```c
#ifndef _LINUX_GENHD_H
#define _LINUX_GENHD_H

/*
 *      genhd.h Copyright (C) 1992 Drew Eckhardt
 *      Generic hard disk header file by
 *              Drew Eckhardt
 *
 *              <drew@colorado.edu>
 */

#include <linux/config.h>

#define CONFIG_MSDOS_PARTITION 1

#ifdef __alpha__
#define CONFIG_OSF_PARTITION 1
#endif

#if defined(__sparc__) || defined(CONFIG_SMD_DISKLABEL)
#define CONFIG_SUN_PARTITION 1
#endif

/*
 * These two have identical behaviour; use the second one if DOS fdisk
 * gets confused about extended/logical partitions starting past
 * cylinder 1023.
 */
#define DOS_EXTENDED_PARTITION 5
#define LINUX_EXTENDED_PARTITION 0x85

#define DM6_PARTITION           0x54    /* has DDO: use xlated geom & offset
*/
#define EZD_PARTITION           0x55    /* EZ-DRIVE:  same as DM6 (we think)
*/
#define DM6_AUX1PARTITION       0x51    /* no DDO:  use xlated geom */
#define DM6_AUX3PARTITION       0x53    /* no DDO:  use xlated geom */

struct partition {
        unsigned char boot_ind;         /* 0x80 - active */
        unsigned char head;             /* starting head */
        unsigned char sector;           /* starting sector */
        unsigned char cyl;              /* starting cylinder */
        unsigned char sys_ind;          /* What partition type */
        unsigned char end_head;         /* end head */
        unsigned char end_sector;       /* end sector */
        unsigned char end_cyl;          /* end cylinder */
        unsigned int start_sect;        /* starting sector counting from 0 */
        unsigned int nr_sects;          /* nr of sectors in partition */
};

struct hd_struct {
        long start_sect;
        long nr_sects;
};

struct gendisk {
        int major;                      /* major number of driver */
        const char *major_name;         /* name of major driver */
        int minor_shift;                /* number of times minor is shifted
```

```
to
                                                     get real minor */
        int max_p;                          /* maximum partitions per device */
        int max_nr;                         /* maximum number of real devices */

        void (*init)(struct gendisk *);      /* Initialization called
                                             * before we do our thing */
        struct hd_struct *part;             /* partition table */
        int *sizes;                         /* device size in blocks,
                                             * copied to blk_size[] */
        int nr_real;                        /* number of real devices */

        void *real_devices;                 /* internal use */
        struct gendisk *next;
};

#ifdef CONFIG_BSD_DISKLABEL
/*
 * BSD disklabel support by Yossi Gottlieb <yogo@math.tau.ac.il>
 */

#define BSD_PARTITION           0xa5    /* Partition ID */

#define BSD_DISKMAGIC   (0x82564557UL)    /* The disk magic number */
#define BSD_MAXPARTITIONS       8
#define BSD_FS_UNUSED           0       /* disklabel unused partition entry
ID */
struct bsd_disklabel {
        __u32   d_magic;                /* the magic number */
        __s16   d_type;                 /* drive type */
        __s16   d_subtype;              /* controller/d_type specific */
        char    d_typename[16];         /* type name, e.g. "eagle" */
        char    d_packname[16];                 /* pack identifier */
        __u32   d_secsize;              /* # of bytes per sector */
        __u32   d_nsectors;             /* # of data sectors per track */
        __u32   d_ntracks;              /* # of tracks per cylinder */
        __u32   d_ncylinders;           /* # of data cylinders per unit */
        __u32   d_secpercyl;            /* # of data sectors per cylinder */
        __u32   d_secperunit;           /* # of data sectors per unit */
        __u16   d_sparespertrack;       /* # of spare sectors per track */
        __u16   d_sparespercyl;         /* # of spare sectors per cylinder */
        __u32   d_acylinders;           /* # of alt. cylinders per unit */
        __u16   d_rpm;                  /* rotational speed */
        __u16   d_interleave;           /* hardware sector interleave */
        __u16   d_trackskew;            /* sector 0 skew, per track */
        __u16   d_cylskew;              /* sector 0 skew, per cylinder */
        __u32   d_headswitch;           /* head switch time, usec */
        __u32   d_trkseek;              /* track-to-track seek, usec */
        __u32   d_flags;                /* generic flags */
#define NDDATA 5
        __u32   d_drivedata[NDDATA];    /* drive-type specific information */
#define NSPARE 5
        __u32   d_spare[NSPARE];        /* reserved for future use */
        __u32   d_magic2;               /* the magic number (again) */
        __u16   d_checksum;             /* xor of data incl. partitions */

                        /* filesystem and partition information: */
        __u16   d_npartitions;          /* number of partitions in following
*/
```

```
        __u32   d_bbsize;               /* size of boot area at sn0, bytes */
        __u32   d_sbsize;               /* max size of fs superblock, bytes
*/
        struct  bsd_partition {         /* the partition table */
                __u32   p_size;         /* number of sectors in partition */
                __u32   p_offset;       /* starting sector */
                __u32   p_fsize;        /* filesystem basic fragment size */
                __u8    p_fstype;       /* filesystem type, see below */
                __u8    p_frag;         /* filesystem fragments per block */
                __u16   p_cpg;          /* filesystem cylinders per group */
        } d_partitions[BSD_MAXPARTITIONS];      /* actually may be more */
};

#endif  /* CONFIG_BSD_DISKLABEL */

extern struct gendisk *gendisk_head;    /* linked list of disks */

/*
 * disk_name() is used by genhd.c and md.c.
 * It formats the devicename of the indicated disk
 * into the supplied buffer, and returns a pointer
 * to that same buffer (for convenience).
 */
char *disk_name (struct gendisk *hd, int minor, char *buf);

#endif
```