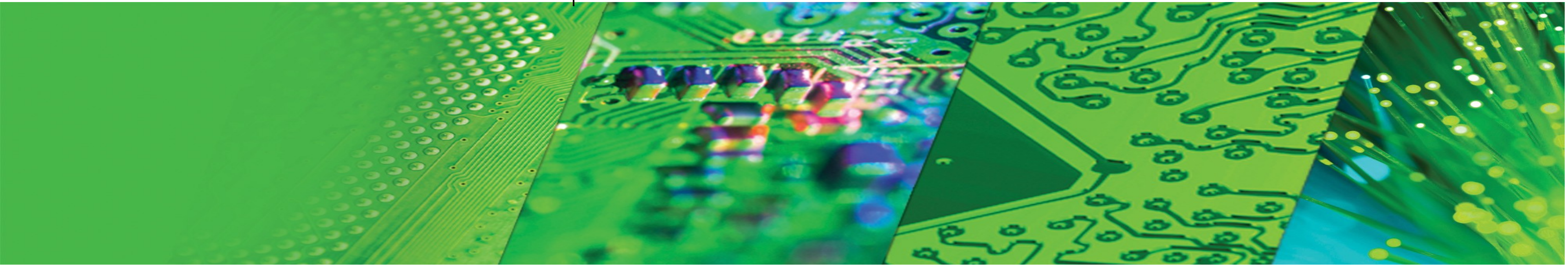
2012
IBM System z Technical University
Enabling the infrastructure for smarter computing

Linux on System z the Toolchain in a Nutshell

zLG10

Susanne Wintenberger



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

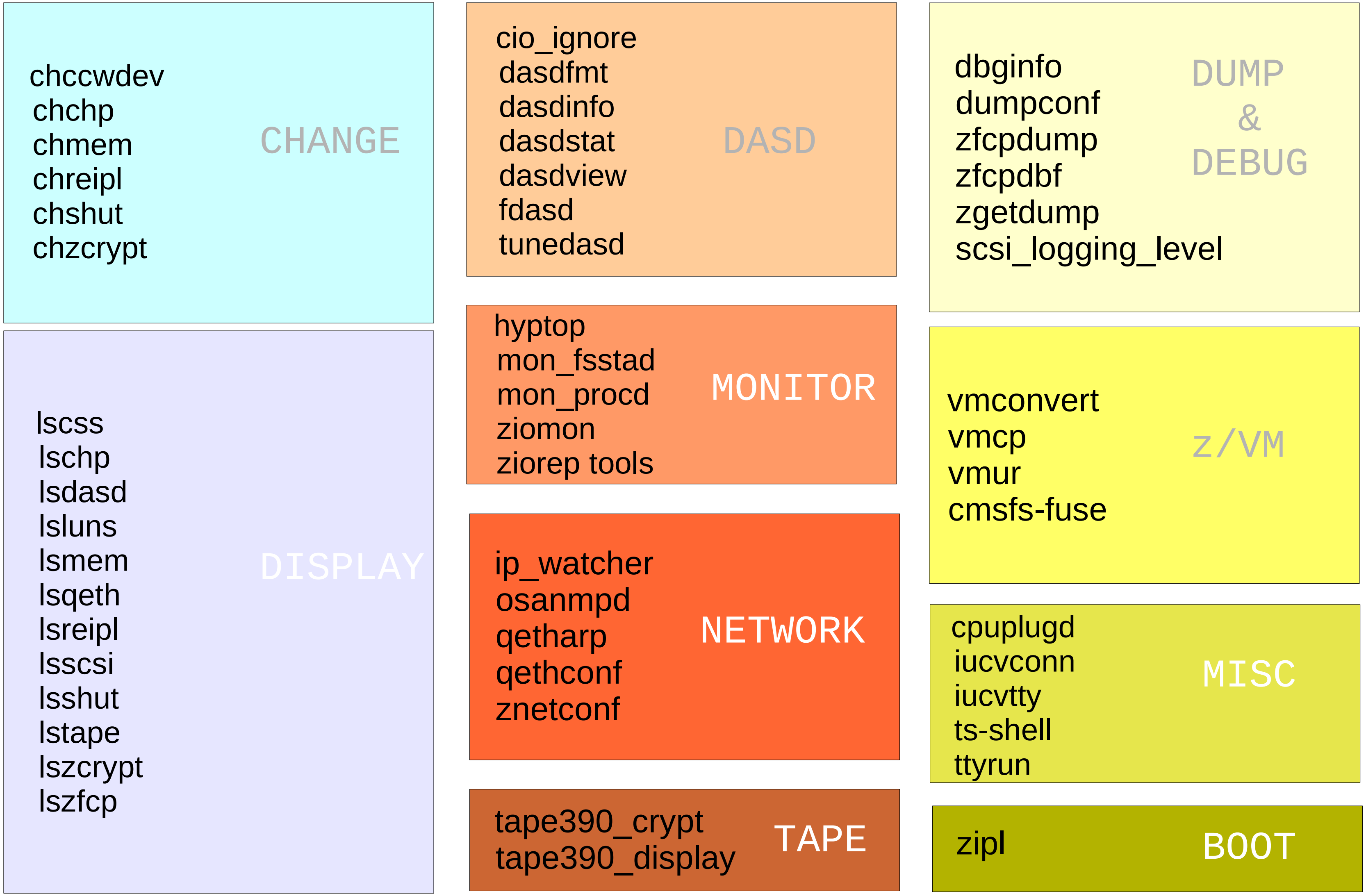
Agenda

- Introduction
  - What is the s390-tools package
  - Contained applications
- Selected Tools
  - dasdstat – DASD performance statistics
  - CMSFS user space file system support
  - hyptop – hypervisor utilization data
  - z/VM CP special messages
  - cpuplugd
  - lsmem/chmem – hotplug memory
  - zipl – integration of device mapper devices
  - makedumptool
  - kdump/kexec support – two stage dumper
  - shutdown action tools
  - IUCV terminal environment
  - cio_ignore
- Where to find more Information

What is the s390-tools package

- s390-tools is a package with a set of user space utilities to be used with the Linux on System z distributions.
- It is the essential tool chain for Linux on System z
  - It contains everything from the boot loader to dump related tools for a system crash analysis.
  - Version 1.16.0 was released in November 2011 and latest version is 1.17.0, released in February 2012
- This software package is contained in all major (and IBM supported) distributions which support s390
    - Red Hat Enterprise Linux 5 (s390-tools-1.8.1 since RHEL 5.4)
    - Red Hat Enterprise Linux 6 (s390-tools-1.8.2)
    - SUSE Linux Enterprise Server 10 (s390-tools-1.6.3 since SLES 10 SP2)
    - SUSE Linux Enterprise Server 11 SP2 (s390-tools-1.15.0)
- Website: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html
- Feedback: linux390@de.ibm.com

The content



CHANGE

chccwdev
chchp
chmem
chreipl
chshut
chzcrypt

DISPLAY

lscss
lschp
lsdasd
lsluns
lsmem
lsqeth
lsreipl
lsscsi
lsshut
lstape
lszcrypt
lszfcp

DASD

cio_ignore
dasdfmt
dasdinfo
dasdstat
dasdview
fdasd
tunedasd

MONITOR

hyptop
mon_fsstad
mon_procd
ziomon
ziorep tools

NETWORK

ip_watcher
osanmpd
qetharp
qethconf
znetconf

TAPE

tape390_crypt
tape390_display

DUMP & DEBUG

dbginfo
dumpconf
zfcpdump
zfcpdbf
zgetdump
scsi_logging_level

z/VM

vmconvert
vmcp
vmur
cmsfs-fuse

MISC

cpuplugd
iucvconn
iucvtty
ts-shell
ttyrun

BOOT

zipl

dasdstat

6.3

- Use the dasdstat command to display DASD performance statistics, including statistics about Parallel Access Volume (PAV) and High Performance Ficon.
- To start data gathering for summary of all available DASDs or for selected devices:

```
root@larsson:~>  dasdstat -e global
root@larsson:~>  dasdstat -e dasda 0.0.b223
```

- To stop data gathering for a single device

```
root@larsson:~>  dasdstat -d dasda
```

- To reset statistic counters for device b223

```
root@larsson:~>  dasdstat -r 0.0.b223
```

