

HOW FORTHBOX *MAY* HANDLE ASYNCHRONOUS I/O

Samuel A. Falvo II
<kc5tja@arrl.net>
2022 Jun 25

DESIDERATA I

- Interactive keyboard and mouse input.
- Background DASD access.
- Ideally, a single, easy to learn I/O model that works with a wide variety of devices.

Example: a CAD program can use keyboard and mouse input together in real-time, while saving UNDO information to disk in the background using interfaces any hacker can build in his or her basement.

DESIDERATA II

- A stable interface across small and large dictionary sizes.
- A stable interface that works across threads *and* processes.
- A better abstraction than explicitly managed queues.

KEY?

- Answers true if KEY (— c) will respond immediately.
- Supports asynchronous keyboard input *IF* used in a centralized event-loop.

```
: eventLoop ( - - )  
  BEGIN  
    KEY? IF handleKey THEN  
    . . .  
  AGAIN  
;
```

9P?

- Answers true if at least one 9P response is ready for processing.
- Should supports 9P processing *IF* used in a centralized event-loop.

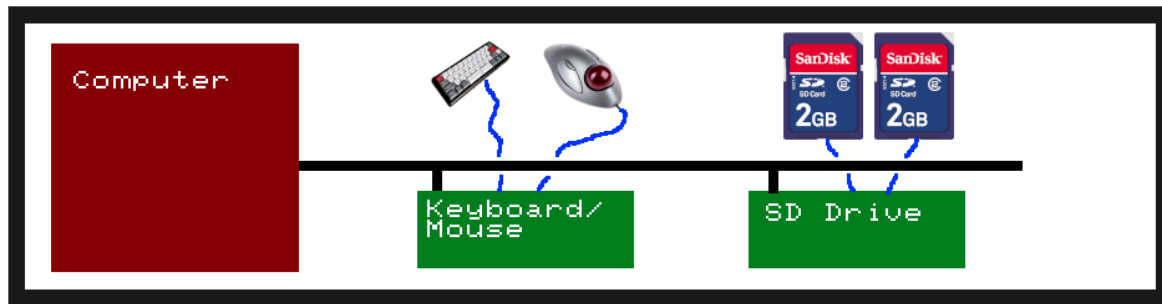
```
: eventLoop ( - - )  
  BEGIN  
    KEY? IF handleKey THEN  
    9P? IF handle9P THEN  
    . . .  
  AGAIN  
;
```

HOW DOES “9P?” KNOW?

- 9P is a protocol, *not a device*.
- Words like KEY? work best with devices; a good facsimile for interrupt status registers.
- Idea: Delegate queries to devices which *may* have 9P traffic.

ONE FORTHBOX EMBODIMENT

- Both keyboard and mouse, and SD card unit on the same SPI I/O bus.
- Keyboard is now subordinate to SPI connection!
- How, then, can KEY? work?



SOLUTIONS...

- Multi-threading, cooperative or preemptive.
- Device driver model introduced last month.
- Punt the problem to the application.
- ...

MULTITHREADING

- Pros
 - We don't strictly need query words anymore.
 - PAUSE causes a task switch; trivial to implement.
- Cons
 - How to create a thread?
 - How does MARKER interact with threads?

DEVICE DRIVERS

- Pros
 - Proven technique that works quite well.
 - Extremely high performance; used for hard real-time systems.
- Cons
 - Not “*Forth-y*”.
 - Needs a large number of support words in dictionary.

PUNT!

- Pros
 - Simplest possible design that could work.
 - DEFER-ed words allows “*lifting*” primitives into more advanced model later.
- Cons
 - *Extremely* hardware configuration dependent.
 - Each application needs to LOAD asynchrony support.

GOING FORWARD

- Looks like I'm going to punt the issue.
- Gather experience before revisiting the problem.
- More information needed!

END OF PART II

THANK YOU