Kdump

A Kexec Based Kernel Crash Dumping Mechanism

Vivek Goyal (vgoyal@in.ibm.com)
Maneesh Soni (maneesh@in.ibm.com)
IBM Linux Technlogy Center

Introduction

- What is Crash Dumping

A mechanism to take system memory snapshot. This memory snapshot is called crash dump and can be used for post crash analysis.

Background

- Dump capture from crashing kernel's context
  - ✓ Resource lockup
  - ✓ Corrupt data structures
- Dedicated dump drivers
  - ✓ Limited number of target devices
  - ✓ Maintenance was a big issue
  - ✓ Dependency on crashing kernel reduced and not eliminated completely

Background contd...

- Stand alone dumpers
  - ✓ Need to maintain low level hardware specific code
  - ✓ Filtering is not possible
- Kernel reboot based dumper
  - ✓ Memory constraint might prevent capturing full dump
  - ✓ Significant amount of code being run in crashing kernel context
  - ✓ Core kernel invasive code

Design Goals of The New Solution

- Simple and minimally invasive into the kernel code
- Highly Reliable
- Available on most architectures
- Easy to Maintain
- Flexibility in terms of dump contents and targets
  - ✓ Full dump or kernel-pages only dump
  - ✓ Dump to disk or across the network
- Ease of Use

Kdump – Overview

- A new kernel, often called capture kernel, is booted after the crash
- Previous kernel's memory is preserved
- Dump is captured from the context of capture kernel
- Kernel-to-kernel boot loader enables booting a new kernel after a crash
- Kexec is underlying kernel to kernel boot-loader

Kexec On Panic

- An extension of Kexec functionality
- Enables booting a new kernel after system crash
- Devices are not shutdown
- New kernel runs from a reserved memory location
    - ✓ Protection against on-going DMA at the time of crash

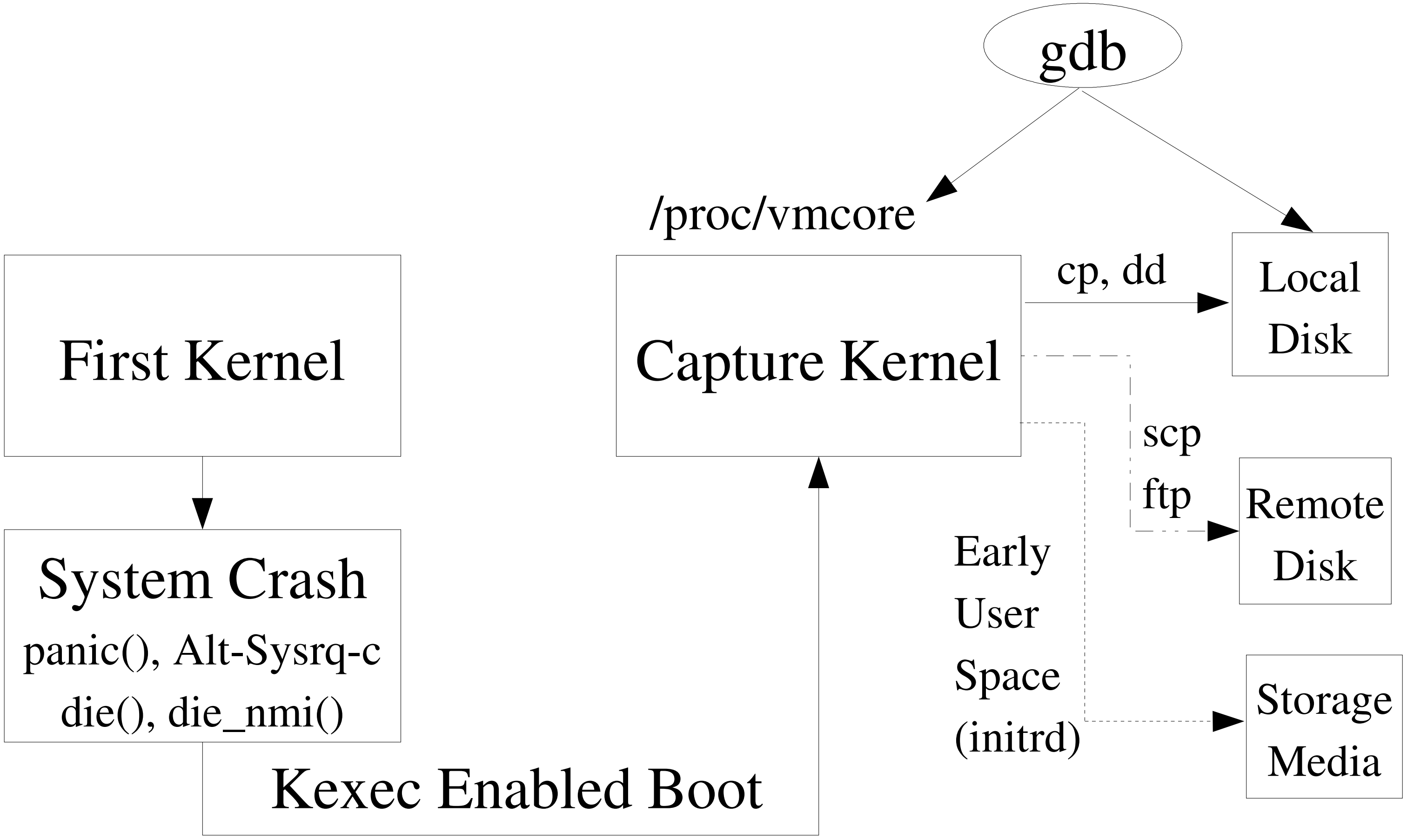
Kexec On Panic Contd..

