

# [PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-08/msg08639.html>

---

- *From:* David Howells <[dhowells@xxxxxxxxxx](mailto:dhowells@xxxxxxxxxx)>
  - *Date:* Thu, 24 Aug 2006 22:33:39 +0100
- 

From: David Howells <[dhowells@xxxxxxxxxx](mailto:dhowells@xxxxxxxxxx)>

Make it possible to disable the block layer. Not all embedded devices require it, some can make do with just JFFS2, NFS, ramfs, etc – none of which require the block layer to be present.

This patch does the following:

(\*) Introduces CONFIG\_BLOCK to disable the block layer, buffering and blockdev support.

(\*) Adds dependencies on CONFIG\_BLOCK to any configuration item that controls an item that uses the block layer. This includes:

(\*) Block I/O tracing.

(\*) Disk partition code.

(\*) All filesystems that are block based, eg: Ext3, ReiserFS, ISOFS.

(\*) The SCSI layer. As far as I can tell, even SCSI chardevs use the block layer to do scheduling.

(\*) Various block-based device drivers, such as IDE, the old CDROM drivers and USB storage.

(\*) MTD blockdev handling and FTL.

(\*) JFFS – which uses set\_bdev\_super(), something it could avoid doing by taking a leaf out of JFFS2's book.

(\*) Makes most of the contents of linux/blkdev.h, linux/buffer\_head.h and linux/elevator.h contingent on CONFIG\_BLOCK being set. sector\_div() is, however, still used in places, and so is still available.

(\*) Also made contingent are the contents of linux/mpage.h, linux/genhd.h and parts of linux/fs.h.

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

- (\*) Makes a number of files in fs/ contingent on CONFIG\_BLOCK.
- (\*) Makes mm/bounce.c (bounce buffering) contingent on CONFIG\_BLOCK.
- (\*) set\_page\_dirty() doesn't call \_\_set\_page\_dirty\_buffers() if CONFIG\_BLOCK is not enabled.
- (\*) fs/no-block.c is created to hold out-of-line stubs and things that are required when CONFIG\_BLOCK is not set:
- (\*) Default blockdev file operations (to give error ENODEV on opening).
- (\*) Makes some /proc changes:
- (\*) /proc/devices does not list any blockdevs.
- (\*) /proc/diskstats and /proc/partitions are contingent on CONFIG\_BLOCK.
- (\*) Makes some compat ioctl handling contingent on CONFIG\_BLOCK.
- (\*) If CONFIG\_BLOCK is not defined, makes sys\_quotactl() return -ENODEV if given command other than Q\_SYNC or if a special device is specified.
- (\*) In init/do\_mounts.c, no reference is made to the blockdev routines if CONFIG\_BLOCK is not defined. This does not prohibit NFS roots or JFFS2.
- (\*) The bdflush, ioprio\_set and ioprio\_get syscalls can now be absent (return error ENOSYS by way of cond\_syscall if so).
- (\*) The seclvl\_bd\_claim() and seclvl\_bd\_release() security calls do nothing if CONFIG\_BLOCK is not set, since they can't then happen.

Signed-Off-By: David Howells <dhowells@xxxxxxxxxx>

---

```
block/Kconfig | 14 ++++++++
block/Kconfig.iosched | 3 ++
block/Makefile | 2 +
drivers/block/Kconfig | 4 +++
drivers/cdrom/Kconfig | 2 +
drivers/char/Kconfig | 1 +
drivers/char/random.c | 4 +++
drivers/ide/Kconfig | 4 +++
drivers/ieee1394/Kconfig | 2 +
drivers/infiniband/ulp/iser/Kconfig | 2 +
drivers/infiniband/ulp/srp/Kconfig | 2 +
drivers/md/Kconfig | 3 ++
drivers/message/i2o/Kconfig | 2 +
drivers/mmc/Kconfig | 2 +
drivers/mmc/Makefile | 3 +-
drivers/mtd/Kconfig | 12 +++++---
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
drivers/mtd/devices/Kconfig | 2 +
drivers/s390/block/Kconfig | 2 +
drivers/scsi/Kconfig | 8 +++--
drivers/usb/storage/Kconfig | 2 +
fs/Kconfig | 32 ++++++++-----
fs/Makefile | 14 ++++++--
fs/compat_ioctl.c | 18 ++++++++
fs/no-block.c | 22 ++++++++
fs/partitions/Makefile | 2 +
fs/proc/proc_misc.c | 11 ++++++
fs/quota.c | 44 ++++++++-----
fs/super.c | 4 +++
fs/xfs/Kconfig | 1 +
include/linux/blkdev.h | 52 ++++++++-----
include/linux/buffer_head.h | 16 ++++++++
include/linux/compat_ioctl.h | 2 +
include/linux/elevator.h | 3 ++
include/linux/fs.h | 25 ++++++++--
include/linux/genhd.h | 4 +++
include/linux/mpage.h | 3 ++
include/linux/raid/md.h | 3 ++
include/linux/raid/md_k.h | 3 ++
include/scsi/scsi_tcq.h | 3 +-
init/Kconfig | 2 +
init/do_mounts.c | 13 ++++++--
kernel/sys_ni.c | 5 +++
mm/Makefile | 2 +
mm/filemap.c | 4 +++
mm/migrate.c | 2 +
mm/page-writeback.c | 8 +++--
security/seclvl.c | 4 +++
47 files changed, 308 insertions(+), 70 deletions(-)
```

```
diff --git a/block/Kconfig b/block/Kconfig
index b6f5f0a..9cc0d0b 100644
--- a/block/Kconfig
+++ b/block/Kconfig
@@ -1,6 +1,18 @@
#
# Block layer core configuration
#
+config BLOCK
+ bool "Enable the block layer"
+ default y
+ help
+ This permits the block layer to be removed from the kernel if it's not
+ needed (on some embedded devices for example). If this option is
+ disabled, then blockdev files will become unusable and some
+ filesystems (such as ext3) will become unavailable. Say Y here unless
+ you know you really don't want to mount disks and suchlike.
+
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
+if BLOCK
+
#XXX – it makes sense to enable this only for 32-bit subarch's, not for x86_64
#for instance.
config LBD
@@ -33,4 +45,6 @@ config LSF
```

If unsure, say Y.

```
+endif
+
source block/Kconfig.iosched
diff --git a/block/Kconfig.iosched b/block/Kconfig.iosched
index 48d090e..903f0d3 100644
--- a/block/Kconfig.iosched
+++ b/block/Kconfig.iosched
@@ -1,3 +1,4 @@
+if BLOCK
```

menu "IO Schedulers"

```
@@ -67,3 +68,5 @@ config DEFAULT_IOSCHED
default "noop" if DEFAULT_NOOP
```

