

DragonFlyBSD - Bug #3200

master: weston screen freezes

08/05/2019 11:26 AM - peeter

Status:	Resolved	Start date:	08/05/2019
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>It has now happened at least four times in a row. After a few days of uptime, weston freezes. I can still ssh into the machine, everything else seems fine except the graphics screen is frozen. ps axlRH shows that weston is in state 'drmev'.</p>			
<pre>1001 1262 1 1261 25 199 0 266664 193568 drmev D0+ v1 7:36.42 /usr/local/bin/weston --use-pixman</pre>			
<p>This is the 'struct event_lock' in 'struct drm_device', which is used among other things in the context of ioctl. kgdb shows this backtrace for weston:</p>			
<pre>--- (kgdb) thread 1096 [Switching to thread 1096 (pid 1262/1, weston)] #0 0xffffffff80665e76 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810 #10 lwkt_switch_return(td->td_switch(ntd)); (kgdb) bt #0 0xffffffff80665e76 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810 #1 0xffffffff8067300e in tsleep (ident=ident@entry=0xffff8037b927c70, flags=flags@entry=1024, wmesg=<optimized out>, timo=timo@entry=0) at /usr/src/sys/kern/kern_synch.c:707 #2 0xffffffff8064040d in lockmgr_exclusive (lkp=lkp@entry=0xffff8037b927c70, flags=flags@entry=2) at /usr/src/sys/kern/kern_lock.c:381 #3 0xffffffff83087a66 in lockmgr (flags=2, lkp=0xffff8037b927c70) at @/sys/lock.h:271 #4 spin_lock_irq (lock=0xffff8037b927c70) at /usr/src/sys/dev/drm/i915/../../../../dev/drm/include/linux/spinlock.h:55 #5 intel_crtc_has_pending_flip (crtc=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:3223 #6 intel_crtc_wait_for_pending_flips (crtc=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:3841 #7 intel_atomic_prepare_commit (nonblock=<optimized out>, state=0xffff8042ca0f340, dev=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:13480 #8 intel_atomic_commit (dev=<optimized out>, state=0xffff8042ca0f340, nonblock=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:13614 #9 0xffffffff83d7b6fc in drm_atomic_helper_set_config (set=0xffff80390f8d678) at /usr/src/sys/dev/drm/drm/./drm_atomic_helper.c:1878 #10 0xffffffff83d54842 in drm_mode_set_config_internal (set=set@entry=0xffff80390f8d678) at /usr/src/sys/dev/drm/drm/./drm_crtc.c:2687 #11 0xffffffff83d54f60 in drm_mode_setcrtc (dev=0xffff8037b927900, data=0xffff80390f8d868, file_priv=<optimized out>) at /usr/src/sys/dev/drm/drm/./drm_crtc.c:2919 #12 0xffffffff83d704f1 in drm_ioctl (ap=<optimized out>) at /usr/src/sys/dev/drm/drm/./drm_ioctl.c:705 #13 0xffffffff80627885 in dev_dioctl (dev=dev@entry=0xffff801581b1c80, cmd=cmd@entry=3228066978, data=data@entry=0xffff80390f8d868 "\360\003\270", fflag=<optimized out>, cred=cred@entry=0xffff8007370f190, msg=msg@entry=0xffff80390f8d960, fp=0xffff8037acb3180) at /usr/src/sys/kern/kern_device.c:244 #14 0xffffffff808f78e5 in devfs_fo_ioctl (fp=0xffff8037acb3180, com=3228066978, data=0xffff80390f8d868 "\360\003\270", ucrd=0xffff8007370f190, msg=0xffff80390f8d960) at /usr/src/sys/vfs/devfs/devfs_vnops.c:1545 #15 0xffffffff80697f6a in fo_ioctl (msg=<optimized out>, cred=<optimized out>, data=<optimized out>, com=<optimized out>, fp=0xffff8037acb3180) at /usr/src/sys/sys/file2.h:84 #16 mapped_ioctl (fd=<optimized out>, com=<optimized out>, uspc_data=<optimized out>, map=0x0, msg=<optimized out>) at /usr/src/sys/kern/sys_generic.c:717 #17 0xffffffff80b96400 in syscall2 (frame=0xffff80390f8d9f8) at /usr/src/sys/platform/pc64/x86_64/trap.c:1308 #18 0xffffffff80b7159d in ?? () at /usr/src/sys/platform/pc64/x86_64/exception.S:450 #19 0x0000000000000009 in ?? () #20 0x00000000c06864a2 in ?? () #21 0x00007ffffdfc8a0 in ?? () #22 0x0000000000000000 in ?? () ---</pre>			

Not sure this is helpful but that's all I've got for now.