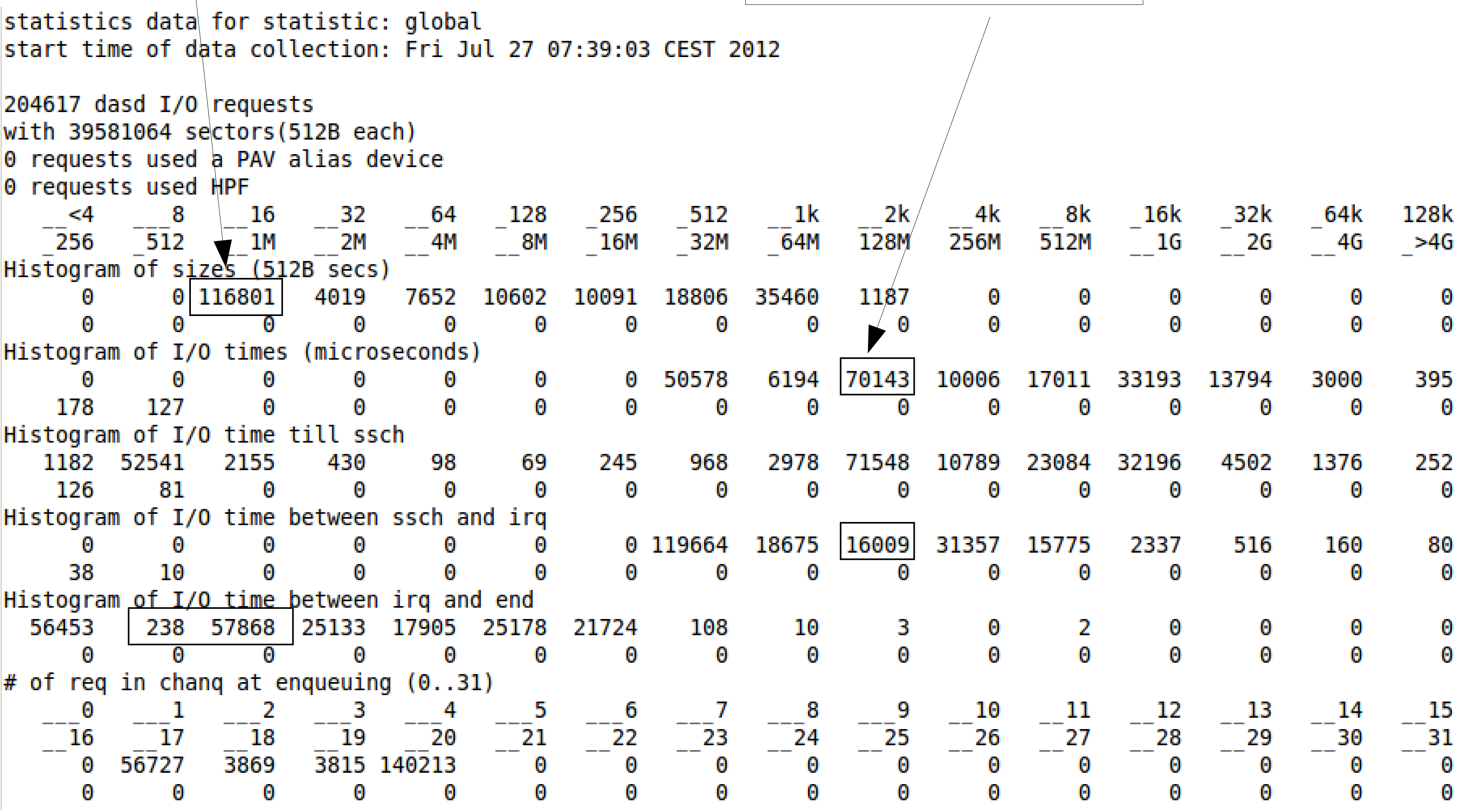
- To read data statistic for all devices and for a single device respectively

```
root@larsson:~>  dasdstat global
root@larsson:~>  dasdstat dasda
```

dasdstat (cont'd)

4 kb <= request size < 8 kb

1 ms <= response time < 2 ms

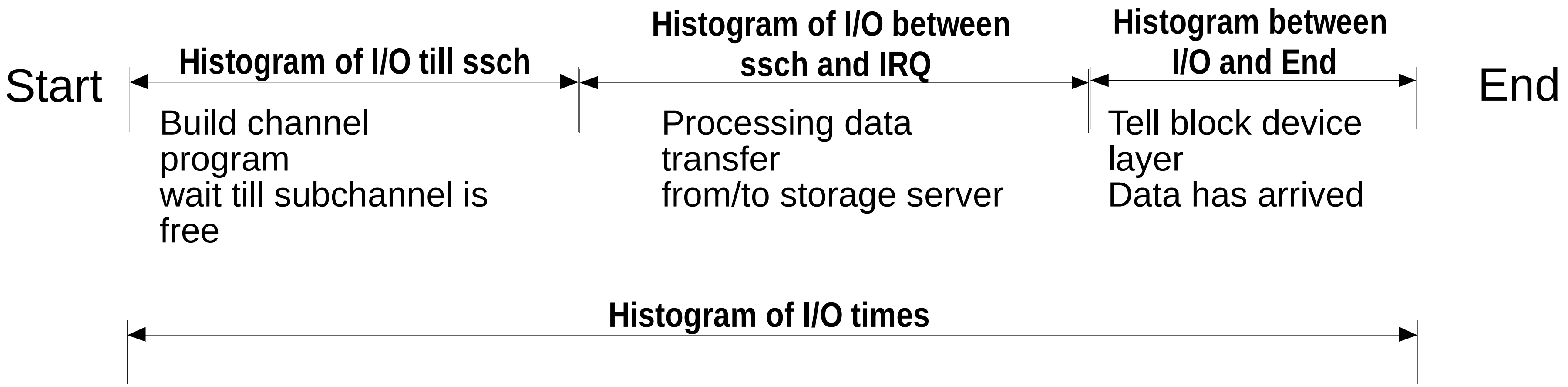


```
statistics data for statistic: global
start time of data collection: Fri Jul 27 07:39:03 CEST 2012

204617 dasd I/O requests
with 39581064 sectors(512B each)
0 requests used a PAV alias device
0 requests used HPF
   __<4   ___8   __16   __32   __64   _128   _256   _512   __1k   __2k   __4k   __8k   _16k   _32k   _64k   128k
   _256   _512   __1M   __2M   __4M   __8M   _16M   _32M   _64M   128M   256M   512M   __1G   __2G   __4G   _>4G
Histogram of sizes (512B secs)
      0      0 116801   4019   7652  10602  10091  18806  35460   1187      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Histogram of I/O times (microseconds)
      0      0      0      0      0      0      0  50578   6194  70143  10006  17011  33193  13794   3000    395
    178    127      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Histogram of I/O time till ssch
   1182  52541   2155    430     98     69    245    968   2978  71548  10789  23084  32196   4502   1376    252
    126     81      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Histogram of I/O time between ssch and irq
      0      0      0      0      0      0      0 119664  18675  16009  31357  15775   2337    516    160     80
     38     10      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Histogram of I/O time between irq and end
  56453    238  57868  25133  17905  25178  21724    108     10      3      0      2      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
# of req in chanq at enqueuing (0..31)
   ___0   ___1   ___2   ___3   ___4   ___5   ___6   ___7   ___8   ___9   __10   __11   __12   __13   __14   __15
   __16   __17   __18   __19   __20   __21   __22   __23   __24   __25   __26   __27   __28   __29   __30   __31
      0  56727   3869   3815 140213      0      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
```

dasdstat (cont'd)

- DASD statistics decomposition
  - Each line represents a histogram of times for a certain operation
  - Operations split up into the following :



Start

Histogram of I/O till ssch

Build channel program
wait till subchannel is free

Histogram of I/O between ssch and IRQ

Processing data transfer
from/to storage server

Histogram between I/O and End

Tell block device layer
Data has arrived

End

Histogram of I/O times

fdasd enhancements

11.2

- The option -c creates several partitions in non-interactive mode, controlled by the plain text configuration file <conf_file>. For example, the following configuration file creates three partitions:

```
[first,500]
[501,1100,swap]
[1101,last]
```

- Create a device with 3 partitions using the following command:

```
root@larsson:~> fdasd -c config /dev/dasde
parsing config file ’config’...
writing volume label...
writing VTOC...
rereading partition table...
```

- This creates partitions as followed:

```
Device start end length Id System
/dev/dasde1 2 500 499 1 Linux native
/dev/dasde2 501 1100 600 2 Linux swap
/dev/dasde3 1101 1499 399 3 Linux native
```

tunedasd enhancements

6.1

11.2

- This command is used to release and reserve a DASD and was extended by the option -Q
- The following sequence of commands first checks the reservation status of a DASD and then reserves it and releases it respectively:

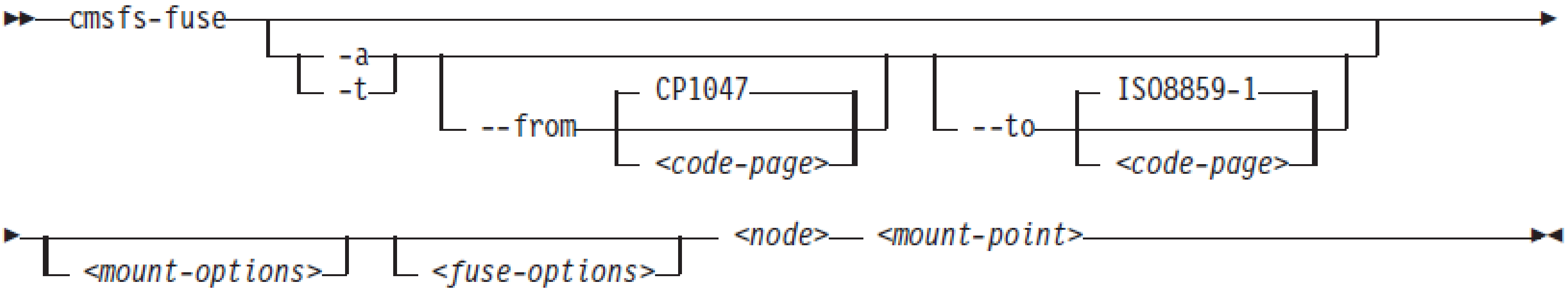
```
root@larsson:~>  tunedasd -Q /dev/dasdb
none
root@larsson:~>  tunedasd -S /dev/dasdb
Reserving device </dev/dasdb>...
Done.
root@larsson:~>  tunedasd -Q /dev/dasdb
reserved
root@larsson:~>  tunedasd -L /dev/dasdb
Releasing device </dev/dasdb>...
Done.
 ....  ....
root@larsson:~>  tunedasd -Q /dev/dasdb
Other
```

CMSFS user space file system support

6.1

11.2

- Allows to mount a z/VM minidisk to a Linux mount point
- z/VM minidisk needs to be in the enhanced disk format (EDF)
- The cmsfs fuse file system transparently integrates the files on the minidisk into the Linux VFS, no special command required



```
root@larsson:~> chccwdev -e 191
root@larsson:~> modprobe fuse
root@larsson:~> cmsfs-fuse /dev/dasdb /mnt/cms
root@larsson:~> ls -la /mnt/cmt/PROFILE.EXEC
-rw-rw----. 1 root root 1258 Dec 10 08:43 /mnt/cms/PROFILE.EXEC
```

CMSFS user space file system support (cont'd)

- By default no conversion is performed
  - Mount with '-t' to convert all text files from EBCDIC to ASCII based on the known text file types. cmsfs-fuse uses the configuration file `/etc/cmsfs-fuse/filetype.conf`

```
root@larsson:~> cmsfs-fuse -t /dev/dasdb /mnt/cms
```

  - To convert all files from EBCDIC to ASCII, mount with '-a'
- Use fusermount to unmount the file system again

```
root@larsson:~> fusermount -u /mnt/cms
```

- Be aware of the following restrictions:
  - Uppercase file names are enforced
  - Writing is only supported at the end of the file
  - Sparse files are not supported
  - The CMS file system does not support empty records. Cmsfs-fuse adds a space to records that consist of a linefeed character only.