endmenu

```
+
+endif
diff --git a/block/Makefile b/block/Makefile
index c05de0e..085e967 100644
--- a/block/Makefile
+++ b/block/Makefile
@@ -2,7 +2,7 @@ #
# Makefile for the kernel block layer
#
```

```
-obj-y := elevator.o ll_rw_blk.o ioctnl.o genhd.o scsi_ioctnl.o
+obj-$(CONFIG_BLOCK) := elevator.o ll_rw_blk.o ioctnl.o genhd.o scsi_ioctnl.o
```

```
obj-$(CONFIG_IOSCHED_NOOP) += noop-iosched.o
obj-$(CONFIG_IOSCHED_AS) += as-iosched.o
diff --git a/drivers/block/Kconfig b/drivers/block/Kconfig
index b5382ce..422e31d 100644
--- a/drivers/block/Kconfig
+++ b/drivers/block/Kconfig
@@ -2,6 +2,8 @@ #
# Block device driver configuration
#
```

```
+if BLOCK
+
menu "Block devices"
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
config BLK_DEV_FD
@@ -468,3 +470,5 @@ config ATA_OVER_ETH
devices like the Coraid EtherDrive (R) Storage Blade.

endmenu
+
+endif
diff --git a/drivers/cdrom/Kconfig b/drivers/cdrom/Kconfig
index ff5652d..4b12e90 100644
--- a/drivers/cdrom/Kconfig
+++ b/drivers/cdrom/Kconfig
@@ -3,7 +3,7 @@ # CDROM driver configuration
#

menu "Old CD-ROM drivers (not SCSI, not IDE)"
- depends on ISA
+ depends on ISA && BLOCK

config CD_NO_IDESCSI
bool "Support non-SCSI/IDE/ATAPI CDROM drives"
diff --git a/drivers/char/Kconfig b/drivers/char/Kconfig
index c40e487..b9c6777 100644
--- a/drivers/char/Kconfig
+++ b/drivers/char/Kconfig
@@ -984,6 +984,7 @@ config GPIO_VR41XX

config RAW_DRIVER
tristate "RAW driver (/dev/raw/rawN) (OBSOLETE)"
+ depends on BLOCK
help
The raw driver permits block devices to be bound to /dev/raw/rawN.
Once bound, I/O against /dev/raw/rawN uses efficient zero-copy I/O.
diff --git a/drivers/char/random.c b/drivers/char/random.c
index 4c3a5ca..b430a12 100644
--- a/drivers/char/random.c
+++ b/drivers/char/random.c
@@ -655,6 +655,7 @@ void add_interrupt_randomness(int irq)
add_timer_randomness(irq_timer_state[irq], 0x100 + irq);
}

+#ifdef CONFIG_BLOCK
void add_disk_randomness(struct gendisk *disk)
{
if (!disk || !disk->random)
@@ -667,6 +668,7 @@ void add_disk_randomness(struct gendisk
}

EXPORT_SYMBOL(add_disk_randomness);
+endif
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
#define EXTRACT_SIZE 10

@@ -918,6 +920,7 @@ void rand_initialize_irq(int irq)
}
}

+#ifdef CONFIG_BLOCK
void rand_initialize_disk(struct gendisk *disk)
{
struct timer_rand_state *state;
@@ -932,6 +935,7 @@ void rand_initialize_disk(struct gendisk
disk->random = state;
}
}
+#endif

static ssize_t
random_read(struct file * file, char __user * buf, size_t nbytes, loff_t *ppos)
diff --git a/drivers/ide/Kconfig b/drivers/ide/Kconfig
index b6fb167..69d627b 100644
--- a/drivers/ide/Kconfig
+++ b/drivers/ide/Kconfig
@@ -4,6 +4,8 @@ #
# Andre Hedrick <andre@xxxxxxxxxxxxxx>
#

+if BLOCK
+
menu "ATA/ATAPI/MFM/RLL support"

config IDE
@@ -1082,3 +1084,5 @@ config BLK_DEV_HD
endif

endmenu
+
+endif
diff --git a/drivers/ieee1394/Kconfig b/drivers/ieee1394/Kconfig
index 1867375..c9d84b9 100644
--- a/drivers/ieee1394/Kconfig
+++ b/drivers/ieee1394/Kconfig
@@ -122,7 +122,7 @@ config IEEE1394_VIDEO1394

config IEEE1394_SBP2
tristate "SBP-2 support (Harddisks etc.)"
- depends on IEEE1394 && SCSI && (PCI || BROKEN)
+ depends on IEEE1394 && BLOCK && SCSI && (PCI || BROKEN)
help
This option enables you to use SBP-2 devices connected to your IEEE
1394 bus. SBP-2 devices include harddrives and DVD devices.
diff --git a/drivers/infiniband/ulp/iser/Kconfig b/drivers/infiniband/ulp/iser/Kconfig
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
index fead87d..f945953 100644
--- a/drivers/infiniband/ulp/iser/Kconfig
+++ b/drivers/infiniband/ulp/iser/Kconfig
@@ -1,6 +1,6 @@
config INFINIBAND_ISER
tristate "ISCSI RDMA Protocol"
- depends on INFINIBAND && SCSI
+ depends on INFINIBAND && BLOCK && SCSI
select SCSI_ISCSI_ATTRS
---help---
Support for the ISCSI RDMA Protocol over InfiniBand. This
diff ---git a/drivers/infiniband/ulp/srp/Kconfig b/drivers/infiniband/ulp/srp/Kconfig
index 8fe3be4..63d7d5a 100644
--- a/drivers/infiniband/ulp/srp/Kconfig
+++ b/drivers/infiniband/ulp/srp/Kconfig
@@ -1,6 +1,6 @@
config INFINIBAND_SRP
tristate "InfiniBand SCSI RDMA Protocol"
- depends on INFINIBAND && SCSI
+ depends on INFINIBAND && BLOCK && SCSI
---help---
Support for the SCSI RDMA Protocol over InfiniBand. This
allows you to access storage devices that speak SRP over
diff ---git a/drivers/md/Kconfig b/drivers/md/Kconfig
index bf869ed..1e91f90 100644
--- a/drivers/md/Kconfig
+++ b/drivers/md/Kconfig
@@ -2,6 +2,8 @@ #
# Block device driver configuration
#

+if CONFIG_BLOCK
+
menu "Multi-device support (RAID and LVM)"

config MD
@@ -251,3 +253,4 @@ config DM_MULTIPATH EMC

endmenu

+endif
diff ---git a/drivers/message/i2o/Kconfig b/drivers/message/i2o/Kconfig
index fef6771..6443392 100644
--- a/drivers/message/i2o/Kconfig
+++ b/drivers/message/i2o/Kconfig
@@ -88,7 +88,7 @@ config I2O_BUS

config I2O_BLOCK
tristate "I2O Block OSM"
- depends on I2O
+ depends on I2O && BLOCK
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

---help---

Include support for the I2O Block OSM. The Block OSM presents disk and other structured block devices to the operating system. If you

diff --git a/drivers/mmc/Kconfig b/drivers/mmc/Kconfig  
index 45bcf09..f540bd8 100644

--- a/drivers/mmc/Kconfig  
+++ b/drivers/mmc/Kconfig  
@@ -21,7 +21,7 @@ config MMC\_DEBUG

config MMC\_BLOCK  
tristate "MMC block device driver"  
- depends on MMC  
+ depends on MMC && BLOCK  
default y

help

Say Y here to enable the MMC block device driver support.

diff --git a/drivers/mmc/Makefile b/drivers/mmc/Makefile  
index d2957e3..b1f6e03 100644

--- a/drivers/mmc/Makefile  
+++ b/drivers/mmc/Makefile  
@@ -24,7 +24,8 @@ obj-\$(CONFIG\_MMC\_AU1X) += au1xmmc.o  
obj-\$(CONFIG\_MMC\_OMAP) += omap.o  
obj-\$(CONFIG\_MMC\_AT91RM9200) += at91\_mci.o

-mmc\_core-y := mmc.o mmc\_queue.o mmc\_sysfs.o  
+mmc\_core-y := mmc.o mmc\_sysfs.o  
+mmc\_core-\$(CONFIG\_BLOCK) += mmc\_queue.o

ifeq (\$(CONFIG\_MMC\_DEBUG),y)  
EXTRA\_CFLAGS += -DDEBUG  
diff --git a/drivers/mtd/Kconfig b/drivers/mtd/Kconfig  
index 1344ad7..188cd37 100644  
--- a/drivers/mtd/Kconfig  
+++ b/drivers/mtd/Kconfig  
@@ -166,7 +166,7 @@ config MTD\_CHAR

config MTD\_BLOCK  
tristate "Caching block device access to MTD devices"  
- depends on MTD  
+ depends on MTD && BLOCK  
---help---

Although most flash chips have an erase size too large to be useful as block devices, it is possible to use MTD devices which are based

@@ -188,7 +188,7 @@ config MTD\_BLOCK

config MTD\_BLOCK\_RO  
tristate "Readonly block device access to MTD devices"  
- depends on MTD\_BLOCK!=y && MTD  
+ depends on MTD\_BLOCK!=y && MTD && BLOCK  
help

This allows you to mount read-only file systems (such as cramfs)

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

from an MTD device, without the overhead (and danger) of the caching  
@@ -199,7 +199,7 @@ config MTD\_BLOCK\_RO

config FTL

tristate "FTL (Flash Translation Layer) support"

- depends on MTD

+ depends on MTD && BLOCK

---help---

This provides support for the original Flash Translation Layer which is part of the PCMCIA specification. It uses a kind of pseudo-

@@ -215,7 +215,7 @@ config FTL

config NFTL

tristate "NFTL (NAND Flash Translation Layer) support"

- depends on MTD

+ depends on MTD && BLOCK

---help---

This provides support for the NAND Flash Translation Layer which is used on M-Systems' DiskOnChip devices. It uses a kind of pseudo-

@@ -238,7 +238,7 @@ config NFTL\_RW

config INFTL

tristate "INFTL (Inverse NAND Flash Translation Layer) support"

- depends on MTD

+ depends on MTD && BLOCK

---help---

This provides support for the Inverse NAND Flash Translation Layer which is used on M-Systems' newer DiskOnChip devices. It

@@ -255,7 +255,7 @@ config INFTL

config RFD\_FTL

tristate "Resident Flash Disk (Flash Translation Layer) support"

- depends on MTD

+ depends on MTD && BLOCK

---help---

This provides support for the flash translation layer known as the Resident Flash Disk (RFD), as used by the Embedded BIOS  