I attach the output of 'ps axlRH', maybe there's something else I should look out for.

Peeter

--

History

#1 - 08/07/2019 04:13 AM - peeter

OK this bug has now become my daily companion. Can't find the deadlock.

#2 - 08/07/2019 04:15 AM - peeter

The machine is intel i7 skylake (i7-6700).

#3 - 10/23/2019 05:03 AM - peeter

I seem to have forgotten to specify how to reproduce the situation:

Create a minimalistic ~/.config/weston.ini

```
---  
[core]  
backend=drm-backend.so  
modules=xwayland.so  
---
```

Then start weston with

```
% weston-launch -- --use-pixman >& log/weston.0.log
```

Click on the terminal icon in the left top corner to start a wayland terminal. Then start chrome manually in that window

```
% chrome &
```

Go to youtube.com and start playing videos. No further interaction needed, sit back and wait until freeze occurs.

#4 - 10/24/2019 01:12 PM - dillon

So far ftigeot has not had any luck reproducing it, but there's almost enough information in the bug report to track the issue down. What we have in your original backtrace is that it's stuck trying to acquire a lock, which means that some other thread likely holds the lock. We have to find that other thread and get a backtrace of *it* as well.

One way to do this is to get into kgdb and get the backtrace you have above, then push into the frame where the struct lock is available (such as frame 2 in the above backtrace), then 'print *lkp'.

Once you have the lock contents, if the lk_lockholder field is not NULL you should be able to track down the process/thread that is holding the lock:

```
print lkp->lk_lockholder  
print lkp->lk_lockholder->td_comm  
print lkp->lk_lockholder->td_proc->p_pid
```

If its a user process it will have a td_proc .. but hopefully from that you can use the 'info thread' output from kgdb to find the thread, then switch to it and get a backtrace of THAT thread too.

-Matt

#5 - 10/25/2019 03:24 AM - peeter

- File weston-drmev-bug-2.md added

I attach the backtraces.

1 - the weston thread is trying acquire a lock which is held by "ithread16 2"
2 - in the "ithread16 2" backtrace, the lock has been acquired in

```
do_intel_finish_page_flip: 10919: spin_lock_irqsave(&dev->event_lock, flags);
```

since spin_lock_irqsave() is defined as follows:

```
spin_lock_irqsave(lock, flags) = spin_lock_irq(lock) = lockmgr(lock, LK_EXCLUSIVE)
```

Note: not sure why in the "ithread16 2" backtrace, do_intel_finish_page_flip() appears twice. Since do_intel_finish_page_flip() acquires dev->event_lock ("drmev"), could it mean the lock is not be released? Or something else is happening. . .

Peeter

--

#6 - 10/26/2019 12:28 AM - peeter

OK I haven't printed out the lock info that the "ithread16 2" is trying to acquire in frame 3. Will do this next time. (I believe I printed it out, but it was "lwe", not "drmev" that "ithread16 2" was trying to acquire in frame 3 in the lockmgr(.))

#7 - 10/30/2019 02:39 AM - peeter

I may have a theory about what is going on.

- "weston" is waiting for "drmev" which is owned by "ithread16 2"

- "ithread16 2" is waiting for "lwq" which is owned by "weston"

- It might seem that there is a slight confusion with the latter statement. This is because the lockholder of "lwq" has td_comm = "weston launch", not "weston". But this seems incorrect because its p_pid = 1064, which is "weston". Also, its p_ppid = 1063 "weston launch". This is consistent with the fact that weston-launch creates weston as its child process.

