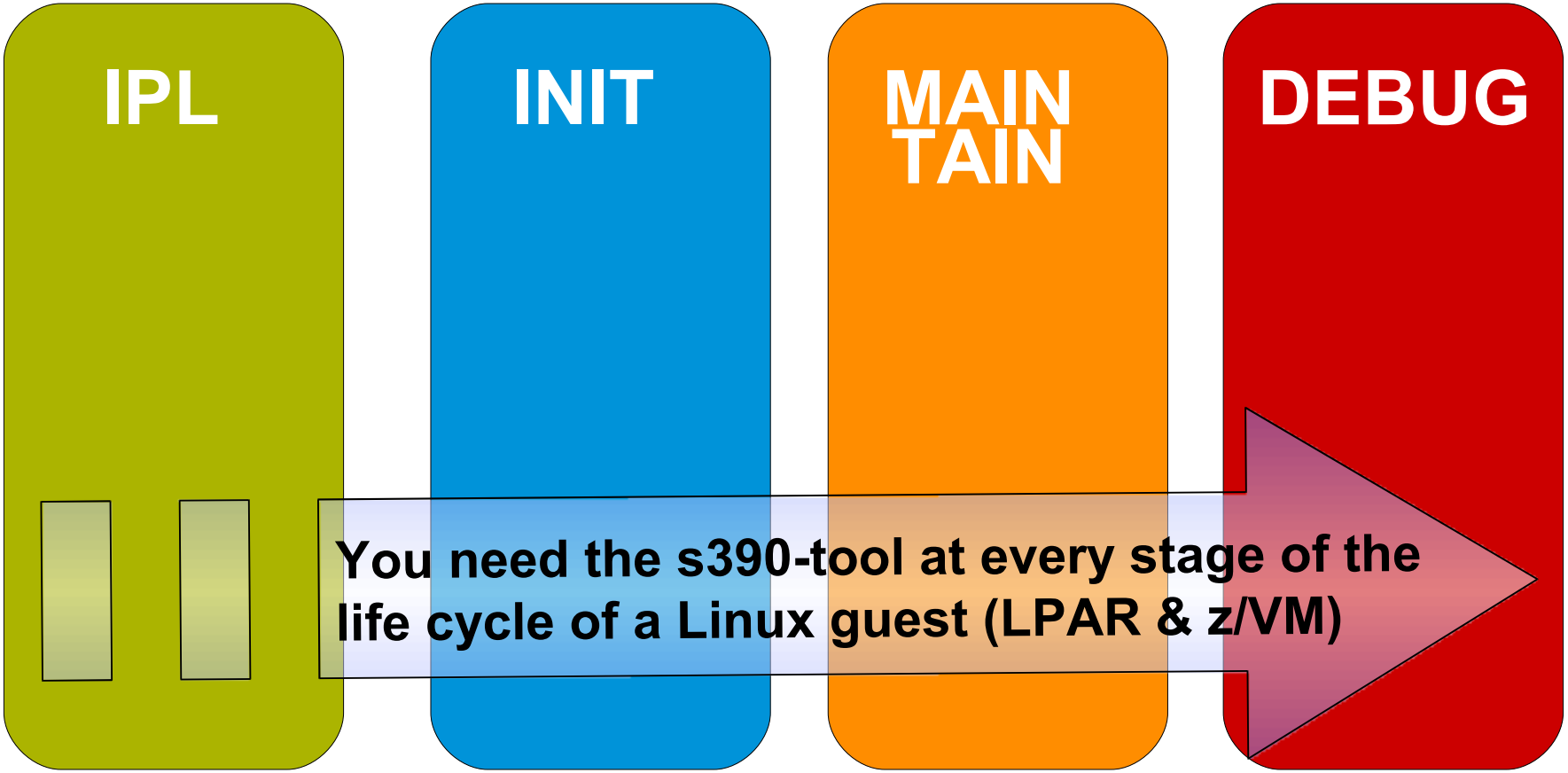
Hans-Joachim Picht, Linux on System z Initiatives <hans@de.ibm.com>

s390-tools: The Swiss Army Knife for Linux on System z System Administration

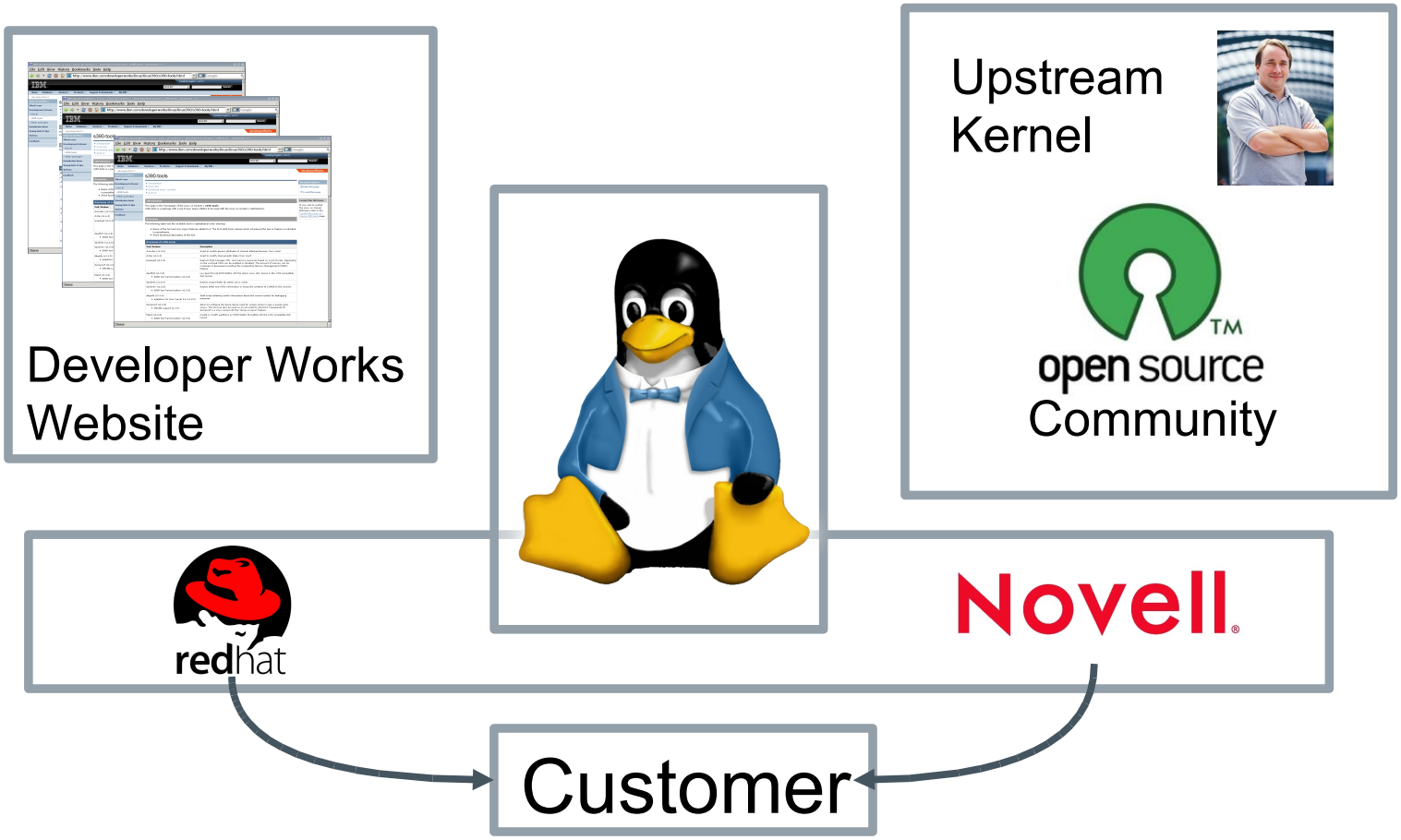
Linux on System z Live Virtual Class
Tuesday, May 4, 2010

