SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED COMPUTING NOTE 512

issued by
D A Duce

VISITS
Notes on a visit to QMC, Professor Coulouris,
Drs Bornat and Abramsky, Mr Page.

17 November 1981

DISTRIBUTION:
R W Witty
F Chambers (Logica Ltd)
D A Duce
Miss G P Jones
Investigators/Coulouris
Investigators/Bornat
Investigators/Page

INTRODUCTION

George Coulouris gave a brief overview of the three DCS projects at QMC.

The MMI project has just been extended by 18 months. This project is aimed at discovering software techniques for effective interaction at an advanced workstation and exploring the use of concurrency and multiwindow display systems. An experimental filing system based on the QMC text terminal has been constructed.

Ian Page is interested in dynamic displays using a DAP-like structure.

Richard Bornat and Samson Abramsky have an approach to the design of distributed systems based on Pascal-M and the associated system organisation. This has been pushed as far as posible using the text terminal and Unix on the 11/34. An interim workstation based on the 8086 and components donated by Nexos was built and a stand-alone Pascal-M system is being developed for this.

There are three RA's employed on the MMI and Pascal-M projects - Steve Cook, Peter Shepherd and Bob Newman (recently recruited from Logica). George has one man year left on the MMI grant.

Recently all effort has been diverted into Pascal-M pending the arrival of PERQs. George was upset about the delay in processing PERQs. However, they have done nothing about purchasing the extra 8086 workstations for which funds were awarded in early summer. Bob Newman is evaluating products.

George asked urgently for:

1. a firm delivery date for 2nd PERQ
2. Ring equipment
3. PERQ-Ring interfaces a.s.a.p and specification a.s.a.p. (Bob Newman will develop his own if they do not like Tony's!)

PASCAL-M

Richard and Samson described the Pascal-M project.

At the outset this was conceived as a theoretical project, involving much thought about language design and very little applications work.

However it turned out that George had a machine and wanted software. The two projects combined to work on fundamental programming issues to provide the kind of interaction George requires. There are two paths being persued:

1. a stand-alone implementation
2. a distributed implementation

Richard, Samson, Peter Shepherd and Bob Newman have been working on the implementation. A demonstration system running on the 8086 workstation is operational. They are working on a filestore system for which process creation is required. The plan is then to build up to a system which can boot itself on the 8086 and then to a stand-alone system.

The 8086 kernel (2Kbyte) is coded in assembler. The system is supported by 'C' programs on the 11/34 which are being rewritten in Pascal. The intention is to microcode the kernel for the PERQ. This will then enable them to perform self-contained applications experiments on PERQ.

Motivation for Pascal-M comes from the notion of (type-checked) message passing as a primitive. They have proposed modifications of Pascal based on CSP. Pascal was chosen as base language for local reasons of convenience (existing compiler, support tools etc) any typed language could have been used.

The research hypothesis is that the primitives proposed are sufficient for writing systems, but do they give elegant solutions? The only way to find out is by writing a real system.

The flavour of the language is guarded commands and mailboxes. Mailboxes may be sent in messages to alter the communication pattern. Once interfaces are separated from processes, a much nicer system description results.

They have succeeded in writing drivers in Pascal-M to control both DMA and character devices without cheating.

CSP only has input guards, but Pascal-M has both send and receive guards which figure largely in the demonstration system and provide clean, flexible event driver programs.

The file system implementation is based on a process representing every active file. The front end of this process talks to the client, the back end to the disc. Any service can be handled in a similar manner.

The system also contains a number of directory processes, for example the window process, which puts an applications process in touch with a window. A second directory process is used to put the keyboard in touch with a window.

Robin Milner has something similar to mailboxes (ports) in his calculus, but has not extended this so that port names can be passed in messages.

Mailboxes in Pascal-M can be treated as ordinary lanaguage objects, special tagging or use of separate segments for Mailbox values is not necessary as protection is performed by the compiler.

The deliberate decision was taken at the outset that no additions would be made to the underlying Pascal, for example message guarded commands have been added, but NOT generalised guarded commands - to have changed Pascal would merely have muddied the conclusions.

One facet of the project plan was to produce an eternal compiler. The notion is that an extensible system can be built if the compiler remembers code already compiled, types, mailboxes etc. The major problem is how to deal with new types - how can a file system written on Tuesday deal with new types, introduced on Wednesday?! The problem is particularly acute with directory processes.

There is a solution based on a special type any, which is an ordinary type wrapped up, but which can only be unwrapped to the original type.

Richard believes that to produce a distributed Pascal-M will require a lot of thought and ingenuity, but will require relatively little code to be written. If two nodes want to communicate using a variety of fixed (known) types there is no problem; the difficulty comes with the desire to define new types. The problem is some version of the distributed database problem. Some of the problems are known already and it is believed they are soluble.