369 pid 1063/1, weston-launch 0xffffffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810

338 pid 1064/1, weston 0xffffffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810

164 kernel ithread16 2 0xffffffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810

(kgdb) thread 338

[Switching to thread 338 (pid 1064/1, weston)]

#0 0xffffffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810

810 lwkt_switch_return(td->td_switch(ntd));

(kgdb) bt

#0 0xffffffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810

#1 0xffffffff80660ece in tsleep (ident=ident@entry=0xffffffff801e0654e70, flags=flags@entry=1024, wmesg=<optimized out>, timo=timo@entry=0) at /usr/src/sys/kern/kern_synch.c:707

#2 0xffffffff8062e27d in lockmgr_exclusive (lkp=lkp@entry=0xffffffff801e0654e70, flags=flags@entry=2) at /usr/src/sys/kern/kern_lock.c:381

#3 0xffffffff837a3b06 in lockmgr (flags=2, lkp=0xffffffff801e0654e70) at @/sys/lock.h:271

#4 spin_lock_irq (lock=0xffffffff801e0654e70) at /usr/src/sys/dev/drm/i915/./././dev/drm/include/linux/spinlock.h:55

#5 intel_crtc_has_pending_flip (crtc=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:3223

#6 intel_crtc_wait_for_pending_flips (crtc=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:3841

#7 intel_atomic_prepare_commit (nonblock=<optimized out>, state=0xffff800cf2ab240, dev=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:13480

#8 intel_atomic_commit (dev=<optimized out>, state=0xffff800cf2ab240, nonblock=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:13614

#9 0xffffffff830567cc in drm_atomic_helper_set_config (set=0xffff801f576f678) at /usr/src/sys/dev/drm/drm/./drm_atomic_helper.c:1875

#10 0xffffffff8302e842 in drm_mode_set_config_internal (set=set@entry=0xffff801f576f678) at /usr/src/sys/dev/drm/drm/./drm_crtc.c:2687

#11 0xffffffff8302ef60 in drm_mode_setcrtc (dev=0xffff801e0654b00, data=0xffff801f576f668, file_priv=<optimized out>) at /usr/src/sys/dev/drm/drm/./drm_crtc.c:2919

#12 0xffffffff8304ab6b in drm_ioctl (ap=<optimized out>) at /usr/src/sys/dev/drm/drm/./drm_ioctl.c:694

#13 0xffffffff80615185 in dev_dioctl (dev=dev@entry=0xffff800d2271dc0, cmd=cmd@entry=3228066978, data=data@entry=0xffff801f576f668 "\360\003\271", fflag=<optimized out>,

cred=cred@entry=0xffff8005c70ef50, msg=msg@entry=0xffff801f576f960, fp=0xffff800d3c6e800) at /usr/src/sys/kern/kern_device.c:244

#14 0xffffffff808e7e25 in devfs_fo_ioctl (fp=0xffff800d3c6e800, com=3228066978, data=0xffff801f576f668 "\360\003\271", ucred=0xffff8005c70ef50, msg=0xffff801f576f960)

at /usr/src/sys/vfs/devfs/devfs_vnops.c:1550

#15 0xffffffff806860ba in fo_ioctl (msg=<optimized out>, cred=<optimized out>, data=<optimized out>, com=<optimized out>, fp=0xffff800d3c6e800) at /usr/src/sys/sys/file2.h:84

#16 mapped_ioctl (fd=<optimized out>, com=<optimized out>, uspc_data=<optimized out>, map=0x0, msg=<optimized out>) at /usr/src/sys/kern/sys_generic.c:717

#17 0xffffffff80b87060 in syscall2 (frame=0xffff801f576f9f8) at /usr/src/sys/platform/pc64/x86_64/trap.c:1308

#18 0xffffffff80b6221d in ?? () at /usr/src/sys/platform/pc64/x86_64/exception.S:450

#19 0x0000000000000009 in ?? ()

#20 0x00000000c06864a2 in ?? ()

#21 0x00007ffffdfcbe0 in ?? ()

#22 0x0000000000000000 in ?? ()

(kgdb) frame 2

#2 0xffffffff8062e27d in lockmgr_exclusive (lkp=lkp@entry=0xffffffff801e0654e70, flags=flags@entry=2) at /usr/src/sys/kern/kern_lock.c:381
381 error = tsleep(lkp, pflags | PINTERLOCKED,

(kgdb) p *lkp

\$36 = {lk_flags = 64, lk_timo = 0, lk_count = 134217729, lk_wmesg = 0xffffffff8306d80b "drmev", lk_lockholder = 0xffffffff801e0662e00}