The Linux on System z Life Cycle



IBM Linux on System z Development

IBM Linux on System z Development contributes in the following areas: Kernel, s390-tools, Open Source Tools (e.g. eclipse, ooprofile), GCC, GLIBC, Binutils



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 system crash analysis
- The latest version is 1.8.4 and was released in March 2010
- This software package is contained in all major (and IBM supported) distributions which support s390
  - RedHat Enterprise Linux 4 (s390-tools-1.3.2)
  - RedHat Enterprise Linux 5 (s390-tools-1.8.1 since RHEL 5.4)
  - SuSE Linux Enterprise Server 10 (s390-tools-1.6.3 since SLES 10 SP2)
  - SuSE Linux Enterprise Server 11 (s390-tools-1.8.0)

- Website: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html

The Content

CHANGE
- chccwdev
- chchp
- chreipl
- chshut
- chzcrypt

DISPLAY
- lscss
- lschp
- lsdasd
- lsluns
- lsqeth
- lsreipl
- lsshut
- lstape
- lszcrypt
- lszfcp

DASD
- dasdfmt
- dasdinfo
- dasdview
- fdasd
- tunedasd

MONITOR
- mon_fsstatd
- mon_procd
- ziomon

NETWORK
- ip_watcher
- osasnmpd
- qetharp
- qethconf

TAPE
- tape390_crypt
- tape390_display

DUMP & DEBUG
- dbginfo
- dumpconf
- zfcpdump
- zfcpdbf
- zgetdump
- scsi_logging_level

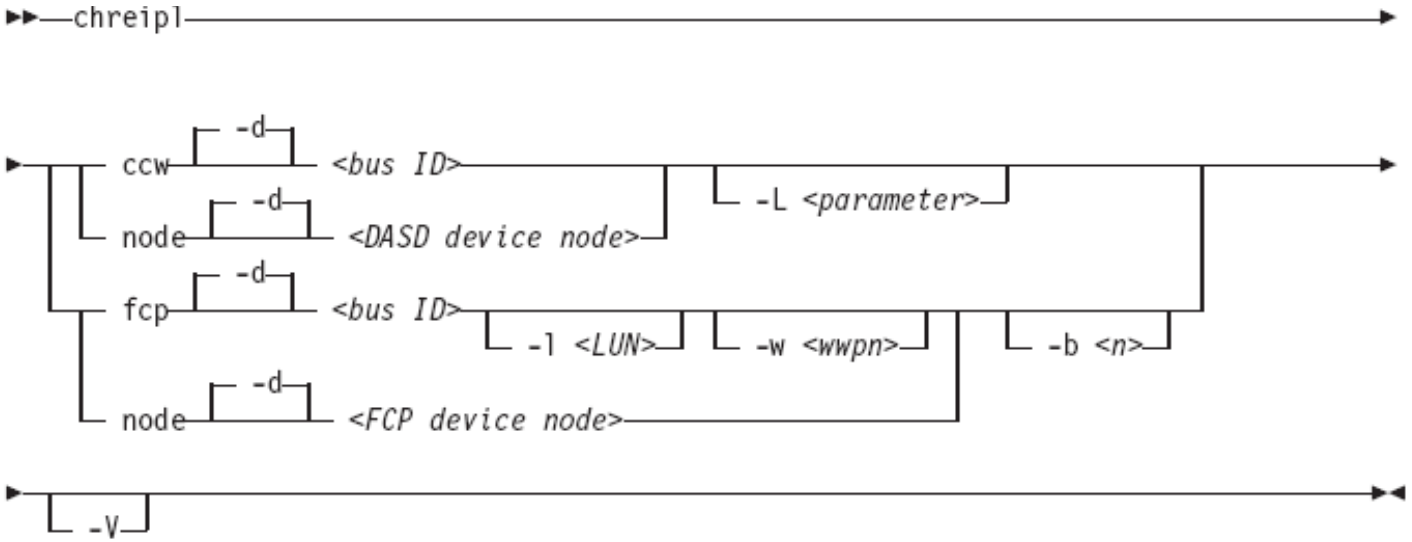
z/VM
- vmconvert
- vmcp
- vmur

MISC
- cpuplugd
- iucvconn
- lucvtty
- ts-shell

BOOT
- zipl

Shutdown action tools

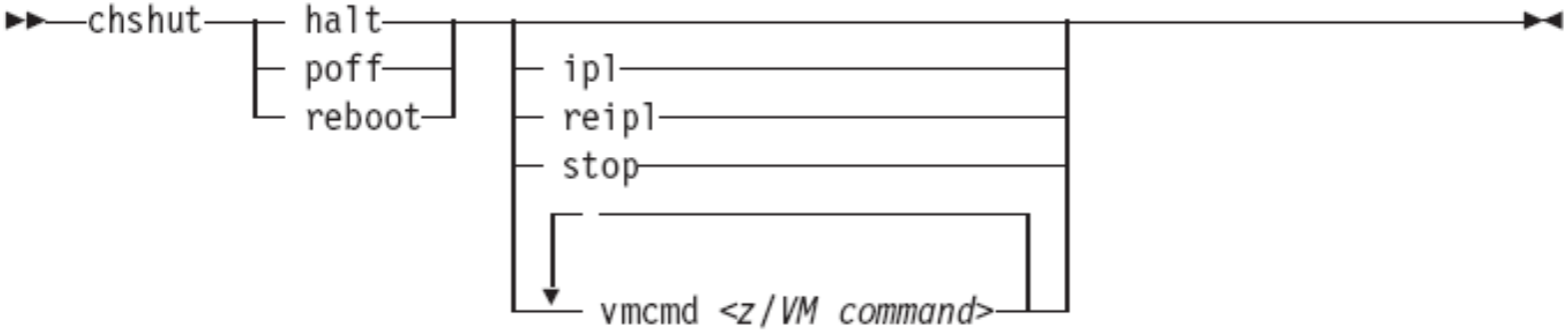
chreipl: Configure a disk or change a an entry in the Boot menu for the next boot cycle.




