LINUX for S/390 - kernel loadable module

LINUX® for S/390®
LCS Device Driver

LINUX for S/390 - kernel loadable module

LINUX® for S/390®
LCS Device Driver

Fifth Edition (May 2000)

This edition applies to the third release of the LINUX for S/390 kernel 2.2.15 patch (made in May 2000) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this book

This book provides information about the LINUX for S/390 LAN channel station (LCS) device driver.

Version 2.2.15 of the LINUX kernel is ported in this release of LINUX for S/390. If you are using a later version of the kernel, the kernel parameter may be different to the parameter described in this document.

This document is an overview of the device driver, anyone requiring more detailed information should read the LINUX for S/390-specific documentation in the kernel source tree.

How this book is organized

This document consists of one chapter describing the LINUX for S/390 LCS device driver. Note that only the differences between the 'standard' LINUX implementations and LINUX for S/390 are described.

Who should read this book

This book is intended for:

- Users who want to utilize the LINUX operating system on their ESA/390 architecture
- Distributors who want to build a LINUX for S/390 distribution
- Developers who want to participate in the enhancement of the LINUX for S/390 kernel

Ideally, these readers should be familiar with LINUX installation on 'standard' platforms, and have an understanding of the ESA/390 architecture. However, other readers who are interested in the installation of LINUX onto the ESA/390 architecture can use this document to gain an overview of the LCS device driver.

Assumptions

The following general assumptions are made about your background knowledge:

- You have an understanding of LINUX and S/390 terminology
- You are familiar with LINUX device driver descriptions
- You are familiar with the S/390 devices attached to your system
- You have an understanding of basic computer architecture, operating systems, and programs

Summary of changes

This revision contains changes to support the LINUX for S/390 kernel loadable module for the LINUX kernel version 2.2.15.

Edition 5 changes

New Information

- Added information describing restrictions concerning running via NFS with the LCS driver.
- Added a note about the limited number (16) of devices supported by OSA/SF. Recommended that devices are not shared with production systems.

Changed Information

- References to *OSA/2* have been changed to *OSA-2*.

Edition 4 changes

New and Changed Information

- There is no new or changed information in Edition 4.

Edition 3 changes

New and Changed Information

- There is no new or changed information in Edition 3.

Edition 2 changes

New Information

- Added a new section describing the module parameter syntax.
- Added a new section describing a potential set up problem where the LCS device driver sometimes selects the same port twice with two different pairs of device numbers.
- Added a description of the module and kernel parameters that allow you to disable the auto-detection of OSA/2 devices.

Changed Information

- The document title and other references to the *OSA/2 device driver* have been changed to *LCS device driver*. The OSA/2 *card* is still referenced by its correct name.

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

LINUX for S/390 LCS device driver

This chapter describes the LINUX for S/390 LCS device driver. For more specific information about the device driver structure, see the documents in the kernel source tree at `.../linux/Documentation/s390`.

The LCS driver supports Ethernet and Token Ring. FDDI was also specified in the interface document but is currently untested.

The LCS network interface has two channels, one read channel and one write channel. This is very similar to the S/390 CTC interface. The read channel is recognized by having an even `cuu` number and model 0x3088. The write channel is recognized by the formula `write cuu=read cuu number + 1` and also a model of 0x3088. Only certain cuu types are supported so as not to clash with a CTC control unit type.

The driver always has a read outstanding on the read subchannel. This is used to receive command replies and network packets (these are differentiated by checking the type field in the LCS header structure). Any network packets that arrive during the startup and shutdown sequence have to be discarded. During normal network I/O, the driver will intermittently retry reads in order to permanently keep reads outstanding on the read channel. This is in case an -EBUSY or similar occurs, in which case the driver would stop receiving network packets).

The default is to use software statistics, with IP checksumming off, this improves performance, network hardware uses a CRC32 (CRC64 for FDDI) which should be adequate to guarantee integrity for normal use. However, financial institutions etc. might like the additional security of IP checksumming.

The additional CUU model types are can be added so that new LCS compatible cards can be supported even though we don't know about them at the time of writing the code.

Features

- Supports Ethernet and Token Ring
- Auto detects whether card is connected to Token Ring or Ethernet
- Configurable to compile as a module or directly into the kernel
- Can be configured from the kernel parameter line or via insmod parameters (if a module).

Module parameter syntax

The following are the LCS device driver module parameters:

use_hw_stats
Gets network statistics from LANSTAT LCS primitive as opposed to doing it in software. This is not used by MVS, it is not recommended, it does not appear to work, and it usually does not work

do_sw_ip_checksumming
Does IP checksumming on inbound packets. Normally not required because Ethernet CRC32 is usually more than adequate (except perhaps for financial institutions).

additional_model_info
Made up of sets of model/maximum relative adapter number pairs. For example:

```
insmod additional_model_info=0x70,3,0x71,5
```

will look for 3 ports on a 0x3088 model 0x70 and will also look for 5 ports on a 0x3088 model 0x71

devno_portno_pairs
Takes `devno,rel_adapter_no(port)` pairs. Relative adapter number of `-1` indicates that you should not use this adapter. This can be used to force certain chipid's to use a particular port number if the LCS protocol tells me the incorrect one (hopefully this will fix a few problems happening in the field). For example:

```
insmod lcs devno_portno_pairs=0x1c00,0,0x1c02,1,0x1d00,-1
```

Tells the LCS device driver to:

- only use `port 0` if available for the device numbers `0x1c00` and `0x1c01`
- only use `port 1` if available for the device numbers `0x1c02` and `0x1c03`
- do not under any circumstances use the device at `0x1d00` and `0x1d01` as an LCS device.

noauto
Set `noauto=1` if you want to set auto-detection to off. You then have to explicitly configure LCS devices with the `devno_portno_pairs` module parameter.

