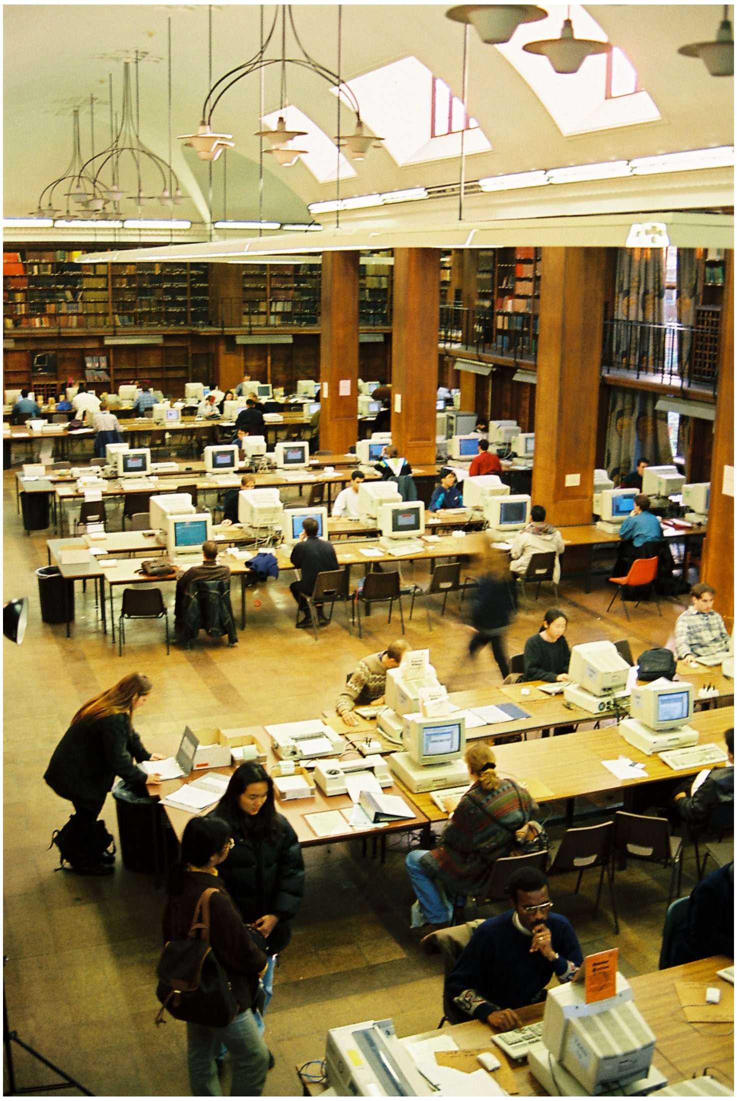
However, the most significant impact was the introduction by IBM – at that time the greatest force in computing in the world – of their own 5150 “**Personal Computer**, also in 1981. This had an open architecture, and led to a huge market for third-party add-ons and applications, and to many competitors all creating "IBM-compatible" machines (“PCs”), which together dominated the market.

The 1983/84 Director’s Report commented: “We have also seen the IBM PC appear in the Laboratory and as it becomes more and more of an industrial standard we expect it to proliferate though it still has rivals that can give better price performance.”

IBM PC (1981)

Apple Macintosh Plus (1986)

In 1984 Apple introduced the Macintosh (**Mac**) computer, the first mass-market personal computer that featured a graphical user interface, built-in screen and mouse, based on work-station developments at the Xerox Palo Alto Research Center. But since the early 1990s, Microsoft operating systems and Intel hardware have dominated much of the personal computer market, first with MS-DOS and then with Windows, a system that adopted many Mac-like features.



The first PC Cluster Room, in the Old Library (c.1990)

At Newcastle as elsewhere, personal computers (especially PCs) came into increasing use, starting in the 1980s, initially as stand-alone devices, but then networked, and large numbers were deployed across the campus for student use in “cluster” rooms, using “managed desktop” software. Now in 2018, the University Computing Service (NUIT) is responsible for 3000 PCs in 40 cluster rooms, and many if not most staff and students have their own PCs or Macs, as well as powerful “smart phones”, such as the Apple iPhone and various types of Android phone. As a result, NUIT is providing support for over 60,000 devices!

