Subject: [PATCH] Print number of oopses in Sysrq-P output
Posted by adobriyan on Thu, 18 Jan 2007 16:58:36 GMT
View Forum Message <> Reply to Message

From: Pavel Emelianov <xemul@openvz.org>

Useful in deciding whether said output should be ignored
in absence of other info. :)

Signed-off-by: Pavel Emelianov <xemul@openvz.org>
Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
---

```
 arch/i386/kernel/process.c   |    4 +++-
 arch/i386/kernel/traps.c     |    2 +-
 arch/x86_64/kernel/process.c |    6 ++++--
 arch/x86_64/kernel/traps.c   |    3 ++-
 4 files changed, 10 insertions(+), 5 deletions(-)

--- a/arch/i386/kernel/process.c
+++ b/arch/i386/kernel/process.c
@@ -292,9 +292,11 @@ __setup("idle=", idle_setup);
 void show_regs(struct pt_regs * regs)
 {
  unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L;
+ extern int die_counter;

  printk("\n");
- printk("Pid: %d, comm: %20s\n", current->pid, current->comm);
+ printk("Pid: %d, comm: %20s, oopses: %d\n",
+  current->pid, current->comm, die_counter);
  printk("EIP: %04x:[<%08lx>] CPU: %d\n",0xffff & regs->xcs,regs->eip, smp_processor_id());
  print_symbol("EIP is at %s\n", regs->eip);

--- a/arch/i386/kernel/traps.c
+++ b/arch/i386/kernel/traps.c
@@ -366,6 +366,7 @@ int is_valid_bugaddr(unsigned long eip)
  return ud2 == 0x0b0f;
 }

+int die_counter = 0;
 /*
  * This is gone through when something in the kernel has done something bad and
  * is about to be terminated.
@@ -381,7 +382,6 @@ void die(const char * str, struct pt_reg
   .lock_owner =  -1,
   .lock_owner_depth = 0
  };
```

```
- static int die_counter;
  unsigned long flags;

  oops_enter();
--- a/arch/x86_64/kernel/process.c
+++ b/arch/x86_64/kernel/process.c
@@ -305,14 +305,16 @@ void __show_regs(struct pt_regs * regs)
  unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L, fs, gs, shadowgs;
  unsigned int fsindex,gsindex;
  unsigned int ds,cs,es;
+ extern int die_counter;

  printk("\n");
  print_modules();
- printk("Pid: %d, comm: %.20s %s %s %.*s\n",
+ printk("Pid: %d, comm: %.20s %s %s %.*s oopses: %d\n",
    current->pid, current->comm, print_tainted(),
    init_utsname()->release,
    (int)strcspn(init_utsname()->version, " "),
-   init_utsname()->version);
+   init_utsname()->version,
+   die_counter);
  printk("RIP: %04lx:[<%016lx>] ", regs->cs & 0xffff, regs->rip);
  printk_address(regs->rip);
  printk("RSP: %04lx:%016lx  EFLAGS: %08lx\n", regs->ss, regs->rsp,
--- a/arch/x86_64/kernel/traps.c
+++ b/arch/x86_64/kernel/traps.c
@@ -519,9 +519,10 @@ void __kprobes oops_end(unsigned long fl
  oops_exit();
 }

+int die_counter = 0;
+
 void __kprobes __die(const char * str, struct pt_regs * regs, long err)
 {
- static int die_counter;
  printk(KERN_EMERG "%s: %04lx [%u] ", str, err & 0xffff,++die_counter);
 #ifdef CONFIG_PREEMPT
  printk("PREEMPT ");
```

---

## Subject: Re: [PATCH] Print number of oopses in Sysrq-P output
Posted by Erik Mouw on Thu, 18 Jan 2007 17:46:31 GMT
View Forum Message <> Reply to Message

On Thu, Jan 18, 2007 at 08:05:22PM +0300, Alexey Dobriyan wrote:
> From: Pavel Emelianov <xemul@openvz.org>
>

> Useful in deciding whether said output should be ignored
> in absence of other info. :)
>
> Signed-off-by: Pavel Emelianov <xemul@openvz.org>
> Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
> ---
>
>  arch/i386/kernel/process.c   |    4 +++-
>  arch/i386/kernel/traps.c     |    2 +-
>  arch/x86_64/kernel/process.c |    6 ++++--
>  arch/x86_64/kernel/traps.c   |    3 ++-
>  4 files changed, 10 insertions(+), 5 deletions(-)

What about the other architectures?

> --- a/arch/i386/kernel/traps.c
> +++ b/arch/i386/kernel/traps.c
> @@ -366,6 +366,7 @@ int is_valid_bugaddr(unsigned long eip)
>   return ud2 == 0x0b0f;
>  }
>
> +int die_counter = 0;

Global variables don't have to be initialised at 0. They live in the
.bss segment so they will automatically initialised at 0 and not take
space in the kernel image.