CMSFS user space file system support (cont'd)

- Create a new empty file in the cmsfs-fuse mounted CMS filesystem

```
root@larsson:~> touch /mnt/cms/test.exec
```

- Extended attributes are used to handle the CMS characteristics of a file.
- To show the record format of the newly created file

```
root@larsson:~> getfattr -n user.record_format
/mnt/cms/TEST.EXEC
user.record_format="V"
```

- To set record length 80 for an empty fixed record format file, TEST.EXEC

```
root@larsson:~> setfattr -n user.record_format -v F
/mnt/cms/TEST.EXEC
user.record_format="F"
root@larsson:~> setfattr -n user.record_lrecl -v 72
/mnt/cms/TEST.EXEC
user.record_lrecl="72"
```

hyptop - hypervisor utilization data

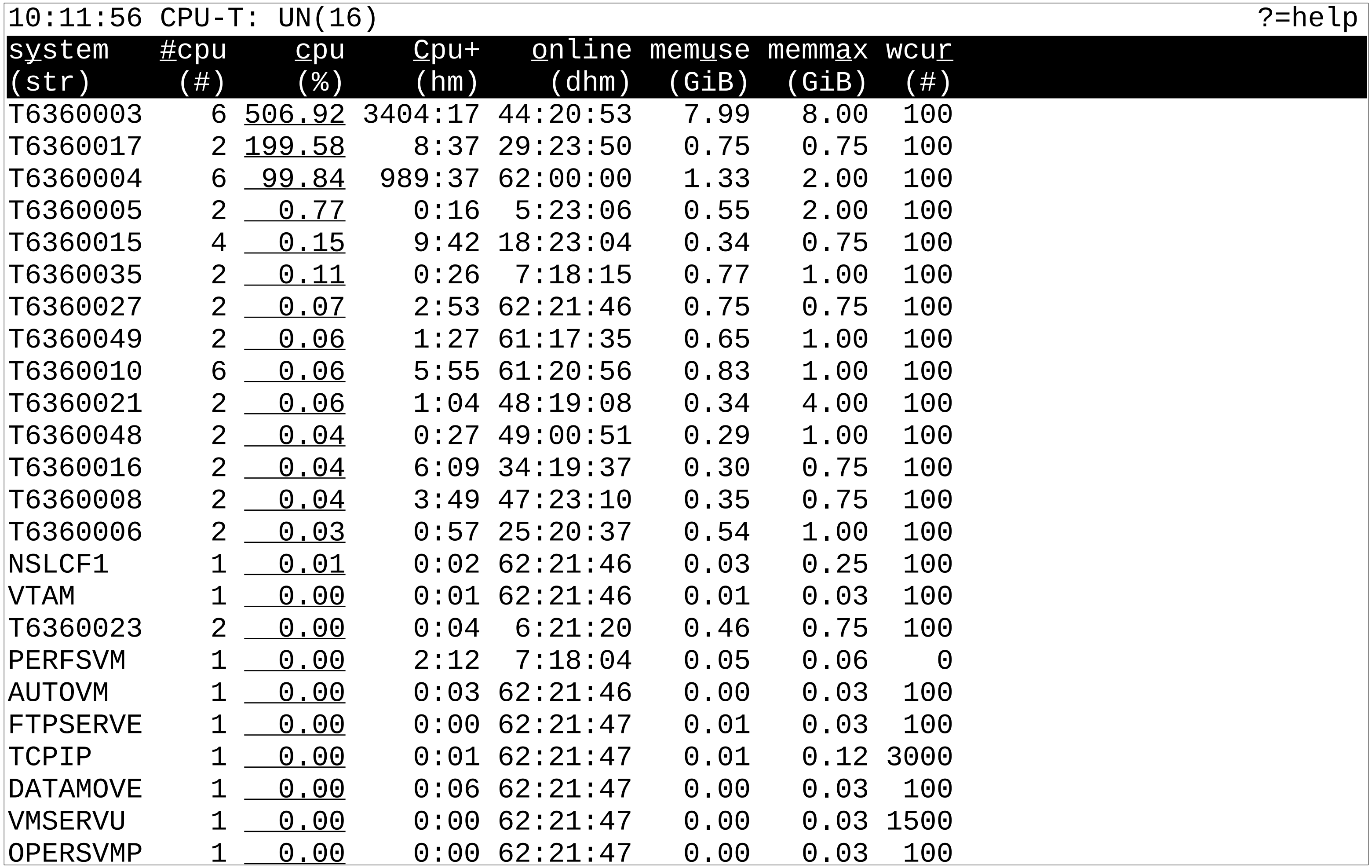
6.1

11.2

- The hyptop command provides a dynamic real-time view of a hypervisor environment on System z.
  - – It works with both the z/VM and the LPAR PR/SM hypervisor.
  - – Depending on the available data it shows, for example, CPU and memory information about
  - – running LPARs or z/VM guest operating systems.
- The following things are required to run hyptop:
  - – The debugfs file system must be mounted.
  - – The hyptop user must have read permission for the required debugfs files:
    - z/VM: <debugfs mount point>/s390_hypfs/diag_2fc
    - LPAR: <debugfs mount point>/s390_hypfs/diag_204
  - – To monitor all LPARs or z/VM guest operating systems of the hypervisor, your system must have additional permissions:
    - For z/VM: The guest must be privilege class B.
    - For LPAR: On the HMC or SE security menu of the LPAR activation profile, select the Global Performance Data Control checkbox.

hyptop - hypervisor utilization data (cont'd)

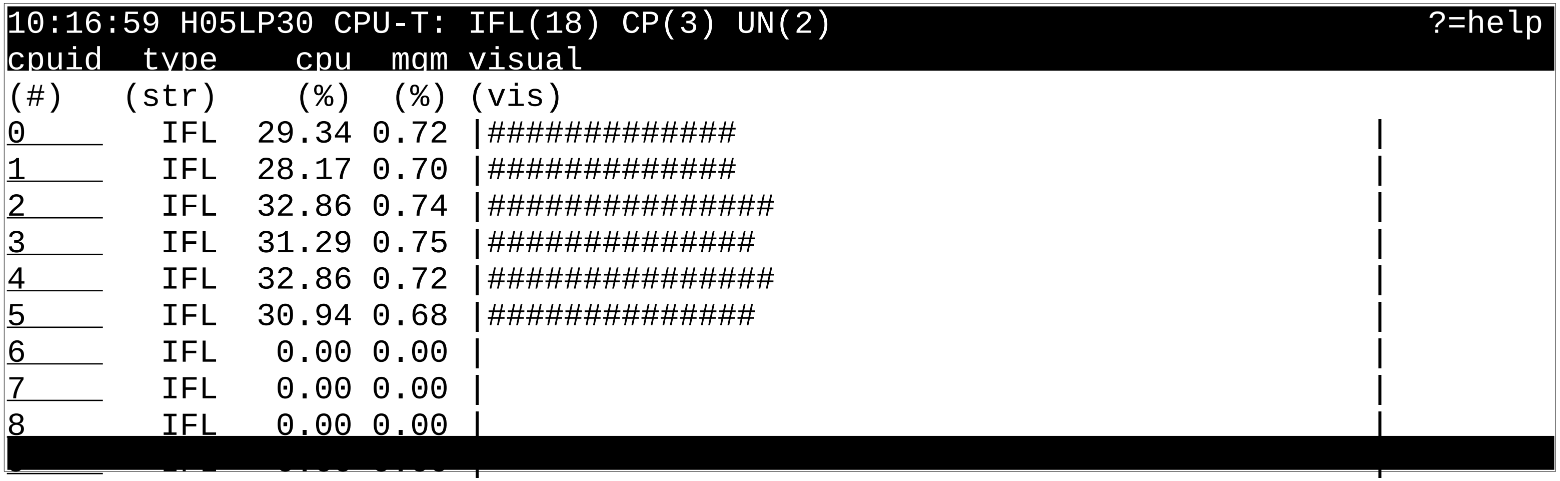
- Example of z/VM utilization data



```
10:11:56 CPU-T: UN(16)                                                    ?=help
system   #cpu    cpu    Cpu+   online memuse memmax wcur
(str)     (#)    (%)    (hm)    (dhm)  (GiB)  (GiB)  (#)
T6360003    6 506.92 3404:17 44:20:53   7.99   8.00  100
T6360017    2 199.58    8:37 29:23:50   0.75   0.75  100
T6360004    6  99.84  989:37 62:00:00   1.33   2.00  100
T6360005    2   0.77    0:16  5:23:06   0.55   2.00  100
T6360015    4   0.15    9:42 18:23:04   0.34   0.75  100
T6360035    2   0.11    0:26  7:18:15   0.77   1.00  100
T6360027    2   0.07    2:53 62:21:46   0.75   0.75  100
T6360049    2   0.06    1:27 61:17:35   0.65   1.00  100
T6360010    6   0.06    5:55 61:20:56   0.83   1.00  100
T6360021    2   0.06    1:04 48:19:08   0.34   4.00  100
T6360048    2   0.04    0:27 49:00:51   0.29   1.00  100
T6360016    2   0.04    6:09 34:19:37   0.30   0.75  100
T6360008    2   0.04    3:49 47:23:10   0.35   0.75  100
T6360006    2   0.03    0:57 25:20:37   0.54   1.00  100
NSLCF1      1   0.01    0:02 62:21:46   0.03   0.25  100
VTAM        1   0.00    0:01 62:21:46   0.01   0.03  100
T6360023    2   0.00    0:04  6:21:20   0.46   0.75  100
PERFSVM     1   0.00    2:12  7:18:04   0.05   0.06    0
AUTOVM      1   0.00    0:03 62:21:46   0.00   0.03  100
FTPSERVE    1   0.00    0:00 62:21:47   0.01   0.03  100
TCPIP       1   0.00    0:01 62:21:47   0.01   0.12 3000
DATAMOVE    1   0.00    0:06 62:21:47   0.00   0.03  100
VMSERVU     1   0.00    0:00 62:21:47   0.00   0.03 1500
OPERSVMP    1   0.00    0:00 62:21:47   0.00   0.03  100
```

hyptop - hypervisor utilization data (cont'd)