Roger Broughton

Prologue

Adèle Davison writes ... During the final weeks of 2016, a group of current and retired staff from the University began working with Roger Broughton to take custodianship of one of his most important projects: The Museum of Computing Artefacts. Some of this amazing collection can be seen here: http://moca.ncl.ac.uk/

Roger was suffering from cancer and knew he didn't have long to live; he was keen to ensure that his beloved museum was placed into safe hands for posterity. Sadly, Roger passed away just before Christmas.

We have put together this special newsletter as a tribute to Roger, who was a dearly loved colleague in the University until he retired in 2002.

John Law, author of this article, writes ... I was an Information / User Education Officer in the Computing Service, 1971-2009. I was asked to join the Museum Project in December 2016. The project is chaired by Dr Will Blewitt of Computing Science; the other members are Emeritus Professor Brian Randell and Sara Bellwood (Computing Science), Dr Clive Gerrard (now retired, formerly the head of our PC Services), Michelle Wright and Adèle Davison (NUIT); and Samantha Gray (Museum Studies).

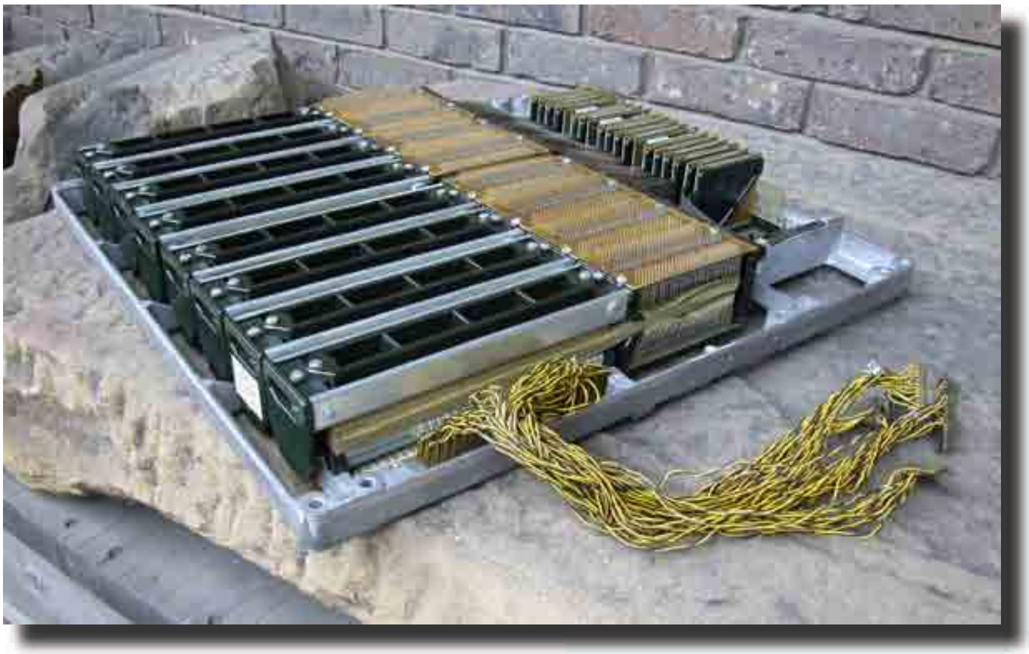
There are several stories that could be told about computing at this University during Roger's time (1967–2002): the Computing Laboratory itself; NUMAC; the University Computing Service and its ground-breaking advances, emulated in the UK and in the world, our pioneering network, and its contribution to the foundation of the Internet.

Those stories may never be written, but in this tribute to Roger, I have touched upon a couple of them. I hope you find time read these notes: it might cheer you.

John Law, 22 March 2017

NUIT Staff Newsletter March 2017

Two artefacts together



This is one of the first photos you see on Roger's "Virtual Museum" website.

It's an example of "firmware", which he describes in detail on the page.
It's from the Control Unit of the disk drives seen later (in this present article) in the "world famous" picture of our first IBM mainframe.

