28 Aug 67
9:45 a.m.

C. Jones
R. Graham
K. Daley
J. Saltzer
N. Liddell joins 10:15

Assumptions necessary to validate scu data.

1. No scu data returned by ring will ever have master mode bit on.
   (User ring handles faults occurring in Master Mode.)
2. (Master mode procedures can produce only system faults.)
   (System faults are handled entirely in ring 0.)

Linker is now called directly by FIM.

a. loading 0
b. ~~⇒~~ fault stack
c. gate & switch ⇒ ring 1
d. loading 1
e. call linker.

1. ~~Linker~~ FIM must repair the brass and reload dsptbr or return from signal.
2. Fim should follow standard stack discipline if plugging is sb10 but sp wrong.
3. Fault stack must follow standard stack discipline in ring 0 unless illegal.
4. Normal path to segment control uses the fault stack.
5. Segment fault occurring on fault stack is handled on the connected stack.

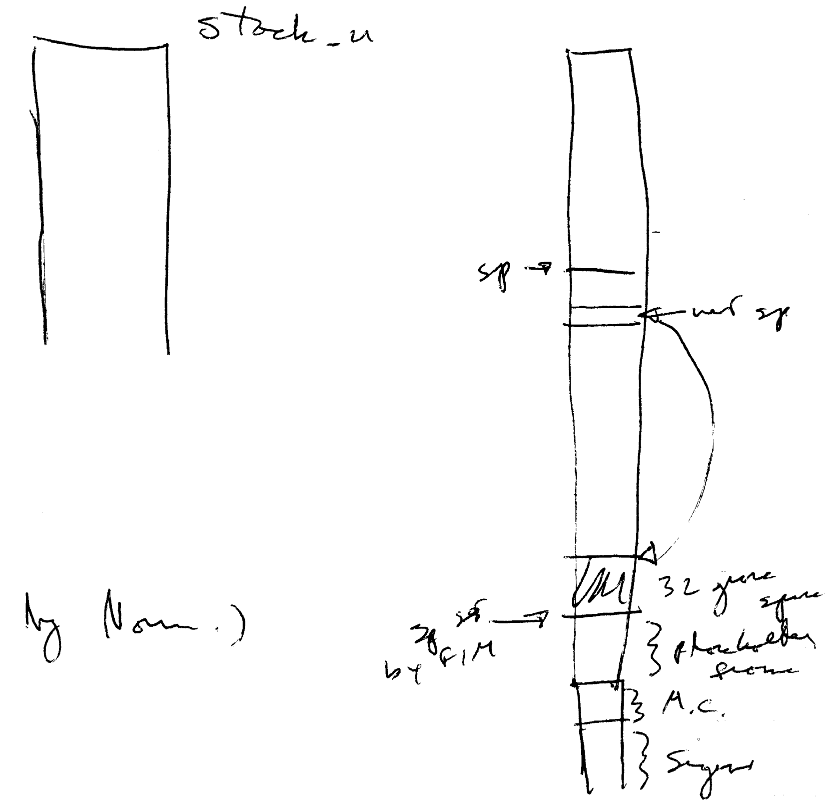
1. Go over interrupt interceptor code
2. Go over ssw/tlb code
3. Go over fault interceptor code
4. Discuss stack switching of gatekeeper when called by FIH.
5. Can we unwind through a fault

~~Wallet~~

Walkthrough of an overflow

While pushing stack_n, no fault can occur which would cause stack_n to be pushed.

fim + gate switch placeholder may be slightly different, since fim placeholder points back to same stack.
(placeholder frames need to be added by Noun.)



Walkthrough of forced ring fault (linker)

fim_stk must be available in all rings, master mode; store in ring 0.

To get a fault handled in ~~a lower privilege~~ an outer ring, an inner ring procedure should provide an in-ring handler which does an explicit signal.

The stack abandoned on a fault must always have a 32-word guard space left in case of a trace call.

Gatekeeper stackswitching.

Linkage format is not coded

Noel's program - sswitch

1. sp|18 is being set after sp is loaded; an ~~timing~~ interrupt will fail (cured inhibit.)

2. sp|18 is being left to point not high enough to cover the entire frame being copied into the concealed stack. Can be fixed by bumping next sp or by only copying 32 words.

3. Not following discipline on stack switching
   sp|28 contains own stack pointer.
   sp|16 should always point back to some stack.
   ~~not~~ (requires a hand coded return)

   OK in BD.9.01

4. One too many indirects on reference to <switch stack>|[return]
   Try eapbp ~~return~~ <switch stack2>|[return]
   (May not work yet?)

~~5. Load registers is loading garbage for registers (poor practice / turn to ...)~~ whatever