- Example of single LPAR utilization data



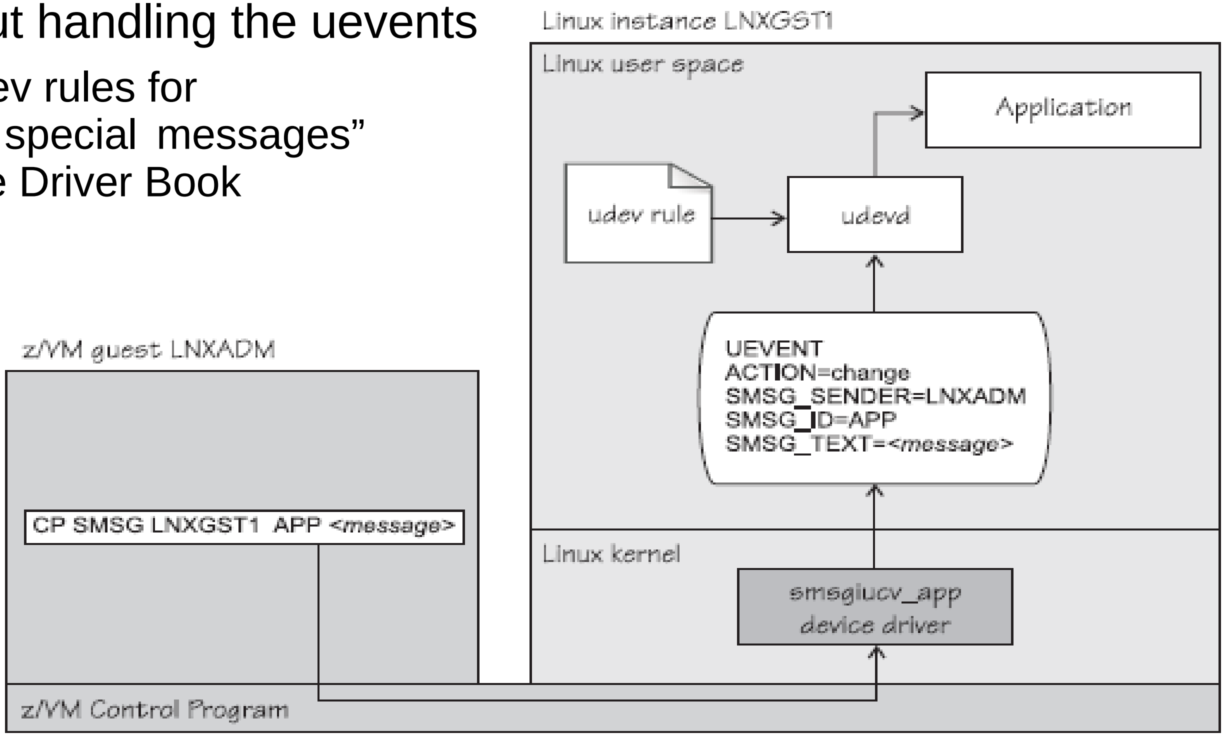
```
10:16:59 H05LP30 CPU-T: IFL(18) CP(3) UN(2)                               ?=help
cpuid  type    cpu  mgm visual
(#)   (str)    (%)  (%) (vis)
0       IFL  29.34 0.72 |#############                                 |
1       IFL  28.17 0.70 |#############                                 |
2       IFL  32.86 0.74 |###############                               |
3       IFL  31.29 0.75 |##############                                |
4       IFL  32.86 0.72 |###############                               |
5       IFL  30.94 0.68 |##############                                |
6       IFL   0.00 0.00 |                                              |
7       IFL   0.00 0.00 |                                              |
8       IFL   0.00 0.00 |                                              |
```

z/VM CP special messages

6.1

11.2

- Linux receives z/VM CP special messages (SMSG) and delivers these messages to user space as udev events (uevents)
  - – Allows to forward SMSG messages starting with “APP” to user space.
  - – udev rules can be used to trigger application specific actions
  - – The special messages cause uevents to be generated.
- More information about handling the uevents
  - – see “Writing udev rules for handling CP special messages” in the Device Driver Book



cpuplugd

5.3

10.2

- Use the cpuplugd command to:
  - Enable or disable CPUs based on a set of rules.
  - The rules can incorporate certain system load variables.
  - Manage memory under z/VM.

- Configuration file: */etc/sysconfig/cpuplugd*

- Init-Script: `'/etc/init.d/cpuplugd {start, stop, restart}'`

cpuplugd enhancements

6.2

11.2

- A set of rules used by the cpuplugd daemon has been improved, and cpuplugd now provides more advanced
  - control of the VM Resource Manager (VMRM) Cooperative Memory Management (CMM) memory balloon
  - adjustment of CPU consumption.
- The daemon now also provides a history function, which allows to retrieve data of up to 100 intervals of the past.
- Data from the `/proc/vmstat`, `/proc/meminfo`, `/proc/stat` and `/proc/loadavg` files can now be used in cpuplugd rules and in user-defined variables.
- A new cpustat.total_ticks variable has been introduced, which simplifies user-defined CPU percentage calculations.

cpuplugd improvements (cont'd)

- Sample configuration file

```
UPDATE="1"
CPU_MIN="1"
CPU_MAX="0"
.........
HOTPLUG="((1 - CP_ActiveAVG) * onumcpus) < 0.08"
HOTUNPLUG="(CP_idleAVG * onumcpus) > 1.15"

CMM_MIN="0"
CMM_MAX="131072"  # 512 MB

pgscan_d="vmstat.pgscan_direct_dma[0] + vmstat.pgscan_direct_normal[0] +
vmstat.pgscan_direct_movable[0]"
pgscan_d1="vmstat.pgscan_direct_dma[1] + vmstat.pgscan_direct_normal[1] +
vmstat.pgscan_direct_movable[1]"
pgscanrate="(pgscan_d - pgscan_d1) / (cpustat.total_ticks[0] - cpustat.total_ticks[1])"
avail_cache="meminfo.Cached - meminfo.Shmem"

CMM_INC="meminfo.MemFree / 40"
CMM_DEC="meminfo.MemTotal / 40"

MEMPLUG="pgscanrate > 20"
MEMUNPLUG="(meminfo.MemFree + avail_cache) > (meminfo.MemTotal / 10)"
```

http://www.ibm.com/developerworks/linux/linux390/perf/tuning_cpuhotplug.html

chmem

6.1

11.1

- Use chmem to set a particular size or a range of hotplug memory offline or online
  - – Setting memory online can fail if the hypervisor does not have enough memory left, for example because memory was overcommitted. Setting memory offline can fail if Linux cannot free the memory.
  - – If only part of the requested memory can be set online or offline, a message tells you how much memory was set online or offline instead of the requested amount
- The -e or –enable option sets memory online
  - – To set 2 GB of memory online

```
root@larsson:~>  chmem -e 2g
```

- The -d or –disable option removes memory
  - – To remove 256 MB (smallest memory block size is 256 MB)

```
root@larsson:~>  chmem --disable 256
```

  - – To remove a memory range of 1 GB (1024 MB)

```
root@larsson:~>  chmem -d 0x00000000b0000000-0x00000000efffffff
```

Configuring standby memory under z/VM

- To see how much central and expanded storage (memory) are installed and allocated to a system use the QUERY STORAGE and QUERY XSTOR commands. For example:

```
#CP Q STORAGE
 STORAGE = 50G CONFIGURED = 50G INC = 256M STANDBY = 0  RESERVED = 0
```

- Modify the directory entry by adding a COMMAND statement. This will give the virtual machine an additional 768 MB of standby memory:

```
USER LINUX01 LNX4VM 256M 2G G
INCLUDE LNXDFLT
COMMAND DEFINE STORAGE 256M STANDBY 768M
OPTION APPLMON
MDISK 100 3390 3339 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
MDISK 101 3390 6677 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
```

lsmem

6.1

11.1

- The lsmem command shows online status information about memory blocks and lists the ranges of available memory with their status.
  - – The listed memory blocks correspond to the memory block representation in sysfs.
  - – The command also shows the memory block size, the device size, and the amount of memory in online and offline state.
- The output of this command, shows ranges of adjacent memory blocks with similar attributes.

```
root@larsson:~>  lsmem
Address Range                          Size (MB)  State    Removable  Device
===============================================================================
0x0000000000000000-0x000000000fffffff        256  online   no         0-255
0x0000000010000000-0x000000006fffffff       1536  online   yes        256-1791
0x0000000070000000-0x000000008fffffff        512  online   no         1792-2303
0x0000000090000000-0x00000000afffffff        512  online   yes        2304-2815
0x00000000b0000000-0x00000000efffffff       1024  offline  -          2816-3839
0x00000000f0000000-0x00000000ffffffff        256  online   no         3840-4095
Memory device size  : 1 MB
Memory block size   : 256 MB
Total online memory : 3072 MB
Total offline memory: 1024 MB
```

zipl – integration of device mapper devices

6

11.1
10.4

- allows installation of and booting from a boot record on logical devices, i.e. devices managed by device mapper, e.g. multipath devices.
  - – physical device is of type DASD or SCSI
  - – all of the device which contains the boot directory must be located on a single physical device (which may be mirrored or accessed through a multipath target – single device-mapper target)
  - – only linear, mirror and multipath targets are supported
  - – access to block 0 of the physical device e.g. /dev/dasda
- zipl provides a helper script, zipl_helper.device-mapper, that detects the required information and provides it to zipl for you.
- To use the helper script run zipl as usual, specifying the parameters for the kernel image, parameter file, initial RAM disk, and target.