- Loading capture kernel

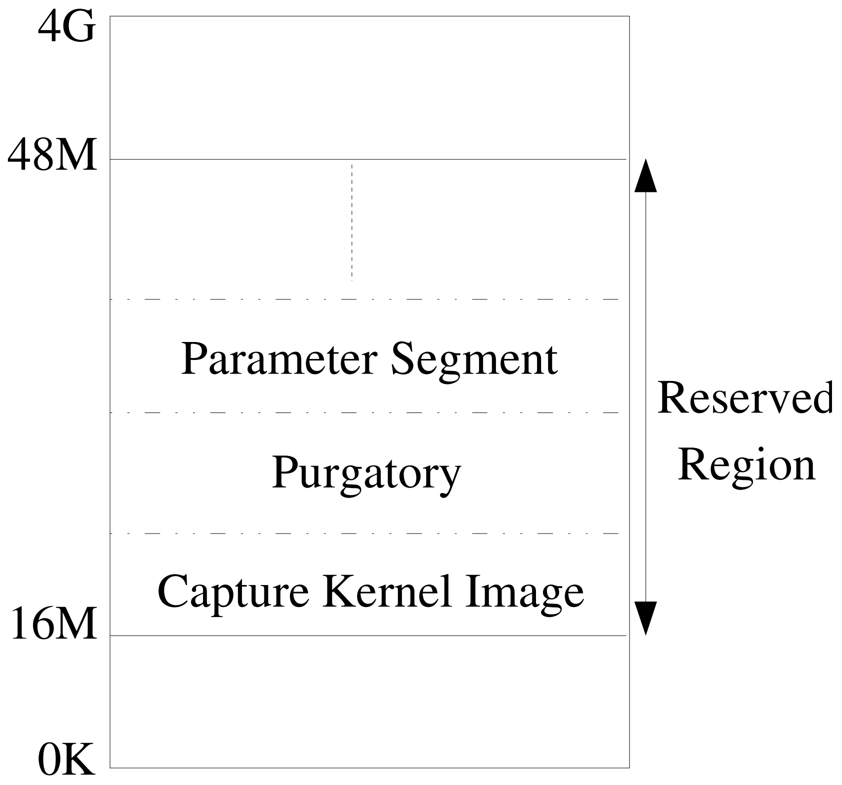
  kexec -p <kernel-image> --append=<options>