Erik

--
+-- Erik Mouw -- www.harddisk-recovery.com -- +31 70 370 12 90 --
| Lab address: Delftechpark 26, 2628 XH, Delft, The Netherlands

---

Subject: Re: [PATCH] Print number of oopses in Sysrq-P output
Posted by Andi Kleen on Thu, 18 Jan 2007 23:19:56 GMT
View Forum Message <> Reply to Message

On Friday 19 January 2007 04:05, Alexey Dobriyan wrote:

> @@ -292,9 +292,11 @@ __setup("idle=", idle_setup);
>  void show_regs(struct pt_regs * regs)
>  {
>   unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L;
> + extern int die_counter;

externs should always be in some .h file, never in a sub scope.

-Andi

Subject: Re: [PATCH] Print number of oopses in Sysrq-P output
Posted by dev on Fri, 19 Jan 2007 11:25:41 GMT
View Forum Message <> Reply to Message

please fix the comment as well.

oops number is very helpful in dealing with people
reports. Very often the first Oops is required to get
understanding of the real problem, so
further oops can be ignored and the first one requested
if the problem is reproducable.

Kirill

> From: Pavel Emelianov <xemul@openvz.org>
>
> Useful in deciding whether said output should be ignored
> in absence of other info. :)
>
> Signed-off-by: Pavel Emelianov <xemul@openvz.org>
> Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
> ---
>
> arch/i386/kernel/process.c | 4 +++-
> arch/i386/kernel/traps.c | 2 +-
> arch/x86_64/kernel/process.c | 6 ++++--
> arch/x86_64/kernel/traps.c | 3 ++-
> 4 files changed, 10 insertions(+), 5 deletions(-)
>
> --- a/arch/i386/kernel/process.c
> +++ b/arch/i386/kernel/process.c
> @@ -292,9 +292,11 @@ __setup("idle=", idle_setup);
> void show_regs(struct pt_regs * regs)
> {
> unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L;
> + extern int die_counter;
>
> printk("\n");
> - printk("Pid: %d, comm: %20s\n", current->pid, current->comm);
> + printk("Pid: %d, comm: %20s, oopses: %d\n",
> + current->pid, current->comm, die_counter);
> printk("EIP: %04x:[<%08lx>] CPU: %d\n",0xffff & regs->xcs,regs->eip, smp_processor_id());
> print_symbol("EIP is at %s\n", regs->eip);
>

```
> --- a/arch/i386/kernel/traps.c
> +++ b/arch/i386/kernel/traps.c
> @@ -366,6 +366,7 @@ int is_valid_bugaddr(unsigned long eip)
>   return ud2 == 0x0b0f;
>  }
>
> +int die_counter = 0;
>  /*
>   * This is gone through when something in the kernel has done something bad and
>   * is about to be terminated.
> @@ -381,7 +382,6 @@ void die(const char * str, struct pt_reg
>    .lock_owner =  -1,
>    .lock_owner_depth = 0
>   };
> - static int die_counter;
>   unsigned long flags;
>
>   oops_enter();
> --- a/arch/x86_64/kernel/process.c
> +++ b/arch/x86_64/kernel/process.c
> @@ -305,14 +305,16 @@ void __show_regs(struct pt_regs * regs)
>   unsigned long cr0 = 0L, cr2 = 0L, cr3 = 0L, cr4 = 0L, fs, gs, shadowgs;
>   unsigned int fsindex,gsindex;
>   unsigned int ds,cs,es;
> + extern int die_counter;
>
>   printk("\n");
>   print_modules();
> - printk("Pid: %d, comm: %.20s %s %s %.*s\n",
> + printk("Pid: %d, comm: %.20s %s %s %.*s oopses: %d\n",
>    current->pid, current->comm, print_tainted(),
>    init_utsname()->release,
>    (int)strcspn(init_utsname()->version, " "),
> -  init_utsname()->version);
> +  init_utsname()->version,
> +  die_counter);
>   printk("RIP: %04lx:[<%016lx>] ", regs->cs & 0xffff, regs->rip);
>   printk_address(regs->rip);
>   printk("RSP: %04lx:%016lx  EFLAGS: %08lx\n", regs->ss, regs->rsp,
> --- a/arch/x86_64/kernel/traps.c
> +++ b/arch/x86_64/kernel/traps.c
> @@ -519,9 +519,10 @@ void __kprobes oops_end(unsigned long fl
>   oops_exit();
>  }
>
> +int die_counter = 0;
> +
>  void __kprobes __die(const char * str, struct pt_regs * regs, long err)
```

> {
> - static int die_counter;
>  printk(KERN_EMERG "%s: %04lx [%u] ", str, err & 0xffff,++die_counter);
> #ifdef CONFIG_PREEMPT
>  printk("PREEMPT ");
>