(kgdb) p *lkp->lk_lockholder

\$37 = {td_threadq = {tqe_next = 0x0, tqe_prev = 0xffff800cad40038}, td_allq = {tqe_next = 0xffff800caec6f00, tqe_prev = 0xffff801e0662790},
td_sleepq = {tqe_next = 0x0,
tqe_prev = 0xffff800cad58430}, td_msgport = {mp_msgq = {tqh_first = 0x0, tqh_last = 0xffff801e0662e30}, mp_msgq_prio = {tqh_first = 0x0, tqh_last
= 0xffff801e0662e40},
mp_flags = 0, mp_cpuid = -1, mp_u = {spin = {counta = 0, countb = 0}, serialize = 0x0, data = 0x0}, mpu_td = 0xffff801e0662e00,
mp_getport = 0xffff8065b9d0 <lwkt_thread_getport>, mp_putport = 0xffff8065bdf0 <lwkt_thread_putport>, mp_waitmsg = 0xffff8065c930
<lwkt_thread_waitmsg>,
mp_waitport = 0xffff8065bf30 <lwkt_thread_waitport>, mp_replyport = 0xffff8065afe0 <lwkt_thread_replyport>, mp_dropmsg = 0xffff8065c310
<lwkt_thread_dropmsg>,
mp_putport_oncpu = 0xffff8065bdf0 <lwkt_thread_putport>}, td_lwp = 0x0, td_proc = 0x0, td_pcb = 0xffff801f2876ac0, td_gd = 0xffff800cad40000,
td_wmesg = 0xffff8380de62 "lwq", td_wchan = 0xffff801e2661f90, td_pri = 28, td_critcount = 4, td_flags = 197760, td_wdomain = 0,
td_preemptable = 0xffff806535e0 <lwkt_preempt>, td_release = 0x0, td_kstack = 0xffff801f2873000 <Address 0xffff801f2873000 out of bounds>,
td_kstack_size = 16384,
td_sp = 0xffff801f28766d8 "240\313\267\200\377\377\377\377\002\002", td_switch = 0xffff80b7cb40, td_uticks = 0, td_sticks = 0, td_iticks =
448523, td_locks = 0,
td_limit = 0x0, td_refs = 0, td_nest_count = 0, td_contended = 0, td_mpflags = 0, td_cscount = 0, td_wakefromcpu = 0, td_upri = 0, td_type = 0,
td_tracker = 0, td_fdcache_lru = 0,
td_unused03 = {0, 0, 0}, td_iosdata = {iorbytes = 0, iowbytes = 0, lastticks = 0}, td_start = {tv_sec = 0, tv_usec = 0}, td_comm = "ithread16
2\000\000\000\000\000",
td_preempted = 0x0, td_ucred = 0x0, td_vmm = 0x0, td_toks_have = 0x0, td_toks_stop = 0xffff801e0662fd8, td_toks_array = {{tr_tok =
0xffff815b0bc0, tr_count = 3,
tr_owner = 0xffff801e0662e00}, {tr_tok = 0xffff815f10c0, tr_count = 3, tr_owner = 0xffff801e0662e00}, {tr_tok = 0x0, tr_count = 0, tr_owner = 0x0}
<repeats 30 times>},
td_fairq_load = 0, td_fairq_count = 0, td_migrate_gd = 0x0, td_fdcache = {{fd = 0, locked = 0, fp = 0x0, lru = 0, unused = {0, 0, 0}}, {fd = 0, locked = 0,
fp = 0x0, lru = 0,
unused = {0, 0, 0}}, {fd = 0, locked = 0, fp = 0x0, lru = 0, unused = {0, 0, 0}}, {fd = 0, locked = 0, fp = 0x0, lru = 0, unused = {0, 0, 0}}, td_linux_task =
0x0, td_mach = {
mtd_cpl = 0, mtd_savefpu = 0xffff801f2876bc0, mtd_savetls = {info = {{base = 0x0, size = 0}, {base = 0x0, size = 0}}}}}
(kgdb) print lkp->lk_lockholder->td_comm
\$38 = "ithread16 2\000\000\000\000\000"

(kgdb) thread 164

[Switching to thread 164 (kernel ithread16 2)]

#0 0xffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810
810 lwkt_switch_return(td->td_switch(ntd));