- Execution of capture kernel
  - ✓ *panic()*
  - ✓ *Alt-Sysrq-c*
  - ✓ *die()*
  - ✓ *die_nmi()*

Kdump Overview

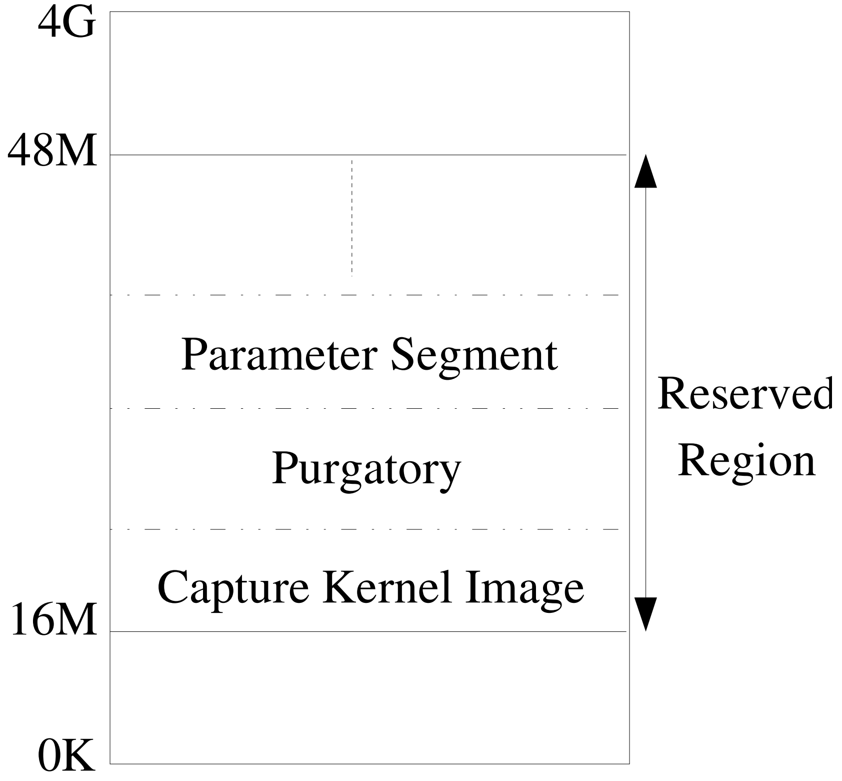


Kexec on Panic - Pre-loading



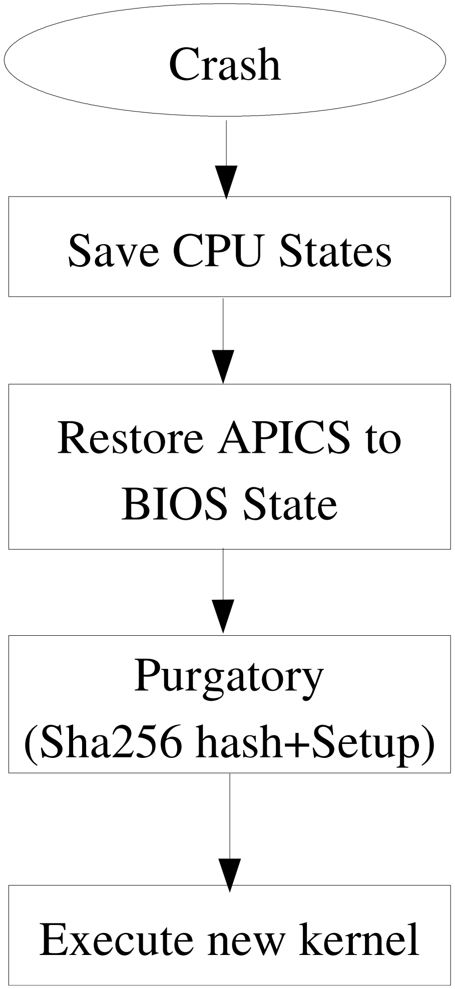
- Reserve memory for capture kernel (crashkernel=X@Y)
- Pre-load the capture kernel
- Capture kernel runs from reserved memory location