zipl – integration of device mapper devices (cont'd)

- Assuming an example device for which the location of the kernel image is /boot/image-5, the location of an initial RAM disk as /boot/initrd-5, a kernel parameter file /boot/parmfile-5, and which writes the required boot loader code to /boot and is a device mapper device, the command then becomes:

```
root@larsson:~>  zipl -i /boot/image-5 -r /boot/initrd-5 -p
/boot/parmfile-5 -t /boot
```

- The corresponding configuration file section becomes:

```
[boot5]
image=/boot/image-5
ramdisk=/boot/initrd-5
parmfile=/boot/parmfile-5
target=/boot
```

Makedumpfile tool

6.1

11.2

- Can be used to compress s390 dumps and exclude memory pages that are not needed for analysis e.g. user space pages, (file) cache pages, free pages, zero pages.
- This substantially reduces the size of dump files and the amount of time needed to transmit them from one location to another.
- Because makedumpfile expects as input dump files in the ELF format, you first have to transform your s390 format dump to ELF format.
- This is best done by mounting the dump using the zgetdump command. Mount the dump in ELF format by performing one of these steps:
  - create virtual elf dump in /mnt/dump.elf from dump partition /dev/dasdb1

```
root@larsson:~>  zgetdump -m -f elf /dev/dasdb1 /mnt
```

  - or from SCSI dump file dump.0

```
root@larsson:~>  zgetdump -m -f elf dump.0 /mnt
```

- Now the dump is available in the file /mnt/dump.elf

Makedumpfile tool (cont'd)

- In order to use the makedumpfile you need the vmlinux file that contains necessary debug information
- Extract the vmlinux debug file from the for your kernel version xyz
  - SLES 11

```
root@larsson:~>  rpm -qlp kernel-default-debuginfo-xyz.rpm | grep vmlinux
/usr/lib/debug/boot/vmlinux-xyz-default.debug
root@larsson:~>  rpm2cpio kernel-default-debuginfo-xyz.rpm | cpio -idv
*vmlinux*
./usr/lib/debug/boot/vmlinux-xyz-default.debug
1224646 blocks
```

  - RHEL 6

```
root@larsson:~>  rpm -qlp kernel-debuginfo-xyz.rpm | grep vmlinux
/usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
root@larsson:~>  rpm2cpio kernel-debuginfo-xyz.rpm | cpio -idv *vmlinux*
./usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
1082264 blocks
```

Makedumpfile tool (cont'd)

- Use the makedumpfile tool to compress the dump and exclude pages, that are typically not needed
  - Use **-d <dump_level>** to indicate which pages are excluded
  - Use **-c** to compress the dump

```
root@larsson:~>  makedumpfile -c -d 31 -x
usr/lib/debug/lib/modules/2.6.32-
131.0.15.el6.s390x/vmlinux  /mnt/dump.elf dump.kdump
Copying data                       : [100 %]
The dumpfile is saved to dump.kdump.
makedumpfile Completed.
```

- For initial problem analysis, extract kernel log and send it to your service organization

```
root@larsson:~>  makedumpfile --dump-dmesg -x
usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
/mnt/dump.elf kernel.log
```

- unmount elf dump

```
root@larsson:~>  zgetdump -u /mnt
```

kdump/kexec support – two stage dumper

6.3

- Two stage dumper / kdump/kexec support
  - Enhanced dump support that is able to
    - reduce dump size and dump time by using page filtering
    - share dump disk space using network dump
    - make dump setup easier
  - Integrated into the System z stand-alone dump tools and shutdown actions framework
- kdump/kexec kernel dumping mechanism is enabled as a Technology Preview
- Any system with more than 4 GB of memory has the kexec/kdump mechanism enabled.
  - To disable kexec/kdump remove the kernel parameter `'crashkernel='`
- Limitations:
  - kdump is limited to DASD or QETH networks as dump devices until kdump on SCSI disk is supported.
  - kdump cannot dump a z/VM named saved system (NSS)
  - For systems running in LPAR kdump consumes memory

kdump/kexec support – two stage dumper (cont'd)

- Use a preloaded Linux crashkernel to create a system dump
- Reserve a memory area
  - Add a `'crashkernel='` parameter to the kernel parameter in */etc/zipl.conf* to reserve a memory area
  - A crashkernel memory size of 64 - 128 MiB is sufficient

```
crashkernel=<size>@<offset>
```

- Boot your system and check the reservation

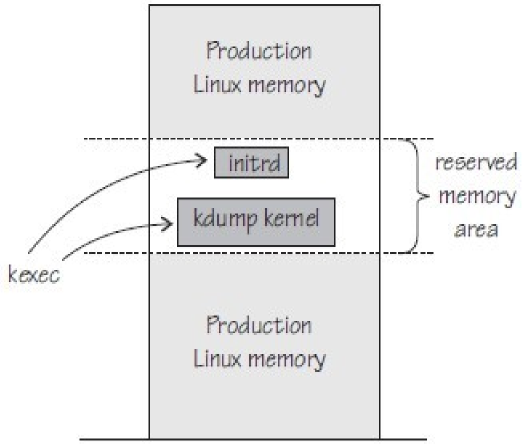
```
root@larsson:~>  cat /proc/iomem
00000000-3fffffff : System RAM
  00000000-005f1143 : Kernel code
  005f1144-00966497 : Kernel data
  00b66000-014c4e9f : Kernel bss
40000000-47ffffff : Crash kernel
48000000-7fffffff : System RAM
```

```
root@larsson:~>  cat /proc/cmdline
dasd=eb90 root=/dev/dasda1  crashkernel=256M@256M
```

kdump/kexec support – two stage dumper (cont'd)

- Load the kdump crashkernel with kexec

```
root@larsson:~>  kexec -p kdump.image --initrd
kdump.initrd
--command-line=”dasd=1234 root=/dev/ram0”
Cannot open `--command-line=dasd=1234
root=/dev/dasda1 maxcpus=1': No such file or
directory
```

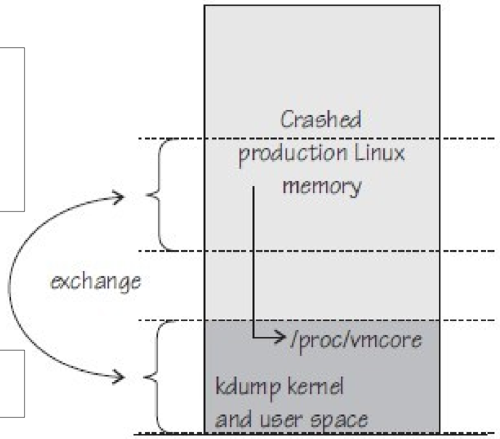


- Verify that a kdump crashkernel is loaded

```
root@larsson:~>  cat
/sys/kernel/kexec_crash_loaded
1
```

- Manually trigger for kdump under z/VM

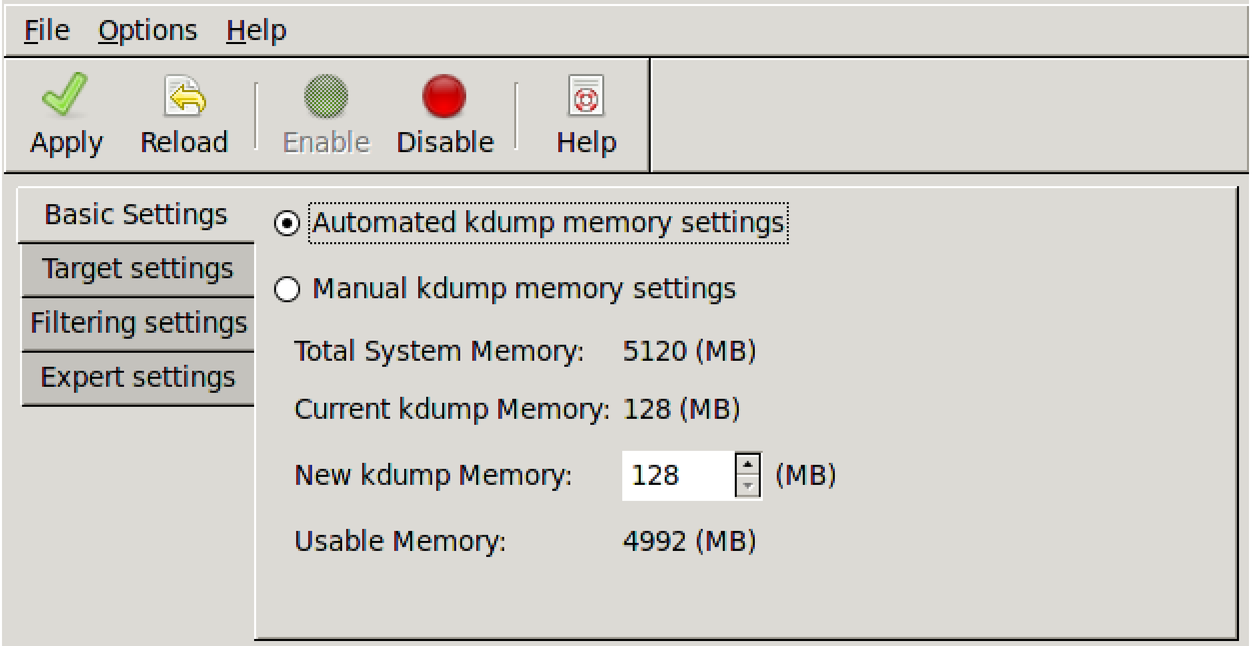
```
#cp system restart
```



- The dump is written to */var/crash/<date>/vmcore*

kdump/kexec support – two stage dumper (cont'd)