(kgdb) bt

#0 0xffff80653d36 in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:810
#1 0xffff806611ee in tsleep (ident=ident@entry=0xffff801e2661f90, flags=flags@entry=1024, wmesg=<optimized out>, timo=timo@entry=0) at
/usr/src/sys/kern/kern_synch.c:720
#2 0xffff8062e27d in lockmgr_exclusive (lkp=lkp@entry=0xffff801e2661f90, flags=flags@entry=2) at /usr/src/sys/kern/kern_lock.c:381
#3 0xffff83799537 in lockmgr (flags=2, lkp=0xffff801e2661f90) at @/sys/lock.h:271
#4 wake_up_all (q=0xffff801e2661f90) at /usr/src/sys/dev/drm/i915/./././dev/drm/include/linux/wait.h:76
#5 page_flip_completed (intel_crtc=intel_crtc@entry=0xffff801e0daeb00) at /usr/src/sys/dev/drm/i915/intel_display.c:3826
#6 0xffff8379a2f4 in do_intel_finish_page_flip (dev=0xffff801e0654b00, crtc=0xffff801e0daeb00) at
/usr/src/sys/dev/drm/i915/intel_display.c:10930
#7 0xffff837ad1e9 in do_intel_finish_page_flip (crtc=<optimized out>, dev=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:10948
#8 intel_finish_page_flip_plane (dev=<optimized out>, plane=<optimized out>) at /usr/src/sys/dev/drm/i915/intel_display.c:10948
#9 0xffff83733370 in ilk_display_irq_handler (de_jir=67108992, dev=0xffff801e0654b00) at /usr/src/sys/dev/drm/i915/i915_irq.c:2188
#10 ironlake_irq_handler (irq=<optimized out>, arg=0xffff801e0654b00) at /usr/src/sys/dev/drm/i915/i915_irq.c:2306
#11 0xffff8065d623 in lwkt_serialize_handler_call (s=0xffff801f2895f78, func=0xffff8305faa0 <linux_irq_handler>, arg=0xffff801f2895f40,
frame=frame@entry=0x0)
at /usr/src/sys/kern/lwkt_serialize.c:175
#12 0xffff80609b72 in ithread_handler (arg=<optimized out>) at /usr/src/sys/kern/kern_intr.c:900
#13 0xffff80654390 in _lwkt_dequeue (td=<error reading variable: Cannot access memory at address 0x8>) at /usr/src/sys/kern/lwkt_thread.c:160
#14 lwkt_deschedule_self (td=<optimized out>) at /usr/src/sys/kern/lwkt_thread.c:328
(kgdb) frame 2
#2 0xffff8062e27d in lockmgr_exclusive (lkp=lkp@entry=0xffff801e2661f90, flags=flags@entry=2) at /usr/src/sys/kern/kern_lock.c:381
381 error = tsleep(lkp, pflags | PINTERLOCKED,

(kgdb) p *lkp

\$39 = {lk_flags = 64, lk_timo = 0, lk_count = 134217729, lk_wmesg = 0xffff8380de62 "lwq", lk_lockholder = 0xffff801f51e3480}