Kexec on Panic - Purgatory



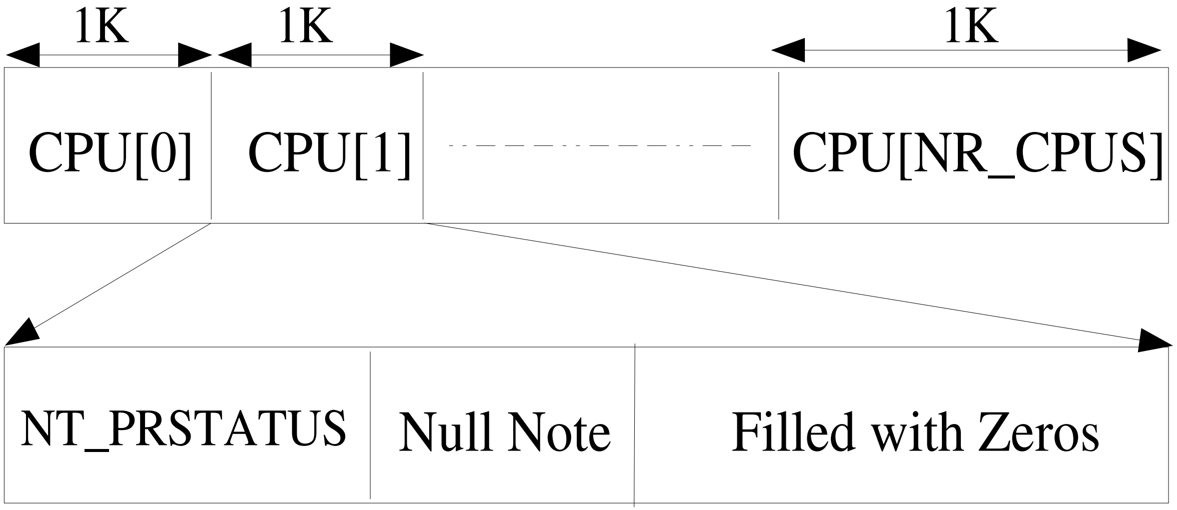
- Purgatory is an ELF relocatable object and it contains setup code and sha256 hash
- Sha256 hash ensures integrity of the new kernel's pre-loaded data

Kexec on Panic – Post Crash (x86)



