LINUX for zSeries

Using the Dump Tools - October 8, 2001

Linux kernel 2. 4

LNUX-1108-00

LINUX for zSeries

Using the Dump Tools - October 8, 2001

Linux kernel 2. 4

LNUX-1108-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 13.

First Edition (October 2001)

This edition applies to Linux kernel 2.4 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Dump tools

Two standalone dump tools are shipped in the s390-tools rpm as part of the zIPL package for generating system memory dumps:

- DASD dump tool for dumps on DASD volumes
- Tape dump tool for dumps on tape

You need to install the tools on the device that you want to use for dumping the memory, the so called dump device. Then you can send the dump to IBM® service to be analyzed. Alternatively, you can use the dump analysis tool **lcrash** (from lkcdutils version 3.1.2 and higher) to analyze the dump

Typically, the system operator initiates a dump after a system crash, but you can take a dump at any time. In order to take a dump, you must IPL the dump device. This is destructive, that is, the running Linux operating system will be killed. The IPL process then writes the system memory to the IPL device.

Supported devices:

- ECKD™ DASDs
  - 3380
  - 3390
- Tape units
  - 3480
  - 3490

Note on examples

In all examples the old device names for tape and DASD are used (/dev/dasdx and /dev/ntibmx or /dev/rtibmx). If you are running Linux with the device filesystem, the names will look different as shown in Table 1:

Table 1. Old vs. device filesystem names

Old	**Device filesystem**
Tape: • /dev/ntibm0 • /dev/rtibm0	Tape: • /dev/tape/<devno>/char/nonrewinding • /dev/tape/<devno>/char/rewinding
DASD: • /dev/dasdx • /dev/dasdx1	DASD: • /dev/dasd/<devno>/disc • /dev/dasd/<devno>/part1

Installing the dump tools

Requirements: Ensure that the dump device has enough space (memory size + 1 MB) to hold the system memory.

Installing the DASD dump tool

You need an unused DASD partition. This section describes how to install the tool on this partition. Dumps will be written to this partition.

Assume /dev/dasdc is the dump device and we want to dump to the first partition /dev/dasdc1.

1. Format DASD (a blocksize of 4K is recommended):
   ```
   dasdfmt -f /dev/dasdc -b 4096
   ```
2. Create a partition with fdasd:
   ```
   fdasd /dev/dasdc
   ```
3. Install the dump tools by editing /etc/zipl.conf and adding the following lines:
   ```
   [dump_eckd]
   target=/boot
   dumpto=/dev/dasdc1
   ```
4. Call zIPL:
   ```
   > zipl dump_eckd
   ```

Alternatively, instead of editing /etc/zipl.conf, you can call zipl with the following command line parameters:

```
> zipl -d /dev/dasdc1 -t /boot
```

Note: When using DASDs formatted with the old Linux disk layout ldl (AIX-compliant), the dump tool must be re-installed using zIPL after each dump.

Installing the tape dump tool

To install the tape dump tool:

1. Insert an empty dump cartridge into your tape device (for example /dev/rtibm0).
2. Ensure that the tape is rewound.
3. Edit /etc/zipl.conf and add the following lines:
   ```
   [dump_tape]
   target=/boot
   dumpto=/dev/rtibm0
   ```
4. Call zIPL:
   ```
   > zipl dump_tape
   ```

Alternatively, instead of editing /etc/zipl.conf, you can call zipl with the following command line parameters:

```
> zipl -d /dev/rtibm0 -t /boot
```

Chapter 2. Dumping on tape

You can accomodate a large dump by using multiple tapes. Only the first tape (that you IPL from) needs to have the tape dump tool installed.

Considerations for native Linux: When initiating the dump process for example from a service element, on the IPL (LOAD) panel select the **Store Status** checkbox. For detailed information on the actual steps to be performed, consult the appropriate manual of your processor.

To take the dump:

1. Set the cartridge loader to AUTO and insert a sufficient number of cartridges.

   Attention: The dump tool loads tapes automatically from the cartridge holder and overwrites any data on them. Ensure that the cartridge holder does not hold tapes with important data.
2. Ensure that the tapes are rewound.
3. If you are working under VM, stop all CPUs.
4. Store status on the IPL CPU.
5. IPL the dump tool on the IPL CPU.

