Frist Pass Linux Kernel Dump

Dyi-Wu Liu

July, 2014

Learning Objectives - Why we are here ?

Care Why

- Effective action plan development for outage case

Know Why

- Direct evidence finding instead of the best guess on action plan development

Know How

- Build it and test it

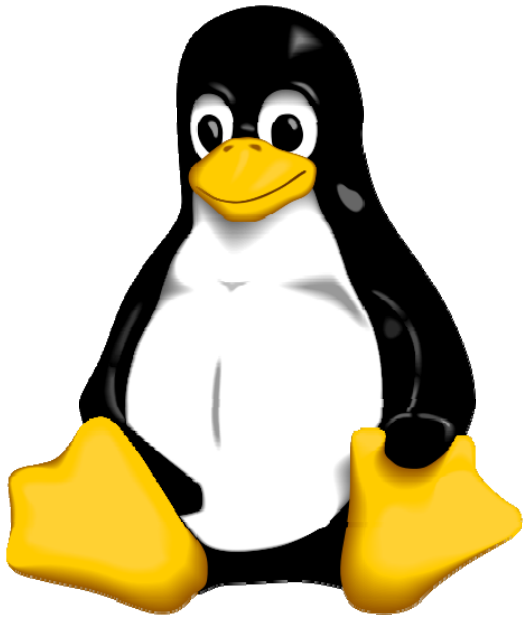
Know What

- What is kdump ?
- How does it work ?

QUINN, James Brian. "The intelligent enterprise a new paradigm." The Executive, 1992, 6.4: 48-63.

Agenda

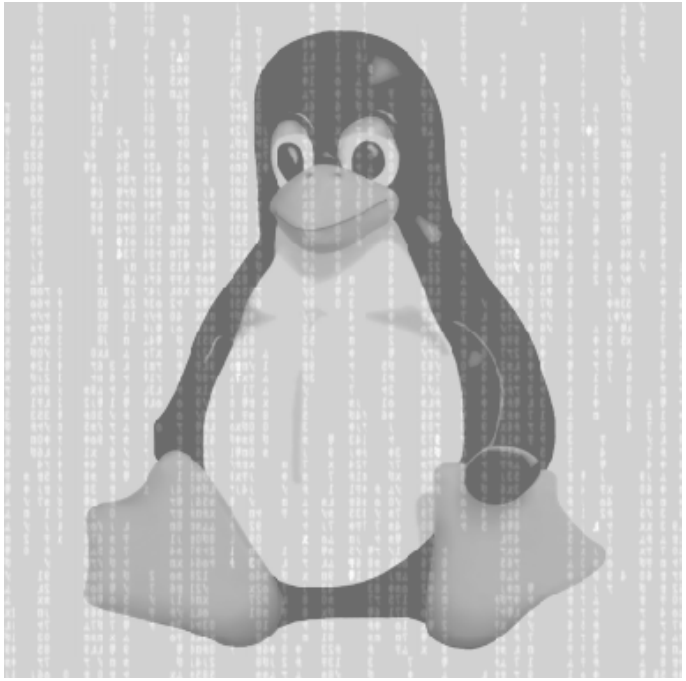
- How kdump work
- Kdump setup/validation
- Serial console
- First Pass Kdump Analysis
- Labs



How kdump work

Kernel Crash Dumper

- Overview
- Kernel to Kernel Boot Loader
- User space tool and kernel system call
- Kexec on panic
- Work flow
- Pre-loading
- Post crash
- In core map
- ELF core file



How kdump work

Overview

- A kexec Based Kernel Crash Dumping Mechanism
- Dump capture from crashing kernel's context
- 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

How kdump work

Kernel to Kernel Boot Loader

- Running kernel acts as a loader for the new kernel
- System directly jumps from one kernel to another
  - Skips BIOS or Firmware stage
- Reboots are extremely fast
- Memory can be preserved across reboots
  - Since BIOS is skipped, it is left to the OS (capture kernel) to retain or erase memory

How kdump work

User space tool and kernel system call

- Allows a Linux kernel to boot another kernel
- Currently available on i386, x86_64, ia64 and ppc64 platforms
- Two components
  - User space tool – kexec(8)
  - Kernel System Call - kexec_load(2)
- Load a new kernel

```
kexec -l <kernel-image> --append=<options>
```

- Exec new kernel

```
kexec -e
```

How kdump work

Kexec on panic

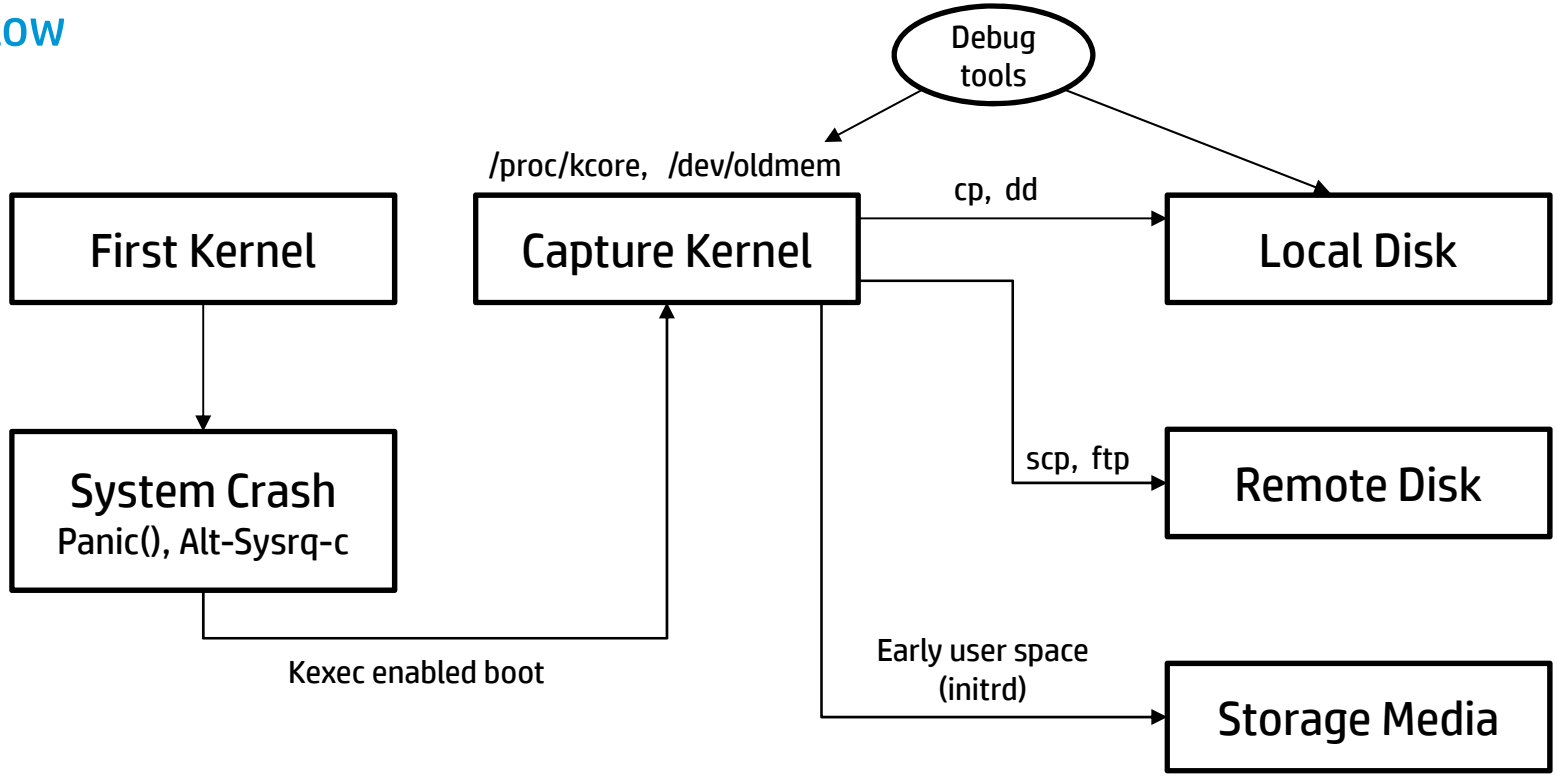
- Enables booting a new kernel after system crash
- Devices are not shutdown
- New kernel runs from a reserved memory location
  - Protection against ongoing DMA at the time of crash
- Loading capture kernel

