



Linux Ubuntu 22.4.5 Manual Pages on command 'ioctl_list.2'

\$ man ioctl_list.2

IOCTL_LIST(2) Linux Programmer's Manual IOCTL_LIST(2)

NAME

ioctl_list - list of ioctl calls in Linux/i386 kernel

DESCRIPTION

This is ioctl List 1.3.27, a list of ioctl calls in Linux/i386 kernel 1.3.27. It

contains 421 ioctls from </usr/include/{asm,linux}/*.h>. For each ioctl, its nu?

merical value, its name, and its argument type are given.

An argument type of const struct foo * means the argument is input to the kernel.

struct foo * means the kernel outputs the argument. If the kernel uses the argu?

ment for both input and output, this is marked with // I-O.

Some ioctls take more arguments or return more values than a single structure.

These are marked // MORE and documented further in a separate section. In addi?

tion, information about some ioctls can be found in the pages listed under SEE ALSO

in ioctl(2).

This list is very incomplete.

ioctl structure

ioctl command values are 32-bit constants. In principle these constants are com?

pletely arbitrary, but people have tried to build some structure into them.

The old Linux situation was that of mostly 16-bit constants, where the last byte is

a serial number, and the preceding byte(s) give a type indicating the driver.

Sometimes the major number was used: 0x03 for the HDIO_* ioctls, 0x06 for the LP*

ioctls. And sometimes one or more ASCII letters were used. For example, TCGETS

has value 0x00005401, with 0x54 = 'T' indicating the terminal driver, and CYGET? TIMEOUT has value 0x00435906, with 0x43 0x59 = 'C' 'Y' indicating the cyclades driver.

Later (0.98p5) some more information was built into the number. One has 2 direction bits (00: none, 01: write, 10: read, 11: read/write) followed by 14 size bits (giving the size of the argument), followed by an 8-bit type (collecting the ioctls in groups for a common purpose or a common driver), and an 8-bit serial number. The macros describing this structure live in <asm/ioctl.h> and are _IO(type,nr) and {_IOR,_IOW,_IOWR}(type,nr,size). They use sizeof(size) so that size is a misnomer here: this third argument is a data type.

Note that the size bits are very unreliable: in lots of cases they are wrong, either because of buggy macros using sizeof(sizeof(struct)), or because of legacy values.

Thus, it seems that the new structure only gave disadvantages: it does not help in checking, but it causes varying values for the various architectures.

RETURN VALUE

Decent ioctls return 0 on success and -1 on error, while any output value is stored via the argument. However, quite a few ioctls in fact return an output value.

This is not yet indicated below.

// Main table.

// <include/asm-i386/socket.h>

0x00008901 FIOSETOWN const int *

0x00008902 SIOCSPGRP const int *

0x00008903 FIOGETOWN int *

0x00008904 SIOCGPGRP int *

0x00008905 SIOCATMAR int *

0x00008906 SIOCGSTAMP timeval *

// <include/asm-i386/termios.h>

0x00005401 TCGETS struct termios *

0x00005402 TCSETS const struct termios *

0x00005403 TCSETSW const struct termios *

0x00005404 TCSETSF const struct termios *

0x00005405 TCGETA struct termio *

0x00005406	TCSETA	const struct termio *	
0x00005407	TCSETAW	const struct termio *	
0x00005408	TCSETAF	const struct termio *	
0x00005409	TCSBRK	int	
0x0000540A	TCXONC	int	
0x0000540B	TCFLSH	int	
0x0000540C	TIOCEXCL	void	
0x0000540D	TIOCNXCL	void	
0x0000540E	TIOCSCTTY	int	
0x0000540F	TIOCGPGRP	pid_t *	
0x00005410	TIOCSPGRP	const pid_t *	
0x00005411	TIOCOUTQ	int *	
0x00005412	TIOCSTI	const char *	
0x00005413	TIOCGWINSZ	struct winsize *	
0x00005414	TIOCSWINSZ	const struct winsize *	
0x00005415	TIOCMGET	int *	
0x00005416	TIOCMBIS	const int *	
0x00005417	TIOCMBIC	const int *	
0x00005418	TIOCMSET	const int *	
0x00005419	TIOCGSOFTCAR	int *	
0x0000541A	TIOCSSOFTCAR	const int *	
0x0000541B	FIONREAD	int *	
0x0000541B	TIOCINQ	int *	
0x0000541C	TIOCLINUX	const char *	// MORE
0x0000541D	TIOCCONS	void	
0x0000541E	TIOCGSERIAL	struct serial_struct *	
0x0000541F	TIOCSSERIAL	const struct serial_struct *	
0x00005420	TIOCPKT	const int *	
0x00005421	FIONBIO	const int *	
0x00005422	TIOCNOTTY	void	
0x00005423	TIOCSETD	const int *	
0x00005424	TIOCGETD	int *	
0x00005425	TCSBRKP	int	

```

0x00005426 TIOCTTYGSTRUCT struct tty_struct *
0x00005450 FIONCLEX      void
0x00005451 FIOCLEX        void
0x00005452 FIOASYNC      const int *
0x00005453 TIOCSERCONFIG  void
0x00005454 TIOCSERGWILD   int *
0x00005455 TIOCSERSWILD  const int *
0x00005456 TIOCGLOCKTRMIO  struct termios *
0x00005457 TIOCSLOCKTRMIO  const struct termios *
0x00005458 TIOCSERGSTRUCT  struct async_struct *
0x00005459 TIOCSERGETLSR   int *
0x0000545A TIOCSERGETMULTI struct serial_multiport_struct *
0x0000545B TIOCSERSETMULTI const struct serial_multiport_struct *
// <include/linux/ax25.h>
0x000089E0 SIOCAX25GETUID  const struct sockaddr_ax25 *
0x000089E1 SIOCAX25ADDUID  const struct sockaddr_ax25 *
0x000089E2 SIOCAX25DELUID  const struct sockaddr_ax25 *
0x000089E3 SIOCAX25NOUID   const int *
0x000089E4 SIOCAX25DIGCTL  const int *
0x000089E5 SIOCAX25GETPARMS struct ax25_parms_struct * // I-O
0x000089E6 SIOCAX25SETPARMS const struct ax25_parms_struct *
// <include/linux/cdk.h>
0x00007314 STL_BINTR   void
0x00007315 STL_BSTART  void
0x00007316 STL_BSTOP   void
0x00007317 STL_BRESET  void
// <include/linux/cdrom.h>
0x00005301 CDROMPAUSE   void
0x00005302 CDROMRESUME  void
0x00005303 CDROMPLAYMSF  const struct cdrom_msf *
0x00005304 CDROMPLAYTRKIND const struct cdrom_ti *
0x00005305 CDROMREADTOCHDR struct cdrom_tochdr *
0x00005306 CDROMREADTOCENTRY struct cdrom_tocentry * // I-O