diff --git a/drivers/mtd/devices/Kconfig b/drivers/mtd/devices/Kconfig

index 16c02b5..440f685 100644

--- a/drivers/mtd/devices/Kconfig

+++ b/drivers/mtd/devices/Kconfig

@@ -136,7 +136,7 @@ config MTD\_RAM\_ABS\_POS

config MTD\_BLOCK2MTD

tristate "MTD using block device"

- depends on MTD

+ depends on MTD && BLOCK

help

This driver allows a block device to appear as an MTD. It would generally be used in the following cases:

diff --git a/drivers/s390/block/Kconfig b/drivers/s390/block/Kconfig

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
index 929d6ff..b250c53 100644
--- a/drivers/s390/block/Kconfig
+++ b/drivers/s390/block/Kconfig
@@ -1,4 +1,4 @@
-if S390
+if S390 && BLOCK
```

```
comment "S/390 block device drivers"
depends on S390
diff --git a/drivers/scsi/Kconfig b/drivers/scsi/Kconfig
index 96a81cd..afcbe19 100644
--- a/drivers/scsi/Kconfig
+++ b/drivers/scsi/Kconfig
@@ -3,11 +3,13 @@ menu "SCSI device support"
config RAID_ATTRS
tristate "RAID Transport Class"
default n
+ depends on BLOCK
---help---
Provides RAID
```

```
config SCSI
tristate "SCSI device support"
+ depends on BLOCK
---help---
If you want to use a SCSI hard disk, SCSI tape drive, SCSI CD-ROM or
any other SCSI device under Linux, say Y and make sure that you know
@@ -43,7 +45,7 @@ comment "SCSI support type (disk, tape,
```

```
config BLK_DEV_SD
tristate "SCSI disk support"
- depends on SCSI
+ depends on SCSI && BLOCK
---help---
If you want to use SCSI hard disks, Fibre Channel disks,
USB storage or the SCSI or parallel port version of
@@ -98,7 +100,7 @@ config CHR_DEV_OSST
```

```
config BLK_DEV_SR
tristate "SCSI CDROM support"
- depends on SCSI
+ depends on SCSI && BLOCK
---help---
If you want to use a SCSI or FireWire CD-ROM under Linux,
say Y and read the SCSI-HOWTO and the CDROM-HOWTO at
@@ -473,7 +475,7 @@ source "drivers/scsi/megaraid/Kconfig.me
```

```
config SCSI_SATA
tristate "Serial ATA (SATA) support"
- depends on SCSI
+ depends on SCSI && BLOCK
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

help

This driver family supports Serial ATA host controllers and devices.

diff --git a/drivers/usb/storage/Kconfig b/drivers/usb/storage/Kconfig

index be9eec2..578aa13 100644

--- a/drivers/usb/storage/Kconfig

+++ b/drivers/usb/storage/Kconfig

@@ -8,7 +8,7 @@ comment "may also be needed; see USB\_STO

config USB\_STORAGE

tristate "USB Mass Storage support"

- depends on USB

+ depends on USB && BLOCK

select SCSI

---help---

Say Y here if you want to connect USB mass storage devices to your

diff --git a/fs/Kconfig b/fs/Kconfig

index 3f00a9f..dc5e69b 100644

--- a/fs/Kconfig

+++ b/fs/Kconfig

@@ -4,6 +4,8 @@ #

menu "File systems"

+if BLOCK

+

config EXT2\_FS

tristate "Second extended fs support"

help

@@ -383,8 +385,11 @@ config MINIX\_FS

partition (the one containing the directory /) cannot be compiled as a module.

+endif

+

config ROMFS\_FS

tristate "ROM file system support"

+ depends on BLOCK

---help---

This is a very small read-only file system mainly intended for initial ram disks of installation disks, but it could be used for

@@ -530,6 +535,7 @@ config FUSE\_FS

If you want to develop a userspace FS, or if you want to use a filesystem based on FUSE, answer Y or M.

+if BLOCK

menu "CD-ROM/DVD Filesystems"

config ISO9660\_FS

@@ -597,7 +603,9 @@ config UDF\_NLS

depends on (UDF\_FS=m && NLS) || (UDF\_FS=y && NLS=y)

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
endmenu
+endif

+if BLOCK
menu "DOS/FAT/NT Filesystems"

config FAT_FS
@@ -782,6 +790,7 @@ config NTFS_RW
It is perfectly safe to say N here.

endmenu
+endif

menu "Pseudo filesystems"

@@ -907,7 +916,7 @@ menu "Miscellaneous filesystems"

config ADFS_FS
tristate "ADFS file system support (EXPERIMENTAL)"
- depends on EXPERIMENTAL
+ depends on BLOCK && EXPERIMENTAL
help
The Acorn Disc Filing System is the standard file system of the
RiscOS operating system which runs on Acorn's ARM-based Risc PC
@@ -935,7 +944,7 @@ config ADFS_FS_RW

config AFFS_FS
tristate "Amiga FFS file system support (EXPERIMENTAL)"
- depends on EXPERIMENTAL
+ depends on BLOCK && EXPERIMENTAL
help
The Fast File System (FFS) is the common file system used on hard
disks by Amiga(tm) systems since AmigaOS Version 1.3 (34.20). Say Y
@@ -957,7 +966,7 @@ config AFFS_FS

config HFS_FS
tristate "Apple Macintosh file system support (EXPERIMENTAL)"
- depends on EXPERIMENTAL
+ depends on BLOCK && EXPERIMENTAL
select NLS
help
If you say Y here, you will be able to mount Macintosh-formatted
@@ -970,6 +979,7 @@ config HFS_FS

config HFSPLUS_FS
tristate "Apple Extended HFS file system support"
+ depends on BLOCK
select NLS
select NLS_UTF8
help
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

@@ -983,7 +993,7 @@ config HFSPLUS\_FS

config BEFS\_FS

tristate "BeOS file system (BeFS) support (read only) (EXPERIMENTAL)"

- depends on EXPERIMENTAL

+ depends on BLOCK && EXPERIMENTAL

select NLS

help

The BeOS File System (BeFS) is the native file system of Be, Inc's

@@ -1010,7 +1020,7 @@ config BEFS\_DEBUG

config BFS\_FS

tristate "BFS file system support (EXPERIMENTAL)"

- depends on EXPERIMENTAL

+ depends on BLOCK && EXPERIMENTAL

help

Boot File System (BFS) is a file system used under SCO UnixWare to

allow the bootloader access to the kernel image and other important

@@ -1032,7 +1042,7 @@ config BFS\_FS

config EFS\_FS

tristate "EFS file system support (read only) (EXPERIMENTAL)"

- depends on EXPERIMENTAL

+ depends on BLOCK && EXPERIMENTAL

help

EFS is an older file system used for non-ISO9660 CD-ROMs and hard

disk partitions by SGI's IRIX operating system (IRIX 6.0 and newer

@@ -1047,7 +1057,7 @@ config EFS\_FS

config JFFS\_FS

tristate "Journaling Flash File System (JFFS) support"

- depends on MTD

+ depends on MTD && BLOCK

help

JFFS is the Journaling Flash File System developed by Axis

Communications in Sweden, aimed at providing a crash/powerdown-safe

@@ -1232,6 +1242,7 @@ endchoice

config CRAMFS

tristate "Compressed ROM file system support (cramfs)"

+ depends on BLOCK

select ZLIB\_INFLATE

help

Saying Y here includes support for CramFs (Compressed ROM File

@@ -1251,6 +1262,7 @@ config CRAMFS

config VXFS\_FS

tristate "FreeVxFS file system support (VERITAS VxFS(TM) compatible)"

+ depends on BLOCK

help

FreeVxFS is a file system driver that support the VERITAS VxFS(TM)

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

file system format. VERITAS VxFS(TM) is the standard file system  
@@ -1268,6 +1280,7 @@ config VXFS\_FS

config HPFS\_FS  
tristate "OS/2 HPFS file system support"  
+ depends on BLOCK  
help  
OS/2 is IBM's operating system for PC's, the same as Warp, and HPFS  
is the file system used for organizing files on OS/2 hard disk  
@@ -1284,6 +1297,7 @@ config HPFS\_FS

config QNX4FS\_FS  
tristate "QNX4 file system support (read only)"  
+ depends on BLOCK  
help  
This is the file system used by the real-time operating systems  
QNX 4 and QNX 6 (the latter is also called QNX RTP).  
@@ -1311,6 +1325,7 @@ config QNX4FS\_RW

config SYSV\_FS  
tristate "System V/Xenix/V7/Coherent file system support"  
+ depends on BLOCK  
help  
SCO, Xenix and Coherent are commercial Unix systems for Intel  
machines, and Version 7 was used on the DEC PDP-11. Saying Y  
@@ -1349,6 +1364,7 @@ config SYSV\_FS

config UFS\_FS  
tristate "UFS file system support (read only)"  
+ depends on BLOCK  
help  
BSD and derivate versions of Unix (such as SunOS, FreeBSD, NetBSD,  
OpenBSD and NeXTstep) use a file system called UFS. Some System V  
@@ -1923,11 +1939,13 @@ config 9P\_FS

endmenu

+if BLOCK  
menu "Partition Types"

source "fs/partitions/Kconfig"

endmenu  
+endif

source "fs/nls/Kconfig"

diff --git a/fs/Makefile b/fs/Makefile  
index 8913542..8071c64 100644  
--- a/fs/Makefile  
+++ b/fs/Makefile

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

@@ -5,12 +5,18 @@ # 14 Sep 2000, Christoph Hellwig <hch@in

# Rewritten to use lists instead of if-statements.

#