   Note: Do not clear storage!

   The dump tool will write messages to the Tape Drive Message Display (not to the operator console). First the number of dumped MB is displayed. When a tape cartridge is full, it will be automatically unloaded and the message `next*vol` is displayed. If more cartridges are available, they will be loaded and the dump continues. If no cartridge is available, the dump tool waits for the operator to load one.

   The dumping can take several minutes depending on the device type you are using and the amount of system memory available. When the dump is complete, the message `dump*end` is displayed and the IPL CPU should go into disabled wait.

   The following PSW indicates that the dump was taken successfully:

   ```
   PSW: 00020000 80000000 00000000 00000000
   ```

   Any other disabled wait PSW indicates an error.
6. Then you can IPL Linux again.

Linux should come up. If not, you can try booting from the reader.

Example: z/VM

If 193 is the tape device:

```
# cp rewind 193
# cp cpu all stop
# cp store status
# cp i 193
```

On VM, a three-processor machine in this example, you will see messages about the disabled wait:

```
01: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
02: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
"CP entered; disabled wait PSW 00020000 80000000 00000000 00000000"
```

IPL the LINUX DASD 192:

```
# cp i 192
```

Tape display messages

next*vol
The dump tool loads the next tape cartridge, or, if none is available, waits for the operator to load one.

number
The number of MB dumped.

dump*end
The dump process ended successfully.

Chapter 3. Dumping on DASD

Considerations for native Linux: When initiating the dump process for example from a service element, on the IPL (LOAD) panel select the **Store Status** checkbox. For detailed information on the actual steps to be performed, consult the appropriate manual of your processor.

To take the dump:

1. If you are working under VM, stop all CPUs.
2. Store status on the IPL CPU.
3. IPL the dump tool on the IPL CPU.

   Note: Do not clear storage!

   The dumping can take several minutes depending on the device type you are using and the amount of system memory. After the dump has completed the IPL CPU should go into disabled wait.

   The following PSW indicates that the dump was taken successfully:

   ```
   PSW: 00020000 80000000 00000000 00000000
   ```

   Any other disabled wait PSW indicates an error.
4. Then you can IPL Linux again.

Linux should come up. If not, you can try booting from the reader.

Example: z/VM

If 193 is the dump device:

```
# cp cpu all stop
# cp store status
# cp i 193
```

On VM, a three-processor machine in this example, you will see messages about the disabled wait:

```
01: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
02: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
"CP entered; disabled wait PSW 00020000 80000000 00000000 00000000"
```

IPL the LINUX DASD 192:

```
# cp i 192
```

Chapter 4. Copying the dump to a filesystem

In order to send the dump to IBM service, you need to first copy it from the dump medium (either DASD or tape) to a filesystem.

Copying the dump from DASD

To copy the dump to a filesystem, you can use the **dd** command or the **zgetdump** tool.

Using the dd command

Assuming the dump device is DASD (for example /dev/dasdx) , use the following command to copy a dump to a filesystem:

```
>dd if=/dev/dasdx1 of=/DUMPDIR/mydump bs=1M count=<memory size in MB plus 1 MB>
```

If you want to compress the dump:

```
>dd if=/dev/dasdx1 bs=1M count=<memory size in MB plus 1 MB> | bzip2 > /DUMPDIR/dump.bz2
```

You can also use other compressing tools, such as gzip.

Note: It can take considerable time, up to a couple of hours, to compress a big dump. The time depends on the processor and the size of the dump.

For general information on dd, see "The dd command" on page 9 or the manpage.

Using the zgetdump tool

By default, the zgetdump tool takes the dump device as input and writes its contents to standard output. As this would be of little use, you must redirect the output to a file.

Assuming that the dump is on DASD device /dev/dasdb1 and you want to copy it to a file named dump_file:

```
> zgetdump /dev/dasdb1 > dump_file
```

For general information on zgetdump, see "The zgetdump tool" on page 10 or the manpage.

Copying the dump from tape

You can use either the **dd** command or the **zgetdump** tool to copy a dump from tape.