```

```

0x00005307 CDROMSTOP      void
0x00005308 CDROMSTART     void
0x00005309 CDROMEJECT     void
0x0000530A CDROMVOLCTRL   const struct cdrom_volctrl *
0x0000530B CDROMSUBCHNL   struct cdrom_subchnl *      // I-O
0x0000530C CDROMREADMODE2 const struct cdrom_msf *      // MORE
0x0000530D CDROMREADMODE1 const struct cdrom_msf *      // MORE
0x0000530E CDROMREADAUDIO const struct cdrom_read_audio * // MORE
0x0000530F CDROMEJECT_SW  int
0x00005310 CDROMMULTISESSION struct cdrom_multisession * // I-O
0x00005311 CDROM_GET_UPC   struct { char [8]; } *
0x00005312 CDROMRESET     void
0x00005313 CDROMVOLREAD    struct cdrom_volctrl *
0x00005314 CDROMREADDRAW  const struct cdrom_msf * // MORE
0x00005315 CDROMREADCOOKED const struct cdrom_msf * // MORE
0x00005316 CDROMSEEK      const struct cdrom_msf *

// <include/linux/cm206.h>

0x00002000 CM206CTL_GET_STAT    int
0x00002001 CM206CTL_GET_LAST_STAT int

// <include/linux/cyclades.h>

0x00435901 CYGETMON      struct cyclades_monitor *
0x00435902 CYGETTHRESH   int *
0x00435903 CYSETTHRESH   int
0x00435904 CYGETDEFTHRESH int *
0x00435905 CYSETDEFTHRESH int
0x00435906 CYGETTIMEOUT  int *
0x00435907 CYSETTIMEOUT  int
0x00435908 CYGETDEFTIMEOUT int *
0x00435909 CYSETDEFTIMEOUT int

// <include/linux/fd.h>

0x00000000 FDCLRPRM      void
0x00000001 FDSETPRM      const struct floppy_struct *
0x00000002 FDDEFPRM      const struct floppy_struct *

```

0x00000003	FDGETPRM	struct floppy_struct *
0x00000004	FDMSGON	void
0x00000005	FDMSGOFF	void
0x00000006	FDFMTBEG	void
0x00000007	FDFMTTRK	const struct format_descr *
0x00000008	FDFMTEND	void
0x0000000A	FDSETEMSGTRESH	int
0x0000000B	FDFLUSH	void
0x0000000C	FDSETMAXERRS	const struct floppy_max_errors *
0x0000000E	FDGETMAXERRS	struct floppy_max_errors *
0x00000010	FDGETDRVTYPE	struct { char [16]; } *
0x00000014	FDSETDRVPRM	const struct floppy_drive_params *
0x00000015	FDGETDRVPRM	struct floppy_drive_params *
0x00000016	FDGETDRVSTAT	struct floppy_drive_struct *
0x00000017	FDPOLLDVSTAT	struct floppy_drive_struct *
0x00000018	FDRESET	int
0x00000019	FDGETFDCSTAT	struct floppy_fdc_state *
0x0000001B	FDWERRORCLR	void
0x0000001C	FDWERRORGET	struct floppy_write_errors *
0x0000001E	FDRAWCMD	struct floppy_raw_cmd * // MORE // I-O
0x00000028	FDTWADDLE	void