```
-obj-y := open.o read_write.o file_table.o buffer.o bio.o super.o \  
- block_dev.o char_dev.o stat.o exec.o pipe.o namei.o fcntl.o \  
+obj-y := open.o read_write.o file_table.o super.o \  
+ char_dev.o stat.o exec.o pipe.o namei.o fcntl.o \  
ioctl.o readdir.o select.o fifo.o locks.o dcache.o inode.o \  
attr.o bad_inode.o file.o filesystems.o namespace.o aio.o \  
- seq_file.o xattr.o libfs.o fs-writeback.o mpage.o direct-io.o \  
- ioprio.o pnode.o drop_caches.o splice.o sync.o \  
+ seq_file.o xattr.o libfs.o fs-writeback.o \  
+ pnode.o drop_caches.o splice.o sync.o \  
+ \  
+ifeq ($(CONFIG_BLOCK),y) \  
+obj-y += buffer.o bio.o block_dev.o direct-io.o mpage.o ioprio.o \  
+else \  
+obj-y += no-block.o \  
+endif
```

```
obj-$(CONFIG_INOTIFY) += inotify.o \  
obj-$(CONFIG_INOTIFY_USER) += inotify_user.o \  
diff --git a/fs/compat_ioctl.c b/fs/compat_ioctl.c \  
index 7b8a9b4..af160e9 100644 \  
--- a/fs/compat_ioctl.c \  
+++ b/fs/compat_ioctl.c \  
@@ -645,6 +645,7 @@ out: \  
} \  
#endif
```

```
+#ifdef CONFIG_BLOCK \  
struct hd_geometry32 { \  
    unsigned char heads; \  
    unsigned char sectors; \  
    @@ -869,6 +870,7 @@ static int sg_grt_trans(unsigned int fd, \  
} \  
return err; \  
} \  
+#endif /* CONFIG_BLOCK */
```

```
struct sock_fprog32 { \  
    unsigned short len; \  
    @@ -992,6 +994,7 @@ static int ppp_ioctl_trans(unsigned int \  
}
```

```
+#ifdef CONFIG_BLOCK \  
struct mtget32 { \  
    compat_long_t mt_type; \  
    compat_long_t mt_resid;
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
@@ -1164,6 +1167,7 @@ static int cdrom_ioctl_trans(unsigned in

return err;
}
+#endif /* CONFIG_BLOCK */

extern int tty_ioctl(struct inode * inode, struct file * file, unsigned int cmd, unsigned long arg);

@@ -1493,6 +1497,7 @@ ret_einval(unsigned int fd, unsigned int
return -EINVAL;
}

+#ifdef CONFIG_BLOCK
static int broken_blkgetsize(unsigned int fd, unsigned int cmd, unsigned long arg)
{
/* The mkswap binary hard codes it to Intel value :-( */
@@ -1527,12 +1532,14 @@ static int blkpg_ioctl_trans(unsigned in

return sys_ioctl(fd, cmd, (unsigned long)a);
}
+#endif

static int ioc_settimeout(unsigned int fd, unsigned int cmd, unsigned long arg)
{
return rw_long(fd, AUTOFS_IOC_SETTIMEOUT, arg);
}

+#ifdef CONFIG_BLOCK
/* Fix sizeof(sizeof()) breakage */
#define BLKBSZGET_32 _IOR(0x12,112,int)
#define BLKBSZSET_32 _IOW(0x12,113,int)
@@ -1553,6 +1560,7 @@ static int do_blkgetsize64(unsigned int
{
return sys_ioctl(fd, BLKGETSIZE64, (unsigned long)compat_ptr(arg));
}
+#endif

/* Bluetooth ioctls */
#define HCIUARTSETPROTO _IOW('U', 200, int)
@@ -1573,6 +1581,7 @@ #define HIDPCONNDEL _IOW('H', 201, int)
#define HIDPGETCONNLIST _IOR('H', 210, int)
#define HIDPGETCONNINFO _IOR('H', 211, int)