- Configure kdump using `system-config-kdump`
  - – Configuration changes are saved in `/etc/kdump.conf`



● Automated kdump memory settings

○ Manual kdump memory settings

Total System Memory: 5120 (MB)

Current kdump Memory: 128 (MB)

New kdump Memory: 128 (MB)

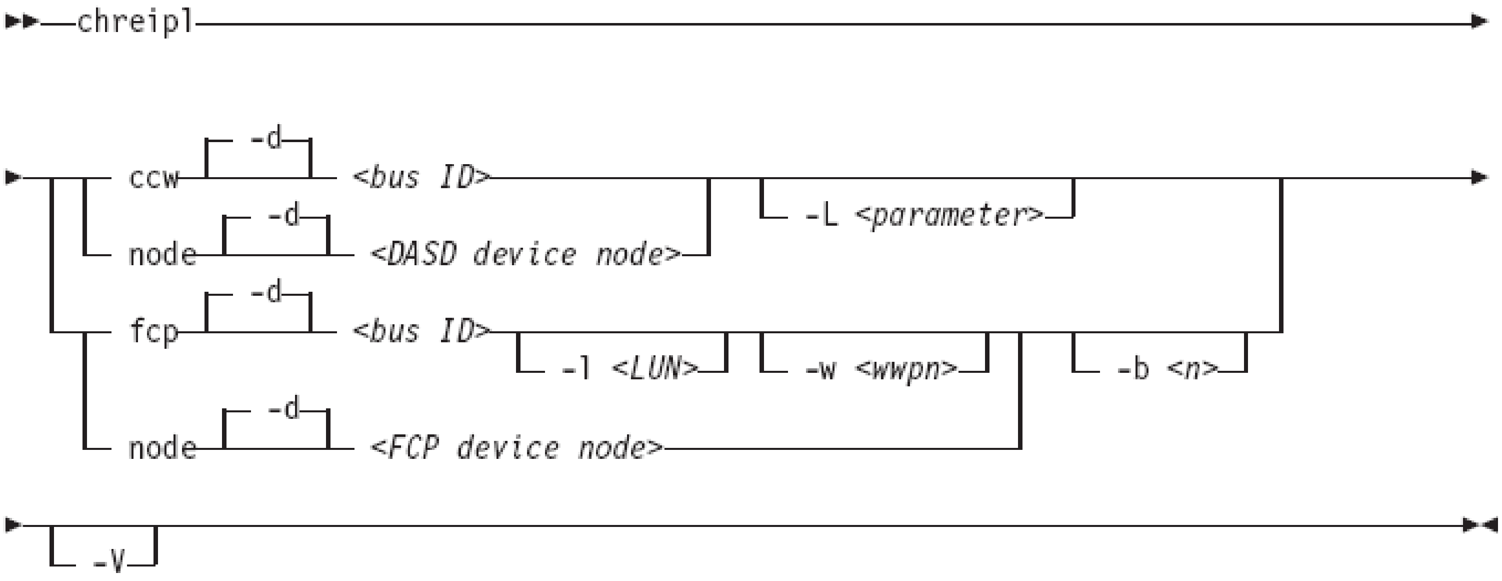
Usable Memory: 4992 (MB)

Shutdown actions tools

5.4

11

chreipl: Configure a disk as well as select an optional entry in the boot menu for the next boot cycle.



```
root@larsson:~> chreipl node /dev/dasda
root@larsson:~> chreipl ccw -d 0.0.7e78 -L 1
root@larsson:~> chreipl fcp --wwpn 0x500507630300c562 \
--lun 0x401040B300000000 -d 0.0.1700
```

chreipl: various enhancements

6.2

11.2

- Add support to re-IPL from named saved systems (NSS)
  - To configure an NSS LINUX1 as the re-IPL device:

```
root@larsson:~> chreipl nss LINUX1
```

- Add support to specify additional kernel parameters for re-IPL

```
root@larsson:~> chreipl 0.0.7e78 -p "mem=512M"
Re-IPL type: ccw
Device: 0.0.7e78
Loadparm: ""
Bootparms: "mem=512M"
```

- Add support to re-IPL from device-mapper multipath devices
  - In the current version <node> can be used to speciy a device node of a DASD, SCSI, or *logical device mapper* re-IPL device.

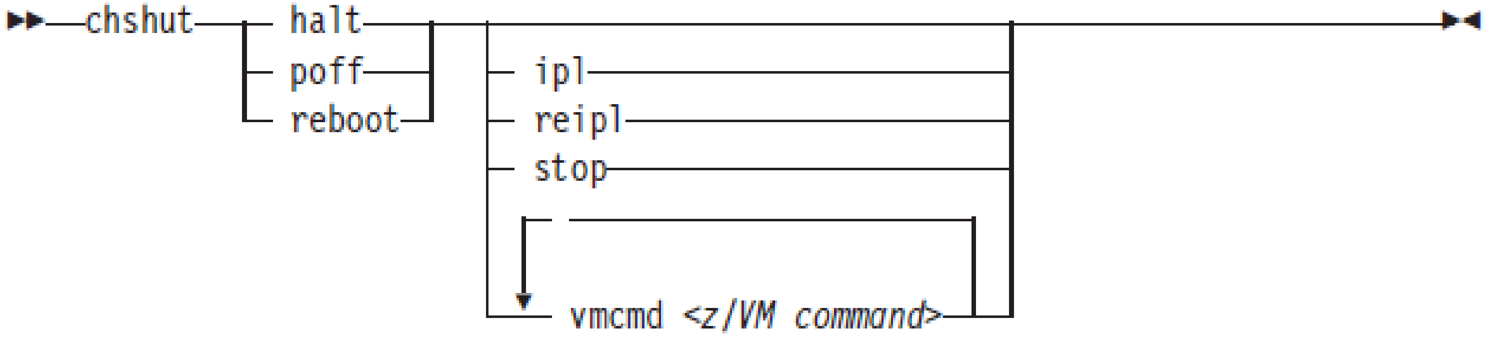
```
root@larsson:~> chreipl node /dev/disk/by-id/scsi-
36005076303ffc56200000000000010b4
```

Shutdown action tools (cont'd)

5.4

11

chshut: controls the system behavior on shutdown actions



```
root@larsson:~> chshut halt ipl
root@larsson:~> chshut halt vmcmd LOGOFF
root@larsson:~> chshut reboot reipl
root@larsson:~> chshut poff vmcmd "MSG MASTER Going down" \
vmcmd "LOGOFF"
```

Shutdown action tools (cont'd)

6.3

lsreipl: command to see from which device your system will boot after you issue the reboot command.

```
root@larsson:~> lsreipl
Re-IPL type:   ccw
Device:        0.0.4bb8
Loadparm:
Bootparms:
```

lsshut: command to see what the system should do in one of the following states.

```
root@larsson:~> lsshut
Trigger          Action
========================
Halt              stop
Power off         vmcmd (LOGOFF)
Reboot            reipl
Restart           kdump,dump_reipl
Panic             kdump,dump_reipl
```

The triggers Restart and Panic are configured using the dumpconf tool.

Automatic dump on panic: dumpconf

5.4

10.1

- The dumpconf tool configures a dump device that is used for automatic dump in case of a kernel panic.
  - The command can be installed as service script under */etc/init.d/dumpconf* or can be called manually.
  - Start service: `# service dumpconf start`
  - It reads the configuration file */etc/sysconfig/dumpconf.*
  - Example configuration for CCW dump device (DASD) and reipl after dump:

```
ON_PANIC=dump_reipl
DUMP_TYPE=ccw
DEVICE=0.0.4711
DELAY_MINUTES=5
```

Automatic dump on panic: dumpconf (cont'd)

5.4

10.1

– Example configuration for FCP dump device (SCSI disk):

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4714
WWPN=0x5005076303004712
LUN=0x4047401300000000
BOOTPROG=0
BR_LBA=0
```

– Example configuration for re-IPL without taking a dump, if a kernel panic occurs:

```
ON_PANIC=reipl
```

– Example of executing a CP command, and rebooting from device 4711 if a kernel panic occurs:

```
ON_PANIC=vmcmd
VMCMD_1="MSG <vmguest> Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```

znetconf

6

11.1

- Allows to list, add, remove & configure System z network devices
- For example: list all potential network devices:

```
root@larsson:~>  znetconf -u
Device Ids                 Type    Card Type   CHPID Drv.
--------------------------------------------------------
0.0.f500,0.0.f501,0.0.f502 1731/01 OSA (QDIO) 00    qeth
0.0.f503,0.0.f504,0.0.f505 1731/01 OSA (QDIO) 01    qeth
```

- To configure the potential network device 0.0.f503 with the layer2 option with the value 0 and the portnumber option with the value 1:

```
root@larsson:~>  znetconf -a f503 -o layer2=0 -o portno=1
```