// <include/linux/fs.h>

0x0000125D	BLKROSET	const int *
0x0000125E	BLKROGET	int *
0x0000125F	BLKRRPART	void
0x00001260	BLKGETSIZE	unsigned long *
0x00001261	BLKFLSBUF	void
0x00001262	BLKRASET	unsigned long
0x00001263	BLKRAGET	unsigned long *
0x00000001	FIBMAP	int * // I-O
0x00000002	FIGETBSZ	int *
0x80086601	FS_IOC_GETFLAGS	int *
0x40086602	FS_IOC_SETFLAGS	int *

```

0x80087601 FS_IOC_GETVERSION    int *
0x40087602 FS_IOC_SETVERSION    int *
0xC020660B FS_IOC_FIEMAP        struct fiemap *
0x40086602 FS_IOC32_SETFLAGS    int *
0x40086602 FS_IOC32_SETFLAGS    int *
0x80047601 FS_IOC32_GETVERSION  int *
0x40047602 FS_IOC32_SETVERSION  int *

// <include/linux/hdreg.h>

0x00000301 HDIO_GETGEO          struct hd_geometry *
0x00000302 HDIO_GET_UNMASKINTR  int *
0x00000304 HDIO_GET_MULTCOUNT  int *
0x00000307 HDIO_GET_IDENTITY    struct hd_driveid *
0x00000308 HDIO_GET_KEEPPSETTINGS int *
0x00000309 HDIO_GET_CHIPSET     int *
0x0000030A HDIO_GET_NOWERR      int *
0x0000030B HDIO_GET_DMA         int *
0x0000031F HDIO_DRIVE_CMD       int *          // I-O
0x00000321 HDIO_SET_MULTCOUNT  int
0x00000322 HDIO_SET_UNMASKINTR  int
0x00000323 HDIO_SET_KEEPPSETTINGS int
0x00000324 HDIO_SET_CHIPSET     int
0x00000325 HDIO_SET_NOWERR      int
0x00000326 HDIO_SET_DMA         int

// <include/linux/if_eq.h>

0x000089F0 EQL_ENSLAVE          struct ifreq * // MORE // I-O
0x000089F1 EQL_EMANCIPATE       struct ifreq * // MORE // I-O
0x000089F2 EQL_GETSLAVECFG      struct ifreq * // MORE // I-O
0x000089F3 EQL_SETSLAVECFG      struct ifreq * // MORE // I-O
0x000089F4 EQL_GETMASTRCFG      struct ifreq * // MORE // I-O
0x000089F5 EQL_SETMASTRCFG      struct ifreq * // MORE // I-O

// <include/linux/if_plip.h>

0x000089F0 SIOCDEVPLIP          struct ifreq * // I-O

// <include/linux/if_ppp.h>

```

```

0x00005490 PPPIOCGFLAGS    int *
0x00005491 PPPIOCSFLAGS    const int *
0x00005492 PPPIOCGASYNCMAP int *
0x00005493 PPPIOCSASYNCMAP const int *
0x00005494 PPPIOCGUNIT     int *
0x00005495 PPPIOCSINPSIG   const int *
0x00005497 PPPIOCSDEBUG    const int *
0x00005498 PPPIOCGDEBUG    int *
0x00005499 PPPIOCGSTAT     struct ppp_stats *
0x0000549A PPPIOCGTIME     struct ppp_ddinfo *
0x0000549B PPPIOCGXASYNCMAP struct { int [8]; } *
0x0000549C PPPIOCSXASYNCMAP const struct { int [8]; } *
0x0000549D PPPIOCSMRU      const int *
0x0000549E PPPIOCRASYNCMAP const int *
0x0000549F PPPIOCSMAXCID   const int *

// <include/linux/ipx.h>

0x000089E0 SIOCAIPXITFCRT  const char *
0x000089E1 SIOCAIPXPRISLT  const char *
0x000089E2 SIOCIPXCFGDATA  struct ipx_config_data *

// <include/linux/kd.h>

0x00004B60 GIO_FONT  struct { char [8192]; } *
0x00004B61 PIO_FONT  const struct { char [8192]; } *
0x00004B6B GIO_FONTX struct console_font_desc *    // MORE // I-O
0x00004B6C PIO_FONTX const struct console_font_desc * //MORE
0x00004B70 GIO_CMAP  struct { char [48]; } *
0x00004B71 PIO_CMAP  const struct { char [48]; }
0x00004B2F KIOCSOUND int
0x00004B30 KDMKTONE  int
0x00004B31 KDGETLED  char *
0x00004B32 KDSETLED  int
0x00004B33 KDGKBTYPE char *
0x00004B34 KDADDIO   int                // MORE
0x00004B35 KDELIO    int                // MORE