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

- Execution of capture kernel
  - panic()
  - Alt-Sysrq-c

How kdump work

Work flow

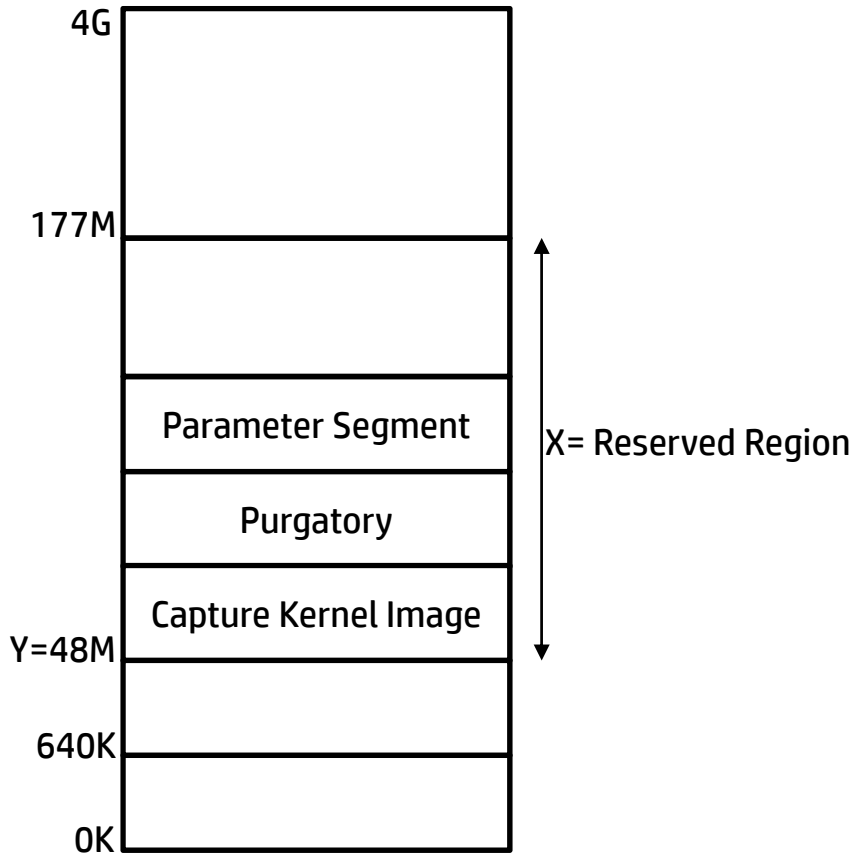


How kdump work

Pre-Loading

- Reserve memory for capture kernel (crashkernel=X@Y)
- Pre-load the capture kernel
- Capture kernel runs from reserved memory location
- For example (RHEL65)

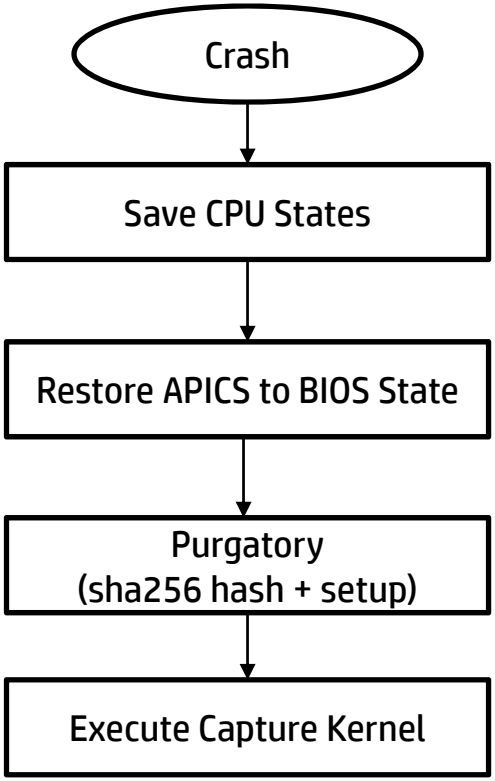
```
/boot/grub/menu.lst
      crashkernel=auto
/proc/cmdline
      crashkernel=129M@0M
# grep Crash /proc/iomem
03000000-0b0fffff : Crash kernel
Y: 0x3000000 (48M)
X: 0xb0fffff-0x3000000+1 (129M)
```



How kdump work

Post Crash

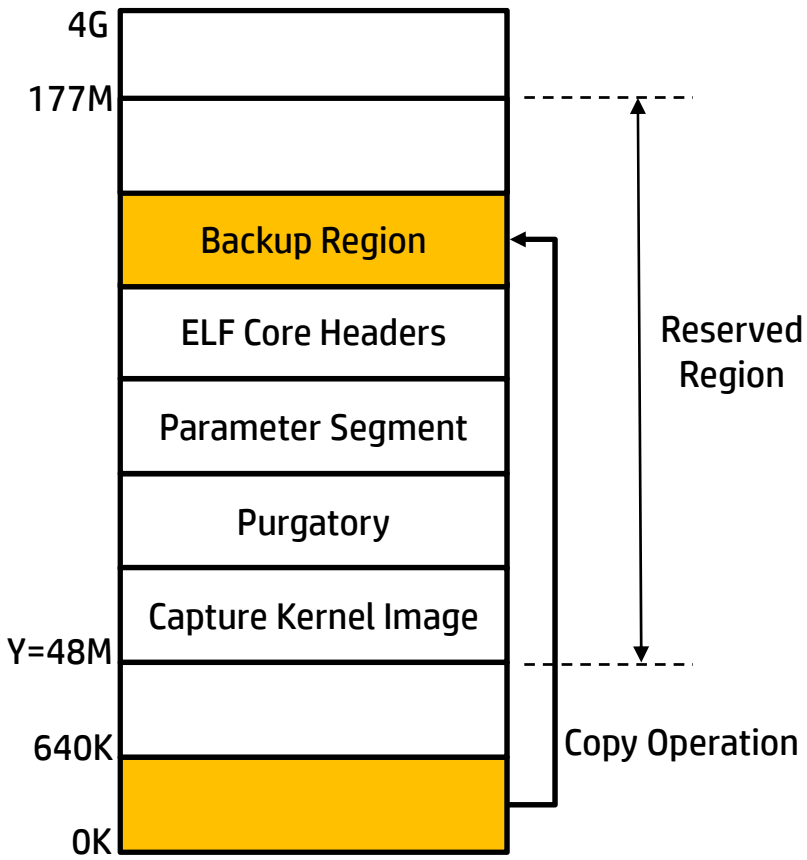
- CUP states are saved and other CPUs are halted
- CPU register states are saved on ELF note format
- 1K memory is reserved statically per CPU
- LAPIC/IOAPIC are disabled and put back into PIC or virtual wire mode
- Purgatory is run and control is transferred to capture kernel



How kdump work

In Core Map

- Dump information exchanged in an ELF format core file across the kernels (/proc/kcore)
- Kexec-tools prepare ELF headers and pre-load them in the reserved region
- Kexec-tools use /proc/iomem to retrieve system RAM information
- Content of first 640K of memory are backed up in backup region for the requirement to boot SMP capture kernel



How kdump work

ELF Core file

- Accessing dump image in ELF Core format
  - /proc/kcore
- Accessing dump image in linear raw format
  - /dev/oldmem
- ELF32/ELF64 format headers
- Physical addresses are filled for all the regions
- Virtual addresses are filled only for linearly mapped memory region

ELF Header	Program Header PT_NOTE	Program Header PT_NOTE	------	Per CPU Register States	Dump Images

kdump setup/validation

Setup

- Required packages
- Magic SysRq key enable
- Configurations

Validation

- Check configuration after kdump enabled
- Force crash by Magic SysRq key c
- Force crash by Non-maskable interrupt (NMI)
- Saved vmcore

kdump setup/validation (RHEL 6.5)

Setup

- Required packages

```
# yum list installed | grep -e kexec -e kdump
kexec-tools.x86_64          2.0.0-273.el6      @anaconda-RedHatEnterpriseLinux-201311111358.x86_64/6.5
system-config-kdump.noarch 2.0.5-15.el6       @anaconda-RedHatEnterpriseLinux-201311111358.x86_64/6.5

# rpm -qa | grep -e kexec -e kdump
system-config-kdump-2.0.5-15.el6.noarch
kexec-tools-2.0.0-273.el6.x86_64
```

- Magic SysRq key enable

```
# grep sysrq /etc/sysctl.conf
kernel.sysrq = 1
```


kdump setup/validation (RHEL 6.5)