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

Shutdown action tools (cont.)

chshut: Change the entries in */sys/firmware* to configure the shutdown behaviour




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

Shutdown action tools (cont.)

lsreipl: command to see from which device your system will boot after you issue the reboot command. Further you can query the system for information about the current boot device.

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

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

```
root@larsson:~> lsshut
Trigger           Action
========================
Halt               stop
Panic              stop
Power off          stop
Reboot             reipl
```

lsluns

Use the **lsluns** command to discover and scan LUNs in Fibre Channel Storage Area Networks (SANs).

This example shows all LUNs for port 0x500507630300c562:

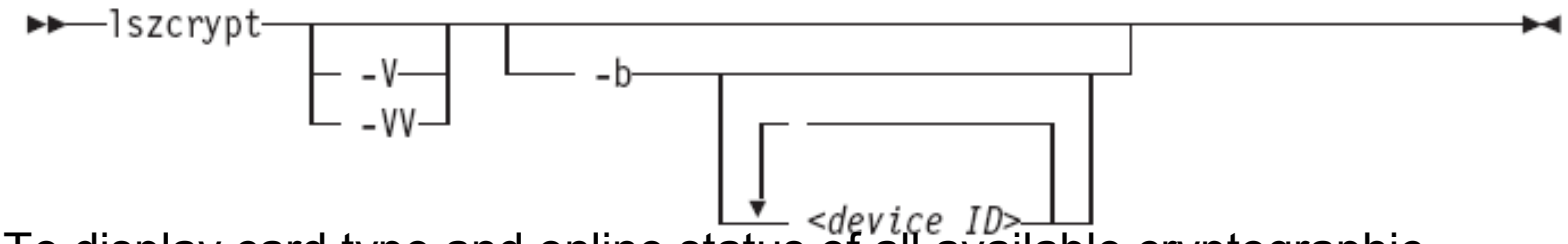
```
root@larsson:~> lsluns --port 0x500507630300c562
Scanning for LUNs on adapter 0.0.5922
at port 0x500507630300c562:
0x4010400000000000
0x4010400100000000
[...]
```

This example shows all LUNs for adapter 0.0.5922:

```
root@larsson:~> lsluns -c 0.0.5922
at port 0x500507630300c562:
0x4010400000000000
[...]
at port 0x500507630303c562:
0x4010400000000000
[...]
```

lszcrypt

Use the **lszcrypt** command to display information about cryptographic adapters managed by zcrypt and zcrypt's AP bus attributes



11

.4

To display card type and online status of all available cryptographic adapters:

```
root@larsson:~> lszcrypt -V
```

To display card type, online status, hardware card type, hardware queue depth, and request count for cryptographic adapters 0, 1, 10, and 12

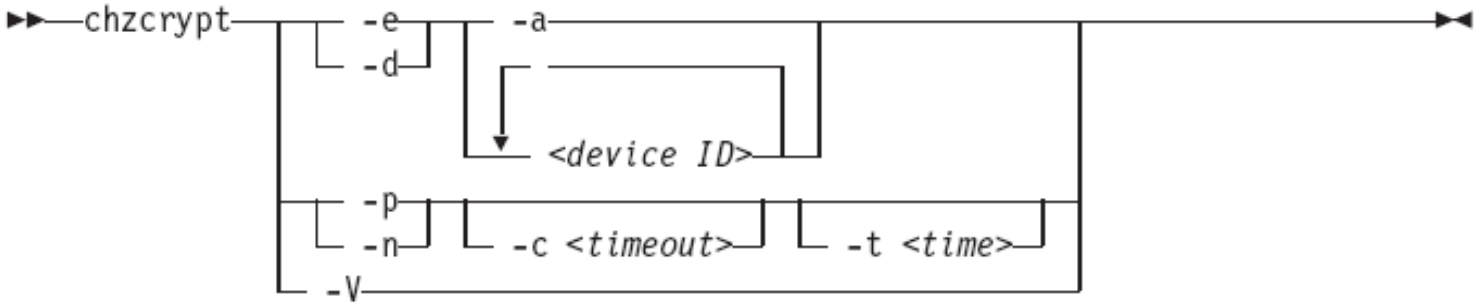
```
root@larsson:~>  lszcrypt -VV 0 1 10 12
```

To display AP bus information:

```
root@larsson:~> lszcrypt -b
```

chzcrypt

Use the **chzcrypt** command to configure cryptographic adapters managed by zcrypt and modify zcrypt's AP bus attributes.



To set the cryptographic adapters 0, 1, 4, 5, and 12 online:

```
root@larsson:~>  chzcrypt -e 0 1 4 5 12
```

To set all available cryptographic adapters offline:

