Inter-Office Memorandum

To	Distribution	Date	November 13, 1978
From	R. Johnsson	Location	Palo Alto
Subject	Microcode Swapping Meeting	Organization	SDD/DE

XEROX

Filed on: [Iris] < Johnsson > Memos > USwap1.bravo

A meeting was held on November 7, 1978 to discuss Microcode swapping as an approach to evolving software and hardware toward non-Alto compatible processors. Attending were Hankins, Johnsson, Koalkin, Lauer, Lynch, McJones, Metcalfe, Sandman, and Wick

How are we going to debug Pilot 3.0 which runs in a non-Alto compatible hardware/microcode environment?

Pilot 3.0 has:
- Devices
  - RDC (Shugart 4000)
  - UTVFC (17" monitor)
  - X-Wire
- PrincOps architecture
  - traps/faults
  - process switching
  - left-to-right opcode bytes
  - not final bytecode set
- Mesa 5.0
  - compiled in /-A mode
- no Nova emulator

Alto/Mesa 5.0 debugger expects:
- Devices
  - RDC (Shugart 4000) to access Pilot files
  - IRDC (Diablo 31) to swap its own code and symbols
  - IUTFP (850 monitor)
  - Ethernet (not essential) to retrieve remote files
- Alto compatible architecture
  - traps/faults
  - process switching
  - right-to-left opcode bytes
  - restricted bytecode set
- Mesa 5.0
  - compiled in /A mode
- Nova emulator
  - to do InLoad/OutLoad
  - process implementation (invisible, replaceable)

The problem is to debug Pilot 3.0 using a Debugger which does not depend on Pilot. Conversion of the Debugger to run on top of Pilot is expected, but not for some time. Wick pointed out that this problem is not unique to Pilot 3.0. We will continue to have to debug new Pilots which run in incompatible environments (new devices, Workstation, PrincOps bytecodes). The Mesa group solves such problems on the Alto by resorting to Swat. This is not possible in the Pilot world without the Nova emulator.

Q: Run the Debugger on the Pilot hardware with what microcode?

A: Change microcode between Pilot and Debugger.

Alternatives:

No changes between Pilot and Debugger
Make one change at a time (no debugger on incompatible changes)
Midas
Remote Debugging

The alternatives were rejected as being too painful or taking too long (in light of current beliefs about schedule). The remainder of the meeting was devoted to outlining the task of swapping microcode.

The proposal is to swap microcode by parameterizing the normal boot sequence. The Debugger (and its Nub in the Pilot world) can then specify to the booting microcode just what microcode and boot format file are to be loaded as well as what initialization of devices/map/memory to perform.

Several problems arise in saving and restoring the state of the world. These are more acute in the Pilot world.

All devices must be stoppable
State of microcode managed by Microcode Exec must be restorable.
Devices and drivers must deal with reinitialization at (almost) any time.

The following outline of what happens when the Pilot world goes to the Debugger was developed:

- Enter nub. No procedure calls allowed -- preallocated frame.
- Save Mesa state (DumpState)
- Save IOCS state
- Stop devices
  Assume that device drivers can tolerate device going away.
- Save Emulator State (WDC, Xfer trap status, ...)
- Save Microcode Exec state
- Save virtual memory state (Map and real memory.)
- Boot Debugger's microcode and Debugger state

When the Debugger proceeds, control continues here:

- Init devices
  What about initialization overlays of microcode?
- Restore emulator state
- Restore state and leave nub (LoadState)

The following outline of what currently happens when the boot button is pressed was developed:

- Hardware reset
- EPROM → CS
- Jump to ram (from this point the boot is controlled by microcode).
- {Disk → CS}*
  Initialization
  Devices
  Diagnostics
  Emulator
- Disk → Memory
- Alto boot sequence

Action Items

McJones/Lynch: define Pilot boot file format. Input from Johnsson on current Alto format.

Hankins: provide more detailed description of what currently happens during a boot.

Johnsson: coordinate, plan, and implement.

Distribution:
Hankins, Koalkin, Lauer, Lynch, McJones, Metcalfe, Sandman, Wick