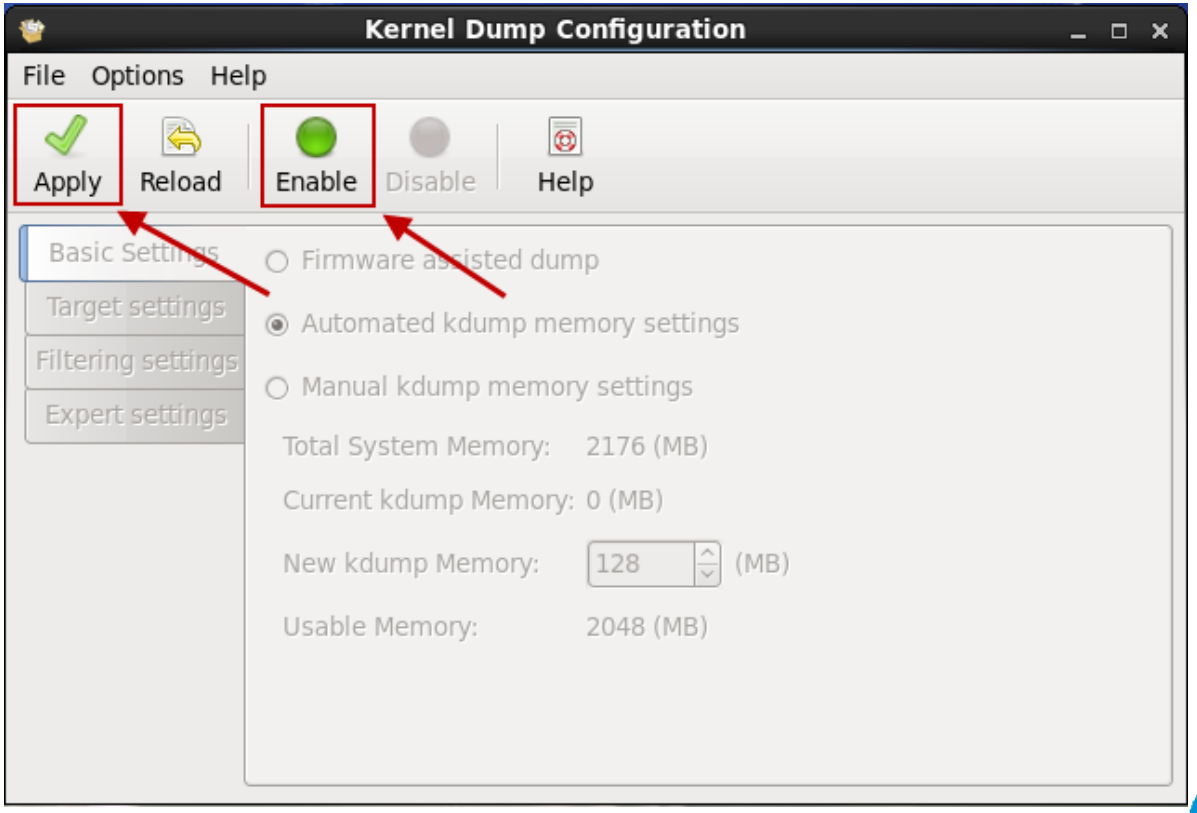
Setup

- Configurations

```
# system-config-kdump
```

System reboot is required

- kdump.conf(5)
- makedumpfile(8)



kdump setup/validation (RHEL 6.5)

Validation

- Check configuration after kdump enabled

```
# grep crashkernel /boot/grub/menu.lst
        kernel /vmlinuz-2.6.32-431.el6.x86_64
root=/dev/mapper/VolGroup-lv_root ro rd_NO_LUKS
LANG=en_US.UTF-8 rd_NO_MD
rd_LVM_LV=VolGroup/lv_swap SYSFONT=latarcyrheb-
sun16 rd_LVM_LV=VolGroup/lv_root
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
crashkernel=auto

# cat /proc/cmdline
root=/dev/mapper/VolGroup-lv_root ro rd_NO_LUKS
LANG=en_US.UTF-8 rd_NO_MD
rd_LVM_LV=VolGroup/lv_swap SYSFONT=latarcyrheb-
sun16 rd_LVM_LV=VolGroup/lv_root
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
crashkernel=129M@0M
```

```
# chkconfig --list kdump
kdump  0:off  1:off  2:on   3:on   4:on   5:on   6:off

# service kdump status
Kdump is operational

# grep -v \# /etc/kdump.conf
path /var/crash
core_collector makedumpfile -c --message-level 1 -d 31

# sysctl kernel.sysrq
kernel.sysrq = 1
```


kdump setup/validation (RHEL 6.5)

Validation

- The c key of Magic SysRq perform a system crash

  Keystroke : Alt + fn + SysRq + c

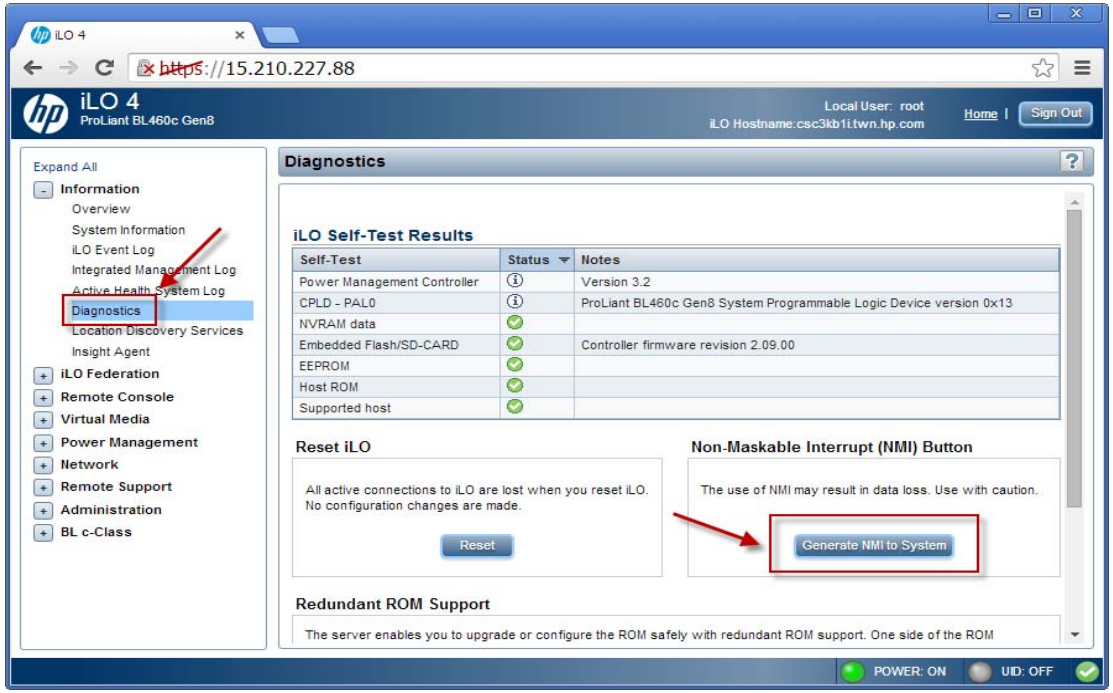
- Force crash by Magic SysRq key c

```
# cat /usr/local/bin/toc
#!/bin/bash
# echo 1 > /proc/sys/kernel/sysrq
echo c > /proc/sysrq-trigger
```


kdump setup/validation (RHEL 6.5)

Validation

- Force crash by Non-maskable interrupt (NMI)



kdump setup/validation (RHEL 6.5)

Validation

- Saved vmcore

```
# ls -lR /var/crash/
/var/crash/:
total 4
drwxr-xr-x 2 root root 4096 Jul  7 12:40 127.0.0.1-2014-07-07-12:40:40

/var/crash/127.0.0.1-2014-07-07-12:40:40:
total 21448
-rw------- 1 root root 21868390 Jul  7 12:40 vmcore
-rw-r--r-- 1 root root    88328 Jul  7 12:40 vmcore-dmesg.txt

# cd /var/crash/127.0.0.1-2014-07-07-12\:40\:40/
# file *
vmcore:           data
vmcore-dmesg.txt: ASCII English text, with very long lines

# strings vmcore| grep OSRELEASE
OSRELEASE=2.6.32-431.el6.x86_64

# uname -r
2.6.32-431.el6.x86_64
```


kdump setup/validation (SLES 11SP3)

Setup

- Required packages

```
# zypper se --installed-only | grep -e kdump -e kexec -e Summary -e ------
S | Name          | Summary                                                                  | Type
--+---------------+--------------------------------------------------------------------------+--------
i | crash         | Crash utility for live systems; netdump, diskdump, LKCD or mcore dumpfiles | package
i | kdump         | Script for kdump                                                         | package
i | kexec-tools   | Tools for fast kernel loading                                            | package
i | yast2-kdump   | Configuration of kdump                                                   | package

# rpm -qa | grep -e kdump -e kexec
kexec-tools-2.0.3-0.15.18
yast2-kdump-2.17.26-0.8.17
kdump-0.8.4-0.29.5
```

- Magic SysRq key enable

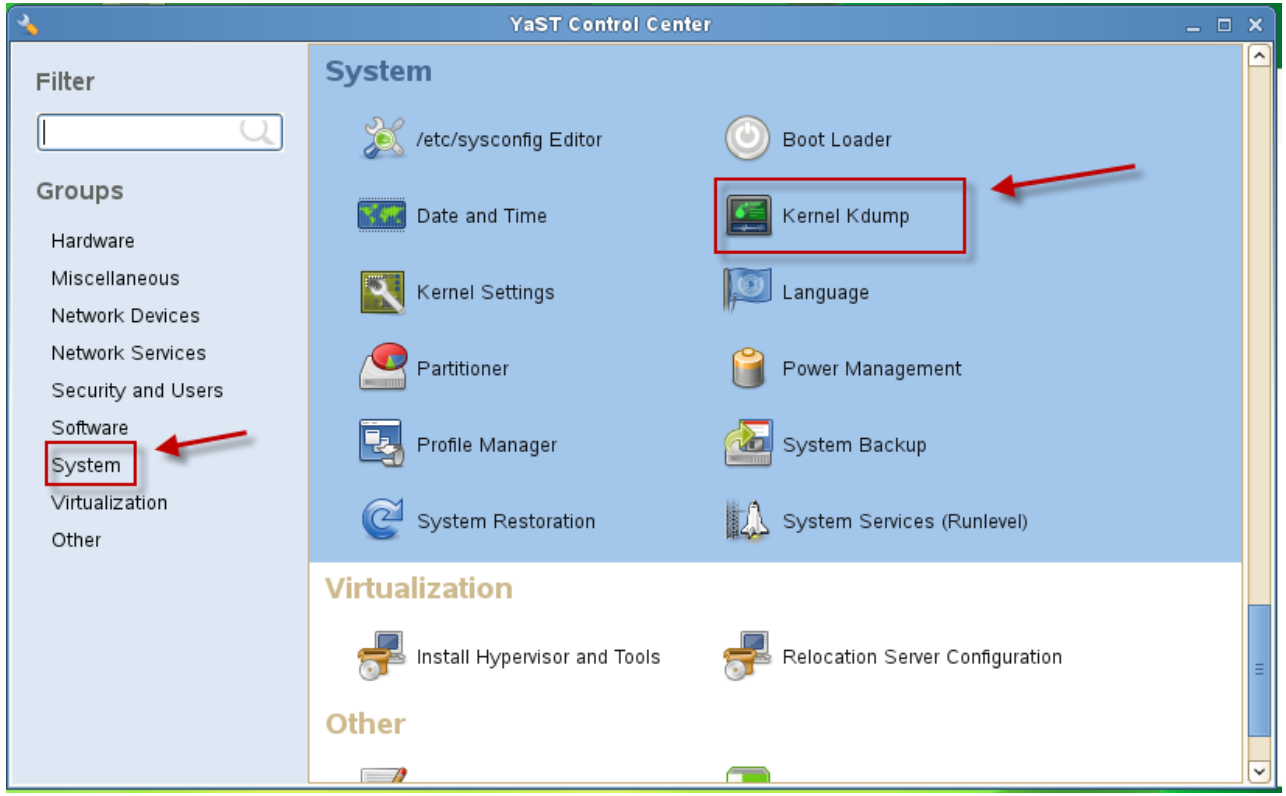
```
# tail -1 /etc/sysctl.conf
kernel.sysrq = 1
```


kdump setup/validation (SLES 11SP3)