- To remove network device 0.0.f503

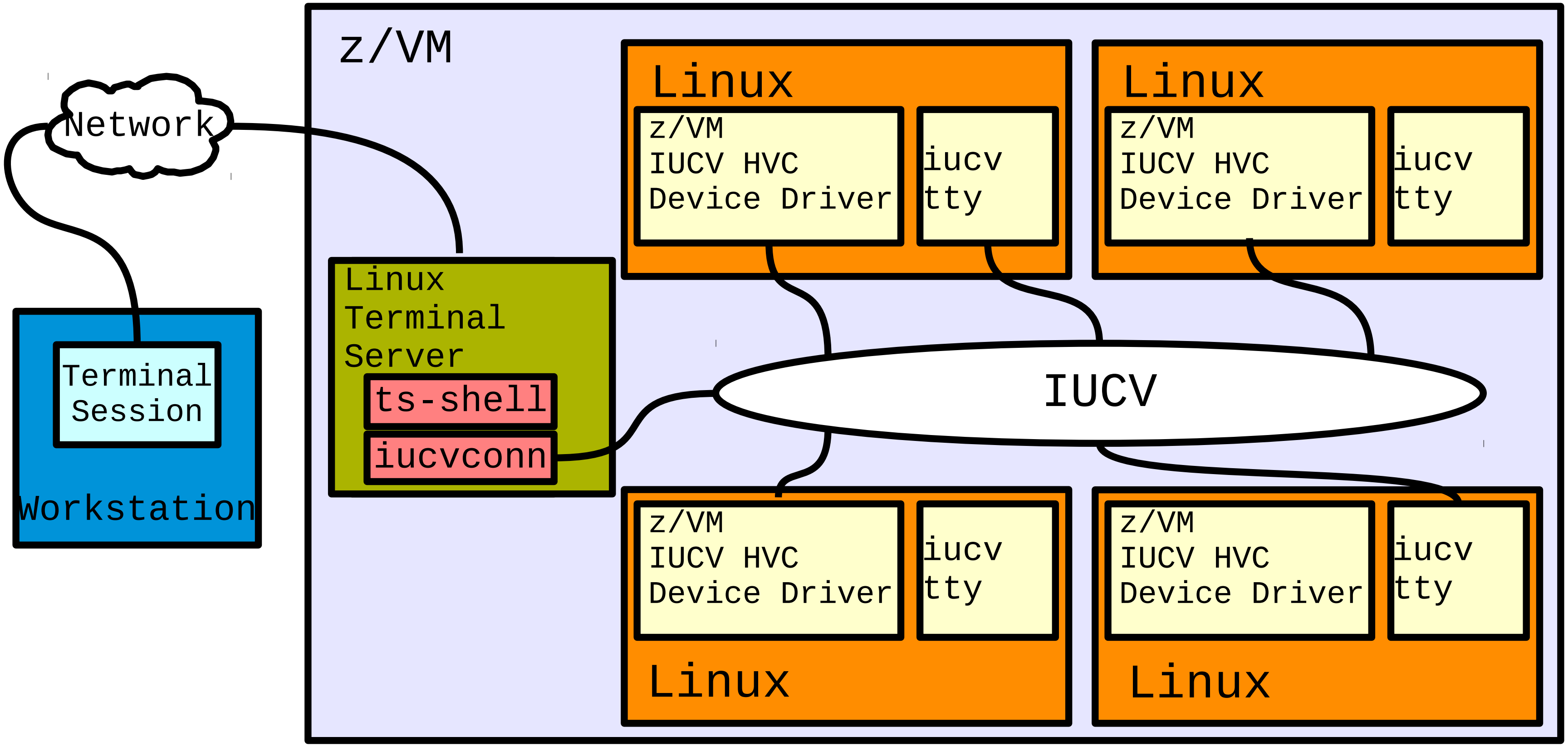
```
root@larsson:~>  znetconf -r 0.0.f503
```

IUCV terminal environment

5.4

10.3

- Full-screen terminal access to Linux guest operating systems on the same z/VM
- Access Linux instances with no external network because IUCV is independent from TCP/IP



IUCV terminal environment (cont'd)

- The IUCV terminal applications consist of:
  - **iucvconn** – Start terminal connection over IUCV
  - **ts-shell** – Login shell for setting up a terminal server over IUCV
  - **iucvtty** – Allow remote logins over IUCV
- Terminal access over IUCV is provided by:
  - **iucvtty**, or
  - z/VM IUCV hypervisor console device driver (Linux kernel)
- For more details, see
  - How to setup a Terminal Server Environment on z/VM (SC34-2596-00)

IUCV terminal environment (cont'd)

- Establishing terminal connections with iucvconn
  - To access the „lxterm1“ terminal on the Linux instance in z/VM guest lnxsys02

```
root@larsson: ssh brueckner@lnxsys01
Password:
brueckner@lnxsys01: iucvconn lnxsys02 lxterm1
login: brueckner
Password:
[brueckner@lnxsys02 ~] ps
  PID TTY          TIME CMD
 1731 pts/0    00:00:00 bash
 1762 pts/0    00:00:00 ps
[brueckner@lnxsys02 ~]
```

  - To access the first z/VM IUCV HVC terminal device on the Linux instance in z/VM guest lnxsys02

```
brueckner@lnxsys01: iucvconn lnxsys02 lnxhvc0
```

  - To create a transcript of the terminal session to the Linux instance in z/VM guest lnxsys99

```
brueckner@lnxsys01: iucvconn -s ~/transcripts/lnxsys99
lnxsys99 lnxhvc0
```

IUCV terminal environment (cont'd)

- Authorizing the z/VM guest virtual machine for IUCV
  - Adding an IUCV user directory statement, for example, IUCV ANY
  - The z/VM user directory for a terminal server might look like:

```
USER LNXTS XSECRETX 768M 1G G
* General statements
IPL 0150
MACH ESA 8
* IUCV authorization
IUCV ANY
OPTION MAXCONN 128
* Generic device statements
CONSOLE 0009 3215 T
SPOOL 000C 2540 READER *
* ...
```

IUCV terminal environment

- Full-screen terminal access to Linux instances on the same z/VM
- Access to Linux instances that are not connected to an Internet Protocol (IP)network
- Use cases
  - Provide an alternative terminal access to 3270 and 3215 line-mode terminals
  - Increase availability by providing emergency access if the network for a system fails
  - Centralize access to systems by providing a terminal server environment
  - Heighten security by separating user networks from administrator networks or by isolating sensitive Linux instances from public IP networks

cio_ignore

- When a Linux on System z instance boots, it senses and analyses all available devices.
- You can use the cio_ignore kernel parameter to specify a list of devices that are to be ignored. The following applies to ignored devices:
  - Ignored devices are not sensed and analyzed. The device cannot be used unless it has been analyzed.
  - Ignored devices are not represented in sysfs.
  - Ignored devices do not occupy storage in the kernel.
  - The subchannel to which an ignored device is attached is treated as if no device were attached.
  - cio_ignore might hide essential devices such as the console under z/VM. The console is typically device number 0.0.0009.
- This example specifies that all devices in the range 0.0.b100 through 0.0.b1ff, and the device 0.0.a100 are to be ignored during boot (kernel parameter option)

```
cio_ignore=0.0.b100-0.0.b1ff,0.0.a100
```

cio_ignore (cont'd)

- Use cio_ignore to manage the I/O device exclusion list
- Use the -l option to display the current exclusion list

```
root@larsson:~>  cio_ignore -l
Ignored devices:
=================
0.0.0000-0.0.0008
0.0.000a-0.0.6365
0.0.6367-0.0.f5ef
0.0.f5f3-0.0.ffff
```

- Use the -L option to display the devices which are accessible

```
root@larsson:~>  cio_ignore -L
Accessible devices:
===================
0.0.0009
0.0.6366
0.0.f5f0-0.0.f5f2
```

cio_ignore (cont'd)

- Use the -r option to remove devices from the exclusion list

```
root@larsson:~>  cio_ignore -r 6366
```

- The the -R option is used to free all devices
- Use the -a option to add devices to the exclusion list

```
root@larsson:~>  cio_ignore -a 4000-5fff
```

- Use the -k option to create the kernel parameter list string

```
root@larsson:~>  cio_ignore -k
cio_ignore=all,!0009,!6366,!f5f0-f5f2
```

- A new option -i / --is-ignored has been added to determine if a device with a given ID is on the blacklist