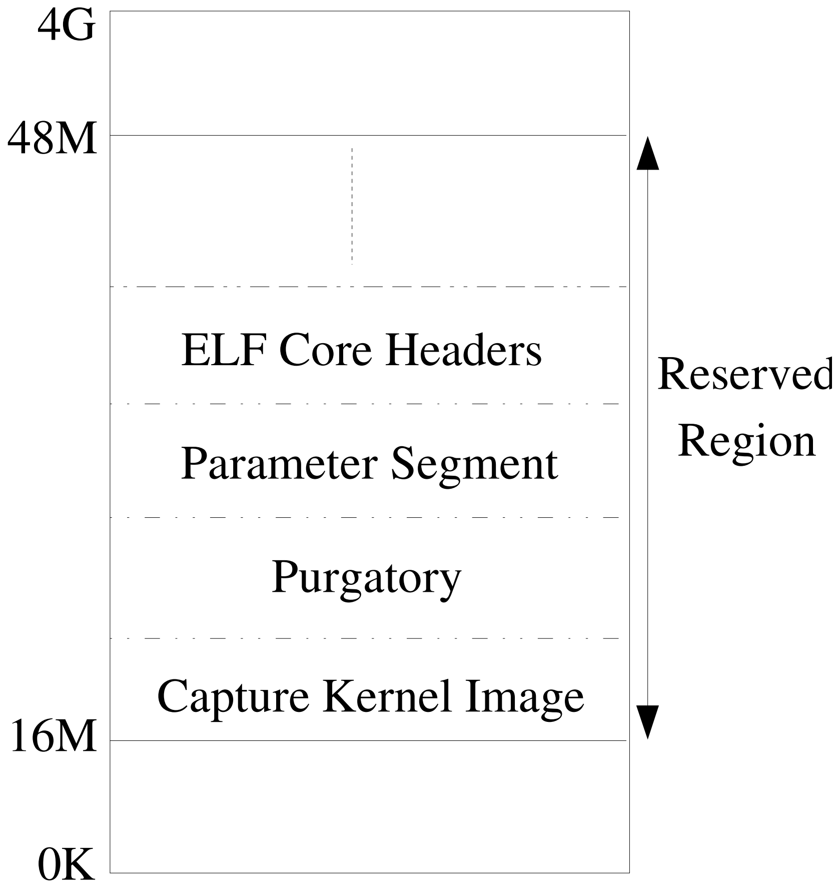
- CPU states are saved and other CPUs are halted using NMI
- LAPIC/IOAPIC are disabled and put back into PIC or virtual wire mode
- Purgatory is run and control is transferred to new kernel

Kexec on Panic – Saving CPU States



- CPU register states are saved in ELF note format
- 1K of memory is reserved statically per CPU

Kdump – ELF Header Generation

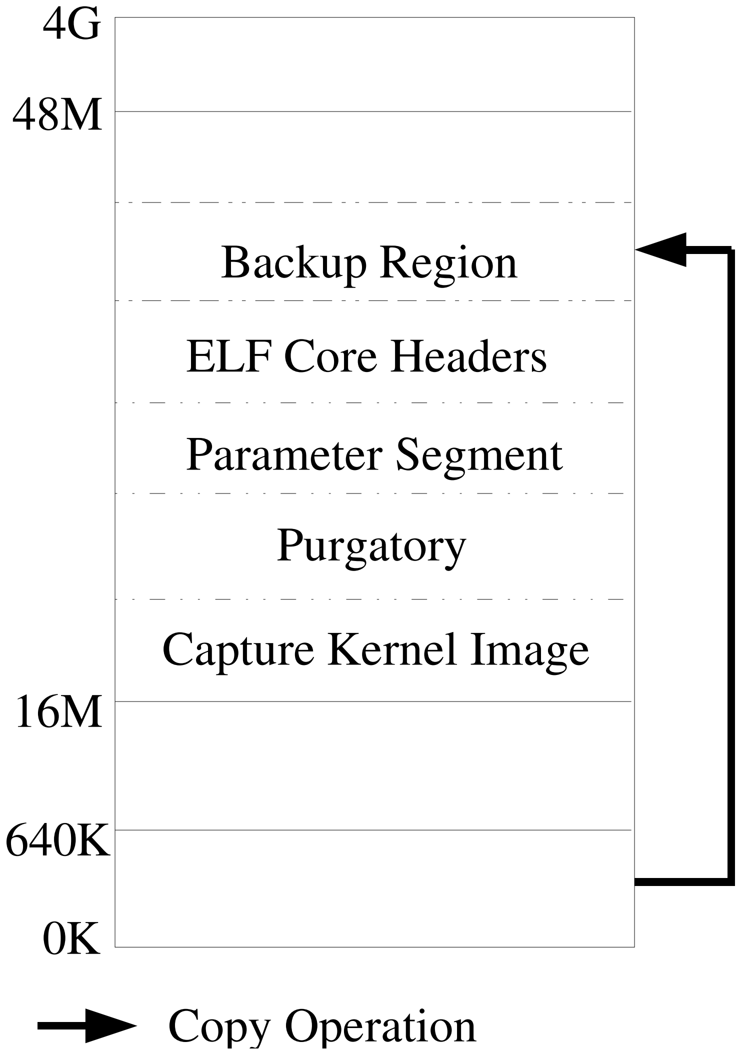


- Dump information across the kernel is exchanged in an ELF format Core file
- Kexec-tools prepare ELF headers and pre-load them in the reserved region