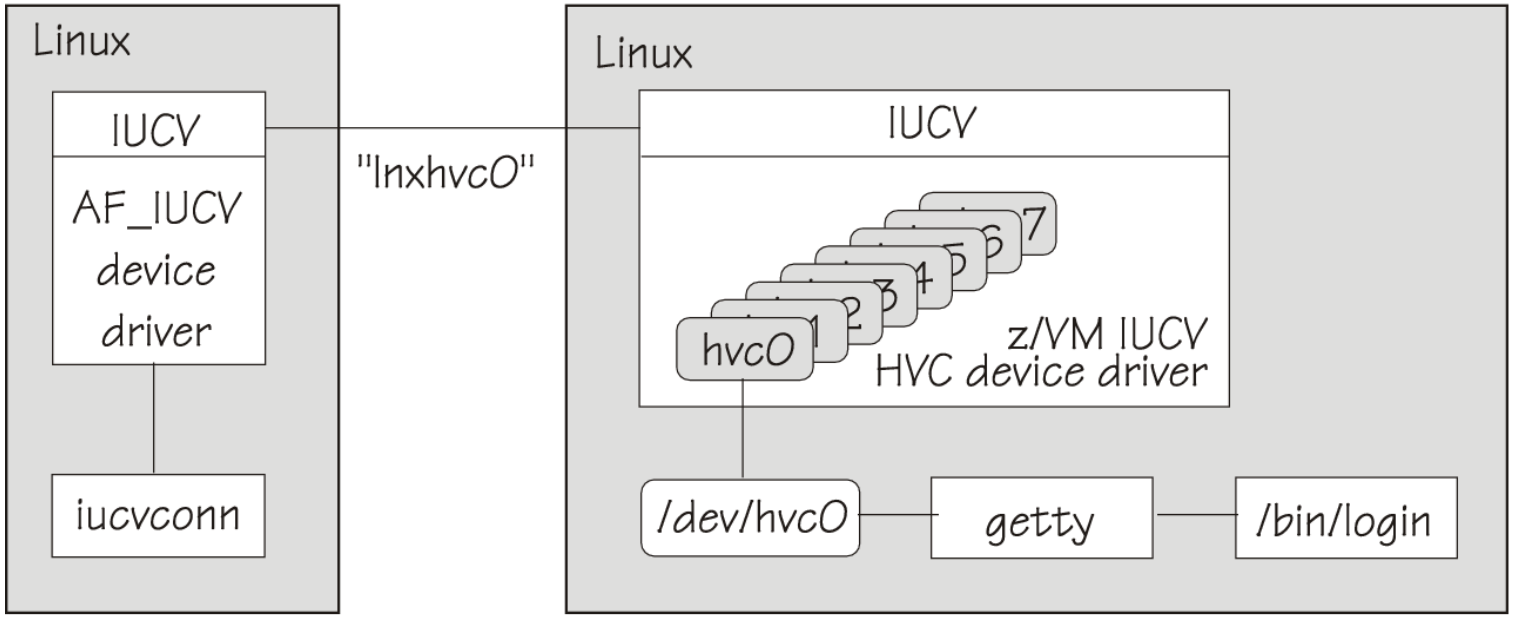
```
root@larsson:~>  chzcrypt -d -a
```

To set the configuration timer for re-scanning the AP bus to 60 seconds and disable zcrypt's poll thread:

```
root@larsson:~>  chzcrypt -c 60 -n
```

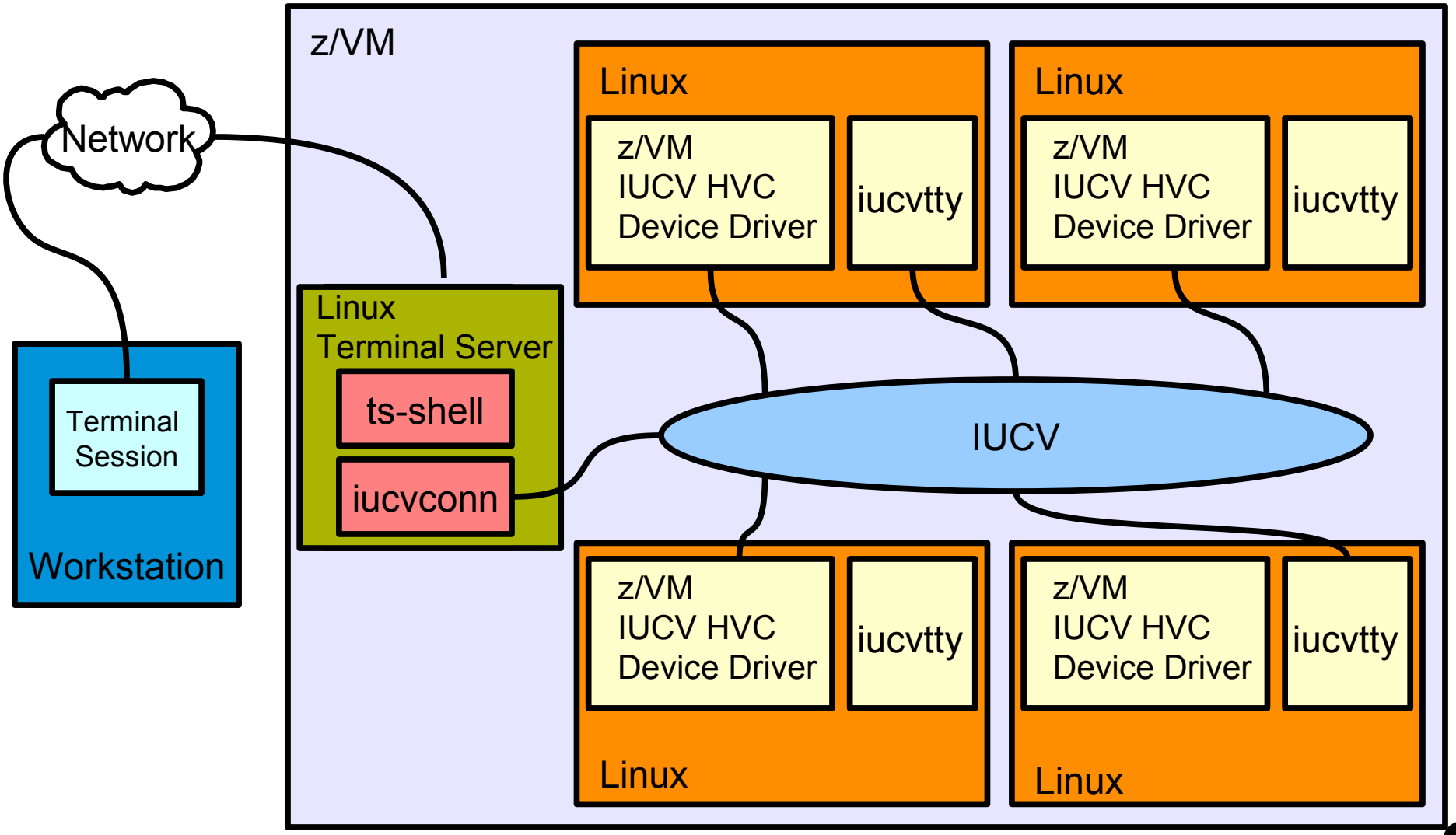
IUCV hypervisor console (HVC) device driver (Linux kernel)

- 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



Target system

IUCV terminal environment



IUCV terminal applications

- The IUCV terminal applications consist of:
  - **iucvconn** – Start terminal connection over IUCV
  - **iucvtty** – Allow remote logins over IUCV
  - **ts-shell** – Login shell for terminal servers over IUCV
- Terminal access over IUCV is provided by:
  - **iucvtty**, or
  - z/VM IUCV hypervisor console device driver (Linux kernel)