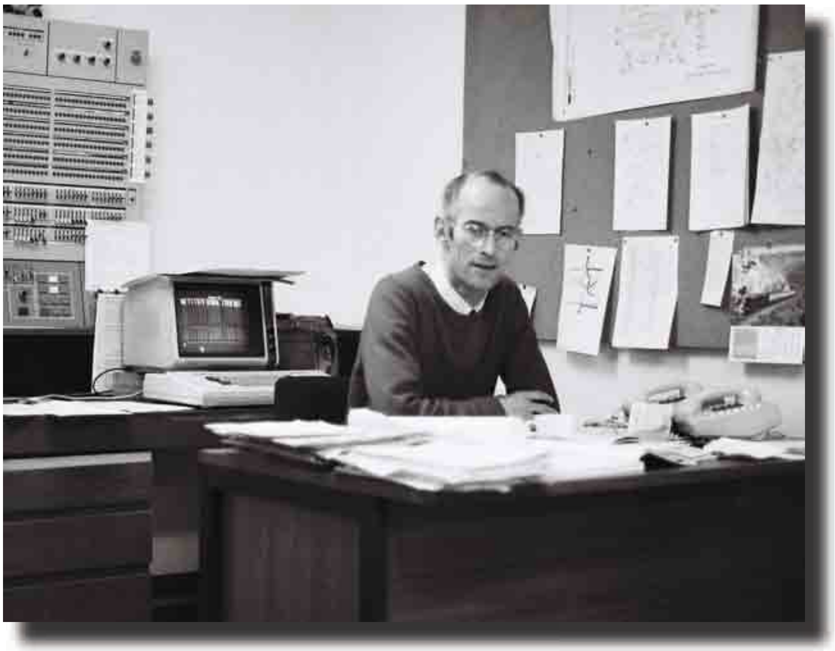
Here, it's been chucked out: but Roger has rescued it. He has photographed it in the Loading Bay (the sunken area beside Claremont Road, overlooked by the entrance outside the Service Desk).

What Roger does **not** say (though I'm certain that he knew) is that it is sitting on a Roman altar, owned by the University's Museum of Antiquities. They had a whole collection of these, and since there aren't many places you can store 100+-kilo blocks of stone, they stored them in the Loading Bay for decades. Those altars (now in a safer place I believe) watched our mighty, cutting-edge mainframes arrive, and watched them depart a few years later.

Roger's own "Museum of Antiquities" has a significance equal to those altars.

Roger E. Broughton (REB)

As our Operations Supervisor, Roger's working life was based below Ground Floor: he spent most of his time in his particular domain, the Sub-Basement Computer Room.



Whenever I went to see him he would greet me with the softly spoken exclamation: "John!", which somehow managed to combine a sense of surprise ("*What are you doing down here?*") with a sense of "*How nice to see you!*" – which always made me feel good.

The Man

On "HMS Computing Laboratory", Roger was the ship's Chief Engineer: in his domain, which for decades was the centre of this University's computing power, nothing went on without Roger's overseeing it.

Up until the 90s, our "customers" very rarely saw him. However, we, his colleagues, did see him: he was at the centre of everything that moved (or rather, sat and hummed). If anyone needed to do something in the Sub-Basement, they had to go through Roger: they could expect a close grilling, but they could also expect a tremendous amount of expert help and friendly cooperation.

He made a point of trying to understand – completely – every single piece of equipment that entered his domain: a new telephone, the core drills used by contractors to bore holes in Claremont Tower's solid concrete walls, the power supplies, up to the immensely complex installation of the mainframes, over which he had charge. This was no burden to him: a love of complex mechanisms and systems was in his DNA.

He had a lifelong love of machines, of any description, size or complexity.

Roger and brother Peter on the famous “tribike”; brother David is taking the picture.
FYI: they are at Heathrow airport!

He had a passion for finding out ***exactly*** how things worked, and if something was discarded for being outdated, or broken, he often delighted in trying to make it work again. On the other hand, whenever a new piece of technology entered our arena, Roger would be right there on it, finding out what it did, and how.

Roger also made a point of getting to know people: he believed – he simply knew – that the wheels of work turn more smoothly if people know and respect each other. Our porters and cleaners, every foreman on every job, the Fire Officers from Newcastle’s Fire Brigade, academic staff throughout the University: all knew him because he went out of his way to get to know them – not in a familiar way, but always in a friendly, professional way.