```

```

0x00004B36 KDENABIO    void                // MORE
0x00004B37 KDDISABIO   void                // MORE
0x00004B3A KDSETMODE   int
0x00004B3B KDGETMODE   int *
0x00004B3C KDMAPDISP   void                // MORE
0x00004B3D KDUNMAPDISP void                // MORE
0x00004B40 GIO_SCRNMAP  struct { char [E_TABSZ]; } *
0x00004B41 PIO_SCRNMAP  const struct { char [E_TABSZ]; } *
0x00004B69 GIO_UNISCRNMAP struct { short [E_TABSZ]; } *
0x00004B6A PIO_UNISCRNMAP const struct { short [E_TABSZ]; } *
0x00004B66 GIO_UNIMAP   struct unimapdesc *    // MORE // I-O
0x00004B67 PIO_UNIMAP   const struct unimapdesc * // MORE
0x00004B68 PIO_UNIMAPCLR const struct unimapinit *
0x00004B44 KDGKBMODE    int *
0x00004B45 KDSKBMODE    int
0x00004B62 KDGKBMETA    int *
0x00004B63 KDSKBMETA    int
0x00004B64 KDGKBLED     int *
0x00004B65 KDSKBLED     int
0x00004B46 KDGKBENT     struct kbentry *    // I-O
0x00004B47 KDSKBENT     const struct kbentry *
0x00004B48 KDGKBSSENT   struct kbsentry *    // I-O
0x00004B49 KDSKBSSENT   const struct kbsentry *
0x00004B4A KDGKBDIACR    struct kbdiacrs *
0x00004B4B KDSKBDIACR    const struct kbdiacrs *
0x00004B4C KDGETKEYCODE  struct kbkeycode *    // I-O
0x00004B4D KDSETKEYCODE  const struct kbkeycode *
0x00004B4E KDSIGACCEPT   int

// <include/linux/lp.h>

0x00000601 LPCHAR       int
0x00000602 LPTIME       int
0x00000604 LPABORT      int
0x00000605 LPSETIRQ     int

```

```

0x00000606 LPGETIRQ    int *
0x00000608 LPWAIT      int
0x00000609 LPCAREFUL   int
0x0000060A LPABORTOPEN int
0x0000060B LPGETSTATUS int *
0x0000060C LRESET      void
0x0000060D LPGETSTATS  struct lp_stats *

// <include/linux/mroute.h>

0x000089E0 SIOCGETVIFCNT struct sioc_vif_req * // I-O
0x000089E1 SIOCGETSGCNT  struct sioc_sg_req * // I-O

// <include/linux/msdos_fs.h> see ioctl_fat(2)

0x82307201 VFAT_IOCTL_READDIR_BOTH  struct dirent [2]
0x82307202 VFAT_IOCTL_READDIR_SHORT struct dirent [2]
0x80047210 FAT_IOCTL_GET_ATTRIBUTES  __u32 *
0x40047211 FAT_IOCTL_SET_ATTRIBUTES  const __u32 *
0x80047213 FAT_IOCTL_GET_VOLUME_ID   __u32 *

// <include/linux/mtio.h>

0x40086D01 MTIOCTOP      const struct mtop *
0x801C6D02 MTIOCGET      struct mtget *
0x80046D03 MTIOCPOS      struct mtpos *
0x80206D04 MTIOCGETCONFIG struct mtconfiginfo *
0x40206D05 MTIOCSETCONFIG const struct mtconfiginfo *

// <include/linux/netrom.h>

0x000089E0 SIOCNRGETPARMS struct nr_parms_struct * // I-O
0x000089E1 SIOCNRSETPARMS const struct nr_parms_struct *
0x000089E2 SIOCNRDECOBS  void
0x000089E3 SIOCNRRTCTL   const int *

// <include/uapi/linux/wireless.h>

// This API is deprecated.

// It is being replaced by nl80211 and cfg80211. See

// https://wireless.wiki.kernel.org/en/developers/documentation/nl80211

x00008b00 SIOCSIWCOMMIT  struct iwreq *
x00008b01 SIOCGIWNAME    struct iwreq *