+#ifdef CONFIG_BLOCK
struct floppy_struct32 {
compat_uint_t size;
compat_uint_t sect;
@@ -1897,6 +1906,7 @@ out:
kfree(karg);
return err;
}

```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
+#endif

struct mtd_oob_buf32 {
u_int32_t start;
@@ -1938,6 +1948,7 @@ static int mtd_rw_oob(unsigned int fd, u
return err;
}

+#ifdef CONFIG_BLOCK
struct raw32_config_request
{
compat_int_t raw_minor;
@@ -2002,6 +2013,7 @@ static int raw_ioctl(unsigned fd, unsign
}
return ret;
}
+#endif /* CONFIG_BLOCK */

struct serial_struct32 {
compat_int_t type;
@@ -2608,6 +2620,7 @@ HANDLE_IOCTL(SIOCBRDELIF, dev_ifsioc)
HANDLE_IOCTL(SIOCRTMSG, ret_einval)
HANDLE_IOCTL(SIOCGSTAMP, do_siocgstamp)
#endif
+#ifdef CONFIG_BLOCK
HANDLE_IOCTL(HDIO_GETGEO, hdio_getgeo)
HANDLE_IOCTL(BLKRAGET, w_long)
HANDLE_IOCTL(BLKGETSIZE, w_long)
@@ -2633,14 +2646,17 @@ HANDLE_IOCTL(FDGETFDCSTAT32, fd_ioctl_tr
HANDLE_IOCTL(FDWERRORGET32, fd_ioctl_trans)
HANDLE_IOCTL(SG_IO,sg_ioctl_trans)
HANDLE_IOCTL(SG_GET_REQUEST_TABLE, sg_grt_trans)
+#endif
HANDLE_IOCTL(PPPIOCGIDLE32, ppp_ioctl_trans)
HANDLE_IOCTL(PPPIOCSCOMPRESS32, ppp_ioctl_trans)
HANDLE_IOCTL(PPPIOCSPASS32, ppp_sock_fprog_ioctl_trans)
HANDLE_IOCTL(PPPIOCSACTIVE32, ppp_sock_fprog_ioctl_trans)
+#ifdef CONFIG_BLOCK
HANDLE_IOCTL(MTIOCGET32, mt_ioctl_trans)
HANDLE_IOCTL(MTIOCPOS32, mt_ioctl_trans)
HANDLE_IOCTL(CDROMREADAUDIO, cdrom_ioctl_trans)
HANDLE_IOCTL(CDROM_SEND_PACKET, cdrom_ioctl_trans)
+#endif
#define AUTOFS_IOC_SETTIMEOUT32_IOWR(0x93,0x64,unsigned int)
HANDLE_IOCTL(AUTOFS_IOC_SETTIMEOUT32, ioc_settimeout)
#ifdef CONFIG_VT
@@ -2679,12 +2695,14 @@ HANDLE_IOCTL(SONET_SETFRAMING, do_atm_io
HANDLE_IOCTL(SONET_GETFRAMING, do_atm_ioctl)
HANDLE_IOCTL(SONET_GETFRSENSE, do_atm_ioctl)
/* block stuff */
+#ifdef CONFIG_BLOCK
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
HANDLE_IOCTL(BLKBSZGET_32, do_blkbszget)
HANDLE_IOCTL(BLKBSZSET_32, do_blkbszset)
HANDLE_IOCTL(BLKGETSIZE64_32, do_blkgetsize64)
/* Raw devices */
HANDLE_IOCTL(RAW_SETBIND, raw_ioctl)
HANDLE_IOCTL(RAW_GETBIND, raw_ioctl)
+ #endif
/* Serial */
HANDLE_IOCTL(TIOCGSERIAL, serial_struct_ioctl)
HANDLE_IOCTL(TIOCSSERIAL, serial_struct_ioctl)
diff --git a/fs/no-block.c b/fs/no-block.c
new file mode 100644
index 0000000..d269a93
--- /dev/null
+++ b/fs/no-block.c
@@ -0,0 +1,22 @@
+ /* no-block.c: implementation of routines required for non-BLOCK configuration
+ *
+ * Copyright (C) 2006 Red Hat, Inc. All Rights Reserved.
+ * Written by David Howells (dhowells@xxxxxxxxxx)
+ *
+ * This program 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.
+ */
+
+ #include <linux/kernel.h>
+ #include <linux/fs.h>
+
+ static int no_blkdev_open(struct inode * inode, struct file * filp)
+ {
+ return -ENODEV;
+ }
+
+ const struct file_operations def_blk_fops = {
+ .open = no_blkdev_open,
+ };
diff --git a/fs/partitions/Makefile b/fs/partitions/Makefile
index d713ce6..67e665f 100644
--- a/fs/partitions/Makefile
+++ b/fs/partitions/Makefile
@@ -2,7 +2,7 @@ #
# Makefile for the linux kernel.
#

-obj-y := check.o
+obj-$(CONFIG_BLOCK) := check.o

obj-$(CONFIG_ACORN_PARTITION) += acorn.o
obj-$(CONFIG_AMIGA_PARTITION) += amiga.o
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
diff --git a/fs/proc/proc_misc.c b/fs/proc/proc_misc.c
index 9f2cfc3..ed8646e 100644
--- a/fs/proc/proc_misc.c
+++ b/fs/proc/proc_misc.c
@@ -268,12 +268,15 @@ static int devinfo_show(struct seq_file
if (i == 0)
seq_printf(f, "Character devices:\n");
chrdev_show(f, i);
- } else {
+ }
+ #ifdef CONFIG_BLOCK
+ else {
i -= CHRDEV_MAJOR_HASH_SIZE;
if (i == 0)
seq_printf(f, "\nBlock devices:\n");
blkdev_show(f, i);
}
+ #endif
return 0;
}

@@ -346,6 +349,7 @@ static int stram_read_proc(char *page, c
}
#endif

+ #ifdef CONFIG_BLOCK
extern struct seq_operations partitions_op;
static int partitions_open(struct inode *inode, struct file *file)
{
@@ -369,6 +373,7 @@ static struct file_operations proc_disks
.llseek = seq_lseek,
.release = seq_release,
};
+ #endif

# ifdef CONFIG_MODULES
extern struct seq_operations modules_op;
@@ -686,7 +691,9 @@ #endif
entry->proc_fops = &proc_kmsg_operations;
create_seq_entry("devices", 0, &proc_devinfo_operations);
create_seq_entry("cpuinfo", 0, &proc_cpuinfo_operations);
+ #ifdef CONFIG_BLOCK
create_seq_entry("partitions", 0, &proc_partitions_operations);
+ #endif
create_seq_entry("stat", 0, &proc_stat_operations);
create_seq_entry("interrupts", 0, &proc_interrupts_operations);
# ifdef CONFIG_SLAB
@@ -698,7 +705,9 @@ #endif
create_seq_entry("buddyinfo", S_IRUGO, &fragmentation_file_operations);
create_seq_entry("vmstat", S_IRUGO, &proc_vmstat_file_operations);
create_seq_entry("zoneinfo", S_IRUGO, &proc_zoneinfo_file_operations);
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
+ #ifdef CONFIG_BLOCK
create_seq_entry("diskstats", 0, &proc_diskstats_operations);
+ #endif
# ifdef CONFIG_MODULES
create_seq_entry("modules", 0, &proc_modules_operations);
# endif
diff --git a/fs/quota.c b/fs/quota.c
index d6a2be8..b9dae76 100644
--- a/fs/quota.c
+++ b/fs/quota.c
@@ -338,6 +338,34 @@ static int do_quotactl(struct super_block
}

/*
+ * look up a superblock on which quota ops will be performed
+ * - use the name of a block device to find the superblock thereon
+ */
+static inline struct super_block *quotactl_block(const char __user *special)
+{
+ #ifdef CONFIG_BLOCK
+ struct block_device *bdev;
+ struct super_block *sb;
+ char *tmp = getname(special);
+
+ if (IS_ERR(tmp))
+ return ERR_PTR(PTR_ERR(tmp));
+ bdev = lookup_bdev(tmp);
+ putname(tmp);
+ if (IS_ERR(bdev))
+ return ERR_PTR(PTR_ERR(bdev));
+ sb = get_super(bdev);
+ bdput(bdev);
+ if (!sb)
+ return ERR_PTR(-ENODEV);
+
+ return sb;
+ #else
+ return ERR_PTR(-ENODEV);
+ #endif
+}
+
+/*
+ * This is the system call interface. This communicates with
+ * the user-level programs. Currently this only supports diskquota
+ * calls. Maybe we need to add the process quotas etc. in the future,
@@ -347,25 +375,15 @@ asmlinkage long sys_quotactl(unsigned in
{
uint cmds, type;
struct super_block *sb = NULL;
- struct block_device *bdev;
- char *tmp;
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
int ret;

cmds = cmd >> SUBCMDSHIFT;
type = cmd & SUBCMDMASK;

if (cmds != Q_SYNC || special) {
- tmp = getname(special);
- if (IS_ERR(tmp))
- return PTR_ERR(tmp);
- bdev = lookup_bdev(tmp);
- putname(tmp);
- if (IS_ERR(bdev))
- return PTR_ERR(bdev);
- sb = get_super(bdev);
- bdput(bdev);
- if (!sb)
- return -ENODEV;
+ sb = quotactl_block(special);
+ if (IS_ERR(sb))
+ return PTR_ERR(sb);
}

ret = check_quotactl_valid(sb, type, cmds, id);
diff --git a/fs/super.c b/fs/super.c
index 22c2fd1..33ce475 100644
--- a/fs/super.c
+++ b/fs/super.c
@@ -570,8 +570,10 @@ int do_remount_sb(struct super_block *sb
{
int retval;

+#ifdef CONFIG_BLOCK
if (!(flags & MS_RDONLY) && bdev_read_only(sb->s_bdev))
return -EACCES;
+#endif
if (flags & MS_RDONLY)
acct_auto_close(sb);
shrink_dcache_sb(sb);
@@ -691,6 +693,7 @@ void kill_litter_super(struct super_bloc

EXPORT_SYMBOL(kill_litter_super);

+#ifdef CONFIG_BLOCK
static int set_bdev_super(struct super_block *s, void *data)
{
s->s_bdev = data;
@@ -786,6 +789,7 @@ void kill_block_super(struct super_block
}

EXPORT_SYMBOL(kill_block_super);
+#endif
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
int get_sb_nODEV(struct file_system_type *fs_type,
int flags, void *data,
diff --git a/fs/xfs/Kconfig b/fs/xfs/Kconfig
index 26b364c..35115bc 100644
--- a/fs/xfs/Kconfig
+++ b/fs/xfs/Kconfig
@@ -1,5 +1,6 @@
config XFS_FS
tristate "XFS filesystem support"
+ depends on BLOCK
help
XFS is a high performance journaling filesystem which originated
on the SGI IRIX platform. It is completely multi-threaded, can