(Almost!) everyone liked him for his forthright honesty in all things. Sometimes he could rub people up the wrong way, but his reasons would usually be because he felt that they were failing in their responsibilities to the job they were being asked to do, or even simply in the responsibilities of being a good human being,

The Computing Laboratory has had many talented people, but I think that Roger was unique. He started with us, having been a mathematician at Swan Hunters (they built ships, remember?) in 1967.

Esso Hibernian at Swan Hunter', seen from Byker, c.1969

Having joined the Laboratory, Roger's first great task was to make sure that the NUMAC IBM 360/67 (see later) would be given its perfect home, according to very demanding specifications.

The Job

Roger's career spanned the age of gargantuan mainframes through to microcomputers and the wonders of networking (it **really was** a wonder: getting your computing service **at home**!). In the 80s and 90s, he was at the "the bleeding edge" of assisting our academic users to work at home – first via modems, and then via ISPs and "that Internet thing" (about which there is another Newcastle story to tell).

From the 1980s onwards, the Computer Room began to be filled, not with two or three juggernauts and their attendant machines, but with scores of smaller (but more powerful) service machines, and servers, and networking equipment. Roger's role did not change – it just got a lot more complicated.

His Operations Team moved, likewise, from feeding punched cards into incredible machines, to becoming experts in shepherding the many systems in the Computer Room ... and subsequently out on campus. Under Roger's tutelage "the Ops" also formed our first phone-in Help Desk which was a terrific service greatly appreciated by all our users from the 80s onwards. (By the way, it was not within our remit to support undergraduate students at all until the early 90s: imagine that.)

The Computing Laboratory: we go back a long way

Newcastle became a University in 1963. Before then, it was King's College, Durham University. King's College was where most of the Science and Engineering departments were located.

Our first Director was Professor Ewan Page (later Vice Chancellor of Reading University). It's very instructive to read his informal CV[1] at the BCS. Here's an edited version of the start, which shows direct links to the prehistory of computing:

> *In 1949 I was a student at Cambridge when Maurice Wilkes[2] offered a short course on programming an electronic computer; ... I was one of about two dozen who attended. Three years later ... I was doing research in Mathematical Statistics and needed a lot of calculating (for those days) to find the properties of some cumulative sum schemes for detecting a change in observations that I had proposed. ... EDSAC[3] was working, and Maurice's committee agreed that I might use it ... A little later in 1954 Durham University appointed me to direct their new University Computing Laboratory, soon to be equipped with a Ferranti Pegasus[4] machine. The experience of EDSAC was key to my appointment; there were very few people who had done any original work at all on a computer in 1954, ...The Pegasus was the only machine in the North East ....*

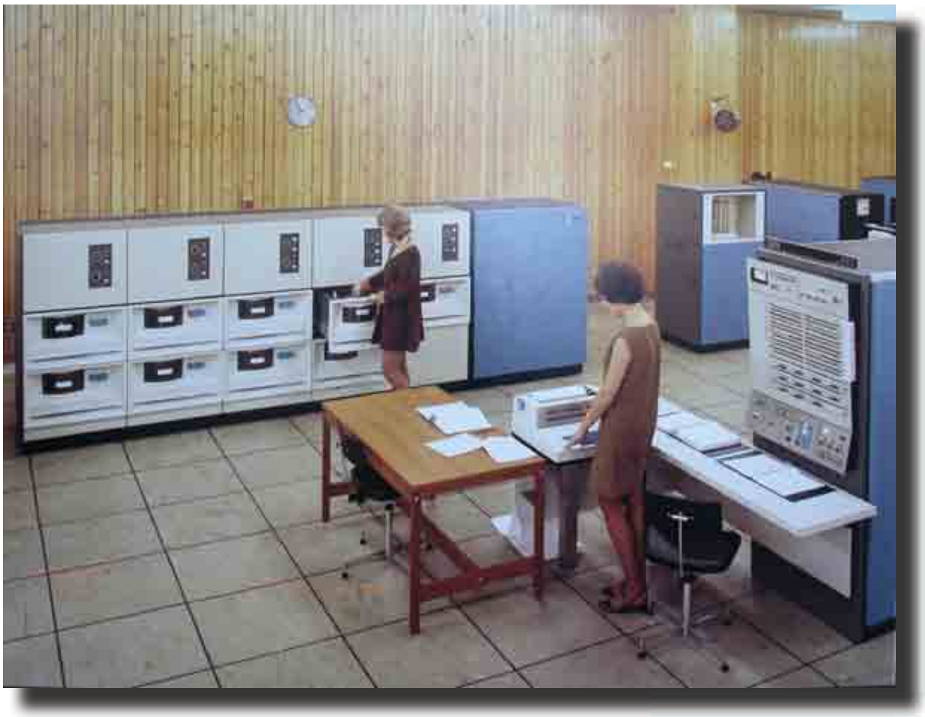
Professor Page was a human dynamo, very ambitious for his pioneering Laboratory. You can see a time-line on the history of the Laboratory at Computing Science's History pages[5]