Setup

- Configurations

```
# yast2
```



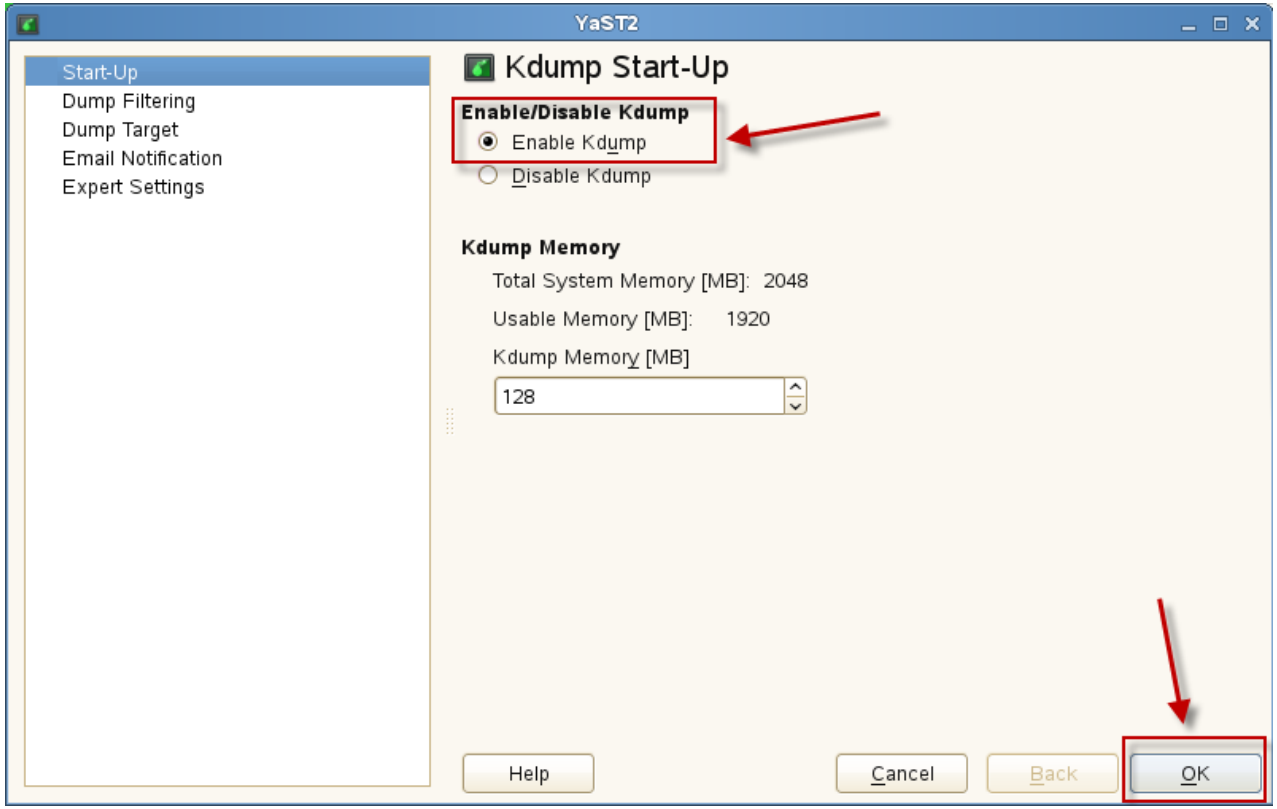
kdump setup/validation (SLES 11SP3)

Setup

- Configurations

  System reboot is required

  - kdump.conf(5)
  - makedumpfile(8)



kdump setup/validation (SLES 11SP3)

Validation

- Check configuration after kdump enabled

```
# grep crashkernel /boot/grub/menu.lst
    kernel /boot/vmlinuz-3.0.76-0.11-default
root=/dev/sda2 resume=/dev/sda1 splash=silent
showopts crashkernel=256M-:128M vga=0x314

# cat /proc/cmdline
root=/dev/sda2 resume=/dev/sda1 splash=silent
crashkernel=256M-:128M vga=0x314

# chkconfig --list boot.kdump
boot.kdump                0:off  1:off  2:off
3:off  4:off  5:off  6:off  B:on

# service boot.kdump status
kdump kernel loaded            running
```

```
# grep -v \# /etc/sysconfig/kdump |grep -v ^$|grep -v \"\"
KDUMP_CPUS="1"
KDUMP_IMMEDIATE_REBOOT="yes"
KDUMP_SAVEDIR="file:///var/crash"
KDUMP_KEEP_OLD_DUMPS="5"
KDUMP_FREE_DISK_SIZE="64"
KDUMP_VERBOSE="3"
KDUMP_DUMPLEVEL="31"
KDUMP_DUMPFORMAT="lzo"
KDUMP_CONTINUE_ON_ERROR="true"
KDUMP_COPY_KERNEL="yes"
KDUMP_NETCONFIG="auto“

# sysctl kernel.sysrq
kernel.sysrq = 1
```


kdump setup/validation (SLES 11SP3)

Validation

- The c key of Magic SysRq perform a system crash

  Keystroke : Alt + fn + SysRq + c

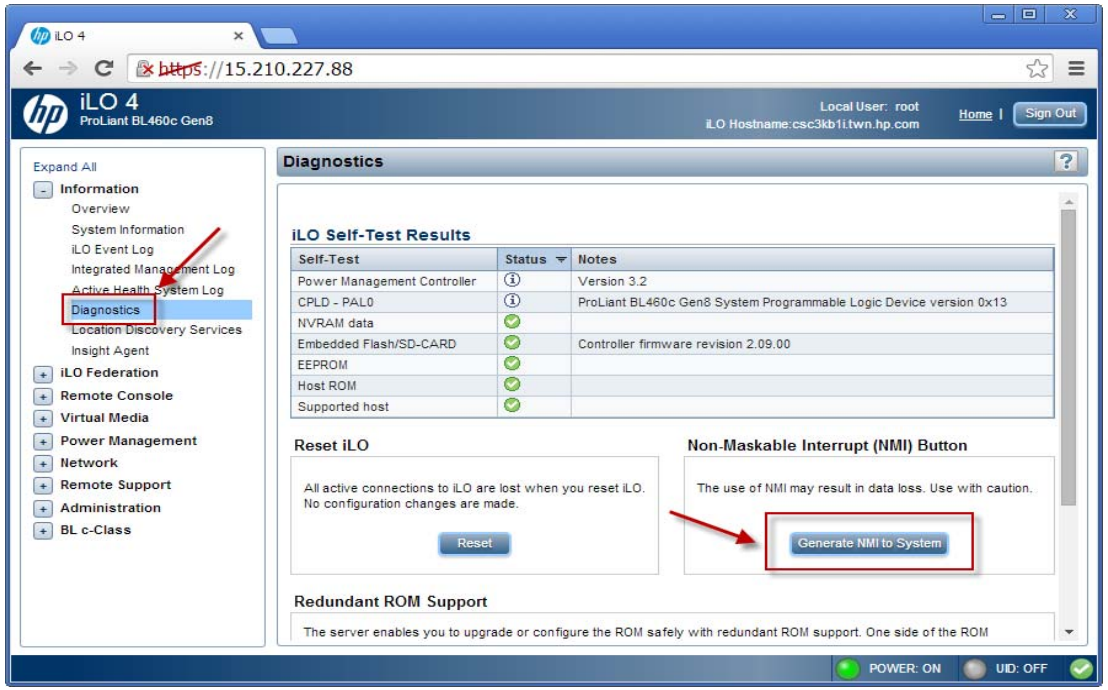
- Force crash by Magic SysRq key c

```
# cat /usr/local/bin/toc
#!/bin/bash
# echo 1 > /proc/sys/kernel/sysrq
echo c > /proc/sysrq-trigger
```


kdump setup/validation (SLES 11SP3)

Validation

- Force crash by Non-maskable interrupt (NMI)



kdump setup/validation (SLES 11SP3)

