DragonFlyBSD - Bug #2931

'gdb' of 'vkernel' unable to print backtrace

07/26/2016 01:33 PM - tofergus

Status:	New	**Start date:**	07/26/2016
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Documentation	**Estimated time:**	0.00 hour
Target version:			

Description

Whilst attempting to look at issue #2390 I came across the 'vkernel' debugging page in the wiki

https://www.dragonflybsd.org/docs/howtos/HowToDebugVKernels/

this noted a failure in the current implementation, which caused lockup (issue #1301). However my STABLE build

[...] 4.4-RELEASE DragonFly v4.4.3.9.ge5cb2-RELEASE #0: Fri Jul 15 17:02:58 UTC 2016 [...]/usr/obj/usr/src/sys/VKERNEL64 x86_64

attaches correctly

$ sudo gdb /var/vkernel/boot/kernel/kernel 8418
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-dragonfly".
For bug reporting instructions, please see:
<http://bugs.dragonflybsd.org/>...
Reading symbols from /var/vkernel/boot/kernel/kernel...done.
Attaching to program: /var/vkernel/boot/kernel/kernel, process 8418
Reading symbols from /lib/libc.so.8...(no debugging symbols found)...done.
Loaded symbols for /lib/libc.so.8
Reading symbols from /libexec/ld-elf.so.2...(no debugging symbols found)...done.
Loaded symbols for /libexec/ld-elf.so.2
0x00000000100a3750 in extpread () from /lib/libc.so.8

but then causes an exception whilst trying to print a backtrace

(gdb) bt
#0 0x00000000100a3750 in extpread () from /lib/libc.so.8
#1 0x00000000101374ab in pread () from /lib/libc.so.8
#2 0x00000000006b9614 in vconsgetc (private=<optimized out>)
at /usr/src/sys/platform/vkernel64/platform/console.c:384
#3 0x00000000005040a6 in cngetc () at /usr/src/sys/kern/tty_cons.c:512
#4 0x0000000000473e6a in db_readline (
lstart=lstart@entry=0xa7f480 <db_line> "", lsize=lsize@entry=120)
at /usr/src/sys/ddb/db_input.c:313
#5 0x00000000004743d2 in db_read_line () at /usr/src/sys/ddb/db_lex.c:55
#6 0x0000000000472ed9 in db_command_loop ()
at /usr/src/sys/ddb/db_command.c:465
#7 0x0000000000475cff in db_trap (type=type@entry=3, code=code@entry=0)
at /usr/src/sys/ddb/db_trap.c:71
#8 0x00000000006ab64e in kdb_trap (type=type@entry=3, code=code@entry=0,
regs=regs@entry=0x802a866a68)
at /usr/src/sys/platform/vkernel64/x86_64/db_interface.c:173
#9 0x00000000006add1c in kern_trap (frame=0x802a866a68)
at /usr/src/sys/platform/vkernel64/x86_64/trap.c:769
#10 0x00000000006aef32 in exc_segfault (signo=<optimized out>,
info=<optimized out>, ctxp=<optimized out>)

at /usr/src/sys/platform/vkernel64/x86_64/exception.c:209
#11 <signal handler called>
#12 0x000000802a8670a0 in ?? ()
Cannot access memory at address 0x1

This causes 'ddb' to exit and the process to halt. Presume this is due to the SIGSTOP that halts the 'db>' prompt but unable to decipher how this might be resolved. '~/.gdbinit' contains the

handle SIGSEGV noprint
handle SIGUSR1 noprint

suggested in the article. Adding SIGSTOP has no effect.

Additionally, connecting to a running 'vkernel' with 'gdb' appears to have a similar effect; the console is disconnected (although the kernel appears to run).

Happy to document if this is purely an information gap on my part.

History

#1 - 07/26/2016 01:51 PM - tofergus

- Category changed from Kernel to Documentation

This appears to be largely a question of understanding. When, instead, starting the 'vkernel' and attaching 'gdb' to the process (and then continuing due from 'signal' handlers), not only does the kernel appear to work normally but the 'panic' is captured, as expected in the 'gdb' session.

NB: the console session initiated for the 'vkernel' disconnects during 'gdb' attachment.