The problem

These two secenairos were described by Colin and Brian in a tech meeing 2008-11-28.

Both involve page faulting involved in handling memory mapped files.

Colin's Scenario#

1. Thread bob1 in user process Bob, opens a file on resmgr nfs
2. Thread bob2 in user process Bob, opens a file on resmgr io_pkt
3. bob1 mmaps its nfs fd, which causes
4. proc to run and lock process Bob's aspace
5. bob1 then reads from its fd which faults
6. proc runs to handle bob1 fault
7. meanwhile, bob2 does a read on io-pkt
8. bob2's read causes io-pkt to run but it faults on bob2's buffer, so enters page wait
9. note that io-pkt cannot exit page wait until Bob's address space becomes unlocked
10. remember bob1? bob1 had faulted reading it's memory mapped nfs file, a proc thread handles the fault by reading nfs
11. nfs happens to read io_pkt, which causes it to send to io_pkt
12. but io_pkt is single threaded, so nfs goes send-blocked
13. note proc wil not release the aspace lock on BOB until io_pkt replies to nfs and returns to proc
14. io_pkt's single thread is in page wait for bob2, which cannot be processed because Bob's aspace is locked
15. bob's your uncle

See the whiteboard snapshot mmap_lockup_1.gif, attacched.

Brian's Scenario

1. user thread Bob mmaps an already open fd: ptr = mmap(...,fd)
2. user thread Bob calls read() on the same fd, using the same ptr as the output buffer: read(fd, ..., ptr)
3. read() is the resmgr library locks the tructure of the file, as part of normal resmgr handling.
4. this is the first read, so the memory access to the file faults
5. a proc thread runs to handle the fault and issues a read on the same fd
6. proc thread calls read(), in the resmgr library, which also tires to lock the same attribute
7. the proc thread hangs forever

See the whiteboard snapshot mmap_lockup_2.gif attached.

Solutions considered#

So far, we have only a brief list (Feel free to flesh these out)

- Judiciously release the lock: may work for Colin's secnario
- require resmgrs to have at least one thread for every mmaped file: requiring behavior changes on usercode is difficult to manage
- map the whole file into memory on open: works but inefficeint.
- clever deadlock detection.
- use multiple finer granularity locks instead of a single resmgr attr clock, and a single aspace lock.

Cultural references quoted #

- we are go for throttle up
- All right, the kernel's secured.
- gaaaame over maaaan