Validation

- Saved vmcore

```
# ls -lR /var/crash/
/var/crash/:
total 4
drwxr-xr-x 2 root root 4096 Jul  6 08:14 2014-07-06-00:32
/var/crash/2014-07-06-00:32:
total 36460
-rw------- 1 root root   112018 Jul  6 00:32 dmesg.txt
-rw-r--r-- 1 root root      179 Jul  6 00:32 README.txt
-rw-r--r-- 1 root root  2067870 Jul  6 00:32 System.map-3.0.76-0.11-default
-rw------- 1 root root 30481332 Jul  6 00:32 vmcore
-rw-r--r-- 1 root root  4605902 Jul  6 00:32 vmlinux-3.0.76-0.11-default.gz

# cd /var/crash/2014-07-06-00\:32/
# file *
dmesg.txt:                      ASCII C++ program text
README.txt:                     ASCII text
System.map-3.0.76-0.11-default: ASCII text
vmcore:                         data
vmlinux-3.0.76-0.11-default.gz: gzip compressed data, from Unix, max compression

# strings vmcore | grep OSRELEASE|head -1
OSRELEASE=3.0.76-0.11-default

# uname -r
3.0.76-0.11-default
```

Serial console

Serial Console on Bare Metal

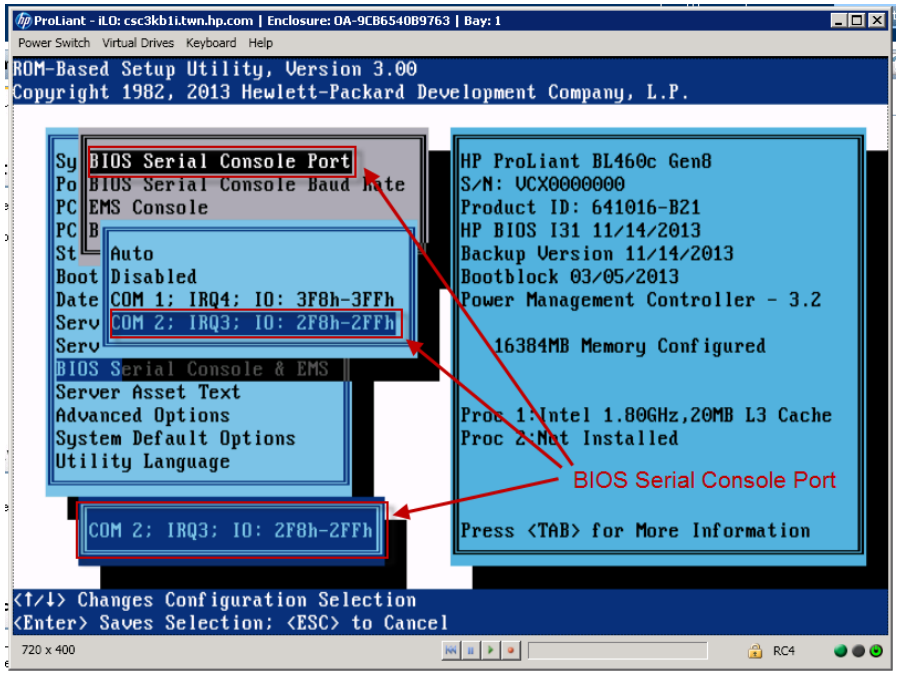
- BIOS configuration
- GRUB configuration
- HP iLO VSP (Virtual Serial Port)
- Connect trough iLO

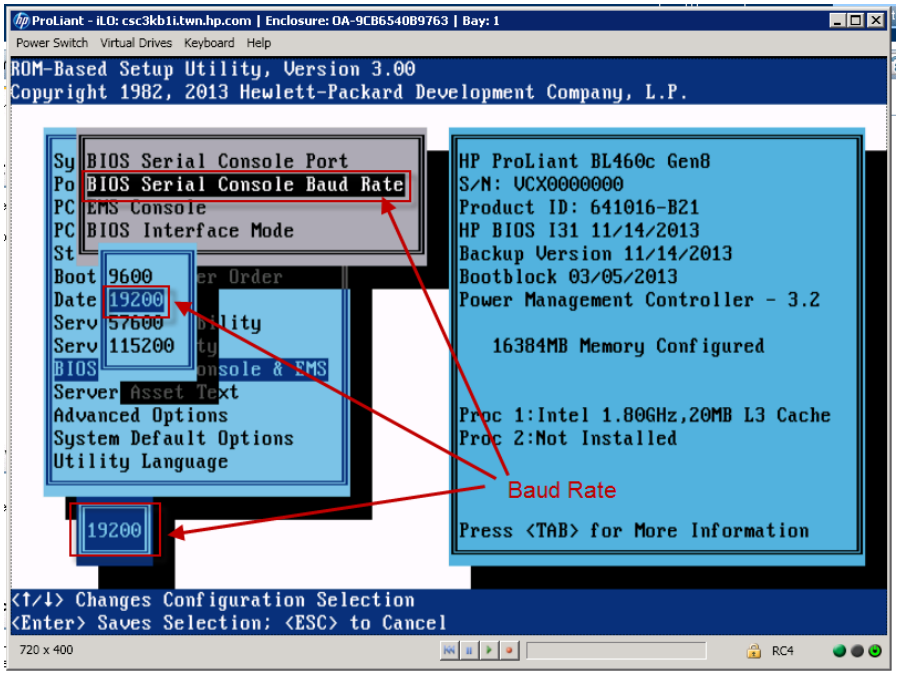
Serial Console on VM Guest

- BIOS configuration
- GRUB configuration
- Serial port on VM Guest
- Connect trough named pipe via putty

Serial Console - Bare Metal

- BIOS Serial Console to COM2
- BIOS Serial Console Baud Rate to 19200





Serial Console - Bare Metal (RHEL 6.5)

Name of COM2, IRQ and I/O Port

```
# setserial -g /dev/ttyS[01]
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 3
```

Kernel line in /boot/grub/menu.lst

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
serial --unit=1 --speed=19200 --word=8 --parity=no --stop=1
terminal --timeout=5 serial console
hiddenmenu
title Red Hat Enterprise Linux (2.6.32-431.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/vg00-root rd_NO_LUKS
rd_LVM_LV=vg00/swap LANG=en_US.UTF-8 rd_LVM_LV=vg00/root rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=128M  KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb console=tty0 console=ttyS1,19200n8
        initrd /initramfs-2.6.32-431.el6.x86_64.img
```


Serial Console - Bare Metal (RHEL 6.5)

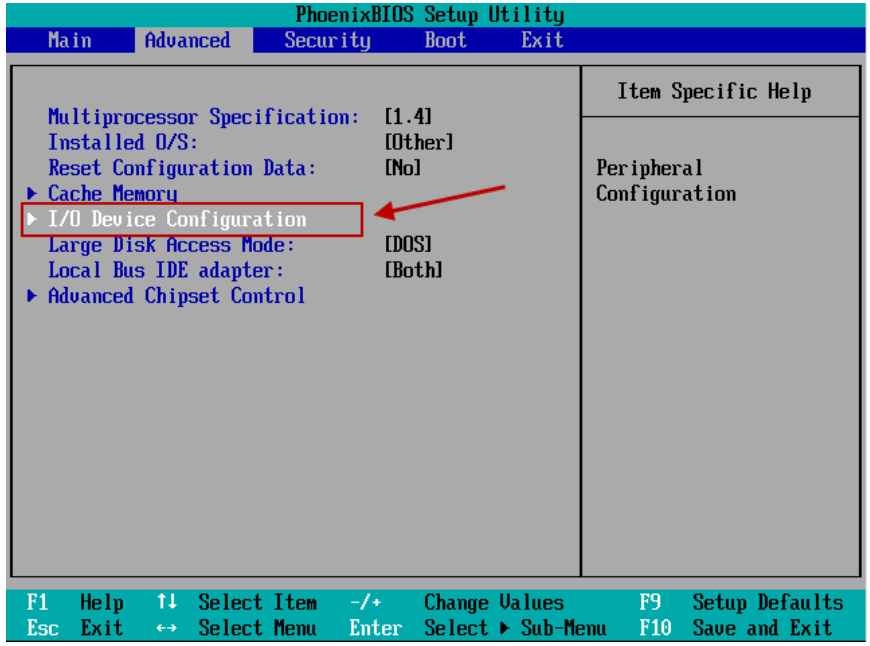
Connect trough iLO

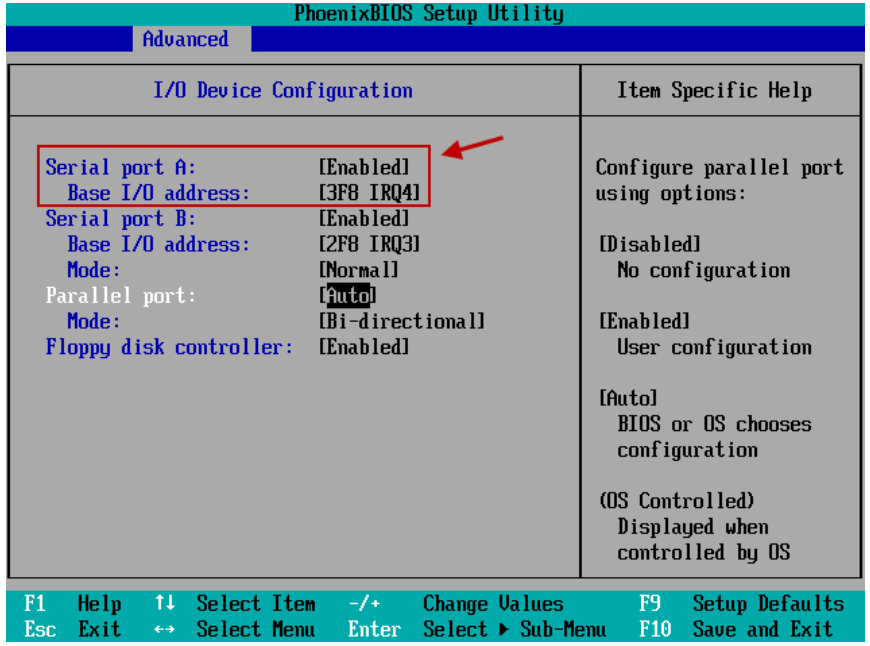
```
# ssh csc3kb1i
root@csc3kb1i's password:
User:root logged-in to csc3kb1i.twn.hp.com(15.210.227.88 / fe80::a65d:36ff:fefe:bb9)
iLO 4 Advanced for BladeSystem 1.32 at  Nov 05 2013
Server Name:
Server Power: On
</>hpiLO-> vsp

Virtual Serial Port Active: COM2
Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.