We were shown Pascal-M demonstration running on the 8086 workstation. The documentation comprises some 5 application processes running in separate windows on the display. The applications are towers of Hanoi, chess, text editor, alarm clock and clock. The system in toto comprises about 30 processes.

DISARRAY

Ian Page spoke about the Disarray project. When Ian joined QMC the text terminal work had just started and he became heavily involved in this.

A graphics capability is obviously missing from the text terminal, which can only work on a character all basis. To support bit map graphics would require some 100 times more processor power. The text terminal uses a purpose designed 16-bit processor built from state of the art bit slice logic. A two orders of magnitude improvement is not feasible, so Ian started looking at parallelism.

The central operation in most raster graphics is the rastmap operation defined by Sproull and others, basically a bit-wise move from one region of memory to another. A logical operation may also be specified which is applied to source and destination values, the result being written into the destination memory location. The problem is how to build a machine to perform this operation very fast. Perq and Alto microcode this operation, but it is still not fast enough.

A DAP-like structure seemed ideal. One can handle bit maps the size of the array in 1 operation. Shift operations also have nice properties.

Ian is building a 16x16 array. Staff problems slowed down progress on the initial grant to the point where only a design (rather than hardware) was completed by the end of the project. Construction is now in progress with the help of RAL.

Ian has been talking to ICL and in fact they have built a powerful personal computer to his design. He thinks they may be interested in building something like Disarray into something like PERQ. Ian sees limited, special purpose, forms of parallelism as a very attractive option. One cannot get another order of magnitude from a "single box" computer because of technology limits and generalised parallelism (1000 microprocessors say) is beyond the reach of present software practice to programme.

Disarray is also attractive because it could be used for matrix, vector operations as well as display operations in a personal computer. It also has potential for fast pattern matching operations.

Reddaway and others from ICL's DAP team have visited Ian and seem willing to learn about Disarray.

Ian has a number of peripheral interests, including VLSI. He has recently been awarded some pump priming facilities at RAL. He is currently building some software tools (in Pascal) and will then be making chips, one of which will be the Dissaray processor, though he has no plans to build an entire system in (V)LSI.

Bell Labs (NJ) have also expressed interest in Disarray and have offered Ian a post, which he seems now to have turned down.

MAN MACHINE INTERFACE

Goerge gave a demonstration of Mike Lammings' filing system running on the text terminal and 11/34.

As mentioned earlier, all efforts have been diverted into Pascal-M and nothing has been done on MMI for the last nine months at least.

George has had contact with Butler Cox and many other companies concerning MMI, but no formal collaborative projects have arisen.

EXPLOITATION

There was a general discussion with all the investigators about exploitation of the QMC work.

Richard Bornat feels that Pascal-M as mounted on PERQ will be a marketable product. There was a lengthy, inconclusive argument between Samson and Richard as to whether it will be an attractive product for less powerful workstations.

Pascal-M is completely portable in the sense that the language compiles into a stack based code (adaption of Tanenbaum's eml machine) so porting the system merely (!) entails writing a translator, interpreter, kernel and device drivers.

Pascal-M has a lot of potential, but much research still remains to be done. It is development work to complete the stand alone uniprocessor implementation (almost), but the distributed case needs more research. The distributed case needs to be demonstrated before the single processor system has any great value. The stand-alone system provides facilities camparable with those of Unix, but is probably not a commercial proposition because Unix is widely used and so much code is available running under Unix.

The stand-alone system is unlikely to be ready before Summer 1982 at the earliest.

William Newman expressed interest in Pascal-M for programming the Logica workstation before he left Logica, but this does not seem to have been persued.

Peter Hall (SERC/DoI Computing Coordinator) has set up a meeting (week commencing 16 November) between Norman Brown and Vic Maller (ICL) and George/Richard/Samson. George thinks the purpose of the meeting is to discuss Pascal-M with a view to putting up a proposal to Wilmott to mount Pascal-M on PERQ. George has in mind an SERC/ICL sponsored team to mount Pascal-M and some tools. Fred Chambers asked that George contact him after the meeting so that Fred can follow it up.

Peter Hall has apparently proposed to the office that they introduce a cooperation research award analogous (almost) to continuous stream procedure. He is suggesting that up to £10,000 should be available for one year projects, with only a memorandum of intent between the two parties. A proposal for such a project would just go to the relevant Committee Chairman (+ referees?). The idea is to get a project off the ground quickly.

George is having a meeting with Rivers-Latham from GEC Hirst Research Centre in the near future. Rivers-Latham has a brief to set up a pilot office system, but George does not think the proposal is very adventurous.

jg-wp