Kdump – ELF Header Generation

- One PT_LOAD type ELF program header is created for every contiguous memory chunk
- Kexec-tools use /proc/iomem to retrieve System RAM information on i386® platform
- Address of the start of ELF header is passed to the capture kernel using command line option "elfcorehdr="

Kdump – Backup Region



- Kernel uses some fixed memory locations to boot
- First 640K of memory is required for booting SMP capture kernel on i386
- Contents of first 640K of memory are backed up in Backup Region

Kdump – Backup Region

- Kexec-tools reserve the memory for backup region while loading capture kernel.
- Purgatory contains the code for backing up first 640K of memory after crash.
- Other architectures can define their own backup region (If need be).

Kdump – Booting into Capture Kernel

- Capture kernel uses limited amount of memory to boot
- Command line option “memmap=exactmap” is used to limit the memory regions capture kernel uses
- Kexec tools append memmap= command line options automatically

Kdump – Capturing the Dump

- Accessing dump image in ELF Core format
  - ✓ /proc/vmcore
- Accessing dump image in linear raw format
  - ✓ /dev/oldmem

Kdump – ELF Format Core File (/proc/vmcore)

ELF Header	Program Header PT_NOTE	Program Header PT_LOAD		Per CPU Register States	Dump Image

- ELF32/ELF64 format headers
- Physical addresses are filled for all the regions
- Virtual addresses are filled only for linearly mapped memory region

Kdump – Analysis Tools

- gdb
  - ✓ Virtual view of memory
  - ✓ Can debug linearly mapped region of memory
  - ✓ User space utility to regenerate ELF headers to create the ELF headers for vmalloc regions
- crash
  - ✓ Physical view of memory

Advantages

- Increased reliability
  - ✓ Dump is captured from a newly booted kernel
- Enhanced flexibility
  - ✓ Dump image can be saved to virtually any storage media supported by kernel
  - ✓ Filtering mechanism can be plugged in

Advantages Contd..

- Ease of use
  - ✓ Standard utilities can be used to save the dump image either locally or remotely
  - ✓ Standard analysis tools like gdb can be directly used for limited debugging

Limitations

- Devices are not shutdown/reset after a crash which might result in a driver initialization failure in capture kernel
- Non-disruptive dumping is not possible

Current Status

- Initial i386 implementation is mainline now (2.6.13-rc1)
- Driver initialization issues are being addressed
  - ✓ Shared Interrupts
    - ➢ irqpoll commandline option, Disabling PCI interrupts etc.
  - ✓ Driver hardening
    - ➢ Reset the device if it is not reset already.

ToDos

- Port kdump to other platforms like x86_64 and ppc64
- Implement kernel pages only filtering mechanism
- Relocatable Kernel for binary image unification
- Initialize APICs before timer initialization

Downloads

- Kdump patches for kexec-tools and test reports are available at:

  http://lse.sourceforge.net/kdump/

Hands-On
(Kdump)

Configure Kdump (Kernel)

- Two kernels need to be built in order to make this feature working
- Build First Kernel
  - Change Makefile to give a version name may be “2.6.13.2-1M”
  - Enable "kexec system call" feature (in Processor type and features) (CONFIG_KEXEC=y)
  - Enable "sysfs file system support" (in Pseudo filesystems (CONFIG_SYSFS=y) (Enabled by default)
  - Enable “Magic SysRq Key” (CONFIG_MAGIC_SYSRQ=y) (in Kernel Hacking/Kernel Debugging)

Configure Kdump (Kernel) Contd..

