

1 Scheduler Activations

The Scheduler Activations paper states that deadlock is potentially an issue when activations perform an upcall:

“One issue we have not yet addressed is that a user-level thread could be executing in a critical section at the instant when it is blocked or preempted...[a] possible ill effect ... [is] deadlock (e.g., the preempted thread could be holding a lock on the user-level thread ready list; if so, deadlock would occur if the upcall attempted to place the preempted thread onto the ready list). (p. 102)”

Why is this not a concern with standard kernel threads, i.e., why do scheduler activations have to worry about this deadlock issue, but standard kernel threads implementations do not have to?

2 LFS

The LFS paper is strongly motivated by perceived trends in both hardware performance and user access patterns. Thus, its improved performance is driven by a set of underlying assumptions.

2.1 Explain what are the assumptions about hardware and access patterns that LFS depends on?

2.2 Explain which mechanisms in LFS depend on each assumption?

For the purposes of the following questions, NVRAM is persistent storage (i.e., will survive the loss of power) and has access times and throughput similar to DRAM.

2.3 At the time the LFS paper was written, non-volatile RAM (NVRAM) was not commonly available. If disks were replaced with NVRAM entirely, would the LFS design still make sense? Explain why or why not and be specific in justifying your answer.

2.4 In real-life the cost per byte of disk is likely to be far cheaper than NVRAM for some time. So instead, consider a situation where some NVRAM is available (e.g., 1/10th of the disk size). This NVRAM might be used for caching reads, caching writes, or storing particular meta-data. Argue which use might be most appropriate for improving the performance of LFS.