Kernel parameter syntax

If the LCS driver is compiled directly into the kernel, the LCS boot parameters are now as follows:

```
lcs=hw-stats,ip-checksumming,additional cu model,max rel adapter no. pairs
```

For example:

```
lcs 1,1,97,8
```

will collect network statistics from the hardware, IP checksumming will be on, model number is 0x61 (97 decimal), and an attempt is made to detect 8 ports on that model.

It is important that the parameters are entered in pairs (2, 4, 6 or 8 parameters) as the `cu model` and `max rel adapter no` are a pair.

lcs_devno=devno,rel_adapter_no pairs
is identical to the `devno_portno_pairs` parameter described for modules. The name `lcs_devno` is quite short because of limitations in the allowable length of kernel parameter names.

lcs_noauto
Put this parameter in the kernel parameter line if you want to set auto-detection to off.

Limitations

- FDDI is untested and the code shipped with kernel patch 2.2.15 is unlikely to work on FDDI
- Because LCS does not appear to tell the driver that a port is busy, it is sometimes necessary to force the driver not to use the ports. The `devno_portno_pairs` or `lcs_devno` kernel parameters are used to do this.
- The most problematic area for the code is starting up and shutting down the driver.

  This is primarily due to the fact that network packets can be received during the startup process before receiving the `lanstat` command to get the mac address. This can happen earlier if the card wasn't previously shut down properly.

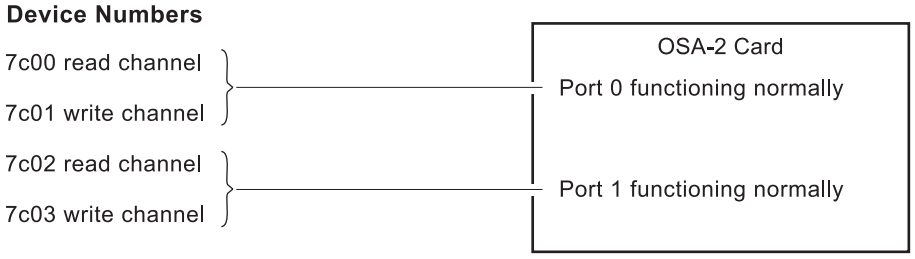
  If the card is being very troublesome, use `ifconfig` to switch it on and off.

  If this fails, compile the driver as a module. Use `insmod` and `rmmod`, as these are guaranteed to call the startup and shutdown routines, whereas the kernel keeps a reference count (doing `ifconfig` up twice will call the startup routine only once).
- LINUX for S/390 cannot run with a root file system mounted via NFS when the network connectivity is established via LCS, because the LCS driver is delivered as an object-code-only module.
- To use OSA-2 devices when running LINUX for S/390 on a basic mode machine (no LPARs) you need to specify an `ipldelay=xyz` boot parameter. We recommend a value between `2m` and `5m` for `xyz` for the OSA-2 card to settle down after `LOAD`.
- Currently, there is only support for up to 16 Token Ring or Ethernet devices. However, we strongly recommend that you do not share devices with production systems.

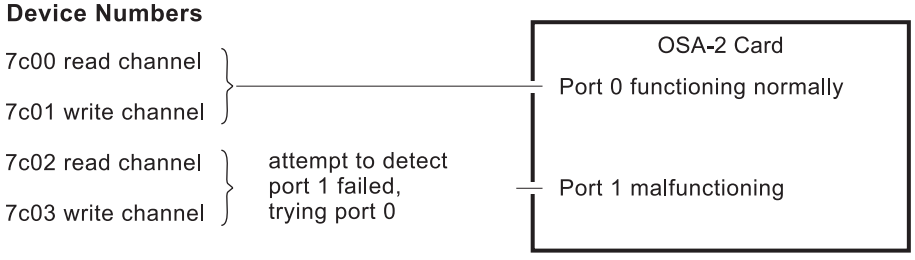
Common LCS set up problem

The LCS device driver sometimes has a set up problem. The same port on the OSA/2 card is taken twice by the LCS driver. This port can be communicated to via two or more pairs of device numbers. The driver attempts to determine the port number from the low byte of the device number, however the LCS microcode does not indicate that the port is already in use. If the first attempt is wrong (port already in use), the driver may use the same port twice with different pairs of device numbers. This can be better explained via the following diagrams.

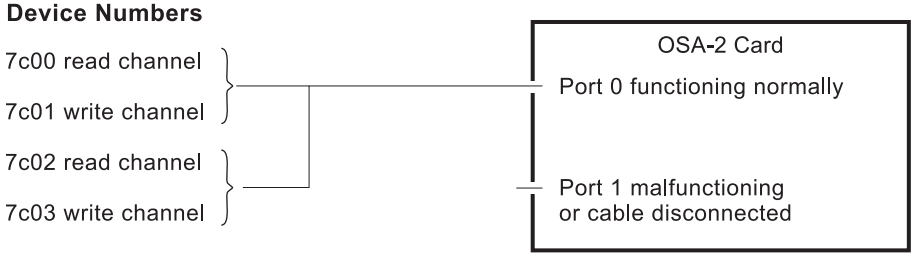
Normal case:



Abnormal case - part 1:



Abnormal case - part 2 (two or more connections on the same port):



The common symtoms of this problem are:

1. Duplicate mac addresses (most common)
2. Two or more network interfaces detected of the same type despite the fact that port X is configured as Token Ring and port Y as Ethernet
3. Disconnecting network cable from one port disables network traffic on two or more ports.

If this problem occurs, use the `devno_portno_pairs` parameter to force the problematical device numbers to specific ports.

Appendix. 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 about 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.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

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

ESA/390	OSA
IBM	OSA-2
MVS	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.