(kgdb) p *lkp->lk_lockholder

\$40 = {td_threadq = {tqe_next = 0x0, tqe_prev = 0xffff81c4b038}, td_allq = {tqe_next = 0xffff800d24c6200, tqe_prev = 0xffff800d24c3490},
td_sleepq = {tqe_next = 0x0,
tqe_prev = 0xffff800d256b020}, td_msgport = {mp_msgq = {tqh_first = 0x0, tqh_last = 0xffff801f51e34b0}, mp_msgq_prio = {tqh_first = 0x0, tqh_last

```
= 0xffff801f51e34c0},
mp_flags = 0, mp_cpuid = -1, mp_u = {spin = {counta = 0, countb = 0}, serialize = 0x0, data = 0x0}, mpu_td = 0xffff801f51e3480,
mp_getport = 0xffff8065b9d0 <lwkt_thread_getport>, mp_putport = 0xffff8065bdf0 <lwkt_thread_putport>, mp_waitmsg = 0xffff8065c930
<lwkt_thread_waitmsg>,
mp_waitport = 0xffff8065bf30 <lwkt_thread_waitport>, mp_replyport = 0xffff8065afe0 <lwkt_thread_replyport>, mp_dropmsg = 0xffff8065c310
<lwkt_thread_dropmsg>,
mp_putport_oncpu = 0xffff8065bdf0 <lwkt_thread_putport>}, td_lwp = 0xffff800cf2a14c0, td_proc = 0xffff800d1d5fa80, td_pcb =
0xffff801f576fac0, td_gd = 0xffff81c4b000,
td_wmesg = 0xffff8306d80b "drmev", td_wchan = 0xffff801e0654e70, td_pri = 10, td_critcount = 3, td_flags = 8521344, td_wdomain = 0,
td_preemptable = 0x0, td_release = 0x0,
td_kstack = 0xffff801f576c000 <Address 0xffff801f576c000 out of bounds>, td_kstack_size = 16384, td_sp = 0xffff801f576f2c0
"\200\310\267\200\377\377\377\377F\002",
td_switch = 0xffff80b7c720, td_uticks = 118410471, td_sticks = 2470528, td_iticks = 0, td_locks = 0, td_limit = 0xffff801f573dbc0, td_refs = 0,
td_nest_count = 0,
td_contended = 0, td_mpflags = 16, td_cscount = 0, td_wakefromcpu = 0, td_upri = -180, td_type = 0, td_tracker = -147, td_fdcache_lru = 192499,
td_unused03 = {0, 0, 0},
td_iosdata = {iorbytes = 0, iowbytes = 0, lastticks = 0}, td_start = {tv_sec = 0, tv_usec = 0}, td_comm = "weston\000launch\000\000\000",
td_preempted = 0x0,
td_ucred = 0xffff8005c70ef50, td_vmm = 0x0, td_toks_have = 0x0, td_toks_stop = 0xffff801f51e3670, td_toks_array = {{tr_tok = 0xffff800cf2a16e0,
tr_count = 3,
tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff800cf2a16e0, tr_count = 3, tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff815646c0, tr_count = 3,
tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff800d2472ac0, tr_count = 3, tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff800d1d5ff10, tr_count = 3,
tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff81ab16d0, tr_count = 3, tr_owner = 0xffff801f51e3480}, {tr_tok = 0xffff8152fcc0, tr_count = 3,
tr_owner = 0xffff801f51e3480}, {tr_tok = 0x0, tr_count = 0, tr_owner = 0x0} <repeats 25 times>}, td_fairq_load = 0, td_fairq_count = 0, td_migrate_gd
= 0x0, td_fdcache = {{
fd = 32, locked = 0, fp = 0xffff800cf22d800, lru = 192496, unused = {0, 0, 0}}, {fd = 4, locked = 0, fp = 0xffff800d3c6e880, lru = 192498, unused = {0,
0, 0}}, {fd = 14,
locked = 0, fp = 0xffff800d3c6e800, lru = 192486, unused = {0, 0, 0}}, {fd = 9, locked = 2, fp = 0xffff800d3c6e800, lru = 192499, unused = {0, 0, 0}}},
td_linux_task = 0xffff8005c5eb2d0, td_mach = {mtd_cpl = 0, mtd_savefpu = 0xffff801f576fbc0, mtd_savetls = {info = {{base = 0x800741b40, size =
-1}, {base = 0x0, size = 0}}}}}
```

```
(kgdb) p lkp->lk_lockholder->td_comm
$41 = "weston\000launch\000\000\000"
```

```
(kgdb) p lkp->lk_lockholder->td_proc->p_pid
$42 = 1064
```

```
(kgdb) p lkp->lk_lockholder->td_proc->p_ppid
$43 = 1063
```

#8 - 11/03/2019 12:52 PM - ftigeot

- Status changed from New to In Progress

"lwg" locks are wait queue locks used by Linux wait_event() and similar macros.

DragonFly's existing implementations of these macros was keeping the lock during the whole sleep cycle instead of releasing it before sleeping like Linux.

This has been changed in master -- commit 85bc18adc3f2e6c133e21d5ba7b8e44bbafd9fe9 .

Hopefully that commit should prevent this deadlock situation from happening again.

#9 - 11/07/2019 12:00 AM - peeter

OK the issue with the name of the thread is a non-issue: the name is a null terminated string, so "weston\000launch\000" = "weston", and the trailing "launch" shows it's a child of weston-launch, exactly as it should be.

Peeter

--

#10 - 11/07/2019 12:02 AM - peeter

- Status changed from In Progress to Resolved

Closed.

Files

File Name	Size	Date	Author
drmev-psalRH.out	159 KB	08/05/2019	peeter
weston-drmev-bug-2.md	54.5 KB	10/25/2019	peeter