```

x00008b02	SIOCSIWNWID	struct iwreq *
x00008b03	SIOCGIWNWID	struct iwreq *
x00008b04	SIOCSIWFREQ	struct iwreq *
x00008b05	SIOCGIWFREQ	struct iwreq *
x00008b06	SIOCSIWMODE	struct iwreq *
x00008b07	SIOCGIWMODE	struct iwreq *
x00008b08	SIOCSIWSENS	struct iwreq *
x00008b09	SIOCGIWSENS	struct iwreq *
x00008b0a	SIOCSIWRANGE	struct iwreq *
x00008b0b	SIOCGIWRANGE	struct iwreq *
x00008b0c	SIOCSIWPRIV	struct iwreq *
x00008b0d	SIOCGIWPRIV	struct iwreq *
x00008b0e	SIOCSIWSTATS	struct iwreq *
x00008b0f	SIOCGIWSTATS	struct iwreq *
x00008b10	SIOCSIWSPY	struct iwreq *
x00008b11	SIOCGIWSPY	struct iwreq *
x00008b12	SIOCSIWTHRSPY	struct iwreq *
x00008b13	SIOCGIWTHRSPY	struct iwreq *
x00008b14	SIOCSIWAP	struct iwreq *
x00008b15	SIOCGIWAP	struct iwreq *
x00008b17	SIOCGIWAPLIST	struct iwreq *
x00008b18	SIOCSIWSCAN	struct iwreq *
x00008b19	SIOCGIWSCAN	struct iwreq *
x00008b1a	SIOCSIWESSID	struct iwreq *
x00008b1b	SIOCGIWESSID	struct iwreq *
x00008b1c	SIOCSIWNICKN	struct iwreq *
x00008b1d	SIOCGIWNICKN	struct iwreq *
x00008b20	SIOCSIWRATE	struct iwreq *
x00008b21	SIOCGIWRATE	struct iwreq *
x00008b22	SIOCSIWRTS	struct iwreq *
x00008b23	SIOCGIWRTS	struct iwreq *
x00008b24	SIOCSIWFRAG	struct iwreq *
x00008b25	SIOCGIWFRAG	struct iwreq *

```

x00008b26 SIOCSIWTXPOW struct iwreq *
x00008b27 SIOCGIWTXPOW struct iwreq *
x00008b28 SIOCSIWRETRY struct iwreq *
x00008b29 SIOCGIWRETRY struct iwreq *
x00008b2a SIOCSIWENCODE struct iwreq *
x00008b2b SIOCGIWENCODE struct iwreq *
x00008b2c SIOCSIWPOWER struct iwreq *
x00008b2d SIOCGIWPOWER struct iwreq *
x00008b30 SIOCSIWGENIE struct iwreq *
x00008b31 SIOCGIWGENIE struct iwreq *
x00008b16 SIOCSIWMLME struct iwreq *
x00008b32 SIOCSIWAUTH struct iwreq *
x00008b33 SIOCGIWAUTH struct iwreq *
x00008b34 SIOCSIWENCODEEXT struct iwreq *
x00008b35 SIOCGIWENCODEEXT struct iwreq *
x00008b36 SIOCSIWPMKSA struct iwreq *

// <include/linux/sbpcd.h>

0x00009000 DDIOCSDBG const int *
0x00005382 CDROMAUDIOBUFSIZ int

// <include/linux/scc.h>

0x00005470 TIOCSCCINI void
0x00005471 TIOCCHANINI const struct scc_modem *
0x00005472 TIOCGKISS struct ioctl_command * // I-O
0x00005473 TIOCSKISS const struct ioctl_command *
0x00005474 TIOCSCCSTAT struct scc_stat *

// <include/linux/scsi.h>

0x00005382 SCSI_IOCTL_GET_IDLUN struct { int [2]; } *
0x00005383 SCSI_IOCTL_TAGGED_ENABLE void
0x00005384 SCSI_IOCTL_TAGGED_DISABLE void
0x00005385 SCSI_IOCTL_PROBE_HOST const int * // MORE

// <include/linux/smb_fs.h>

0x80027501 SMB_IOC_GETMOUNTUID uid_t *

// <include/uapi/linux/sockios.h> see netdevice(7)

```

0x0000890B	SIOCADDRT	const struct rtenry *	// MORE
0x0000890C	SIOCDELRT	const struct rtenry *	// MORE
0x00008910	SIOCGIFNAME	char []	
0x00008911	SIOCSIFLINK	void	
0x00008912	SIOCGIFCONF	struct ifconf *	// MORE // I-O
0x00008913	SIOCGIFFLAGS	struct ifreq *	// I-O
0x00008914	SIOCSIFFLAGS	const struct ifreq *	
0x00008915	SIOCGIFADDR	struct ifreq *	// I-O
0x00008916	SIOCSIFADDR	const struct ifreq *	
0x00008917	SIOCGIFDSTADDR	struct ifreq *	// I-O
0x00008918	SIOCSIFDSTADDR	const struct ifreq *	
0x00008919	SIOCGIFBRDADDR	struct ifreq *	// I-O
0x0000891A	SIOCSIFBRDADDR	const struct ifreq *	
0x0000891B	SIOCGIFNETMASK	struct ifreq *	// I-O
0x0000891C	SIOCSIFNETMASK	const struct ifreq *	
0x0000891D	SIOCGIFMETRIC	struct ifreq *	// I-O
0x0000891E	SIOCSIFMETRIC	const struct ifreq *	
0x0000891F	SIOCGIFMEM	struct ifreq *	// I-O
0x00008920	SIOCSIFMEM	const struct ifreq *	
0x00008921	SIOCGIFMTU	struct ifreq *	// I-O
0x00008922	SIOCSIFMTU	const struct ifreq *	
0x00008923	OLD_SIOCGIFHWADDR	struct ifreq *	// I-O
0x00008924	SIOCSIFHWADDR	const struct ifreq *	// MORE
0x00008925	SIOCGIFENCAP	int *	
0x00008926	SIOCSIFENCAP	const int *	
0x00008927	SIOCGIFHWADDR	struct ifreq *	// I-O
0x00008929	SIOCGIFSLAVE	void	
0x00008930	SIOCSIFSLAVE	void	
0x00008931	SIOCADDMULTI	const struct ifreq *	
0x00008932	SIOCDELMULTI	const struct ifreq *	
0x00008940	SIOCADDRTOLD	void	
0x00008941	SIOCDELRTOLD	void	
0x00008950	SIOCDDARP	const struct arpreq *	

```

0x00008951 SIOCGARP      struct arpreq *      // I-O
0x00008952 SIOCSARP      const struct arpreq *
0x00008960 SIOCDRARP     const struct arpreq *
0x00008961 SIOCGRARP     struct arpreq *      // I-O
0x00008962 SIOCSRARP     const struct arpreq *
0x00008970 SIOCGIFMAP    struct ifreq *       // I-O
0x00008971 SIOCSIFMAP    const struct ifreq *

// <include/linux/soundcard.h>

0x00005100 SNDCTL_SEQ_RESET void
0x00005101 SNDCTL_SEQ_SYNC void
0xC08C5102 SNDCTL_SYNTH_INFO struct synth_info * // I-O
0xC0045103 SNDCTL_SEQ_CTRLRATE int *           // I-O
0x80045104 SNDCTL_SEQ_GETOUTCOUNT int *
0x80045105 SNDCTL_SEQ_GETINCOUNT int *
0x40045106 SNDCTL_SEQ_PERCMODE void
0x40285107 SNDCTL_FM_LOAD_INSTR const struct sbi_instrument *
0x40045108 SNDCTL_SEQ_TESTMIDI const int *
0x40045109 SNDCTL_SEQ_RESETSAMPLES const int *
0x8004510A SNDCTL_SEQ_NRSYNTHS int *
0x8004510B SNDCTL_SEQ_NRMIDIS int *
0xC074510C SNDCTL_MIDI_INFO struct midi_info * // I-O
0x4004510D SNDCTL_SEQ_THRESHOLD const int *
0xC004510E SNDCTL_SYNTH_MEMAVL int *           // I-O
0x4004510F SNDCTL_FM_4OP_ENABLE const int *
0xCFB85110 SNDCTL_PMGR_ACCESS struct patmgr_info * // I-O
0x00005111 SNDCTL_SEQ_PANIC void
0x40085112 SNDCTL_SEQ_OUTOFBAND const struct seq_event_rec *
0xC0045401 SNDCTL_TMR_TIMEBASE int *           // I-O
0x00005402 SNDCTL_TMR_START void
0x00005403 SNDCTL_TMR_STOP void
0x00005404 SNDCTL_TMR_CONTINUE void
0xC0045405 SNDCTL_TMR_TEMPO int *              // I-O
0xC0045406 SNDCTL_TMR_SOURCE int *            // I-O