NUMAC

The most significant single step in that timeline (i.e. in terms of computing power, all the considerable academic achievements aside) was the acquisition in 1967 of the mainframe computer, the IBM 360/67 (read all about it)[6].

Initially, only half a dozen of these machines were built by IBM in the world. They were capable of something unique: they used virtual memory to enable a new concept – time-sharing. IBM had played with the idea, were about to drop it, and then agreed to make a few examples under pressure from the computer scientists at MIT and at the University of Michigan. Our computer scientists knew their computer scientists, and hence the Computing Laboratory became the proud owners of a machine unique outside the USA.

[1] http://www.bcs.org/content/ConWebDoc/54755
[2] https://en.wikipedia.org/wiki/Maurice_Wilkes
[3] https://en.wikipedia.org/wiki/Electronic_delay_storage_automatic_calculator
[4] https://en.wikipedia.org/wiki/Ferranti_Pegasus
[5] http://www.ncl.ac.uk/computing/about/history/
[6] https://en.wikipedia.org/wiki/IBM_System/360_Model_67



One of the best known pictures on the Internet (if you're googling for IBM 360/67). This is our Computer Room. Roger says: "This is set up: the computer room was never that tidy, and that 7th switch from the left on the front of the processor is down, which means 'DISABLE INTERVAL TIMER' i.e. the processor was doing nothing."

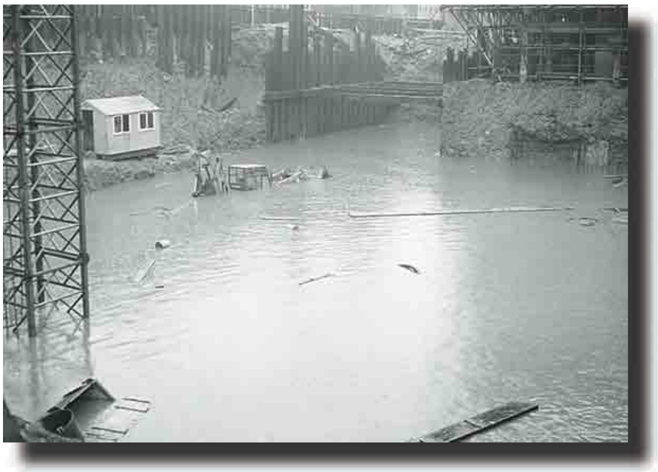
Because of the continuing close cooperation between Newcastle and Durham Universities, a joint bid was put forward to install a 360/67 in the Computer Room of the newly opened Claremont Tower. This computer cost well over £1,000,000 – over £16,000,000 in 2017 terms (money for computers was at that time awarded directly, on merit, by Government).

An organisation was created to run this computer and all its unprecedented services: Northumbrian Universities Multiple Access Computer. NUMAC. Newcastle provided 70% of the resources, Durham the other 30%.

In 1967 the 360/67 was one of the biggest and most powerful computers in the UK. Incidentally the main computing system weighed about 12 tons; its successor in 1975, the IBM 370/168, weighed 24 tons; but its successor in 1985, the Amdahl 5860, weighed a mere 17 tons (*maybe something was happening, huh?*). The 360/67 required 200m^2 floor space, about half of our very large Computer Room. The Computer Room had (and still has) its own power supply. The 360 also required its own air conditioning plant, which was located in other Sub-Basement rooms. (All this kit was of course craned in via the Loading Bay, which is why it is there: the architects designed a Sub-Basement Computer Room, with all necessities. Although they did make one little, fundamental mistake: see below.)