IUCV terminal applications – examples

Using the iucvconn program:

To access the first z/VM IUCV HVC terminal on the Linux instance in z/VM guest LNXSYS02

```
root@larsson:~> iucvconn LNXSYS02 lnxhvc0
```

To create a transcript of the terminal session to the Linux instance in z/VM guest LNXSYS99

```
root@larsson:~> iucvconn -s ~/transcripts/lnxsys99
LNXSYS99 lnxhvc0
```

Using the iucvtty program:

To allow remote logins using the terminal identifier „lnxterm“

```
root@larsson:~> iucvtty lnxterm
```

To access the „lnxterm“ terminal on the Linux instance in z/VM guest LNXSYS01

```
root@larsson:~>  iucvconn LNXSYS01 lnxterm
```

To use /sbin/sulogin instead of /bin/login for terminal “suterm”

```
root@larsson:~>  iucvtty suterm -- /sbin/sulogin
```


cpuplugd

Use the **cpuplugd** command to:

- Enable or disable CPUs based on a set of rules. This increases the performance of single threaded applications within a z/VM or LPAR environment with multiple CPUs.
- 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: Example Configuration

```
UPDATE="60"

CPU_MIN="2"
CPU_MAX="10"

HOTPLUG = "(loadavg > onumcpus +0.75) & (idle < 10.0)"
HOTUNPLUG = "(loadavg < onumcpus -0.25) | (idle > 50)"


CMM_MIN="0"
CMM_MAX="8192"
CMM_INC="256"

MEMPLUG = "swaprate > freemem+10 & freemem+10 < apcr"
MEMUNPLUG = "swaprate > freemem + 10000"
```

dbginfo.sh

dbginfo.sh is a script to collect various system related files, for debugging purposes.

- It generates a tar-archive which can be attached to PMRs / Bugzilla entries
- It is similar to the RedHat tools sosreport

```
root@larsson:~>  dbginfo.sh
Create target directory /tmp/DBGINFO-2010-04-25-22-06-20-
t6345057
Change to target directory /tmp/DBGINFO-2010-04-25-22-06-
20-t6345057
Get procfs entries
Saving runtime information into runtime.out
Get file list of /sys
Get entries of /sys
[...]
```

Please use the data from this tool is you open a Bugzilla (Novell/ RedHat) or a PMR !!!

Multi Volume Dump

zipl can now dump to multiple DASDs. It is now possible to dump system images, which are larger than a single DASD.

10.3

5.4

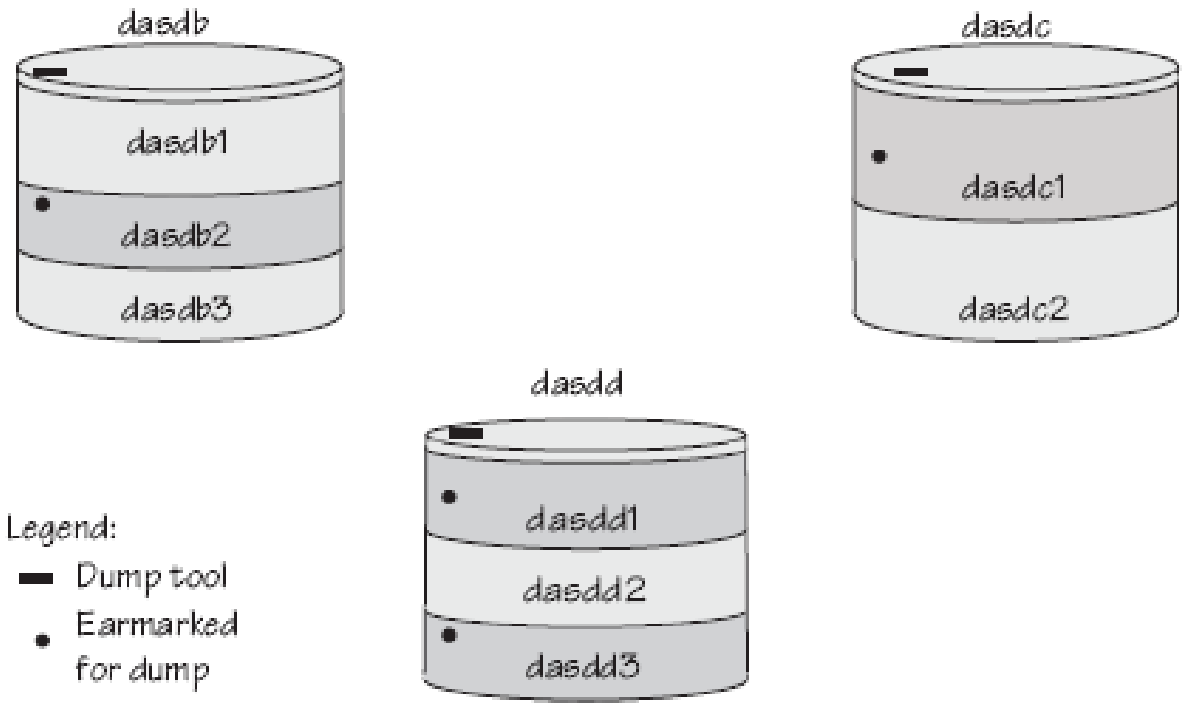
You can specify up to 32 ECKD DASD partitions for a multi-volume dump

What are dumps good for?

Full snapshot of a system state taken at any point in time.

Can be used to analyse system state beyond messages written to syslog / the console.

Allows access to Internal data structures not exported anywhere



Multi Volume Dump

How to prepare a set of ECKD DASD devices for a multi volume dump? (64-bit systems only).