diff --git a/include/linux/blkdev.h b/include/linux/blkdev.h
index e3f30d5..fb85d63 100644
--- a/include/linux/blkdev.h
+++ b/include/linux/blkdev.h
@@ -16,6 +16,21 @@ #include <linux/stringify.h>

#include <asm/scatterlist.h>

+#ifdef CONFIG_LBD
+# include <asm/div64.h>
+# define sector_div(a, b) do_div(a, b)
+#else
+# define sector_div(n, b)( \
+{ \
+ int _res; \
+ _res = (n) % (b); \
+ (n) /= (b); \
+ _res; \
+} \
+)
+#endif
+
+#ifdef CONFIG_BLOCK
extern struct super_block *blockdev_superblock;

#define sb_is_blkdev_sb(sb) ((sb) == blockdev_superblock)
@@ -825,20 +840,6 @@ struct work_struct;
int kblockd_schedule_work(struct work_struct *work);
void kblockd_flush(void);

-#ifdef CONFIG_LBD
-# include <asm/div64.h>
-# define sector_div(a, b) do_div(a, b)
-#else
-# define sector_div(n, b)( \
- { \
- int _res; \
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
- _res = (n) % (b); \
- (n) /= (b); \
- _res; \
-} \
-)
-#endif
-
#define MODULE_ALIAS_BLOCKDEV(major,minor) \
MODULE_ALIAS("block-major-" __stringify(major) "-" __stringify(minor))
#define MODULE_ALIAS_BLOCKDEV_MAJOR(major) \
@@ -846,4 +847,27 @@ #define MODULE_ALIAS_BLOCKDEV_MAJOR(majo

extern void bdev_cache_init(void);

+##else /* CONFIG_BLOCK */
+/*
+ * stubs for when the block layer is configured out
+ */
+#define buffer_heads_over_limit 0
+
+static inline long blk_congestion_wait(int rw, long timeout)
+{
+ return timeout;
+}
+
+static inline long nr_blockdev_pages(void)
+{
+ return 0;
+}
+
+static inline void bdev_cache_init(void) {}
+static inline void exit_io_context(void) {}
+
+#define sb_is_blkdev_sb(sb) 0
+
+##endif /* CONFIG_BLOCK */
+
+##endif
diff --git a/include/linux/buffer_head.h b/include/linux/buffer_head.h
index 64b508e..131ffd3 100644
--- a/include/linux/buffer_head.h
+++ b/include/linux/buffer_head.h
@@ -14,6 +14,8 @@ #include <linux/pagemap.h>
#include <linux/wait.h>
#include <asm/atomic.h>

+#ifndef CONFIG_BLOCK
+
enum bh_state_bits {
BH_Uptodate, /* Contains valid data */
BH_Dirty, /* Is dirty */
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
@@ -301,4 +303,18 @@ static inline void lock_buffer(struct bu
}

extern int __set_page_dirty_buffers(struct page *page);
+
+ #else /* CONFIG_BLOCK */
+
+ static inline void buffer_init(void) {}
+ static inline int try_to_free_buffers(struct page *page) { return 1; }
+ static inline int sync_blockdev(struct block_device *bdev) { return 0; }
+ static inline int inode_has_buffers(struct inode *inode) { return 0; }
+ static inline void invalidate_inode_buffers(struct inode *inode) {}
+ static inline int remove_inode_buffers(struct inode *inode) { return 1; }
+ static inline int sync_mapping_buffers(struct address_space *mapping) { return 0; }
+ static inline void invalidate_bdev(struct block_device *bdev, int destroy_dirty_buffers) {}
+
+
+ #endif /* CONFIG_BLOCK */
#endif /* _LINUX_BUFFER_HEAD_H */
diff --git a/include/linux/compat_ioctl.h b/include/linux/compat_ioctl.h
index 13cea44..307f2db 100644
--- a/include/linux/compat_ioctl.h
+++ b/include/linux/compat_ioctl.h
@@ -90,6 +90,7 @@ COMPATIBLE_IOCTL(FDTWADDLE)
COMPATIBLE_IOCTL(FDFMTTRK)
COMPATIBLE_IOCTL(FDRAWCMD)
/* 0x12 */
+ #ifdef CONFIG_BLOCK
COMPATIBLE_IOCTL(BLKRASET)
COMPATIBLE_IOCTL(BLKROSET)
COMPATIBLE_IOCTL(BLKROGET)
@@ -103,6 +104,7 @@ COMPATIBLE_IOCTL(BLKTRACESETUP)
COMPATIBLE_IOCTL(BLKTRACETEARDOWN)
ULONG_IOCTL(BLKRASET)
ULONG_IOCTL(BLKFRASET)
+ #endif
/* RAID */
COMPATIBLE_IOCTL(RAID_VERSION)
COMPATIBLE_IOCTL(GET_ARRAY_INFO)
diff --git a/include/linux/elevator.h b/include/linux/elevator.h
index 1713ace..d2f4b0a 100644
--- a/include/linux/elevator.h
+++ b/include/linux/elevator.h
@@ -1,6 +1,8 @@
#ifndef _LINUX_ELEVATOR_H
#define _LINUX_ELEVATOR_H

+ #ifdef CONFIG_BLOCK
+
typedef int (elevator_merge_fn) (request_queue_t *, struct request **,
struct bio *);
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
@@ -150,4 +152,5 @@ enum {

#define rq_end_sector(rq) ((rq)->sector + (rq)->nr_sectors)

+msgid /* CONFIG_BLOCK */
+msgid
diff --git a/include/linux/fs.h b/include/linux/fs.h
index 7339d41..b72e3d0 100644
--- a/include/linux/fs.h
+++ b/include/linux/fs.h
@@ -1477,6 +1477,7 @@ #else
extern void putname(const char *name);
+msgid

+#ifdef CONFIG_BLOCK
extern int register_blkdev(unsigned int, const char *);
extern int unregister_blkdev(unsigned int, const char *);
extern struct block_device *bdget(dev_t);
@@ -1485,11 +1486,15 @@ extern void bd_forget(struct inode *inode)
extern void bdput(struct block_device *);
extern struct block_device *open_by_devnum(dev_t, unsigned);
extern struct block_device *open_partition_by_devnum(dev_t, unsigned);
-extern const struct file_operations def_blk_fops;
extern const struct address_space_operations def_blk_aops;
+msgid
+static inline void bd_forget(struct inode *inode) {}
+msgid
+extern const struct file_operations def_blk_fops;
extern const struct file_operations def_chr_fops;
extern const struct file_operations bad_sock_fops;
extern const struct file_operations def_fifo_fops;
+#ifdef CONFIG_BLOCK
extern int ioctl_by_bdev(struct block_device *, unsigned, unsigned long);
extern int blkdev_ioctl(struct inode *, struct file *, unsigned, unsigned long);
extern long compat_blkdev_ioctl(struct file *, unsigned, unsigned long);
@@ -1505,6 +1510,7 @@ #else
#define bd_claim_by_disk(bdev, holder, disk) bd_claim(bdev, holder)
#define bd_release_from_disk(bdev, disk) bd_release(bdev)
+msgid
+msgid

/* fs/char_dev.c */
#define CHRDEV_MAJOR_HASH_SIZE 255
@@ -1518,14 +1524,19 @@ extern int chrdev_open(struct inode *, s
extern void chrdev_show(struct seq_file *, off_t);

/* fs/block_dev.c */
-#define BLKDEV_MAJOR_HASH_SIZE 255
#define BDEVNAME_SIZE 32 /* Largest string for a blockdev identifier */
+
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
+#ifdef CONFIG_BLOCK
+#define BLKDEV_MAJOR_HASH_SIZE 255
extern const char *__bdevname(dev_t, char *buffer);
extern const char *bdevname(struct block_device *bdev, char *buffer);
extern struct block_device *lookup_bdev(const char *);
extern struct block_device *open_bdev_excl(const char *, int, void *);
extern void close_bdev_excl(struct block_device *);
extern void blkdev_show(struct seq_file *, off_t);
+#else
+#define BLKDEV_MAJOR_HASH_SIZE 0
+#endif

extern void init_special_inode(struct inode *, umode_t, dev_t);

@@ -1539,6 +1550,7 @@ extern const struct file_operations rdwr

extern int fs_may_remount_ro(struct super_block *);

+#ifdef CONFIG_BLOCK
/*
 * return READ, READA, or WRITE
 */
@@ -1550,9 +1562,10 @@ #define bio_rw(bio) ((bio)->bi_rw & (RW
#define bio_data_dir(bio) ((bio)->bi_rw & 1)

extern int check_disk_change(struct block_device *);
-extern int invalidate_inodes(struct super_block *);
extern int __invalidate_device(struct block_device *);
extern int invalidate_partition(struct gendisk *, int);
+#endif
+extern int invalidate_inodes(struct super_block *);
unsigned long invalidate_mapping_pages(struct address_space *mapping,
pgoff_t start, pgoff_t end);
unsigned long invalidate_inode_pages(struct address_space *mapping);
@@ -1585,7 +1598,9 @@ extern void emergency_sync(void);
extern void emergency_remount(void);
extern int do_remount_sb(struct super_block *sb, int flags,
void *data, int force);
+#ifdef CONFIG_BLOCK
extern sector_t bmap(struct inode *, sector_t);
+#endif
extern int notify_change(struct dentry *, struct iattr *);
extern int permission(struct inode *, int, struct nameidata *);
extern int generic_permission(struct inode *, int,
@@ -1668,9 +1683,11 @@ static inline void insert_inode_hash(str
extern struct file * get_empty_filp(void);
extern void file_move(struct file *f, struct list_head *list);