Roger and his colleagues worked hand-in-glove with IBM's engineers to prepare the Computer Room, oversee the delivery, and then the commissioning. After that it was simply a case of maintaining 100% running, 365 days a year .... when the service failed, operational faults should not be the cause!

Of course there were failures – regularly: compared to modern times, this was steam age computing (sometimes almost literally if we had a flood in Claremont Tower), and so the systems (i.e. on-call experts) for dealing with them were sophisticated.



Our very first flood, c.1964?: the builders of Claremont Tower discover the subterranean Pandon Burn[7] (which has been running underneath Claremont Road since the 19th century). This one wasn't our problem, but we've had several floods since, one caused by the Burn, others caused by the Tower's central heating.

For Roger, any operational emergency was both a pain in the neck (no matter who was on call, he almost always became involved) and also a challenge, in which he usually took a certain pleasure.

The Split

The other very significant date in Computing Science's time-line[8] is 1991: it was then that the University split the Computing Laboratory into the Department of Computing Science, and the University Computing Service (UCS, later ISS, now NUIT). The split was regretted by the staff on both sides. During the 90s and 00s, the Computing Service absorbed the University's Admin systems, and soon acquired SAP (a consultant-led decision on the part of a committee set up by the University, which quite deliberately did not have any members with computing expertise).

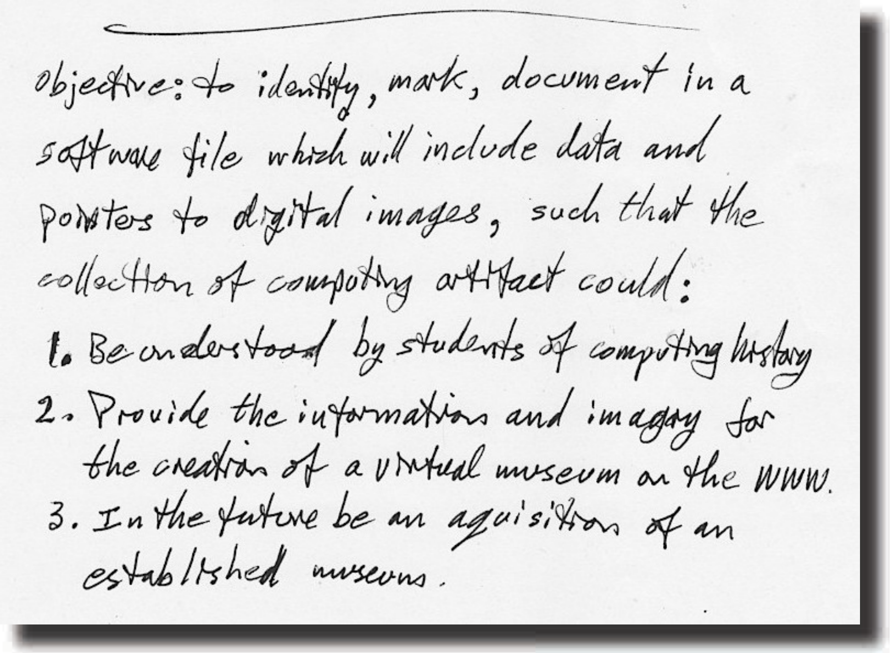
[7] http://blog.twmuseums.org.uk/the-real-barras-bridge-and-newcastles-beautiful-lost-dean/
[8] http:///www.ncl.ac.uk/computing/about/history/

Roger’s Collection

Back to the Museum Of Computing Artefacts ... throughout his career Roger collected machines, and bits of machines, that had become redundant. His natural inclinations aside, he knew that these items were too important just to be discarded. Being privy to places that few others knew about, he stashed them wherever space could be found.

He began to catalogue his collection after he retired, using a database which is popular with small museums. He has over 500 items catalogued; each one is marked with its number to the recommended standards; each database entry has space for up to 30 fields of information. The more that I, myself, find out about this database and the collection, the greater my admiration for Roger’s skills increases: within the limitations of his budget (i.e. zero), there are no half-measures.