Red Hat Enterprise Linux Server release 6.5 (Santiago)
Kernel 2.6.32-431.el6.x86_64 on an x86_64
csc3kb1.twn.hp.com login:
```

Console session log in text

Serial Console – VM Guest (RHEL 6.5)

- Enable serial port A
- Base I/O address, IRQ number





Serial Console – VM Guest (RHEL 6.5)

Name of port A, IRQ and I/O Port

```
# setserial -g /dev/ttyS[01]
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 3
```

Kernel line in /boot/grub/menu.lst

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
serial --unit=0 --speed=19200 --word=8 --parity=no --stop=1
terminal --timeout=5 serial console
# hiddenmenu
title Red Hat Enterprise Linux (2.6.32-431.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/vg00-root rd_NO_LUKS
rd_LVM_LV=vg00/swap LANG=en_US.UTF-8 rd_LVM_LV=vg00/root rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=128M  KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb console=tty0 console=ttyS0,19200n8
        initrd /initramfs-2.6.32-431.el6.x86_64.img
```


Serial Console – VM Guest (RHEL 6.5)

Enable ttyS0 and console device in /etc/securetty

```
# grep -e console -e ttyS0 /etc/securetty
ttyS0
console
```

Check the agetty process

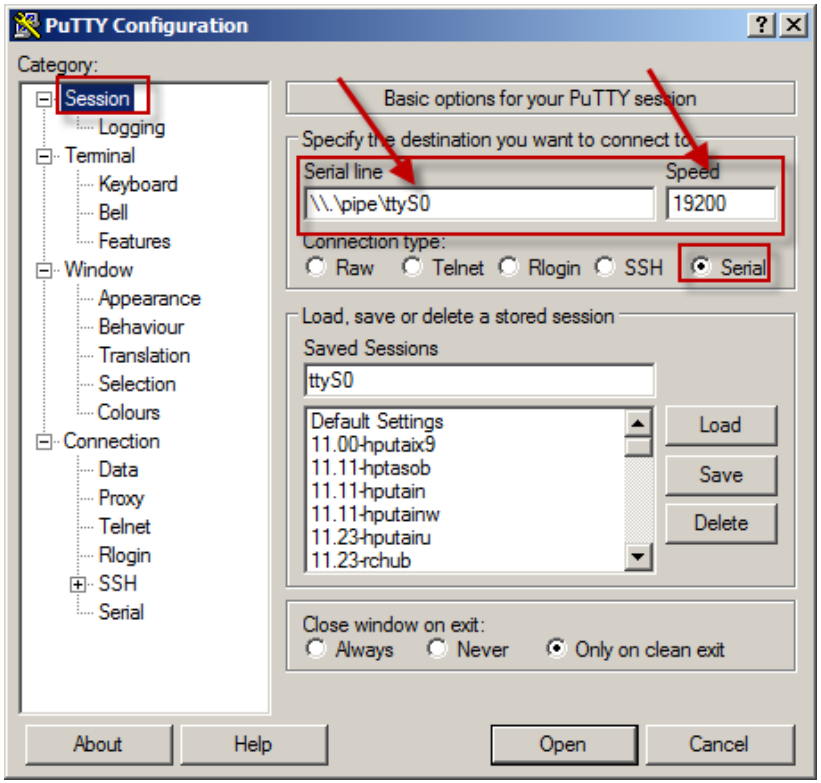
```
# initctl status serial DEV=ttyS0
serial (ttyS0) start/running, process 2321
# ps -ef | grep agetty | grep -v grep
root      2321     1  0 13:39 ttyS0    00:00:00 /sbin/agetty /dev/ttyS0 19200 vt100-navgrep console
```

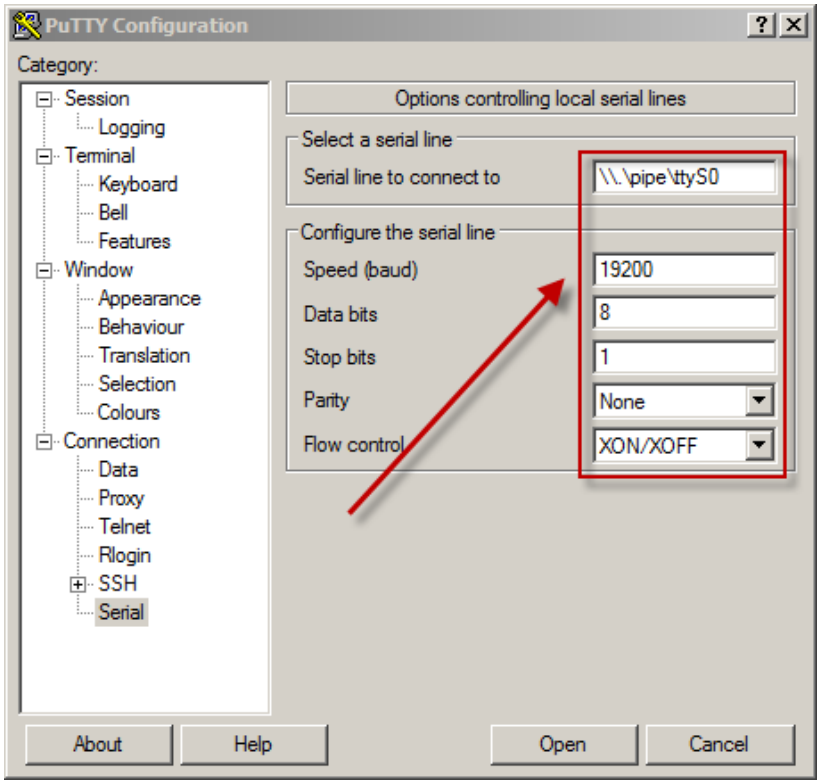
Connect trough putty

```
Red Hat Enterprise Linux Server release 6.5 (Santiago)
Kernel 2.6.32-431.el6.x86_64 on an x86_64

linux3.asiapacific.hpqcorp.net login: root
Password:
Last login: Mon Jul  7 13:39:37 from kcz.twn.hp.com
# tty
/dev/ttyS0
```


Serial Console – putty client (RHEL 6.5)

Named pipe connection from putty





Serial Console – VM Guest (SLES 11 SP3)

Name of port A, IRQ and I/O Port

```
# setserial -g /dev/ttyS[01]
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 3
```

Port A entry in /etc/init.d/setserial

```
# grep 0x3F8 /etc/init.d/setserial
     run_setserial /dev/ttyS0 uart 16550A port 0x3F8 irq 4
```

Kernel line in /boot/grub/menu.lst

```
##YaST - generic_mbr
# gfxmenu (hd0,1)/boot/message
serial --unit=0 --speed=19200
terminal --timeout=5 serial console
##YaST - activate

###Don't change this comment - YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11 SP3 - 3.0.76-0.11
    root (hd0,1)
    kernel /boot/vmlinuz-3.0.76-0.11-default root=/dev/sda2 resume=/dev/sda1 splash=silent showopts
crashkernel=256M-:128M vga=0x314 console=tty0 console=ttyS0,19200
    initrd /boot/initrd-3.0.76-0.11-default
```


Serial Console – VM Guest (SLES 11 SP3)

Enable ttyS0 and console device in /etc/securetty

```
# tail -2 /etc/securetty
ttyS0
console
```

