

Nooks

Improving the Reliability of Commodity Operating Systems

Swift, Bershad, and Levy

SOSP'03

1. What was the motivation for Nooks?
2. What were the design principles and goals of Nooks?
3. What are the components of the Nooks Isolation Manager (NIM)?
4. Why is Isolation needed? At a high level, how is it provided?
5. Why is Interposition needed? At a high level, how is it provided?
6. Why is Object Tracking needed? At a high level, how is it provided?
7. Why is Recovery needed? At a high level, how is it provided?
8. How much work was it to implement Nooks in Linux 2.4.18?
9. For memory management, what memory rights does an extension have? What memory rights does the main Linux kernel have? How is this protection provided?
10. Why is a synchronized copy of the kernel page table needed for each domain? Are there any implications of this? Why does the Nooks design prevent bugs but not malicious extensions? What is the performance cost of switching between lightweight protection domains?
11. How is control between an extension and kernel domain handled with XPC? What is the purpose of a deferred call?
12. Did the Linux kernel need to be modified to support isolation?
13. In Linux, extensions sometimes directly access global data structures. How is this handled? When is XPC used? When is it not? How should one determine which approach to use?
14. What does a wrapper need to do? What is the difference between call-by-value-result and call-by-reference?
15. How much work is it to write a wrapper?
16. What must the Object Tracker track?
17. What does the Nooks recovery manager do?
18. What are some limitations of Nooks?
19. Does Nooks meet its goals of Isolation, Recovery, and Backward Compatibility?
20. Is the performance of Nooks acceptable?
21. Conclusions?