extern void file_kill(struct file *f);
+#ifdef CONFIG_BLOCK
struct bio;
extern void submit_bio(int, struct bio *);
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
extern int bdev_read_only(struct block_device *);
+#endif
extern int set_blocksize(struct block_device *, int);
extern int sb_set_blocksize(struct super_block *, int);
extern int sb_min_blocksize(struct super_block *, int);
@@ -1751,6 +1768,7 @@ static inline void do_generic_file_read(
actor);
}

+#ifdef CONFIG_BLOCK
ssize_t __blockdev_direct_IO(int rw, struct kiocb *iocb, struct inode *inode,
struct block_device *bdev, const struct iovec *iov, loff_t offset,
unsigned long nr_segs, get_block_t get_block, dio_iDONE_t end_io,
@@ -1788,6 +1806,7 @@ static inline ssize_t blockdev_direct_IO
return __blockdev_direct_IO(rw, iocb, inode, bdev, iov, offset,
nr_segs, get_block, end_io, DIO_OWN_LOCKING);
}
+#endif

extern const struct file_operations generic_ro_fops;

diff --git a/include/linux/genhd.h b/include/linux/genhd.h
index e4af57e..41f276f 100644
--- a/include/linux/genhd.h
+++ b/include/linux/genhd.h
@@ -11,6 +11,8 @@ #define _LINUX_GENHD_H

#include <linux/types.h>

+#ifdef CONFIG_BLOCK
+
enum {
/* These three have identical behaviour; use the second one if DOS FDISK gets
confused about extended/logical partitions starting past cylinder 1023. */
@@ -420,3 +422,5 @@ static inline struct block_device *bdget
#endif

#endif
+
+
diff --git a/include/linux/mpage.h b/include/linux/mpage.h
index 517c098..cc5fb75 100644
--- a/include/linux/mpage.h
+++ b/include/linux/mpage.h
@@ -9,6 +9,7 @@
* (And no, it doesn't do the #ifdef __MPAGE_H thing, and it doesn't do
* nested includes. Get it right in the .c file).
*/
+#ifdef CONFIG_BLOCK

struct writeback_control;
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
typedef int (writepage_t)(struct page *page, struct writeback_control *wbc);
@@ -20,3 +21,5 @@ int mpage_writepages(struct address_spac
struct writeback_control *wbc, get_block_t get_block);
int mpage_writepage(struct page *page, get_block_t *get_block,
struct writeback_control *wbc);
+
+ #endif
diff --git a/include/linux/raid/md.h b/include/linux/raid/md.h
index eb3e547..c588709 100644
--- a/include/linux/raid/md.h
+++ b/include/linux/raid/md.h
@@ -53,6 +53,8 @@ #include <linux/raid/md_p.h>
#include <linux/raid/md_u.h>
#include <linux/raid/md_k.h>

+ #ifdef CONFIG_MD
+
+ /*
+ * Different major versions are not compatible.
+ * Different minor versions are only downward compatible.
@@ -95,5 +97,6 @@ extern void md_new_event(mddev_t *mddev)

extern void md_update_sb(mddev_t * mddev);

+ #endif /* CONFIG_MD */
# endif

diff --git a/include/linux/raid/md_k.h b/include/linux/raid/md_k.h
index d288902..920b94f 100644
--- a/include/linux/raid/md_k.h
+++ b/include/linux/raid/md_k.h
@@ -18,6 +18,8 @@ #define _MD_K_H
/* and dm-bio-list.h is not under include/linux because.... ??? */
#include "../drivers/md/dm-bio-list.h"

+ #ifdef CONFIG_BLOCK
+
+ #define LEVEL_MULTIPATH (-4)
+ #define LEVEL_LINEAR (-1)
+ #define LEVEL_FAULTY (-5)
@@ -362,5 +364,6 @@ static inline void safe_put_page(struct
if (p) put_page(p);
}

+ #endif /* CONFIG_BLOCK */
# endif

diff --git a/include/scsi/scsi_tcq.h b/include/scsi/scsi_tcq.h
index e47e36a..bc34746 100644
--- a/include/scsi/scsi_tcq.h
+++ b/include/scsi/scsi_tcq.h
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
@@ -5,7 +5,6 @@ #include <linux/blkdev.h>
#include <scsi/scsi_cmnd.h>
#include <scsi/scsi_device.h>

-
#define MSG_SIMPLE_TAG 0x20
#define MSG_HEAD_TAG 0x21
#define MSG_ORDERED_TAG 0x22
@@ -13,6 +12,7 @@ #define MSG_ORDERED_TAG 0x22
#define SCSI_NO_TAG (-1) /* identify no tag in use */

+#ifdef CONFIG_BLOCK

/**
 * scsi_get_tag_type - get the type of tag the device supports
@@ -131,4 +131,5 @@ static inline struct scsi_cmnd *scsi_fin
return sdev->current_cmnd;
}

+#endif /* CONFIG_BLOCK */
#endif /* _SCSI_SCSI_TCQ_H */
diff --git a/init/Kconfig b/init/Kconfig
index a099fc6..814bacc 100644
--- a/init/Kconfig
+++ b/init/Kconfig
@@ -92,7 +92,7 @@ config LOCALVERSION_AUTO

config SWAP
bool "Support for paging of anonymous memory (swap)"
- depends on MMU
+ depends on MMU && BLOCK
default y
help
This option allows you to choose whether you want to have support
diff --git a/init/do_mounts.c b/init/do_mounts.c
index 94aee7..dbb2604 100644
--- a/init/do_mounts.c
+++ b/init/do_mounts.c
@@ -284,7 +284,11 @@ void __init mount_block_root(char *name,
{
char *fs_names = __getname();
char *p;
+#ifdef CONFIG_BLOCK
char b[BDEVNAME_SIZE];
+#else
+ const char *b = name;
+#endif

get_fs_names(fs_names);
retry:
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
@@ -303,7 +307,9 @@ retry:
* Allow the user to distinguish between failed sys_open
* and bad superblock on root device.
*/
+#ifdef CONFIG_BLOCK
__bdevname(ROOT_DEV, b);
+#endif
printk("VFS: Cannot open root device \"%s\" or %s\n",
root_device_name, b);
printk("Please append a correct \"root=\" boot option\n");
@@ -315,7 +321,10 @@ retry:
for (p = fs_names; *p; p += strlen(p)+1)
printk(" %s", p);
printk("\n");
- panic("VFS: Unable to mount root fs on %s", __bdevname(ROOT_DEV, b));
+#ifdef CONFIG_BLOCK
+ __bdevname(ROOT_DEV, b);
+#endif
+ panic("VFS: Unable to mount root fs on %s", b);
out:
putname(fs_names);
}
@@ -386,8 +395,10 @@ #ifdef CONFIG_BLK_DEV_FD
change_floppy("root floppy");
}
#endif
+#ifdef CONFIG_BLOCK
create_dev("/dev/root", ROOT_DEV);
mount_block_root("/dev/root", root_mountflags);
+#endif
}

/*
diff --git a/kernel/sys_ni.c b/kernel/sys_ni.c
index 6991bec..7a3b2e7 100644
--- a/kernel/sys_ni.c
+++ b/kernel/sys_ni.c
@@ -134,3 +134,8 @@ cond_syscall(sys_madvise);
cond_syscall(sys_mremap);
cond_syscall(sys_remap_file_pages);
cond_syscall(compat_sys_move_pages);
+
+/* block-layer dependent */
+cond_syscall(sys_bdflush);
+cond_syscall(sys_ioprio_set);
+cond_syscall(sys_ioprio_get);
diff --git a/mm/Makefile b/mm/Makefile
index 84fff32..3af3154 100644
--- a/mm/Makefile
+++ b/mm/Makefile
@@ -12,7 +12,7 @@ obj-y := bootmem.o filemap.o mempool.o
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
readahead.o swap.o truncate.o vmscan.o \  
prio_tree.o util.o mmzone.o vmstat.o $(mmu-y)  
  
-obj-y += bounce.o  
+obj-$(CONFIG_BLOCK) += bounce.o  
obj-$(CONFIG_SWAP) += page_io.o swap_state.o swapfile.o thrash.o  
obj-$(CONFIG_HUGETLBFS) += hugetlb.o  
obj-$(CONFIG_NUMA) += mempolicy.o  
diff --git a/mm/filemap.c b/mm/filemap.c  
index a5ea7e0..88d9cd1 100644  
--- a/mm/filemap.c  
+++ b/mm/filemap.c  
@@ -2003,6 +2003,7 @@ inline int generic_write_checks(struct f  
if (unlikely(*pos + *count > inode->i_sb->s_maxbytes))  
*count = inode->i_sb->s_maxbytes - *pos;  
} else {  
+ifdef CONFIG_BLOCK  
loff_t isize;  
if (bdev_read_only(I_BDEV(inode)))  
return -EPERM;  
@@ -2014,6 +2015,9 @@ inline int generic_write_checks(struct f  
  
if (*pos + *count > isize)  
*count = isize - *pos;  
+else  
+ return -EPERM;  
+endif  
}  
return 0;  
}  
diff --git a/mm/migrate.c b/mm/migrate.c  
index 0227163..bedc0ed 100644  
--- a/mm/migrate.c  
+++ b/mm/migrate.c  
@@ -409,6 +409,7 @@ int migrate_page(struct address_space *m  
}  
EXPORT_SYMBOL(migrate_page);  
  
+ifdef CONFIG_BLOCK  
/*  
* Migration function for pages with buffers. This function can only be used  
* if the underlying filesystem guarantees that no other references to "page"  
@@ -466,6 +467,7 @@ int buffer_migrate_page(struct address_s  
return 0;  
}  
EXPORT_SYMBOL(buffer_migrate_page);  
+endif  
  
/*  
* Writeback a page to clean the dirty state  
diff --git a/mm/page-writeback.c b/mm/page-writeback.c
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