```

```

0x40045407 SNDCTL_TMR_METRONOME  const int *
0x40045408 SNDCTL_TMR_SELECT      int *          // I-O
0xCFB85001 SNDCTL_PMGR_IFACE      struct patmgr_info * // I-O
0xC0046D00 SNDCTL_MIDI_PRETIME    int *          // I-O
0xC0046D01 SNDCTL_MIDI_MPUMODE    const int *
0xC0216D02 SNDCTL_MIDI_MPUCMD     struct mpu_command_rec * // I-O
0x00005000 SNDCTL_DSP_RESET       void
0x00005001 SNDCTL_DSP_SYNC        void
0xC0045002 SNDCTL_DSP_SPEED       int * // I-O
0xC0045003 SNDCTL_DSP_STEREO      int * // I-O
0xC0045004 SNDCTL_DSP_GETBLKSIZE  int * // I-O
0xC0045006 SOUND_PCM_WRITE_CHANNELS int * // I-O
0xC0045007 SOUND_PCM_WRITE_FILTER int * // I-O
0x00005008 SNDCTL_DSP_POST        void
0xC0045009 SNDCTL_DSP_SUBDIVIDE   int * // I-O
0xC004500A SNDCTL_DSP_SETFRAGMENT int * // I-O
0x8004500B SNDCTL_DSP_GETFMTS     int *
0xC0045005 SNDCTL_DSP_SETFMT      int * // I-O
0x800C500C SNDCTL_DSP_GETOSPACE   struct audio_buf_info *
0x800C500D SNDCTL_DSP_GETISPACE   struct audio_buf_info *
0x0000500E SNDCTL_DSP_NONBLOCK    void
0x80045002 SOUND_PCM_READ_RATE    int *
0x80045006 SOUND_PCM_READ_CHANNELS int *
0x80045005 SOUND_PCM_READ_BITS    int *
0x80045007 SOUND_PCM_READ_FILTER  int *
0x00004300 SNDCTL_COPR_RESET      void
0xCFB04301 SNDCTL_COPR_LOAD       const struct copr_buffer *
0xC0144302 SNDCTL_COPR_RDATA      struct copr_debug_buf * // I-O
0xC0144303 SNDCTL_COPR_RCODE      struct copr_debug_buf * // I-O
0x40144304 SNDCTL_COPR_WDATA      const struct copr_debug_buf *
0x40144305 SNDCTL_COPR_WCODE      const struct copr_debug_buf *
0xC0144306 SNDCTL_COPR_RUN        struct copr_debug_buf * // I-O
0xC0144307 SNDCTL_COPR_HALT       struct copr_debug_buf * // I-O