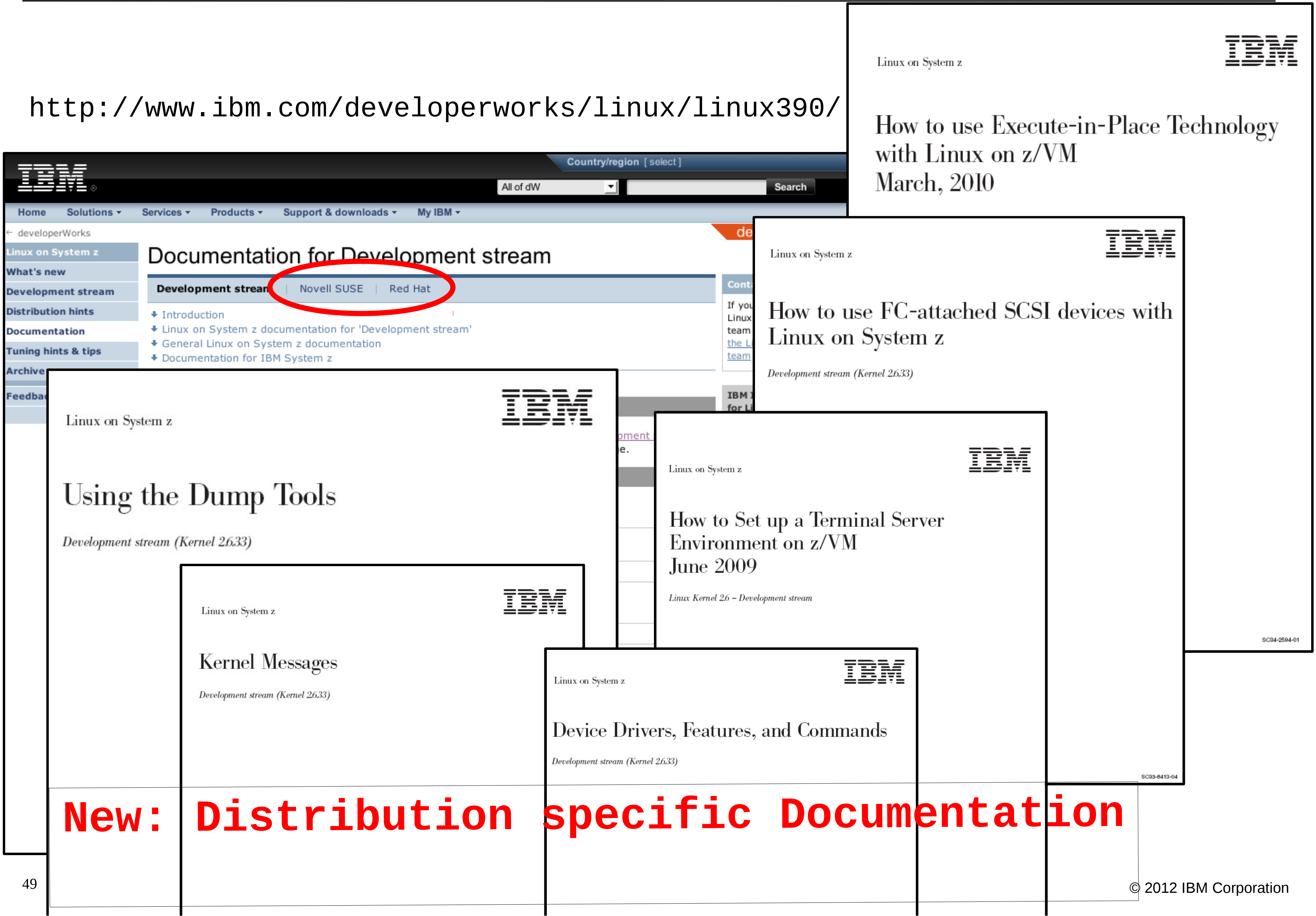
6.1

11.2

```
root@larsson:~> cio_ignore --is-ignored 0.0.0190
Device 0.0.0190 is ignored
root@larsson:~> cio_ignore -i 0.0.0009
Device 0.0.0009 is not ignored
```

More information

http://www.ibm.com/developerworks/linux/linux390/



Documentation for Development stream

- Introduction
- Linux on System z documentation for 'Development stream'
- General Linux on System z documentation
- Documentation for IBM System z

Linux on System z

How to use Execute-in-Place Technology with Linux on z/VM
March, 2010

SC34-2594-01

Linux on System z

How to use FC-attached SCSI devices with Linux on System z

Development stream (Kernel 2.6.33)

SC33-8413-04

Linux on System z

Using the Dump Tools

Development stream (Kernel 2.6.33)

Linux on System z

How to Set up a Terminal Server Environment on z/VM
June 2009

Linux Kernel 2.6 – Development stream

Linux on System z

Kernel Messages

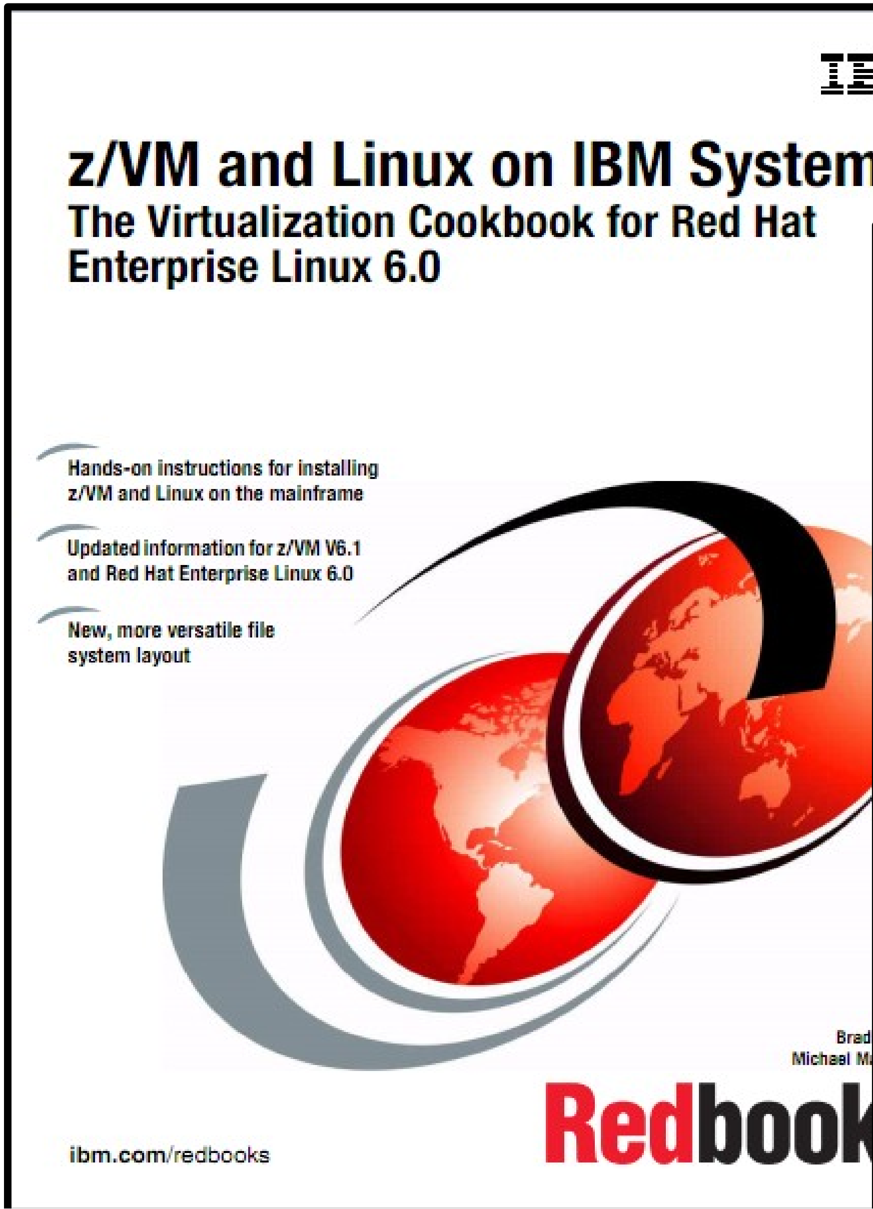
Development stream (Kernel 2.6.33)

Linux on System z

Device Drivers, Features, and Commands

Development stream (Kernel 2.6.33)

New: Distribution specific Documentation



z/VM and Linux on IBM System z

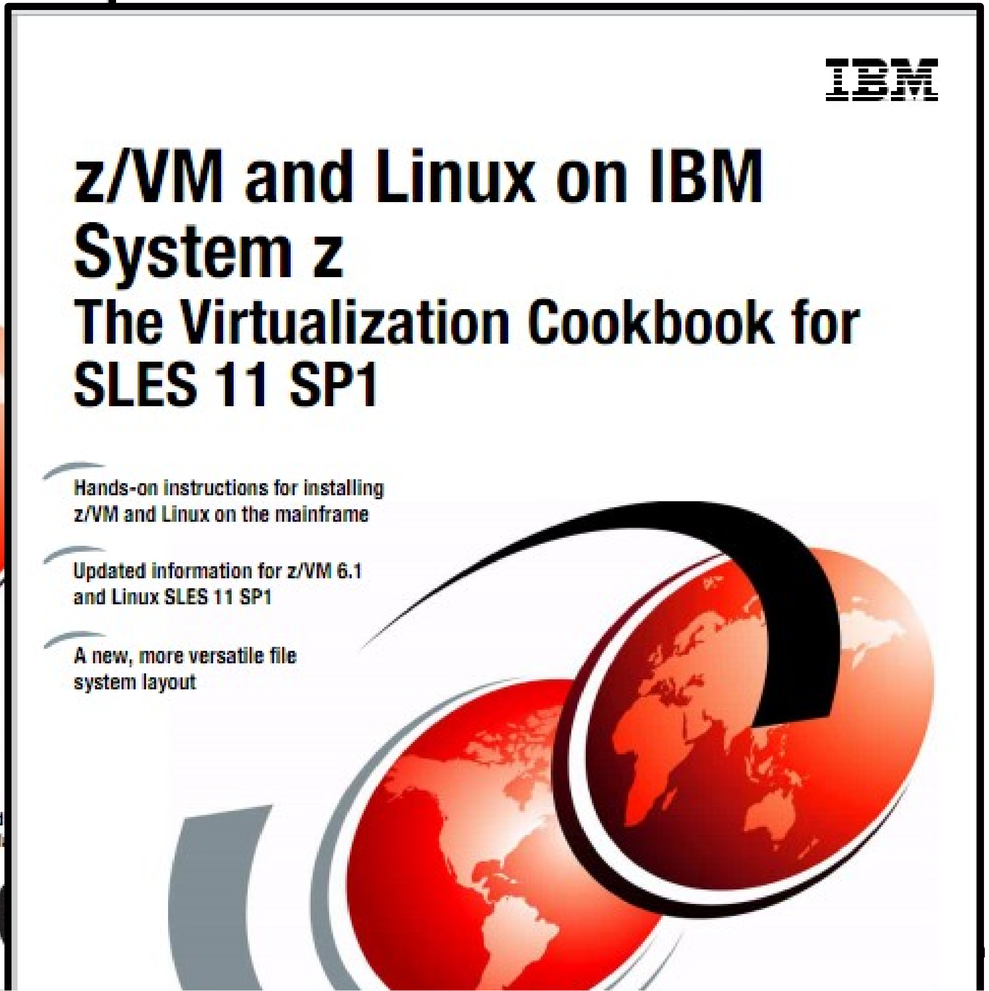
The Virtualization Cookbook for Red Hat Enterprise Linux 6.0

- Hands-on instructions for installing z/VM and Linux on the mainframe
- Updated information for z/VM V6.1 and Red Hat Enterprise Linux 6.0
- New, more versatile file system layout

Brad
Michael M

Redbook

ibm.com/redbooks



z/VM and Linux on IBM System z

The Virtualization Cookbook for SLES 11 SP1

- Hands-on instructions for installing z/VM and Linux on the mainframe
- Updated information for z/VM 6.1 and Linux SLES 11 SP1
- A new, more versatile file system layout

Questions?

Susanne Wintenberger

Certified IT Specialist

Linux on System z

Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-3514
swinten@de.ibm.com