Note: For multi-volume dumps, you must use the **zgetdump** tool.

First you must rewind the tape, and find the correct position on the tape to start copying from. Use the mt tool to do this.

Prerequisite: Install the **mt** utility.

Assuming the tape device is /dev/ntibm0:

1. Rewind the tape:

   ```
   >mt -f /dev/ntibm0 rewind
   ```

2. Skip the first file on the tape (this is the dump tool itself):

   ```
   >mt -f /dev/ntibm0 fsf
   ```

Now you can use either the **dd** command or the **zgetdump** tool to continue.

Using the dd command

Assuming that the tape is in the correct position as described above and is on tape device /dev/ntibm0, use the following command to copy the dump from tape to a filesystem:

```
>dd if=/dev/ntibm0 of=/DUMPDIR/mydump bs=32k
```

or, to compress the dump:

```
>dd if=/dev/ntibm0 bs=32k | bzip2 > /DUMPDIR/mydump.bz2
```

Note: It can take considerable time, up to a couple of hours, to compress a big dump. The time depends on the processor and the size of the dump.

For general information on dd, see “The dd command” on page 9 or the manpage.

Using the zgetdump tool

In the case of a multi-volume dump, the cartridges must be loaded in the right order, starting with the first volume. You can use the cartridge loader for automatic loading. When **zgetdump** finds the end of a volume (that is not the end of the dump) it will write a message and attempt to load the next cartridge.

By default, the zgetdump tool takes the dump device as input and writes its contents to standard output. As this would be of little use, you must redirect the output to a file.

Note: Always use the AUTO setting on your tape device when working with zgetdump.

Assuming that the tape is in the correct position as described above and is on tape device /dev/ntibm0, use the following command to copy the dump from tape to a file dump_file in the filesystem:

```
zgetdump /dev/ntibm0 > dump_file
```

For general information on zgetdump, see “The zgetdump tool” on page 10 or the manpage.

Checking whether a dump is valid

To check whether a single-volume or a multivolume dump is valid, use the **-i** option. For multi-volume dumps, use the **-a** option to signal that you are working with a multi-volume dump.For example, to check whether a multi-volume dump is valid:

1. Ensure that the volumes are loaded in the correct sequence.
2. Skip the first file on the first tape (this is the dump tool itself):

   ```
   >mt -f /dev/ntibm0 fsf
   ```

3. Enter:

   ```
   zgetdump -i -a /dev/ntibm0
   ```

   zgetdump will go through all the volumes until it reaches the end of the dump. See also “Using zgetdump to make a multi-volume dump” on page 11.

Command summary

The descriptions of the commands contain only the relevant options and parameters, for a full description refer to the man pages.

The dd command

If dd is installed on your system, the command **info dd** should give you the full manual.

dd Command

```
►►──dd──────────────────────────────────────────────────────────────────────────►

►──┬─count──=──MEMSIZE──bs──=──BYTES──if──=──DUMPDEVICE──of──=──FILE─┬──────────►◄
   ├───--help────────────────────────────────────────────────────────┤
   └───--version─────────────────────────────────────────────────────┘
```

where:

if=*DUMPDEVICE*
: Denotes the dump device. For example: `if=/dev/dasdx1`

of=*FILE*
: Denotes the filesystem to which the dump will be copied. For example: `of=/DUMPDIR/mydump`

bs=*BYTES*
: Denotes the blocksize. For example: `bs=1M`.

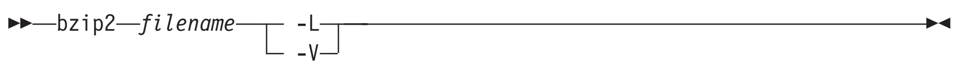
count= *MEMSIZE*
: Denotes the memory size in MB plus 1 MB.

The bzip2 tool

The bzip2 tool compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors. The command-line options are deliberately very similar to those of GNU gzip, but they are not identical.