Enable agetty in /etc/inittab

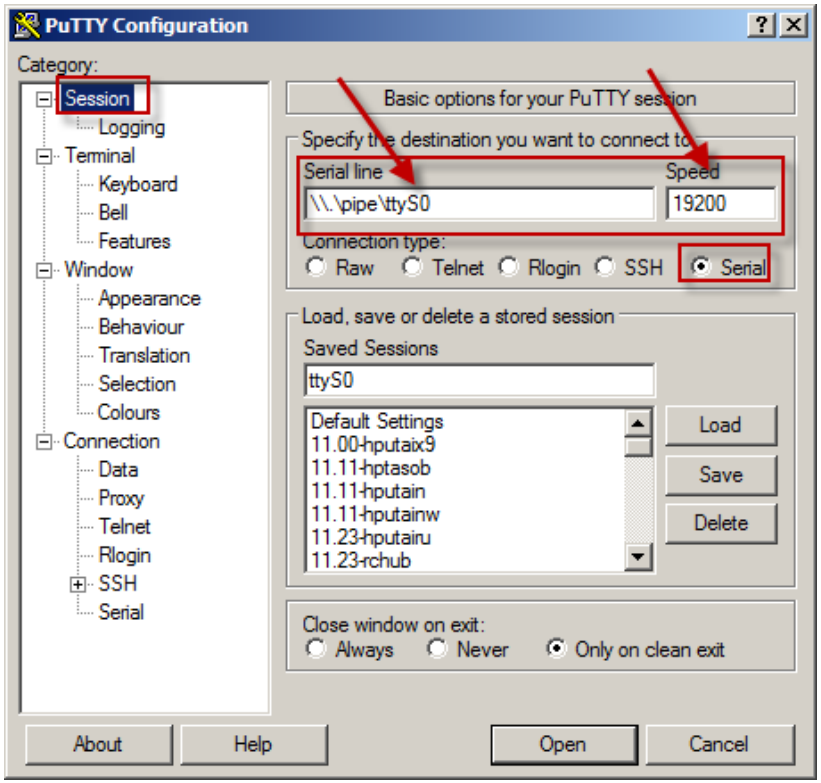
```
# grep console /etc/inittab
cons:12345:respawn:/sbin/smart_agetty -L 19200 console
```

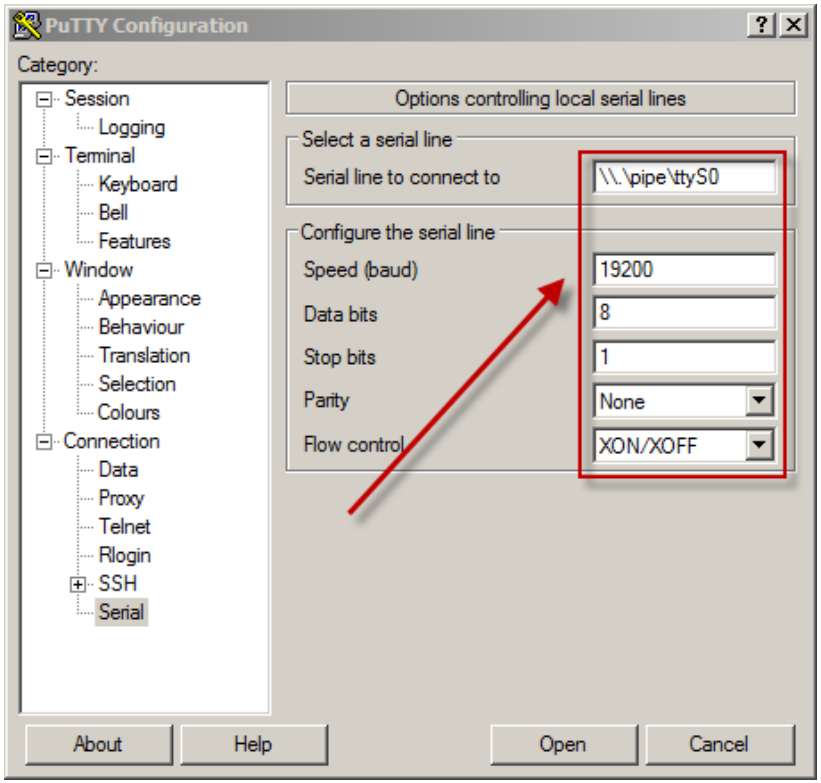
Connect trough putty

```
Welcome to SUSE Linux Enterprise Server 11 SP3  (x86_64) - Kernel 3.0.76-0.11-default (console).

linux-c4o5 login: root
Password:
Last login: Sun Jul  6 23:50:56 CST 2014 on console
# tty
/dev/console
```

Serial Console – VM Guest (SLES 11 SP3)

Named pipe connection from putty





First Pass Kdump Analysis

Types of hangs

Type of crashes

Vmcore analysis tools

- crash64
- mpykdump64.so
- crashinfo
- Remote crash
  - ✓ crash-server
  - ✓ crash-client – text version
  - ✓ crash-client – web version

Moment of ecstasy-

Doc Debugalov finds the elusive bug corrupting the heap...

First Pass Kdump Analysis

Type of hangs

- Network hang
  - loss of network connectivity but system still running
  - Does not respond to a ping
  - Users are unable to access system with ssh
  - The system can be accessed via the iLO, or serial port
  - Partial network hanging? check switches / routers
- I/O hang
  - Some storage has failed and the system is waiting for it
  - System responds to ping
  - Users hang after connecting
- Complete hang
  - System unresponsive to network and console

First Pass Kdump Analysis

Type of crashes

- Oops
  - One process gets killed immediate
  - A subsystem is left in an unknown state
- Panic
  - Something happens in the kernel that it is unable to continue from
- BUG
  - Macro in the kernel testing for a condition that should not happen
- OOM Killer
  - Kernel memory starvation where the kernel must kill processes to keep
- NMI
  - Hardware generated or virtual button on the iLO
- Other
  - The perhaps overlooked loss of power causing an unexplained system outage

First Pass Kdump Analysis

Vmcore analysis tools

- crash64
- mpykdump64.so
- crashinfo
- crash-server
- crash-client – text version
- crash-client – web version
- ftp://kczb.twn.hp.com/dist/debuginfo/tools/
- http://sourceforge.net/projects/pykdump/

First Pass Kdump Analysis

crash64

```
# crash64
crash64 7.0.3
crash64: cannot find booted kernel -- please enter namelist argument
Usage:
  crash [OPTION]... NAMELIST MEMORY-IMAGE  (dumpfile form)
  crash [OPTION]... [NAMELIST]             (live system form)
Enter "crash64 -h" for details.
```

- NAMELIST
  - The debuginfo file from RedHat for RHEL or from Novell for SLES
  - http://bl460node3.alf.cpqcorp.net/debuginfo/
  - ftp://kczb.twn.hp.com/dist/debuginfo
- MEMORY-IMAGE
  - Saved vmcore from kdump
  - Live system /dev/crash

First Pass Kdump Analysis

crash64 invocation on RHEL

```
# ll
total 152036
-rw------- 1 root root  21868390 Jul  7 12:40 vmcore
-rw-r--r-- 1 root root     88328 Jul  7 12:40 vmcore-dmesg.txt
-rw-r--r-- 1 root root 133715563 Jun 19 14:33 vmlinux-2.6.32-431.el6.x86_64

# crash64 vmlinux-2.6.32-431.el6.x86_64 vmcore
crash64 7.0.3
Copyright (C) 2002-2013  Red Hat, Inc.
Snipped …
crash64> help
*               files           mach            repeat          timer
alias           foreach         mod             runq            tree
ascii           fuser           mount           search          union
bt              gdb             net             set             vm
btop            help            p               sig             vtop
dev             ipcs            ps              struct          waitq
dis             irq             pte             swap            whatis
eval            kmem            ptob            sym             wr
exit            list            ptov            sys             q
extend          log             rd              task
```


First Pass Kdump Analysis

crash64 invocation on SLES

```
# crash64 -s vmlinux-3.0.76-0.11-default.debug vmcore
crash64: vmlinux-3.0.76-0.11-default.debug: no text and data contents

crash64: The namelist argument supplied in this case is a debuginfo file,
which must be accompanied by the kernel file from which it was derived.

# ll
total 147192
-rw------- 1 root root   112018 Jul  6 00:32 dmesg.txt
-rw-r--r-- 1 root root      179 Jul  6 00:32 README.txt
-rw-r--r-- 1 root root  2067870 Jul  6 00:32 System.map-3.0.76-0.11-default
-rw------- 1 root root 30481332 Jul  6 00:32 vmcore
-rw-r--r-- 1 root root 22701625 Jul  6 00:32 vmlinux-3.0.76-0.11-default
-rw-r--r-- 1 root root 95177924 Jul  7 00:30 vmlinux-3.0.76-0.11-default.debug

# crash64 -s vmlinux-3.0.76-0.11-default vmcore
crash64> sys
      KERNEL: vmlinux-3.0.76-0.11-default
    DUMPFILE: vmcore  [PARTIAL DUMP]
        CPUS: 2
        DATE: Sun Jul  6 00:32:10 2014
Snipped …
```


First Pass Kdump Analysis

crashinfo – mpykdump64.so

```
# ll /usr/local/lib
total 3068
-rwxr-xr-x 1 root root 3140837 Nov 16  2013 mpykdump64.so

# crash64 -s vmlinux-2.6.32-431.el6.x86_64 vmcore
crash64> extend /usr/local/lib/mpykdump64.so
/usr/local/lib/mpykdump64.so: shared object loaded
crash64> help
*              extend         mach           repeat         timer
alias          files          mod            runq           tree
ascii          foreach        mount          search         union
bt             fuser          net            set            vm
btop           gdb            nfsshow        sig            vtop
crashinfo      help           p              struct         waitq
dev            ipcs           ps             swap           whatis
dis            irq            pte            sym            wr
epython        kmem           ptob           sys            xportshow
eval           list           ptov           task           q
exit           log            rd             taskinfo
crash64> crashinfo –v > crashinfo.txt
crash64> crashinfo | grep Dump
>---------------------| How This Dump Has Been Created |---------------------<
Dump has been initiated: with sysrq
```


First Pass Kdump Analysis

Command file -i myfirstpass

```
# cat myfirstpass
sys
sys -c
kmem -i
kmem -f
bt
bt -a
foreach bt | grep spin
ps | grep -v " IN "
runq | grep PID
mod
log
quit

# crash64 -s -i myfirstpass vmlinux-2.6.32-431.el6.x86_64 vmcore > my.txt
```

Have fun with my.txt !!!