  - Enable “Compile the kernel with debug info” (CONFIG_DEBUG_INFO=y)
  - For simplicity, build appropriate disk and network driver in kernel.
  - Boot into first kernel with the command line parameter "*crashkernel=Y@X*". Use appropriate values for X and Y. Y denotes how much memory to reserve for the second kernel, and X denotes at what physical address the reserved memory section starts. For example: "*crashkernel=48M@16M*".

- Build Second Kernel
  - Change Makefile to give a version name may be “2.6.13.2-16M”

Configure Kdump (Kernel) Contd...

- Take arch/i386/defconfig as .config file
- Enable "Configure standard kernel feature (for small systems)" (under General setup).
- Enable “High Memory Support” (4GB) (CONFIG_HIGHMEM=y) & (CONFIG_HIGHMEM4G=y)
- Enable "kernel crash dumps" feature (in Processor type and features). (CONFIG_CRASH_DUMP=y)
- Specify a suitable value for "Physical address where the kernel is loaded" (in Processor type and features). Typically this value should be same as X (See option d) above, e.g., 16 MB or 0x1000000. (CONFIG_PHYSICAL_START=0x1000000)

Configure Kdump (Kernel) Contd..

- Enable "/proc/vmcore support" (Optional, in Pseudo filesystems (CONFIG_PROC_VMCORE=y)
- Disable SMP support and build a UP kernel (CONFIG_SMP=n)
- Enable "Local APIC support on uniprocessors". (CONFIG_X86_UP_APIC=y)
- Enable "IO-APIC support on uniprocessors" (CONFIG_X86_UP_IOAPIC=y)
- make; make modules_install

Configure Kdump (User Space)

- Pre-load capture kernel
  - *kexec -p <capture kernel vmlinux image> --args-linux --elf32-core-headers --append="root=<root-dev> init 3 irqpoll"*

  Note: Specify suitable kernel command options and initrd, if need be.

- Force Panic
  - Drop to execution level 3 (*init 3*)
  - *echo c > /proc/sysrq-trigger*

  System Boots into capture kernel.

Access/Save Core Image

- Save Dump (/proc/vmcore)
  - *cp   /proc/vmcore   <destination directory>*
- Save Dump (/dev/oldmem)
  - *mknod /dev/oldmem c 1 12*
  - *dd if=/dev/oldmem of=<destination directory>*

Dump Capture Using initrd

- First Kernel Preparation
  - Enable “Loopback device support” (BLK_DEV_LOOP=y) (Under Device Drivers/Block Devices)
  - make; make modules_install
  - Boot into first kernel
- Capture Kernel Preparation
  - Enable “Initial RAM disk (initrd) support” (BLK_DEV_INITRD=y) (Under Device Drivers/Block Devices)
  - make; make modules_install

Dump capture using initrd

- Download and apply appropriate initrd patch
  - *http://lse.sf.net/kdump/patches/*
- Generate initrd for capture kernel based on distribution used
  - *mkinitrd -k <capture kernel vmlinux> -i <capture kernel initrd> -v <dump device> -f <dump device file system>*

  or

  - *mkinitrd --dump-dev=<dump-dev> --dump-fs<fs> <initrd-image> <capture-kernel-version>*

Dump capture using initrd

- Pre-load capture kernel
  - *kexec -p <capture kernel vmlinux image> --args-linux --initrd=<capture kernel initrd> --elf32-core-headers --append="root=<root-dev> init 3 irqpoll"*
- Force Panic
  - Drop to execution level 3 (*init 3*)
  - *echo c > /proc/sysrq-trigger*

System Boots into capture kernel. Initrd automatically saves the dump to dump device and system reboots back.

Legal Statement

-
- This work represents the view of the author and does not necessarily represent the view of IBM.
- IBM, and the IBM logo, are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linux Torvalds in the United States, other countries, or both.
- i386 is trademark of Intel Corporation in the United States, other countries or both.
- Other company, product, and service names may be trademarks or service marks of others.
- References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.
- This document is provided "AS IS" with no express or implied warranties. Use the information in this document at your own risk.

Questions?