Pseudocode

```
Queue
void on message available():
                                                                RPC Call 0 (part 0) RPC Call 0 (part 1) RPC Call 0 (part 2)
    msg = receive_msg(); // pop msg
    // read message and determine
    // whether we need to pop more
    // messages in order to receive
    // the whole rpc call
                                                                  Message
                                                                   Queue
    // soft stack overflow here
                                                                RPC Call 0 (part 1) RPC Call 0 (part 2)
void on pagefault():
    // pagefault in stack region
    // so we request a frame and map
    // it at the faulting location
    rpc_call_get_frame();
void rpc call get frame():
    send get frame message();
    // wait for response
    while(!msg_available());
                                                                  Message
                                                                   Queue
    response = receive msg();
                                                                RPC Call 0 (part 1) RPC Call 0 (part 2) Response (frame)
    // response is now the second part
    // of the original rpc call
    // instead of the response to our
    // ram request
```

Message