WORKING NOTE D2

Alan Creak
3 May 1986

PROPOSED PROJECT : A VIRTUAL MACHINE AS A STANDARD.

At present, there is no standard computer for use by disabled people even within New Zealand. Apples predominate, but there is a sizeable body of Polys around Wellington, and a scattering of other systems throughout the country (1). The Technical Aid Trust has suggested that the IBM personal computer should be adopted as standard (2); this seems to me to be a bad idea, as its main immediate effect would be to introduce yet another sort of machine into an already overpopulated field, while in the long term it would anchor development efforts to obsolete hardware.

Indeed, I suggest that to base a standard on any particular piece of hardware would be misguided. Such a course would certainly not make it any easier to take advantage of new developments in microprocessor engineering as new and more powerful processors appear on the scene; and it would make it equally difficult to take advantage of cheaper systems, which may be less powerful in some sense than the IBM personal computer, but may turn out to be perfectly adequate for some purposes.

An alternative route to standardisation could be based on an agreement to write programmes in some commonly accepted programming language. The language C has been suggested for this purpose (3), for example. I think that this suggestion is an improvement on any attempt based on hardware : in principle, at least, a C compiler could be written for any computer, so some degree of transportability is certainly possible. Even this proposition, though, can't cope too well with the considerable differences between different systems in terms of the architectures of their microcomputers, and their diverse ways of handling peripheral devices; and other machine-dependent features would probably prove troublesome.

What we require is a standard environment which is nevertheless compatible with any microcomputer great or small, existing or yet to be invented. We would like our standard environment to be well adapted to the needs of programmes for the disabled : while we're designing our own universe, we may as well design a nice one. We would like facilities to handle a wide variety of input and output devices and modes of use, including such features as good graphics, communications, and control functions for robotic applications. We would also like our environment to be flexible, so that it can evolve to meet new needs and to take advantage of new methods and facilities.

THE PROPOSAL.

It is possible to construct such an environment as a "virtual machine" (or "abstract machine" – I prefer the first term, and will use it systematically). This is a computer programme which simulates the action of a computer in that it interprets a "code" stream, and performs actions represented by the code on the data in its "memory", just as if the code were being executed by a real computer. The advantage of this approach is that you can write a standard virtual machine for any real computer; while the same is true of, say, a C compiler, the virtual machine also takes care of differences between the detailed behaviour of the two computers at the "hardware" level, so the code is truly transportable. The major disadvantage of a virtual machine is that, as execution is interpretive, it is correspondingly slow. With good design, though, the machine can be made to operate fairly slowly rather than very slowly, and for many applications speed is not a critical requirement.

This approach has been successfully used in other fields where software is required to run effectively on many varieties of hardware : perhaps the most notable example is the UCSD Pascal P-machine. The result of such a standard would not necessarily be that everyone could do anything on any machine; the real hardware still imposes limitations. It does not offer a solution to the requirement that disabled people should be able to use programmes written for able users, as specially written – or, at least, specially compiled – programmes are still required; further extension of the virtual machine approach in such a direction is a possibility (4). It does not, in itself, rationalise the present scene; but existing programmes in common languages could be run on the virtual machine at the cost of writing a compiler to translate that language into code which runs on the virtual machine.

The approach does offer a good prospect of software compatibility over a wide range of different computers, and a hope of advancing into the future by orderly evolution rather than by starting again from scratch at each new hardware or peripheral release (5).

To proceed along these lines, one could follow a programme like this :

1. Find out what sort of facilities should be provided on a computer specifically designed for use by disabled people. (For example : ready conversion to use a wide range of input devices, special modes of presentation of output, multiple programmes active simultaneously [6], etc.)

2. Design a prototype virtual machine incorporating these requirements.

3. Write and test the virtual machine. The programming language at this stage could be something like C or Pascal, for which compilers exist on a range of microprocessors; later it should be possible to express the machine in a language specially developed for the purpose.

4. Improve the basic functions of the virtual machine in the light of experience gathered in the previous step, but without yet writing extensive programmes.

5. Transport the virtual machine to several different type of microcomputer. This exercise will probably suggest further improvements, or dictate necessary compromises. (From this point on, all work done on the system is automatically accessible on all machines for which a virtual machine exists.)

6. Write a compiler for a suitable language. (This could be either an existing language, such as Basic or Pascal, which would obviously be valuable to run existing programmes; or one designed for the purpose, which would be better for writing the virtual machine's own software. Expediency suggests that Basic would probably be the best place to start.)

7. Work on improving the compilers, etc. The system is now up and running, and further advances are in the nature of development.

REFERENCES

MBS B.T. McMahon, E.J. Burkhead, J.P. Sampson : "*Computer access and applications to career counseling with vocational rehabilitation clients*" (D:ATA Institute, The Catholic University of America, Washington DC, 1985)

AC Adrianne Cranshaw : "*Computer use by disabled people*" (New Zealand Council for Educational Research, 1985)

[1] AC p6.
[2] AC p7 ; Neil Scott, letter to Alan Creak, 6 January 1986.
[3] Neil Scott, conversation with Alan Creak, 22 January 1986.
[4] MBS p28; Alan Creak, Working note D6 (1986).
[5] MBS p27.
[6] MBS p29.