First Pass Kdump Analysis

Remote crash

- Time consuming on two transfers on the same vmcore

```
Customer  ---> ftp.usa.hp.com ---> ERT
vmcore            vmcore          vmcore
```

- Install debuginfo rpm and crash-server on customer site
- Open one port, ie tcp/5111 port
- Run crash-client and look at the vmcore remotely
- No vmcore transfer

```
crash-server  <--- tcp/5111 ---> crash-client
```

Remote crash components

- Crash-server
- Crash-client – text version
- Crash-client – web version

First Pass Kdump Analysis

Crash-server

- To be run on the system where vmlinux and vmcore resides
- Reads input (commands) from and writes output (result) to a socket
- Listens on one port between 5111 and 5130
- 20 instances can be run on a single system
- Versions
  - crash-server32 (x86)
  - crash-server64 (x86_64)
  - crash-serveria64 (ia64)
- Produces a log file /tmp/wtec-log-username-port#

First Pass Kdump Analysis

Crash-server

```
# pwd
/var/crash/127.0.0.1-2014-07-07-12:40:40
# ll
total 152040
-rw-r--r-- 1 root root       108 Jul  7 16:30 myfirstpass
-rw------- 1 root root  21868390 Jul  7 12:40 vmcore
-rw-r--r-- 1 root root     88328 Jul  7 12:40 vmcore-dmesg.txt
-rw-r--r-- 1 root root 133715563 Jun 19 14:33 vmlinux-2.6.32-431.el6.x86_64

# crash-server64 -s vmlinux-2.6.32-431.el6.x86_64 vmcore
7ff6f000crash-server ready. Listening on port # 5111
All commands will be recorder on file /tmp/wtec-log-root-5111
The web client can be started by going to:
http://linuxdb.corp.hp.com/crash-client/main.php?remote_host=linuxbox.abc.com&port=5111
Processing sys ....Done
Processing bt ....Done
Processing extend /usr/local/lib/mpykdump64.so ....Done
Processing crashinfo  ....Done
```


First Pass Kdump Analysis

Crash-client – text version

- Communicates with server via a socket (tcp/5111 ~ tcp/5130)
- Basic and limited tty (text) app
- Loops prompting for crash commands
- Works for all regular crash commands
- Supports pipes | and redirections >
- Support readline and history

First Pass Kdump Analysis

Crash-client – text version

```
# crash-client64 linuxbox.abc.com 5111
Enter file name to log crash output: /tmp/abc.log
crash-client Version 1.0
To report bugs or enhancements please email
a detail report stating the problem and vmcore location to:
eddie.quinteros@hp.com

      KERNEL: vmlinux-2.6.32-431.el6.x86_64
    DUMPFILE: vmcore  [PARTIAL DUMP]
        CPUS: 2
        DATE: Mon Jul  7 12:40:37 2014
      UPTIME: 00:15:36
LOAD AVERAGE: 0.00, 0.00, 0.00
       TASKS: 212
    NODENAME: linuxbox.abc.com
     RELEASE: 2.6.32-431.el6.x86_64
     VERSION: #1 SMP Sun Nov 10 22:19:54 EST 2013
     MACHINE: x86_64  (2494 Mhz)
      MEMORY: 2 GB
       PANIC: "Oops: 0002 [#1] SMP " (check log for details)

crash-client> sys
crash-client> bt
crash-client> extend /usr/local/lib/mpykdump64.so
crash-client> crashinfo > crashinfo.txt
```

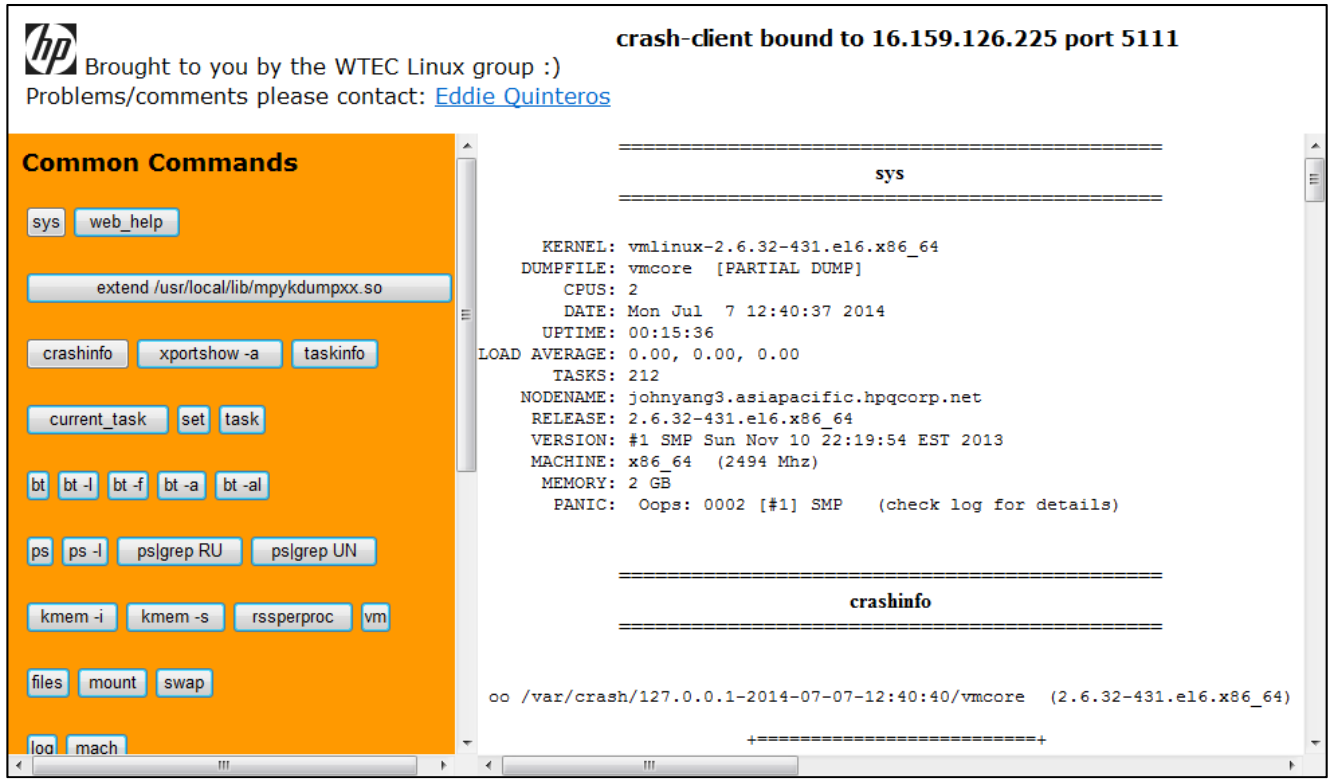

First Pass Kdump Analysis

Crash-client – web version

- More advanced than the text version
- Easier to read vmcore by producing links on most command
- Tied to linuxdb.corp.hp.com
- Links to functions and source code lines on linuxdb.corp.hp.com
- It can be expanded to include customer commands and links

First Pass Kdump Analysis

- http://linuxdb.corp.hp.com/crash-client/main.php?remote_host=16.159.126.225&port=5111



Labs

Exercises

- kdump setup and validation
- Enable the serial console
- vmcore analysis
  - crash64
  - crashinfo
  - crash-server
  - crash-client – text version
  - crash-client – web version

Labs (cont.)

RedHat 6.5 guest VM

- VMWare Workstation 9.0 virtual machine
- SELinux, Ipv4/ipv6 packet filter disabled
- Yum repository on ftp://127.0.0.1/dist/linux/RedHat/65Server/os/x86_64

SLES 11 SP3 guest VM

- VMWare Workstation 9.0 virtual machine
- SELinux, Ipv4/ipv6 packet filter disabled
- Zypper repository on ftp://127.0.0.1/dist/linux/SLES/sles11sp3

Tools

- ERT kernels source code Page - http://linuxdb.corp.hp.com/
- debuginfo repository on ftp://kczb.twn.hp.com/dist/debuginfo

AHA !!!

Learning Objectives Review - Take away

Care Why

- Effective action plan development for outage case
- With the vmcore on hand, do you feel more confidence on cause finding and action plan development ?

Know Why

- Direct evidence finding instead of best guess on action plan development
- Have the vmcore provide you the opportunities to find the direct evidences ?

Know How

- Build it and test it
- Can you make a linux box with kdump enabled ?

Know What

- What is kdump ?
- How does it work ?
- Have you a clean picture about how does kdump work?

QUINN, James Brian. "The intelligent enterprise a new paradigm. The Executive," 1992, 6.4: 48-63.

Next Chapter

Courses at Grow @hp

- Course ID: 00809694 - Crash Information Capture (1.5h)
- Course ID: 00809441 - Crash(1) Commands (1.5h)
- Course ID: 00809695 - Dump Analysis Fundamentals (1.5h)
- Course ID: 00809696 - Linuxdb and web-crash (1.5h)

The Kernel books

- *Linux Kernel Development, Robert Love* ISBN: 978-0672329463
- *Understanding the Linux Kernel, Daniel P. Bovet* ISBN: 978-0596005658
- *Linux Device Drivers, Jonathan Corbet* ISBN: 978-0596005900

Thank you