

SCIENCE AND ENGINEERING RESEARCH COUNCIL  
RUTHERFORD & APPLETON LABORATORIES

COMPUTING DIVISION

D I S T R I B U T E D   C O M P U T I N G   N O T E   4 7 0

PERQ  
DISCUSSION PAPER 10

issued by  
A S Williams

Unix implementation on PERQ

17 September 1981

DISTRIBUTION:

- R W Witty
- L O Ford
- C Prosser
- A S Williams
- F R A Hopgood
- Perq/General File

Several strategies are presented below. For the serious contenders, major work items are listed with estimated timescales, and arguments are presented.

1. RAL staff implement Bell Labs Unix V7
  - a. C compiler with Q code assembler-linker 3 months for 2 full-time staff who are experts. CMU have a compiler which may save us much of this.
  - b. translate Unix PDP-11 asm into Q code (linker required).
  - c. translate Unix PDP-11 object code into Q code.

a, b and c all require effort in implementing the Unix kernel:

  - memory management
  - protection (microcode needed)
  - device drivers
  - signals, interrupts, user-kernel communication
  - swapping of processes
  - fork and other primitives stack manipulation needed
  - expansion of data areas.

We estimate 3 months for 2 experts.

- d. Interpreter for PDP-11 object code written in Pascal.
  - e. Microcode interpreters for PDP-11 timescale inestimable.
2. Cooperate with HCR Inc.
 

Same effort as 1a, but shared with HCR. HCR's timescales are long.

3. Implement the Unix-C program interface using the SPICE-81 kernel functions.

This requires a C compiler as in 1a.

In place of the kernel development in 1a, a library of functions is required. More detailed study of the spice kernel is needed, but we estimate 3 man months for this.

A study of the Unix utilities which make assumptions (about eg file system formats) is required. Non-critical utilities could be implemented over a longer period, after Unix is in use.

4. As 1 but implement a virtual machine aimed at the C language.

Additional work (prior to that in 1a) would be to design the virtual machine (1-6 months) and write the microcode to implement it.

#### ARGUMENTS

Con 1d. One can expect 2 orders of magnitude performance degradation.

Con 1e. We have no expertise in Perq microcode.

Con 1d,  
1e, 1b, 1c. No long term benefit accrues.

Con 1b, 1c. Instructions like `cmp (r0)+`, `(r0)+` require on the order of 5 memory cycles, both signed and unsigned tests (the latter does not exist in Q code). Condition codes, segment faults, jump address resolution all cause problems. Optimisation would require colossal effort.

Con 1a. - limited address space  
- Unix filestore incompatible with spice  
- process swapping rather than paging  
- Perq Pascal compiler will not run under Unix- language compatibility is lost

Pro 1a  
and 3 C compiler work would be useful for 'any other Pcode machine' C compiler.

Pro 3 - 32 bit paged virtual memory per process  
- long term compatibility with SPICE  
- CMU did preliminary study of this and foresee no problems  
- Spice kernel available December or January  
  
- Ada/Pascal compiler compatibility

Con 3 - relies on CMU: critical path not under our control  
- Spice kernel might be unstable for a while, requiring maintenance, release procedures etc.

Pro 4 Under our control.

Could lead to a more efficient implementation of C.

Con 4 3R software will not run, in particular the Pascal Compiler,  
and SPICE software (Ada, LISP, Scribe, Canvas etc).

Support would be required for the microcode.

We do not have expertise on Perq microcode.

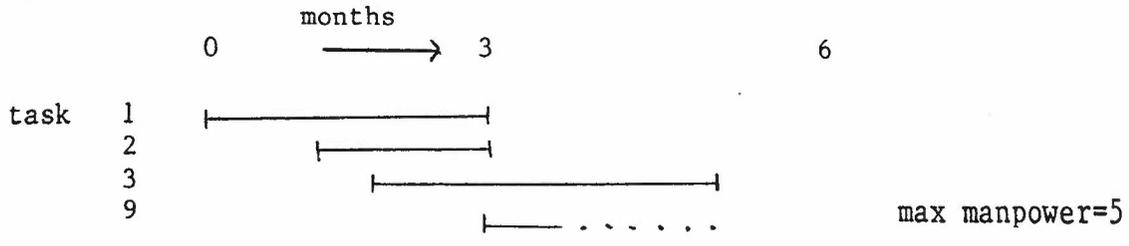
Only a single language is well supported.

APPENDIX

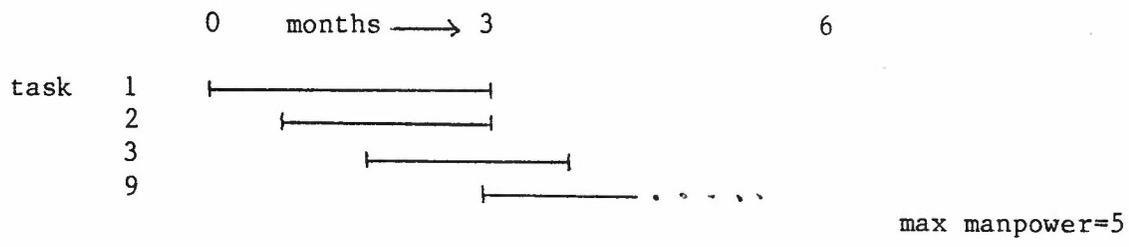
<u>Timescales</u>	options			task
	1a/2	3	4	
C - Qcode Compiler	3x2	3x2		1
Qcode assembler and linker	2x1	2x1		2
Unix kernel m/c dependant parts	3x2		3x2	3
Unix system interface		1.5x2		4
Design C machine			4x2	5
Microcode C machine			3x2	6
C - C machine compiler			3x2	7
C machine linker			2x1	8
Unix shell and Utilities				9

Figures are months x men

option 1a/2:



option 3



option 4