```

0x4FA44308	SNDCTL_COPR_SENDMSG	const struct copr_msg *
0x8FA44309	SNDCTL_COPR_RCVMSG	struct copr_msg *
0x80044D00	SOUND_MIXER_READ_VOLUME	int *
0x80044D01	SOUND_MIXER_READ_BASS	int *
0x80044D02	SOUND_MIXER_READ_TREBLE	int *
0x80044D03	SOUND_MIXER_READ_SYNTH	int *
0x80044D04	SOUND_MIXER_READ_PCM	int *
0x80044D05	SOUND_MIXER_READ_SPEAKER	int *
0x80044D06	SOUND_MIXER_READ_LINE	int *
0x80044D07	SOUND_MIXER_READ_MIC	int *
0x80044D08	SOUND_MIXER_READ_CD	int *
0x80044D09	SOUND_MIXER_READ_IMIX	int *
0x80044D0A	SOUND_MIXER_READ_ALTPCM	int *
0x80044D0B	SOUND_MIXER_READ_RECLEV	int *
0x80044D0C	SOUND_MIXER_READ_IGAIN	int *
0x80044D0D	SOUND_MIXER_READ_OGAIN	int *
0x80044D0E	SOUND_MIXER_READ_LINE1	int *
0x80044D0F	SOUND_MIXER_READ_LINE2	int *
0x80044D10	SOUND_MIXER_READ_LINE3	int *
0x80044D1C	SOUND_MIXER_READ_MUTE	int *
0x80044D1D	SOUND_MIXER_READ_ENHANCE	int *
0x80044D1E	SOUND_MIXER_READ_LOUD	int *
0x80044DFF	SOUND_MIXER_READ_RECSRC	int *
0x80044DFE	SOUND_MIXER_READ_DEVMASK	int *
0x80044DFD	SOUND_MIXER_READ_RECMASK	int *
0x80044DFB	SOUND_MIXER_READ_STEREODEVS	int *
0x80044DFC	SOUND_MIXER_READ_CAPS	int *
0xC0044D00	SOUND_MIXER_WRITE_VOLUME	int * // I-O
0xC0044D01	SOUND_MIXER_WRITE_BASS	int * // I-O
0xC0044D02	SOUND_MIXER_WRITE_TREBLE	int * // I-O
0xC0044D03	SOUND_MIXER_WRITE_SYNTH	int * // I-O
0xC0044D04	SOUND_MIXER_WRITE_PCM	int * // I-O
0xC0044D05	SOUND_MIXER_WRITE_SPEAKER	int * // I-O

```

0xC0044D06 SOUND_MIXER_WRITE_LINE int * // I-O
0xC0044D07 SOUND_MIXER_WRITE_MIC int * // I-O
0xC0044D08 SOUND_MIXER_WRITE_CD int * // I-O
0xC0044D09 SOUND_MIXER_WRITE_IMIX int * // I-O
0xC0044D0A SOUND_MIXER_WRITE_ALTPCM int * // I-O
0xC0044D0B SOUND_MIXER_WRITE_RECLEV int * // I-O
0xC0044D0C SOUND_MIXER_WRITE_IGAIN int * // I-O
0xC0044D0D SOUND_MIXER_WRITE_OGAIN int * // I-O
0xC0044D0E SOUND_MIXER_WRITE_LINE1 int * // I-O
0xC0044D0F SOUND_MIXER_WRITE_LINE2 int * // I-O
0xC0044D10 SOUND_MIXER_WRITE_LINE3 int * // I-O
0xC0044D1C SOUND_MIXER_WRITE_MUTE int * // I-O
0xC0044D1D SOUND_MIXER_WRITE_ENHANCE int * // I-O
0xC0044D1E SOUND_MIXER_WRITE_LOUD int * // I-O
0xC0044DFF SOUND_MIXER_WRITE_RECSRC int * // I-O

// <include/linux/timerfd.h> see timerfd_create(2)

0x40085400 TFD_IOC_SET_TICKS uint64_t *

// <include/linux/umsdos_fs.h>

0x000004D2 UMSDOS_READDIR_DOS struct umsdos_ioctl * // I-O
0x000004D3 UMSDOS_UNLINK_DOS const struct umsdos_ioctl *
0x000004D4 UMSDOS_RMDIR_DOS const struct umsdos_ioctl *
0x000004D5 UMSDOS_STAT_DOS struct umsdos_ioctl * // I-O
0x000004D6 UMSDOS_CREAT_EMD const struct umsdos_ioctl *
0x000004D7 UMSDOS_UNLINK_EMD const struct umsdos_ioctl *
0x000004D8 UMSDOS_READDIR_EMD struct umsdos_ioctl * // I-O
0x000004D9 UMSDOS_GETVERSION struct umsdos_ioctl *
0x000004DA UMSDOS_INIT_EMD void
0x000004DB UMSDOS_DOS_SETUP const struct umsdos_ioctl *
0x000004DC UMSDOS_RENAME_DOS const struct umsdos_ioctl *

// <include/linux/vt.h>

0x00005600 VT_OPENQRY int *
0x00005601 VT_GETMODE struct vt_mode *
0x00005602 VT_SETMODE const struct vt_mode *

```

```

0x00005603 VT_GETSTATE    struct vt_stat *
0x00005604 VT_SENDSIG     void
0x00005605 VT_RELDISP     int
0x00005606 VT_ACTIVATE    int
0x00005607 VT_WAITACTIVE  int
0x00005608 VT_DISALLOCATE int
0x00005609 VT_RESIZE      const struct vt_sizes *
0x0000560A VT_RESIZEX     const struct vt_consize *

```

// More arguments. Some ioctl's take a pointer to a structure which contains addi?

tional pointers. These are documented here in alphabetical order.