The bzip2 tool expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name ″original_name.bz2″.

bzip2 Command



Where:

filename
The file to be compressed

-L Displays the license

-V Displays the version

Example

Using bzip2 to compress a dump file

To compress a dump file on device /dev/ntibm0 to a file /DUMPDIR/mydump.bz2 enter:

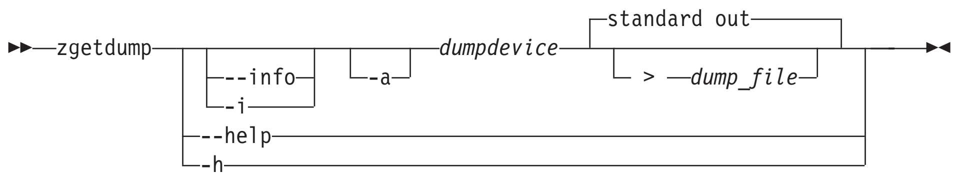
```
>dd if=/dev/ntibm0 bs=32k | bzip2 > /DUMPDIR/mydump.bz2
```

Note: It can take considerable time, up to a couple of hours, to compress a big dump. The time depends on the processor and the size of the dump.

The zgetdump tool

The zgetdump tool reads a dump from the given dump device and writes it contents to standard out, unless the operator redirects it to a file.

zgetdump Command



Where:

dumpdevice
The device where the dump is located.

dump_file
The file the output is re-directed to. The default is standard out.

- -info or -i
Displays the header information from the dump.

-a Signal that the dump is multi-volume.

- -help or -h
Displays the help information for the command.

Examples

Using zgetdump to copy a dump

Assuming that the dump is on DASD device /dev/dasdb1 and you want to copy it to a file named dump_file:

```
> zgetdump /dev/dasdb1 > dump_file
```

Using zgetdump to make a multi-volume dump

Making a multi-volume dump:

```
> zgetdump /dev/ntimb0 > dump_file
Dump device: /dev/ntibm0

Tape Volume 0

>>>  Dump header information  <<<
Dump created on: Wed Aug  1 17:20:16 2001
Magic number:    0xa8190173618f23fd
Version number:  2
Header size:     4096
Page size:       4096
Physical memory: 268435456
Number of pages: 65536
cpu id:          0xff02453096720000
System Arch:     s390 (ESA)
Build Arch:      s390
>>>  End of Dump header  <<<

Reading dump content ...........................
End of Volume reached.

Waiting for next volume to be loaded... done

Tape Volume 1 of a multi volume dump.
Reading dump content ......
Dump ended on:   Wed Aug  1 17:23:31 2001

Dump End Marker found: this dump is valid.
```

Checking if a multi-volume dump is valid

Assuming that the tape device is /dev/ntibm0:

```
zgetdump -i -a /dev/ntibm0

 "zgetdump -i -a" checks if a multi-volume tape dump is valid.
 Please make sure that all volumes are loaded in sequence.

 Dump device: /dev/ntibm0

 Tape Volume 0

 >> Dump header information <<<
 ...

 >> End of Dump header <<<

 Checking if the dump is valid - this might take a while...
 Reached End of Volume 0.
 Waiting for Volume 1 to be loaded... done

 Tape Volume 1 of a multi volume dump.
 Dump End found: This Dump is valid.
```

Checking whether a single-volume dump is valid

Assuming that the tape device is /dev/ntibm0:

```
zgetdump -i /dev/ntibm0

"zgetdump -i" checks if a dump on either
a dasd volume or single tape is valid.
If the tape is part of a multi-volume tape dump,
it checks if it is a valid portion of the dump.

Dump device: /dev/ntibm0

Tape Volume 0

>>> Dump header information <<<
...
>>> End of Dump header <<<

Checking if the dump is valid - this might take a while...
Checking for End of Volume...
Reached End of Volume 0 of a multi-volume tape dump.
This part of the dump is valid.
```

Displaying the help text for zgetdump

To display the help text:

```
zgetdump --help
```

or

```
zgetdump -h
```

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This 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 time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

This information is for planning purposes only. The information herein is subject to change before the products described become available.

Trademarks

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

ECKD
IBM
S/390

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

LINUX is a registered trademark of Linus Torvalds and others.

Other company, product, and service names may be trademarks or service marks of others.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

LNUX-1108-00