```
index 668716c..b450520 100644
--- a/mm/page-writeback.c
+++ b/mm/page-writeback.c
@@ -800,9 +800,11 @@ int fastcall set_page_dirty(struct page
if (likely(mapping)) {
int (*spd)(struct page *) = mapping->a_ops->set_page_dirty;
- if (spd)
- return (*spd)(page);
- return __set_page_dirty_buffers(page);
+#ifdef CONFIG_BLOCK
+ if (!spd)
+ spd = __set_page_dirty_buffers;
+#endif
+ return (*spd)(page);
}
if (!PageDirty(page)) {
if (!TestSetPageDirty(page))
diff --git a/security/seclvl.c b/security/seclvl.c
index c26dd7d..fc00df2 100644
--- a/security/seclvl.c
+++ b/security/seclvl.c
@@ -377,6 +377,7 @@ static int seclvl_settime(struct timespe
/* claim the blockdev to exclude mounters, release on file close */
static int seclvl_bd_claim(struct inode *inode)
{
+#ifdef CONFIG_BLOCK
int holder;
struct block_device *bdev = NULL;
dev_t dev = inode->i_rdev;
@@ -389,12 +390,14 @@ static int seclvl_bd_claim(struct inode
/* claimed, mark it to release on close */
inode->i_security = current;
}
+#endif
return 0;
}

/* release the blockdev if you claimed it */
static void seclvl_bd_release(struct inode *inode)
{
+#ifdef CONFIG_BLOCK
if (inode && S_ISBLK(inode->i_mode) && inode->i_security == current) {
struct block_device *bdev = inode->i_bdev;
if (bdev) {
@@ -403,6 +406,7 @@ static void seclvl_bd_release(struct ino
inode->i_security = NULL;
}
}
+#endif
}
```

[PATCH 17/17] BLOCK: Make it possible to disable the block layer [try #2]

/\*\*

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>