CDROMREADAUDIO takes an input pointer `const struct cdrom_read_audio *`. The `buf` field points to an output buffer of length `nframes * CD_FRAME_SIZE_RAW`.

CDROMREADCOOKED, CDROMREADMODE1, CDROMREADMODE2, and CDROMREADRAW take an input pointer `const struct cdrom_msf *`. They use the same pointer as an output pointer to `char []`. The length varies by request. For CDROMREADMODE1, most drivers use `CD_FRAME_SIZE`, but the Optics Storage driver uses `OPT_BLOCK_SIZE` instead (both have the numerical value 2048).

```

CDROMREADCOOKED  char [CD_FRAME_SIZE]
CDROMREADMODE1   char [CD_FRAME_SIZE or OPT_BLOCK_SIZE]
CDROMREADMODE2   char [CD_FRAME_SIZE_RAW]
CDROMREADRAW     char [CD_FRAME_SIZE_RAW]

```

EQL_ENSLAVE, EQL_EMANCIPATE, EQL_GETSLAVECFG, EQL_SETSLAVECFG, EQL_GETMASTERCFG, and EQL_SETMASTERCFG take a `struct ifreq *`. The `ifr_data` field is a pointer to an? other structure as follows:

```

EQL_ENSLAVE      const struct slaving_request *
EQL_EMANCIPATE   const struct slaving_request *
EQL_GETSLAVECFG  struct slave_config *          // I-O
EQL_SETSLAVECFG  const struct slave_config *
EQL_GETMASTERCFG struct master_config *
EQL_SETMASTERCFG const struct master_config *

```

FDRAWCMD takes a `struct floppy_raw_cmd *`. If `flags & FD_RAW_WRITE` is nonzero, then data points to an input buffer of length `length`. If `flags & FD_RAW_READ` is non? zero, then data points to an output buffer of length `length`.

GIO_FONTX and PIO_FONTX take a struct console_font_desc * or a const struct console_font_desc *, respectively. chardata points to a buffer of char [charcount].

This is an output buffer for GIO_FONTX and an input buffer for PIO_FONTX.

GIO_UNIMAP and PIO_UNIMAP take a struct unimapdesc * or a const struct unimapdesc *, respectively. entries points to a buffer of struct unipair [entries]. This is an output buffer for GIO_UNIMAP and an input buffer for PIO_UNIMAP.

KDADDIO, KDELIO, KDDISABIO, and KDENABIO enable or disable access to I/O ports.

They are essentially alternate interfaces to 'ioperm'.

KDMAPDISP and KDUNMAPDISP enable or disable memory mappings or I/O port access.

They are not implemented in the kernel.

SCSI_IOCTL_PROBE_HOST takes an input pointer const int *, which is a length. It uses the same pointer as an output pointer to a char [] buffer of this length.

SIOCADDRT and SIOCDELRT take an input pointer whose type depends on the protocol:

Most protocols	const struct rtenv *
AX.25	const struct ax25_route *
NET/ROM	const struct nr_route_struct *
INET6	const struct in6_rtmsg *

SIOCGIFCONF takes a struct ifconf *. The ifc_buf field points to a buffer of length ifc_len bytes, into which the kernel writes a list of type struct ifreq [].

SIOCSIFHWADDR takes an input pointer whose type depends on the protocol:

Most protocols	const struct ifreq *
AX.25	const char [AX25_ADDR_LEN]

TIOCLINUX takes a const char *. It uses this to distinguish several independent subcases. In the table below, N + foo means foo after an N-byte pad. struct se?

lection is implicitly defined in drivers/char/selection.c

TIOCLINUX-2	1 + const struct selection *
TIOCLINUX-3	void
TIOCLINUX-4	void
TIOCLINUX-5	4 + const struct { long [8]; } *
TIOCLINUX-6	char *
TIOCLINUX-7	char *
TIOCLINUX-10	1 + const char *

// Duplicate ioctls

This list does not include ioctls in the range SIOCDEVPRIVATE and SIOCPROTOPRIVATE.

0x00000001	FDSETPRM	FIBMAP
0x00000002	FDDEFPRM	FIGETBSZ
0x00005382	CDROMAUDIOBUFSIZ	SCSI_IOCTL_GET_IDLUN
0x00005402	SNDCTL_TMR_START	TCSETS
0x00005403	SNDCTL_TMR_STOP	TCSETSW
0x00005404	SNDCTL_TMR_CONTINUE	TCSETSF

SEE ALSO

ioctl(2), ioctl_fat(2), netdevice(7)

COLOPHON

This page is part of release 5.05 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.

Linux	2019-11-19	IOCTL_LIST(2)
-------	------------	---------------