We use two DASDs in this example:

```
root@larsson:~> dasdfmt -f /dev/dasdc -b 4096
root@larsson:~> dasdfmt -f /dev/dasdd -b 4096
```

Create the partitions with fdasd. The sum of the partition sizes must be sufficiently large (the memory size + 10 MB):

```
root@larsson:~> fdasd /dev/dasdc
root@larsson:~> fdasd /dev/dasdd
```

Create a file called sample_dump_conf containing the device nodes (e.g. /dev/dasda1) of the two partitions, separated by one or more line feed characters

Prepare the volumes using the zipl command.

```
root@larsson:~>  zipl -M sample_dump_conf
[...]
```


How to obtain a dump

To obtain a dump with the multi-volume DASD dump tool, perform the following steps:

- Stop all CPUs, Store status on the IPL CPU.
- IPL the dump tool using one of the prepared volumes, either 4711 or 4712.
- After the dump tool is IPLed, you'll see a messages that indicates the progress of the dump. Then you can IPL Linux again

```
==> cp cpu all stop
==> cp cpu 0 store status
==> cp ipl 4711
```

- Copying a multi-volume dump to a file
- Use zgetdump command without any option to copy the dump parts to a file:

```
root@larsson:~>  zgetdump /dev/dasdc > mv_dump_file
```


How to obtain information about a multi volume dumps

Display information on the involved volumes:

```
root@larsson:~>  zgetdump -d /dev/dasdc
'/dev/dasdc' is part of Version 1 multi-volume dump,which is
spread along the following DASD volumes:
0.0.4711 (online, valid)
0.0.4712 (online, valid)
[...]
```

Display information about the dump itself:

```
root@larsson:~>  zgetdump -i /dev/dasdc
Dump device: /dev/dasdc
>>>  Dump header information  <<<
Dump created on: Thu Feb  25 15:12:41 2010
[...]
Multi-volume dump: Disk 1 (of 2)
Reading dump contents from
0.0.4711.................................
Dump ended on:   Thu Feb  25 15:12:52 2010
Dump End Marker found: this dump is valid.
```

dumpconf

- 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
```


dumpconf (cont.)

Example configuration for FCP dump device (SCSI disk):

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4714
WWPN=0x500507630300471 2
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 (MASTER is a VM Guest Name):

```
ON_PANIC=vmcmd
VMCMD_1="MSG MASTER Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```


Dump Tools Summary

Tool	Stand alone tools			VMDUMP
	DASD	Tape	SCSI	
Environment	VM&LPAR		LPAR	VM
Preparation	Zipl -d /dev/<dump_dev>		Mkdir /dumps/mydumps zipl -D /dev/sda1 ...	---
Creation	Stop CPU & Store status ipl <dump_dev_CUU>			Vmdump
Dump medium	ECKD or FBA	Tape cartridges	LINUX file system on a SCSI disk	VM reader
Copy to filesystem	Zgetdump /dev/<dump_dev> > dump_file		---	Dumpload ftp ... vmconvert ...
Viewing	Lcrash or crash			

See “Using the dump tools” book at http://www.ibm.com/developerworks/linux/linux390/

vmcp

Using the z/VM CP interface device driver (vmcp), you can send control program (CP) commands to the VM hypervisor and display VM's response.

```
root@larsson:~> modprobe vmcp
root@larsson:~> vmcp "q dasd"|grep T6345057
DASD 4DE0 ATTACHED TO T6345057 4DE0 R/W 0X4DE0
DASD 4DE1 ATTACHED TO T6345057 4DE1 R/W 0X4DE1
DASD 4DE2 ATTACHED TO T6345057 4DE2 R/W 0X4DE
DASD 4DE3 ATTACHED TO T6345057 4DE3 R/W 0X4DE3
```

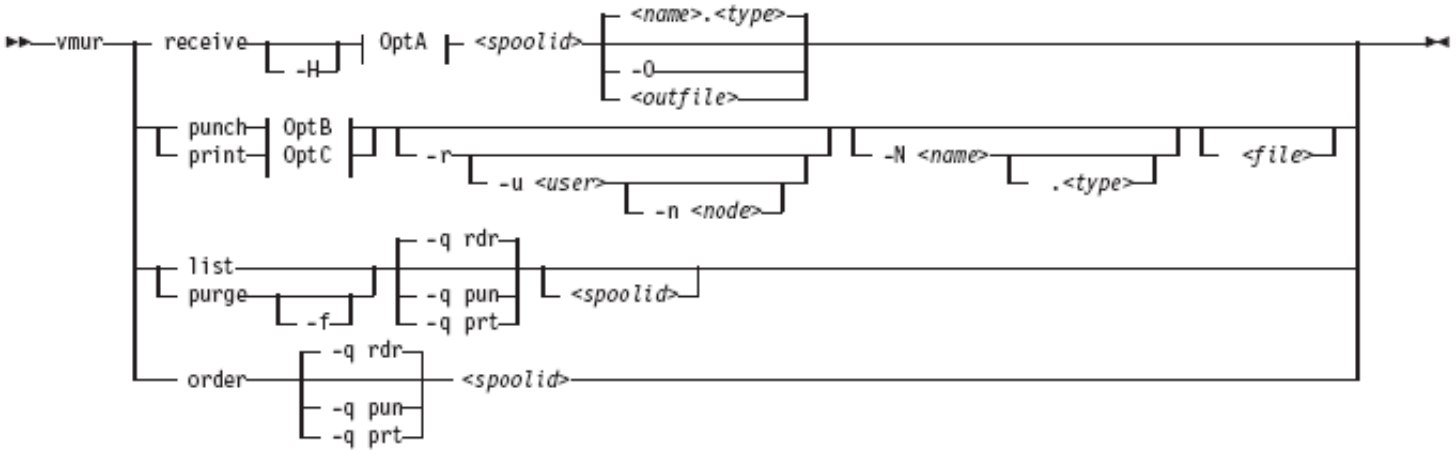
vmur

The **vmur** command provides all functions required to work with z/VM spool file queues:

10.1

5.2

- Receive: Read data from the z/VM reader file queue
- Punch or print: Write data to the z/VM punch or printer file queue and transfer it to another user's virtual reader, optionally on a remote z/VM node.
- List: Display detailed information about one or all files on the specified spool file queue.
- Purge: Remove one or all files on the specified spool file queue.
- Order: Position a file at the top of the specified spool file queue.



vmur: Produce and read Linux guest machine dump

Produce guest machine dump:

```
root@larsson:~>  vmcp vmdump
```

Find spool ID of VMDUMP spool file in the output of the `vmur li` command:

```
root@larsson:~> vmur li
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE
DIST T6360025 0463 V DMP 00020222 001 NONE 06/11 15:07:42
VMDUMP FILE T6360025
```

Move vmdump file to top of reader queue with the vmur order command:

```
root@larsson:~>  vmur or 463
```

Read and convert the vmdump file to a file on the Linux file system in the current working directory and close the virtual reader

```
root@larsson:~> chccwdev -e 000c
root@larsson:~> vmconvert /dev/vmrdr-0.0.000c linux_dump
root@larsson:~> vmcp cl c
```


vmur: Log and read Linux guest machine console

Begin console spooling:

```
root@larsson:~>  vmcp sp cons start
```

Produce output to VM console (for example, with CP TRACE).
Close the console file and transfer it to the reader queue, find the spool ID behind the FILE keyword in the corresponding CP message.

```
root@larsson:~>   vmcp sp cons clo \* rdr
RDR FILE 0398 SENT FROM T6360025 CON WAS 0398 RECS 1872
CPY 001 T NOHOLD NOKEEP
```

