

Writing s390 channel device drivers

Cornelia Huck <cornelia.huck@de.ibm.com>

Writing s390 channel device drivers

by Cornelia Huck

Copyright © 2007 IBM Corp.

This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

For more details see the file COPYING in the source distribution of Linux.

Table of Contents

1. Introduction	1
2. The ccw bus	2
I/O functions for channel-attached devices	2
ccw devices	16
The channel-measurement facility	49
3. The ccwgroup bus	55
ccw group devices	55
4. Generic interfaces	65
register_adapter_interrupt	66
unregister_adapter_interrupt	67
airq_iv_create	68
airq_iv_release	69
airq_iv_alloc	70
airq_iv_free	71
airq_iv_scan	72

Chapter 1. Introduction

This document describes the interfaces available for device drivers that drive s390 based channel attached I/O devices. This includes interfaces for interaction with the hardware and interfaces for interacting with the common driver core. Those interfaces are provided by the s390 common I/O layer.

The document assumes a familiarity with the technical terms associated with the s390 channel I/O architecture. For a description of this architecture, please refer to the "z/Architecture: Principles of Operation", IBM publication no. SA22-7832.

While most I/O devices on a s390 system are typically driven through the channel I/O mechanism described here, there are various other methods (like the diag interface). These are out of the scope of this document.

Some additional information can also be found in the kernel source under Documentation/s390/driver-model.txt.

Chapter 2. The ccw bus

The ccw bus typically contains the majority of devices available to a s390 system. Named after the channel command word (ccw), the basic command structure used to address its devices, the ccw bus contains so-called channel attached devices. They are addressed via I/O subchannels, visible on the css bus. A device driver for channel-attached devices, however, will never interact with the subchannel directly, but only via the I/O device on the ccw bus, the ccw device.

I/O functions for channel-attached devices

Some hardware structures have been translated into C structures for use by the common I/O layer and device drivers. For more information on the hardware structures represented here, please consult the Principles of Operation.

Name

struct erw_eadm — EADM Subchannel extended report word

Synopsis

```
struct erw_eadm {  
    __u32 b:1;  
    __u32 r:1;  
};
```

Members

b aob error

r arsb error

Name

struct esw0 — Format 0 Extended Status Word (ESW)

Synopsis

```
struct esw0 {  
    struct sublog sublog;  
    struct erw erw;  
    __u32 faddr[2];  
    __u32 saddr;  
};
```

Members

sublog	subchannel logout
erw	extended report word
faddr[2]	failing storage address
saddr	secondary ccw address

Name

struct esw1 — Format 1 Extended Status Word (ESW)

Synopsis

```
struct esw1 {  
    __u8 zero0;  
    __u8 lpum;  
    __u16 zero16;  
    struct erw erw;  
    __u32 zeros[3];  
};
```

Members

zero0	reserved zeros
lpum	last path used mask
zero16	reserved zeros
erw	extended report word
zeros[3]	three fullwords of zeros

Name

struct esw2 — Format 2 Extended Status Word (ESW)

Synopsis

```
struct esw2 {  
    __u8 zero0;  
    __u8 lpum;  
    __u16 dcti;  
    struct erw erw;  
    __u32 zeros[3];  
};
```

Members

zero0	reserved zeros
lpum	last path used mask
dcti	device-connect-time interval
erw	extended report word
zeros[3]	three fullwords of zeros

Name

`pathmask_to_pos` — find the position of the left-most bit in a pathmask

Synopsis

```
u8 pathmask_to_pos (u8 mask);
```

Arguments

mask pathmask with at least one bit set

ccw devices

Devices that want to initiate channel I/O need to attach to the ccw bus. Interaction with the driver core is done via the common I/O layer, which provides the abstractions of ccw devices and ccw device drivers.

The functions that initiate or terminate channel I/O all act upon a ccw device structure. Device drivers must not bypass those functions or strange side effects may happen.

int_class interruption class to use for accounting interrupts

Name

`ccw_device_siosl` — initiate logging

Synopsis

```
int ccw_device_siosl (struct ccw_device * cdev);
```

Arguments

cdev ccw device

Description

This function is used to invoke model-dependent logging within the channel subsystem.

Chapter 3. The ccwgroup bus

The ccwgroup bus only contains artificial devices, created by the user. Many networking devices (e.g. qeth) are in fact composed of several ccw devices (like read, write and data channel for qeth). The ccwgroup bus provides a mechanism to create a meta-device which contains those ccw devices as slave devices and can be associated with the netdevice.

ccw group devices

Chapter 4. Generic interfaces

Some interfaces are available to other drivers that do not necessarily have anything to do with the busses described above, but still are indirectly using basic infrastructure in the common I/O layer. One example is the support for adapter interrupts.

Name

`unregister_adapter_interrupt` — unregister adapter interrupt handler

Synopsis

```
void unregister_adapter_interrupt (struct airq_struct * airq);
```

Arguments

airq pointer to adapter interrupt descriptor