As his collection grew, he began thinking about creating a Virtual Museum: a website based upon the artefacts. **Why?** Well, among his papers we found the actual bit of paper ... ...



objective: to identify, mark, document in a software file which will include data and pointers to digital images, such that the collection of computing artifact could:

1. Be understood by students of computing history
2. Provide the information and imagery for the creation of a virtual museum on the WWW.
3. In the future be an aquisition of an established museums.

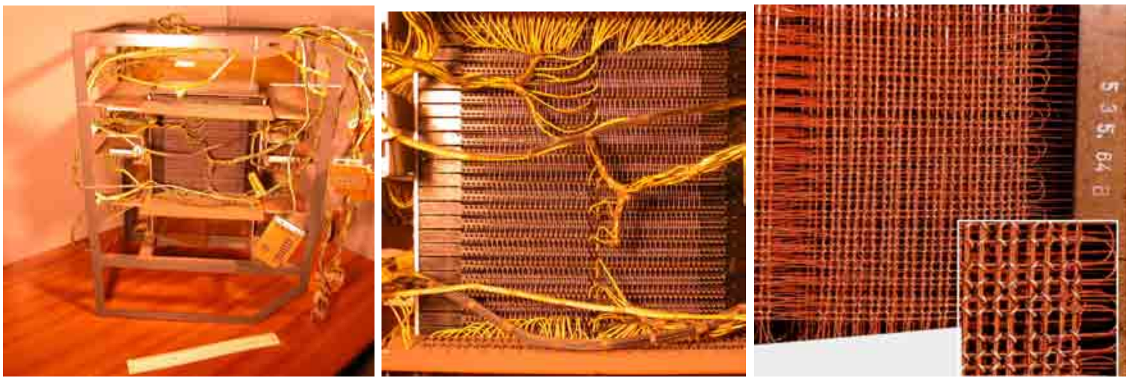
... ... on which he had jotted down his original objectives (perhaps it was a recommendation, to do this, from the creators of the museum catalogue software that he acquired):

The Virtual Museum, and the real one

As Adèle said at the start, Roger's Virtual Museum is now at http://moca.ncl.ac.uk/. He started his wesbite in 1994 – and it shows(!), being plain old HTML. If you trace a path through his museum you will sometimes find yourself "in a maze of twisty little passages" .... however (unlike the original Colossal Cave[9]) these passages are not all alike, but bejewelled with nuggets of fascination.

A few more glimpses of Roger's Collection ...

We discovered this lump in the corner of one of REB's store rooms. It's not catalogued; Clive thinks it might be part of a disk drive from the Amdahl; it weighs 158lb (there's a warning label on it); we haven't moved it yet.



In contrast, here's some of the core storage from the KDF9 (vintage 1963). Read about the incredible, complex, delicacy of this form of storage here[10]. and see more pictures of this artefact here[11]

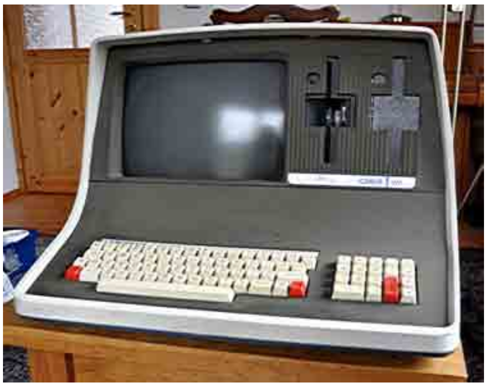
[9] https://en.wikipedia.org/wiki/Colossal_Cave_Adventure

[10] http://moca.ncl.ac.uk/corestore/h3wcsw.htm

[11] http://moca.ncl.ac.uk/corestore/KDF9cs.htm

Disk store: there's a great little web page here[12] *about the first photograph, which rather boggles you.*

Contrast Picture 2 with "The Lump" on the previous page. This item (160GB) must be really recent but I haven't found the web page about it yet.



Finally, one of the first ever micros that we acquired. Yes it really was called a Superbrain. Some years ago, we found the invoice for it: it cost about £4000. But – come on! – not only did this baby have ***two*** *(count 'em) floppy disk drives: it had a hard disk!*

Working with Professor Brian Randell and Dr Will Blewitt of Computing Science, Roger had already organised physical displays of historical computing equipment. One of these is a permanent exhibit in Claremont Bridge, where there are display cabinets containing some of the artefacts (go have a look: it's outside the Tower end of "The Rack" on Floor 6).

