

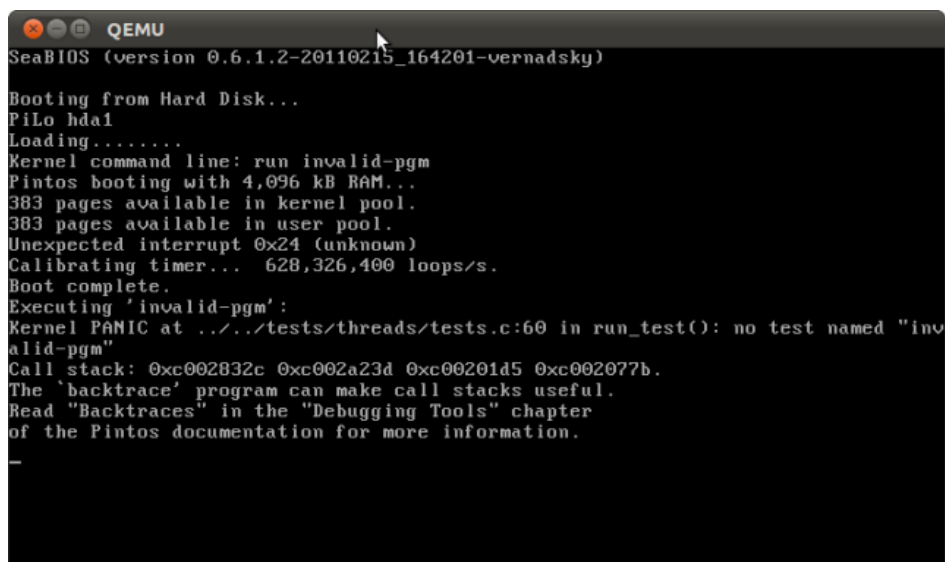
### Pintos : Using backtrace utility for debugging

Posted on [September 24, 2012](#)

While adding new functionality to Pintos kernel, you may face a kernel panic. Kernel panic means kernel is not able to proceed with normal execution. When ever you face such a problem, it is useful to know the exact function that caused the panic to diagnose the problem.

When ever kernel panics, it prints a call stack which is a sequence of hexadecimal numbers. These are the memory location of the functions called. To know the function names, a utility called “backtrace” is provided. Now, we will see how to use it.

```
$ pintos run invalid-pgm
```



```
QEMU
SeaBIOS (version 0.6.1.2-20110215_164201-vernadsky)

Booting from Hard Disk...
PiLo hda1
Loading.....
Kernel command line: run invalid-pgm
Pintos booting with 4,096 kB RAM...
383 pages available in kernel pool.
383 pages available in user pool.
Unexpected interrupt 0x24 (unknown)
Calibrating timer... 628,326,400 loops/s.
Boot complete.
Executing 'invalid-pgm':
Kernel PANIC at ../../tests/threads/tests.c:60 in run_test(): no test named "invalid-pgm"
Call stack: 0xc002832c 0xc002a23d 0xc00201d5 0xc002077b.
The 'backtrace' program can make call stacks useful.
Read "Backtraces" in the "Debugging Tools" chapter
of the Pintos documentation for more information.
```

Note the lines in the image:

```
Kernel PANIC at ../../tests/threads/tests.c:60 in run_test(): no test named "invalid-pgm"
Call stack: 0xc002832c 0xc002a23d 0xc00201d5 0xc002077b.
```

Copy the Call stack and use it as follows:

```
$ cd $HOME/os-pg/pintos/src/utils
$ backtrace ../threads/build/kernel.o 0xc002832c 0xc002a23d 0xc00201d5 0xc002077b

0xc002832c: debug_panic (../../lib/kernel/debug.c:38)
0xc002a23d: fail (../../tests/threads/tests.c:84)
0xc00201d5: run_task (../../threads/init.c:292)
0xc002077b: run_actions (../../threads/init.c:341)
```

The second line refers to `fail()`, the function that handles failure in `tests.c` which is in turn called by `run_task()` that was not able to find “invalid-pgm” to run.

So, this is how you can track down the source of kernel panic and easily resolve the issue knowing its exact source and call trace. Hope it helps you....

Happy Coding!!

– Rasesh

---

Share this:

Tweet

Share 0

Like

Be the first to like this.



#### About Rasesh Mori

I am Rasesh Mori, Software Development Engineer at Amazon since July 2013 after completing masters in Computer Science and Engineering from IIIT, Hyderabad.

[View all posts by Rasesh Mori →](#)

This entry was posted in [Pintos](#) and tagged [assignment](#), [backtrace](#), [call stack](#), [debug](#), [failure](#), [kernel](#), [kernel fail](#), [kernel panic](#), [kernel panics](#), [Operating System](#), [os](#), [panic](#), [Pintos](#). Bookmark the [permalink](#).

## 8 Responses to *Pintos : Using backtrace utility for debugging*



**akshay** says:

February 5, 2013 at 10:21 am

sir plz can u tell me how to create a file in pintos ...using create() system call present in syscall.h

[Reply](#)



---

**Rasesh Mori** says:

February 5, 2013 at 12:03 pm

@Akshay: You cannot create a file directly in Pintos i.e. system calls for that are not implemented and you are expected to implement it. I think below links will help you:-

<http://stackoverflow.com/questions/12147956/system-call-implementation-in-pintos>

[http://www.stanford.edu/class/cs140/projects/pintos/pintos\\_3.html#SEC32](http://www.stanford.edu/class/cs140/projects/pintos/pintos_3.html#SEC32)

[Reply](#)



**akshay** says:

February 6, 2013 at 9:18 am

sir i m not able to create a file using create(const char\*file ,unsigned filesize) system call and which header file to include ...plz help ...

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

February 6, 2013 at 2:03 pm

I have already replied to that query in the previous comment. Please check it and the links.

[Reply](#)



**KillerPollo** says:

November 11, 2014 at 10:09 pm

help! when I try to do this instructions:  
pintos-mkdisk filesys.dsk -filesys-size=2  
pintos -f -q  
I get an error when I try to format a disc

“Enter to Parse Options:Kernel PANIC at ../../threads/init.c:266 in parse\_options(); unknown option `f’ (use -h for help)  
Call stack: 0xc002813e.”

[Reply](#)



**praneeth** says:

November 24, 2014 at 11:59 am

Its because -f format is not getting recognised by the “FILESYS” . Change the path in pintos-pm from /home/praneeth/.../threads/build/loader.bin to /home/praneeth/.../userprog/build/loader.bin .Also set the same path even for kernel.bin in “pintos” perl script of pintos/src/utls directory.

This should work!!

[Reply](#)



**[vu2seeema](#)** says:

May 24, 2015 at 6:30 pm

Hi gdb is not working, in my pintos, here is my screenshot regarding the same  
<https://drive.google.com/file/d/0B2JLFvh4frRSbE11WkJRUDlSbjA/view?usp=sharing>

[Reply](#)



**[vu2seeema](#)** says:

May 24, 2015 at 6:31 pm

Just wanted to add run alarm-multiple seems to be working fine

[Reply](#)