Stress Testing The FreeBSD Kernel
Stress Testing The FreeBSD Kernel

Peter Holm
<pho@FreeBSD.org>

The FreeBSD Project
What Is Stress Testing?

● Deliberately trying to break the kernel
  – By stressing individual system calls
  – By ensuring resources are low
  – By supplying random input data
  – By applying random changes to valid input
  – By applying as much random values to
    ● Number of invocations
    ● Startup delay
    ● Test program mix
The Origin of This Interest

• An incident 15 years ago
  – Crash dump analysis
• Lessons learned
  – More than 80% of users do not report errors
  – On site debugging is time consuming
  – Kernel stress testing is a must
  – So is real life testing
The Key Concept of a Test

- Test a specific area, for example a system call
- Make sure it gets called a lot
- Have random incarnations running
- Introduce a random startup delay
- Add resource hogging programs
The First Test Program

- Inspired by on site debugging
- Bug in a user mode program
- 1st test program with 50 lines of code
The First FreeBSD Test Suite

- Shell scripts and a few small programs
- Not a pretty sight
- Change of strategy
  - Moved from analyzing to reporting due to the number of different panics
- Number of reports
  - 2003  42
  - 2004  102
  - 2005  162
The Bug Reports

Stress Test Log

All tests are performed on source checked out with the HEAD tag. The checkout ended at the UTC timestamp. The kernel configuration is GENERIC with "BREAK_TO_DEBUGGER" added. The test are run on this setup and the source code has been argumented with the following changes.

<table>
<thead>
<tr>
<th>Date</th>
<th>CVS Date</th>
<th>Problem</th>
<th>Reference(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-01-30</td>
<td>01-28 13:41</td>
<td>Liveloek</td>
<td>console log</td>
</tr>
<tr>
<td>2006-01-27</td>
<td>01-27 19:01</td>
<td>panic: No vop_inactive(0xc534e3f0, 0xe764f74c)</td>
<td>console log</td>
</tr>
<tr>
<td>2006-01-26</td>
<td>01-26 16:14</td>
<td>panic: make_dev() by driver pts on pre-existing device (min=2, name=ttyp2)</td>
<td>console log</td>
</tr>
<tr>
<td>2006-01-22</td>
<td>01-21 18:16</td>
<td>panic: No vop_lock(0xca456a80, 0xe7a3c788)</td>
<td>console log</td>
</tr>
<tr>
<td>2005-12-22</td>
<td>12-21 16:43</td>
<td>panic: mutex nfsd_mtx not owned at ../../nfs/nfs_srvsock.c:145</td>
<td>console log</td>
</tr>
<tr>
<td>2005-12-20</td>
<td>12-19 15:56</td>
<td>Fatal trap 12: page fault while in kernel mode</td>
<td>console log</td>
</tr>
<tr>
<td>2005-12-18</td>
<td>12-18 15:25</td>
<td>panic: m_prepend: MH_ALIGN not PKTHDR mbuf</td>
<td>console log</td>
</tr>
<tr>
<td>2005-11-30</td>
<td>11-28 19:10</td>
<td>panic: witness_warn</td>
<td>console log</td>
</tr>
<tr>
<td>2005-11-04</td>
<td>11-04 09:15</td>
<td>panic: mb_dtor_pack: ref_cnt != 1</td>
<td>console log</td>
</tr>
</tbody>
</table>

Done
A Bug Report

 GENERIC 7.0-CURRENT from Dec 21 16:43 UTC, vmcore.35
 Test with loopback NFS mount. Panic in nfsserver/nfs_srvsock.c:145.

 KOB: debugger backends: ddb
 KOB: current backend: ddb
 Copyright (c) 1992-2005 The FreeBSD Project.
 The Regents of the University of California. All rights reserved.
 philcrashbox.osted.lan:/usr/src/sys/i386/compile/PHO
 WARNING: WITNESS option enabled, expect reduced performance.
 ACPI APIC Table: <A M I OEMAPIC>
 Timecounter "i8254" frequency 1193182 Hz quality 0
 CPU: Intel(R) XEON(TM) CPU 1.80GHz (1790.80-MHz 686-class CPU)
 Origin = "GenuineIntel" Id = 0xf24 Stepping = 4
 Features=0x3febfbf<PUD,VM,DE,PSE,TSC,MSR,PAE, MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CLFLUSH,DTL,ACPI,MMX,
 Logical CPUs per core: 2
 real memory = 1073675288 (1023 MB)
 avail memory = 1040891904 (992 MB)
:
 Trying to mount root from ufs:/dev/ad0s1a
 Tmp0: link state changed to UP
 panic: mutex nfsd_mtx not owned at ../../../nfsserver/nfs_srvsock.c:145
 cpuuid = 0
 KOB: enter: panic
 [thread pid 401 tid 100074 ]
 Stopped at kdb_enter+0x2b: nop
db> where
 Tracing pid 401 tid 100074 td 0xc4e88900
 kdb_enter(c087e434) at kdb_enter+0x2b
 panic:(c087e48f,c080ff1,c081db4,91.0) at panic+0x14b
 mtx_assert(c09ac998.1,c091db4,91.8) at mtx_assert+0x67
 nfs_renhead(0xc5a1b00,48,75x1060,75x4964) at nfs_renhead+0x25
Type of Problems Found