The School of Computing Science will move in September this year to the Urban Sciences Building at Science City. 2017 is their official 60th birthday! They plan to have computing artefacts displayed prominently in the entrance hall of the building. Most of these will be from Roger's collection. Roger was discussing the plans with

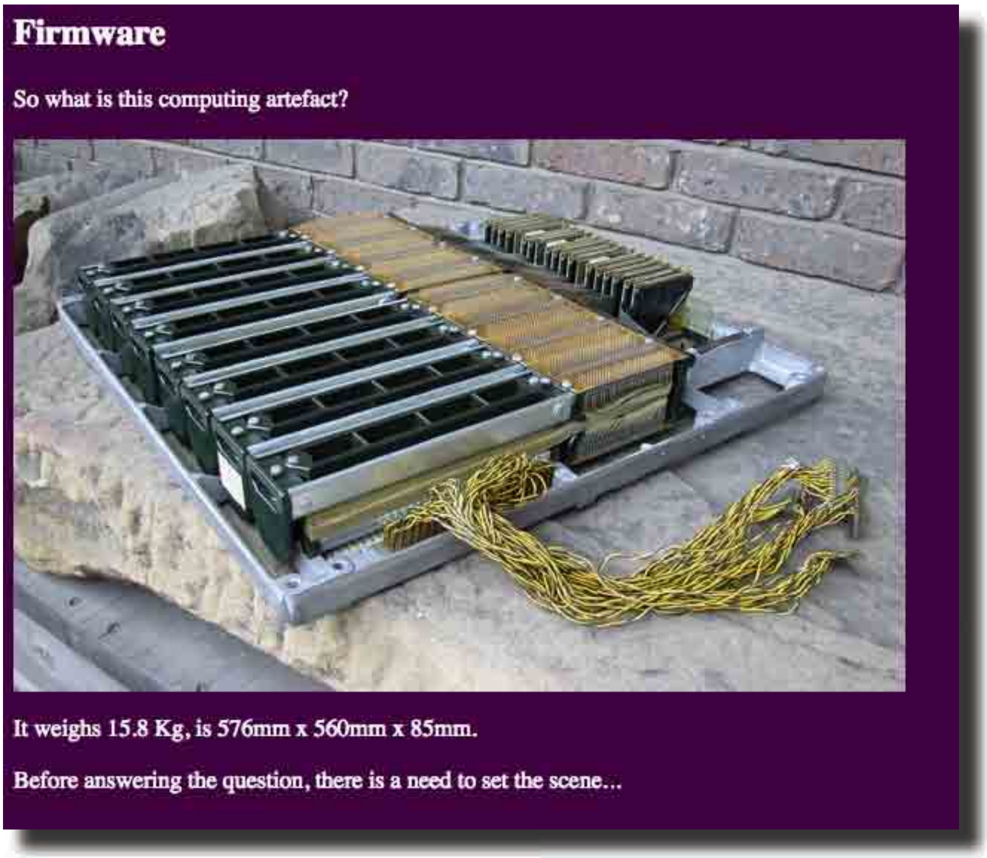
[12] http://moca.ncl.ac.uk/corestore/512MB2004.htm

CompSci in 2016, but he didn't live to see (or more interestingly for him: help to organise) the display.

Our tasks in the Museum Project are:

- Gather together Roger's collection into a safe, contiguous space; Jason has kindly found us such a space.
- Verify Roger's catalogue (over 500 objects so far) against what we have collected.
- And then in effect hand over the Roger Broughton Museum of Computing Artefacts to Computing Science, for their use and direction, in the ways in which Roger envisaged in his original objectives.
- At present, the questions of future storage and curatorship are too far in the future to be concerned about :-)

We must now continue without our dear friend. His website is about to be taken over by Computing Science (who have exciting plans for a VR rendition). They will make it somewhat fancier, and it will get a rigorous editing, but the substance of it will remain the same: Roger was all about **content** and **accuracy**, not style.



Firmware

So what is this computing artefact?

It weighs 15.8 Kg, is 576mm x 560mm x 85mm.

Before answering the question, there is a need to set the scene...