

ANNOUNCE: kexec 2.6.8.1-kexec3

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-08/5833.html>

From: Eric W. Biederman (ebiederm_at_xmission.com)

Date: 08/20/04

To: fastboot@osdl.org, linux-kernel@vger.kernel.org, LinuxBIOS linuxbios@clustermatic.org

Date: 20 Aug 2004 01:44:19 -0600

<http://www.xmission.com/~ebiederm/files/kexec/2.6.8.1-kexec3/>

<http://www.xmission.com/~ebiederm/files/kexec/kexec-tools-1.96.tgz>

- Current ports now include i386 ppc and x86_64
Between the three of them I think there are some good examples of what is needed to implement kexec. i386 and ppc can turn off their mmu while x86_64 cannot, yet they all implement identity mapped memory for code running in the processors native mode.
- The code has been reviewed in preparation to sending to Andrew Morton and hopefully mainline kernel inclusion.
- All uses of `init_mm` have been removed from the generic code

- Three new architecture specific hooks have been added.
`int machine_kexec_prepare(struct kimage *image);`
This does whatever architecture specific setup such as allocating page tables that is needed for a given image.

`void machine_kexec_cleanup(struct kimage *image);`
When the image is removed instead of executed this does the necessary cleanup.

`void machine_shutdown(void);`
`device_shutdown` cannot do everything. There are some pieces of hardware that can only be shutdown by architecture specific code. That needed architecture specific shutdown code is generally common between `machine_restart` and `machine_kexec`, and it needs a function to live in. In addition the coming `kexec_on_panic` code path needs to do nothing what is absolutely necessary, making the architecture specific shutdown code inappropriate.

`machine_shutdown` is designed to hold the architecture specific shutdown code.

REBOOT_CMD_KEXEC now calls machine_shutdown just before machine_kexec. Which means the coming kexec_on_panic implementation can skip the unnecessary code by simply not calling machine_shutdown.

- The i386 port now no longer uses init_mm and the implementation actually got simpler, and is much more robust in the face of changing kernel infrastructure. Even making it work with the 4G/4G patch should be simple.
- The x86_64 port has been tested on both Opterons and Xeons and it works fine. Support for the 32bit x86 system call has not been done but I have a design for it if anyone cares.
- The jointly released port of kexec-tools-1.96 adds support for x86_64, fixes a small glitch so it works on even a lowly 386 (tested) and adds support for linux style arguments to the i386 bootloader.

And a quick summary of the broken-out patches:

i8259-shutdown.i386.patch

i8259-sysfs.x86_64.patch

For i386 all I needed to do was add a shutdown the PIC.

For x86_64 sysfs support had not even been added for the legacy PIC, so I ported the appropriate code from x86.

apic-virtwire-on-shutdown.i386.patch

apic-virtwire-on-shutdown.x86_64.patch

The x86 Multiprocessor Specification says the local apic needs to be in either pic_mode (i.e. disabled) or it needs to be in virtual wire mode. This patch restores the local apic to virtual wire mode when appropriate.

ioapic-virtwire-on-shutdown.i386.patch

ioapic-virtwire-on-shutdown.x86_64.patch

The x86 Multiprocessor Specification says access to the legacy pic can either be direct to the cpu or it can be through a io_apic programed in virtual wire mode.

This patch examines the ioapic interrupt routing and if an i8259 is connected to an ioapic in external interrupt mode it places the given ioapic in virtual wire mode instead of disabling it completely.

e820-64bit.x86_64.patch

Someone overzealously copied the resource reservation code from x86 and was filter out 64bit io and memory resources. Ouch! x86_64 needs this if it is going to allocation 64bit resources properly and I need the 64bit memory resources if I want to see all of the memory through /proc/iomem.

kexec-generic.patch

This patch simply holds the generic part of kexec.

machine_shutdown.x86_64.patch

kexec.x86_64.patch

The x86_64 port, I simply factor out machine_shutdown from machine_restart before I complete the port.

machine_shutdown.i386.patch

kexec.i386.patch

The i386 port. While factoring out machine_shutdown I make rebooting on the bootstrap cpu the default as required by the Multiprocessor Specification and expected by linux. With a number of motherboard specific fixups become unnecessary.

use_mm.patch

kexec.ppc.patch

homebrew-dol-support.ppc.patch [EXPERIMENTAL]

This is the ppc port. It still uses init_mm. So I first make use_mm no-static. The port is not as mature as the x86 port but it should still work in the majority of cases. I don't think the homebrew Dolphin OS support is clean enough to be in the stable kernel but it is included in case people need it and as a starting point for something better.

vmlinux-lds.i386.patch [EXPERIMENTAL]

highbzImage.i386.patch [EXPERIMENTAL]

In the discussions on how to implement kexec_on_panic a key question has been can we build a kernel that executes in memory the kernel that called panic had reserved.

These two patches are my proof of concept that we can make an x86 kernel that will execute when loaded at a different memory address. The first patch fixes up vmlinux.lds.S so it vmlinux exports the appropriate physical addresses in it's ELF program header and adds an option what memory location you want to boot from. This is needed for the kexec_on_panic case if we want the panic kernel to load somewhere else.

The patch makes the bzImage of an i386 kernel built with a load_address other than bootable. This means you don't have to use kexec to boot one of these kernels.

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>