Read the guest machine console file into a file on the Linux file system in the current working directory:

```
root@larsson:~>  chccwdev -e 000c
root@larsson:~>  vmur re -t 398 linux_cons
```


vmur: Prepare z/VM reader to IPL Linux image

Send parmfile to VM punch and transfer it to the reader queue and find the parmfile spool id message

```
root@larsson:~>  vmur pun -r /boot/parmfile
[...]
Reader file with spoolid 0465 created.
```

Send image to VM punch and transfer it to reader queue:

```
root@larsson:~> vmur pun -r /boot/vmlinuz -N image
```

Move image to first and parmfile to the second position in the reader queue:

```
root@larsson:~>  vmur or 465
root@larsson:~>  vmur or 466
```

Prepare re-IPL from the VM reader and boot the Linux image

```
root@larsson:~>  chreipl ccw 0.0.000c
root@larsson:~>  reboot
```


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.

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


cio_ignore (cont.)

Display ignored devices:

```
root@larsson:~>  cat /proc/cio_ignore
0.0.0000-0.0.78ff
0.0.f503-0.0.ffff
```

Free a individual device from the ignore list

```
root@larsson:~>  echo free 0.0.4711 >/proc/cio_ignore
```

Free all devices from the ignore list

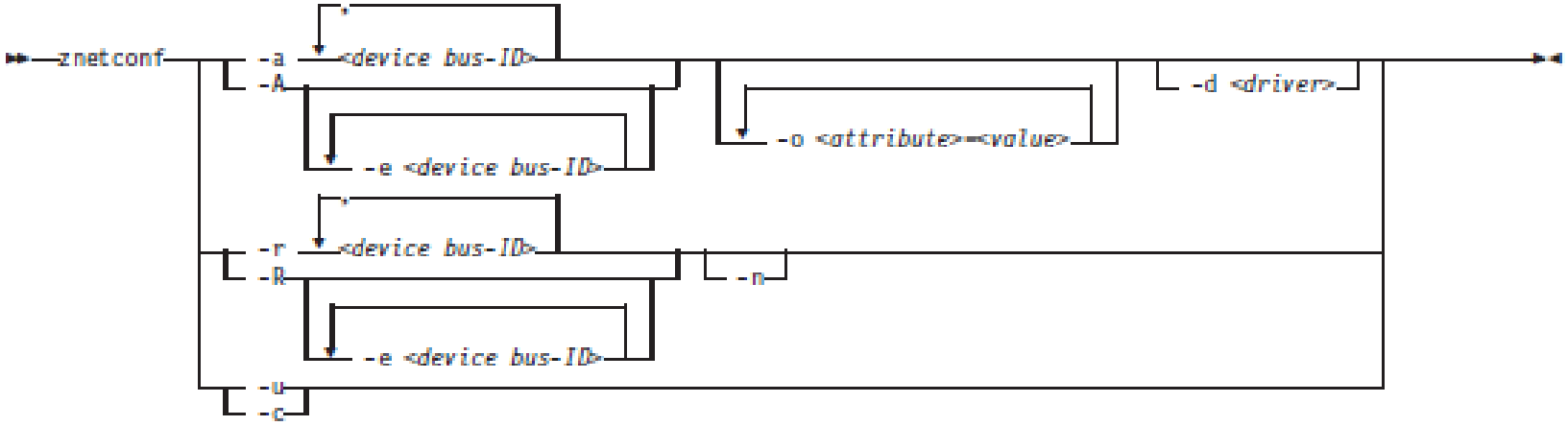
```
root@larsson:~>  echo free all >/proc/cio_ignore
```


Future Linux on System z Technology

Software which has already been developed and externally published – but is **not** yet available in any Enterprise Linux Distribution

znetconf

Use the znetconf command to list and configure network devices.



To list all configured network devices:

```
root@larsson:~>  znetconf -c
Device IDs                Type    Card Type  CHPID Drv. Name        State

------------------------------------------------------------------------------
0.0.f500,0.0.f501,0.0.f502 1731/01 OSD_100      76 qeth eth0        online
```

To configure the potential network device 0.0.f503 with the layer2 option with the value 0 and the portname option with the value myname:

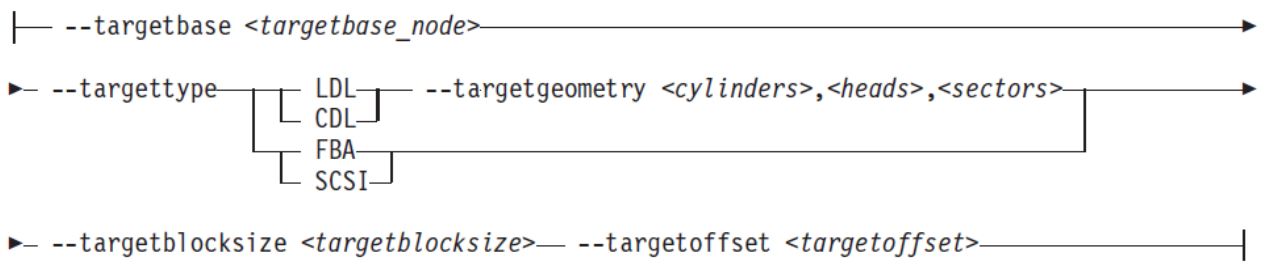
```
root@larsson:~>  znetconf -a f503 -o layer2=0 -o
portname=myname
```

Support for device mapper devices: zipl

Lately zipl allows installation of and booting from a boot record on logical devices, i.e. devices managed by device mapper, e.g. multipath devices.

zipl - additional command line parameters for logical boot devices

Target base parameters:



- A physical device is of type DASD or SCSI
- All of the devices which contains the directory must be located on a single physical device (which may be mirrored or accessed through a multipath target) only linear, mirror and multipath targets are supported
- The boot directory is located on a device consisting of a single device-mapper target

zipl (cont'd)

Example for using additional parameters

```
root@larsson:~>  zipl --dry-run -i /boot/image -r
/boot/initrd -p /boot/parmfile  -t /boot --targetbase /
dev/dm-0 --targettype CDL --targetgeometry 3339,15,12
--targetblocksize=4096 --targetoffset 24 -V
```