- Panics (my favorite)
- Livelocks
- Deadlocks
- Memory leak
- Other resource leaks
Teamwork

- Jeff Roberson
- Headed the GIANT-lock free VFS work. The VFS layer provides the abstractions in the kernel that describe file objects.
  - From Nov. 2004 to Jun. 2005
  - More than 90 reports
The NG Stress Test Suite

- Inspired by talks with Jeff Roberson
- Design goals
  - Do away with most of the scripts
  - More user friendly
  - Easy to extend
Design

- Each test program must implement three procedures: setup(), cleanup() and test()
- Argument parsing and randomization is handled by the library
- The initial version with only three (two) test programs caused lots of panics
```c
int i, master, slave;
char buf[512], sname[1025];
struct termios tios;

for (i = 0; i < 10 && done_testing == 0; i++) {
    if (openpty(&master, &slave, sname, NULL, NULL) == -1)
        err(1, "openpty");
    if (close(master) == -1)
        err(1, "close(master)");
    if (close(slave) == -1)
        err(1, "close(%s", sname);
}

if (openpty(&master, &slave, sname, NULL, NULL) == -1)
    err(1, "openpty");
if (tcgetattr(slave, &tios) < 0)
    err(1, "tcgetattr(%s", sname);
    cfmakeraw(&tios);
if (tcsetattr(slave, TCSADFLUSH, &tios) < 0)
    err(1, "tcsetattr(%s", sname);

for (i = 0; i < 64 && done_testing == 0; i++) {
    if (write(master, TXT, sizeof(TXT)) == -1)
        err(1, "write");
    if (read(slave, buf, sizeof(TXT)) == -1)
        err(1, "read(%s", sname);
}

close(master);
close(slave);
return (0);
```
Other Tests

- UFS2 snapshots
- Mangling ELF headers
- Mangling file systems
Inspiration for Tests

• Most inspiration comes from the current mailing list

• I get email about problems worth looking into
Tuning

- The test suite must be tuned so it loads the system, but does not overload it.
- The environment can be changed, for example UFS1, UFS2 (with and without snapshots) and NFS.
- The mix of test programs.
Test Setup

- I have two test boxes
  - A Tyan S2730 with two 1.8 MHz Zeon CPUs and 1 Gb RAM
  - An ASUS P5A with a 1.8 Mhz Celeron and 256 Mb RAM
- Both boxes run with a serial console connected to a third box running FreeBSD 6.0
Testing

• Crash dumps and the corresponding source code is saved for follow up debugging

• Once a panic becomes a candidate for the log, a HTML document is generated and the RSS feed is updated
The Real World

- Stress testing is only a nice tool for finding the most obvious errors
- It is no substitute for real world testing
  - During one release the only way to provoke a scheduling problem, was by transferring money between two of my bank accounts
The FreeBSD Package Cluster

- More than 14,000 packages
- Kris Kennaway's work is a fantastic test, due to the dynamic nature of the ports
The 6.0 Release

- The TODO page got a link to the Stress Test Log page
- High focus on bug fixing
- All known bugs got fixed
The todo page

Documentation items that must be resolved for 6.1

<table>
<thead>
<tr>
<th>Issue</th>
<th>Status</th>
<th>Responsible</th>
<th>Description</th>
</tr>
</thead>
</table>

Testing foci for 6.1-RELEASE

<table>
<thead>
<tr>
<th>Issue</th>
<th>Status</th>
<th>Responsible</th>
<th>Description</th>
</tr>
</thead>
</table>

Stress Test Panics

The system is continuously being subjected to Peter Holm's Kernel Stress Test Suite. The following issues have recently been discovered from this test suite:

- panic: No vop_inactive(0xc534e3f0, 0xe764f74c)
- panic: make_dev() by driver pts on pre-existing device (min=2, name=tty2)
- panic: No vop_lock(0xc4f5e80, 0xe7a3788)
- panic: mutex nfsd_mbx not owned at ../../nfserv/srcssock.c:145
- Fatal trap 12: page fault while in kernel mode
- panic: m_prepend: MH_ALIGN not PLTHDR mbuf
- panic: witness warn
- panic: mb_dtor_pack: ref_cnt != 1
- panic: filesystem goof, vop_panic[vop_readlink]
- Fatal trap 12: page fault while in kernel mode
Portability

- It was never the intention to make the test suite portable.
- It has however been run on
  - Dragonfly
  - OS X
  - Linux
  - HP-UX
- However, no feedback
Related work

- A lot of work in this area has been done by other people
- The Linux Test Project
  - http://ltp.sourceforge.net/
Links

- The source
- The log files
- Slides
  - http://people.freebsd.org/~pho/linuxforum06/