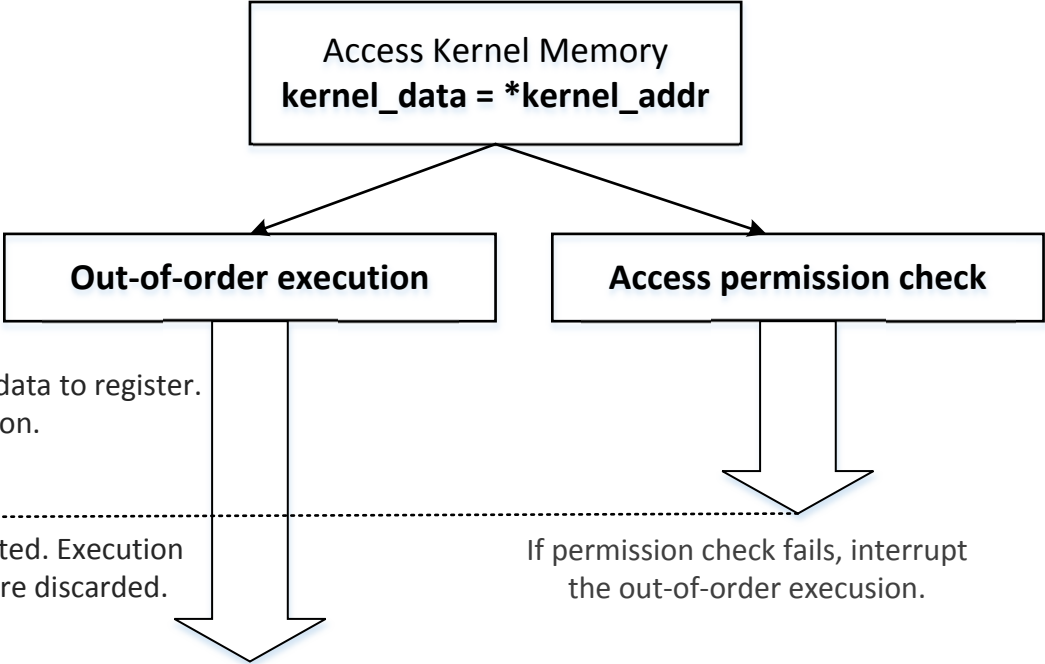


Access Kernel Memory
`kernel_data = *kernel_addr`



```
graph TD; A["Access Kernel Memory  
kernel_data = *kernel_addr"] --> B["Out-of-order execution"]; A --> C["Access permission check"]; B --> D["Bring the kernel data to register.  
Continue execution."]; C --> E["If permission check fails, interrupt  
the out-of-order execution."]; D -.-> F["Interrupted. Execution  
results are discarded."]; E -.-> F;
```

The diagram illustrates the execution flow for accessing kernel memory. It starts with a box containing the instruction 'Access Kernel Memory' and the code 'kernel_data = *kernel_addr'. This instruction branches into two parallel paths: 'Out-of-order execution' and 'Access permission check'. The 'Out-of-order execution' path leads to a step where kernel data is brought to the register and execution continues. The 'Access permission check' path leads to a decision point. If the check fails, it triggers an interrupt that discards the results of the out-of-order execution. A horizontal dashed line separates the successful execution path from the interruption path.

Out-of-order execution

Access permission check

Bring the kernel data to register.
Continue execution.

Interrupted. Execution
results are discarded.

If permission check fails, interrupt
the out-of-order execution.