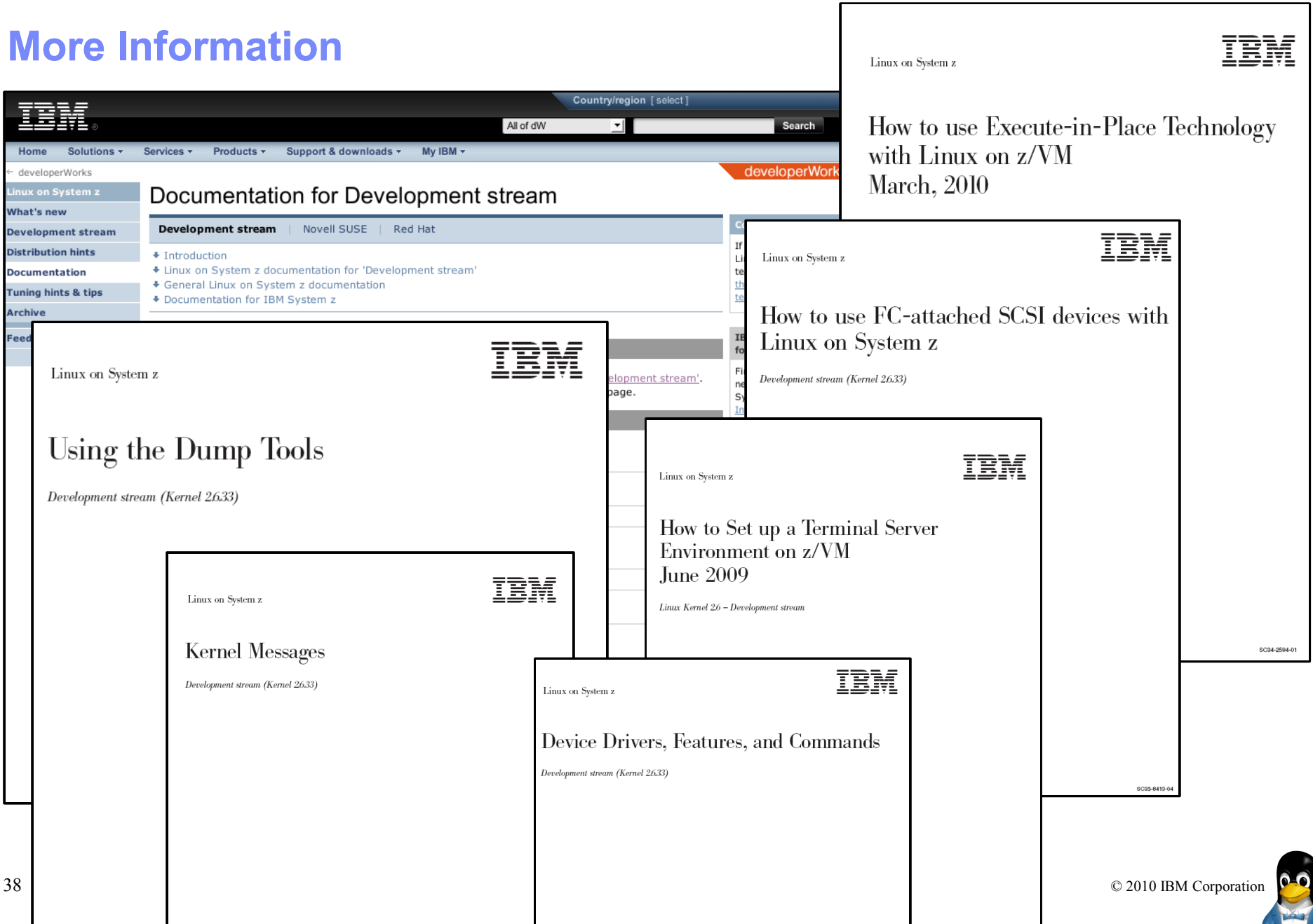
Example section in the zipl configuration file

```
[boot5]
image=/boot/image
ramdisk=/boot/initrd
parmfile=/boot/parmfile
target=/boot
targetbase=253:0
targettype=CDL
targetgeometry=3339,15,12
targetblocksize=4096
targetoffset=24
```


zipl (cont'd)

```
root@larsson:~>  zipl -V
Using config file '/etc/zipl.conf'
Run /lib/s390-tools/zipl_helper.device-mapper /boot/zipl
Target device information
  Device.........................: fd:00
  Device name....................: dm-0 *)
  Device driver name.............: device-mapper
  Type...........................: disk device
  Disk layout....................: ECKD/compatible disk layout *)
  Geometry - heads...............: 15 *)
  Geometry - sectors.............: 12 *)
  Geometry - cylinders...........: 3339 *)
  Geometry - start...............: 24 *)
  File system block size.........: 4096
  Physical block size............: 4096 *)
  Device size in physical blocks..: 600996
  *) Data provided by script.
Building bootmap in '/boot/zipl'
Adding IPL section 'ipl' (default)
  kernel image......: /boot/image-2.6.16.60-0.21-default
  kernel parmline...: 'root=/dev/disk/by-id/ccw-IBM.75000000092461.2f00.0c-part1
TERM=dumb '
  initial ramdisk...: /boot/initrd-2.6.16.60-0.21-default
  component address:
    kernel image....: 0x00010000-0x006d4fff
    parmline........: 0x006d5000-0x006d5fff
    initial ramdisk.: 0x006e0000-0x00a24fff
    internal loader.: 0x0000a000-0x0000afff
Preparing boot device: dm-0.
Syncing disks...
Done.
```


More Information



Questions?

Hans-Joachim Picht

Linux Technology Center

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@de.ibm.com

Your Linux on System z Requirements?

Are you missing a certain feature, functionality or tool?
I'd love to hear from you!

We will evaluate each request and (hopefully) develop the additional functionality you need.

Trademarks & Disclaimer

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

IBM, the IBM logo, BladeCenter, Calibrated Vectored Cooling, ClusterProven, Cool Blue, POWER, PowerExecutive, Predictive Failure Analysis, ServerProven, System p, System Storage, System x , System z, WebSphere, DB2 and Tivoli are trademarks of IBM Corporation in the United States and/or other countries. For a list of additional IBM trademarks, please see http://ibm.com/legal/copytrade.shtml.

The following are trademarks or registered trademarks of other companies: Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both Microsoft, Windows,Windows NT and the Windows logo are registered 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 or both. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc. InfiniBand is a trademark of the InfiniBand Trade Association.
Other company, product, or service names may be trademarks or service marks of others.

NOTES: Linux penguin image courtesy of Larry Ewing (lewing@isc.tamu.edu) and The GIMP

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment. IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided “AS IS” without warranty of any kind. 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 are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any