Package syscall go1.15.2 Latest

Published: **Sep 9, 2020** | License: BSD-3-Clause | Standard library

Overview

Package syscall contains an interface to the low-level operating system primitives. The details vary depending on the underlying system, and by default, godoc will display the syscall documentation for the current system. If you want godoc to display syscall documentation for another system, set $GOOS and $GOARCH to the desired system. For example, if you want to view documentation for freebsd/arm on linux/amd64, set $GOOS to freebsd and $GOARCH to arm. The primary use of syscall is inside other packages that provide a more portable interface to the system, such as "os", "time" and "net". Use those packages rather than this one if you can. For details of the functions and data types in this package consult the manuals for the appropriate operating system. These calls return err == nil to indicate success; otherwise err is an operating system error describing the failure. On most systems, that error has type syscall.Errno.

Deprecated: this package is locked down. Callers should use the corresponding package in the golang.org/x/sys repository instead. That is also where updates required by new systems or versions should be applied. See https://golang.org/s/go1.4-syscall for more information.

Constants

```
const (
    AF_ALG                          = 0x26
    AF_APPLETALK                    = 0x5
    AF_ASH                          = 0x12
    AF_ATMPVC                       = 0x8
    AF_ATMSVC                       = 0x14
    AF_AX25                         = 0x3
    AF_BLUETOOTH                    = 0x1f
    AF_BRIDGE                       = 0x7
    AF_CAIF                         = 0x25
    AF_CAN                          = 0x1d
    AF_DECnet                       = 0xc
    AF_ECONET                       = 0x13
    AF_FILE                         = 0x1
    AF_IEEE802154                   = 0x24
    AF_INET                         = 0x2
    AF_INET6                        = 0xa
    AF_IPX                          = 0x4
    AF_IRDA                         = 0x17
    AF_ISDN                         = 0x22
    AF_IUCV                         = 0x20
    AF_KEY                          = 0xf
    AF_LLC                          = 0x1a
```

```
AF_LOCAL                        = 0x1
AF_MAX                          = 0x27
AF_NETBEUI                      = 0xd
AF_NETLINK                      = 0x10
AF_NETROM                       = 0x6
AF_PACKET                       = 0x11
AF_PHONET                       = 0x23
AF_PPPOX                        = 0x18
AF_RDS                          = 0x15
AF_ROSE                         = 0xb
AF_ROUTE                        = 0x10
AF_RXRPC                        = 0x21
AF_SECURITY                     = 0xe
AF_SNA                          = 0x16
AF_TIPC                         = 0x1e
AF_UNIX                         = 0x1
AF_UNSPEC                       = 0x0
AF_WANPIPE                      = 0x19
AF_X25                          = 0x9
ARPHRD_ADAPT                    = 0x108
ARPHRD_APPLETLK                 = 0x8
ARPHRD_ARCNET                   = 0x7
ARPHRD_ASH                      = 0x30d
ARPHRD_ATM                      = 0x13
ARPHRD_AX25                     = 0x3
ARPHRD_BIF                      = 0x307
ARPHRD_CHAOS                    = 0x5
ARPHRD_CISCO                    = 0x201
ARPHRD_CSLIP                    = 0x101
ARPHRD_CSLIP6                   = 0x103
ARPHRD_DDCMP                    = 0x205
ARPHRD_DLCI                     = 0xf
ARPHRD_ECONET                   = 0x30e
ARPHRD_EETHER                   = 0x2
ARPHRD_ETHER                    = 0x1
ARPHRD_EUI64                    = 0x1b
ARPHRD_FCAL                     = 0x311
ARPHRD_FCFABRIC                 = 0x313
ARPHRD_FCPL                     = 0x312
ARPHRD_FCPP                     = 0x310
ARPHRD_FDDI                     = 0x306
ARPHRD_FRAD                     = 0x302
ARPHRD_HDLC                     = 0x201
ARPHRD_HIPPI                    = 0x30c
ARPHRD_HWX25                    = 0x110
ARPHRD_IEEE1394                 = 0x18
ARPHRD_IEEE802                  = 0x6
ARPHRD_IEEE80211                = 0x321
ARPHRD_IEEE80211_PRISM          = 0x322
ARPHRD_IEEE80211_RADIOTAP       = 0x323
ARPHRD_IEEE802154               = 0x324
ARPHRD_IEEE802154_PHY           = 0x325
ARPHRD_IEEE802_TR               = 0x320
ARPHRD_INFINIBAND               = 0x20
```

```
ARPHRD_IPDDP                    = 0x309
ARPHRD_IPGRE                    = 0x30a
ARPHRD_IRDA                     = 0x30f
ARPHRD_LAPB                     = 0x204
ARPHRD_LOCALTLK                 = 0x305
ARPHRD_LOOPBACK                 = 0x304
ARPHRD_METRICOM                 = 0x17
ARPHRD_NETROM                   = 0x0
ARPHRD_NONE                     = 0xfffe
ARPHRD_PIMREG                   = 0x30b
ARPHRD_PPP                      = 0x200
ARPHRD_PRONET                   = 0x4
ARPHRD_RAWHDLC                  = 0x206
ARPHRD_ROSE                     = 0x10e
ARPHRD_RSRVD                    = 0x104
ARPHRD_SIT                      = 0x308
ARPHRD_SKIP                     = 0x303
ARPHRD_SLIP                     = 0x100
ARPHRD_SLIP6                    = 0x102
ARPHRD_TUNNEL                   = 0x300
ARPHRD_TUNNEL6                  = 0x301
ARPHRD_VOID                     = 0xffff
ARPHRD_X25                      = 0x10f
BPF_A                           = 0x10
BPF_ABS                         = 0x20
BPF_ADD                         = 0x0
BPF_ALU                         = 0x4
BPF_AND                         = 0x50
BPF_B                           = 0x10
BPF_DIV                         = 0x30
BPF_H                           = 0x8
BPF_IMM                         = 0x0
BPF_IND                         = 0x40
BPF_JA                          = 0x0
BPF_JEQ                         = 0x10
BPF_JGE                         = 0x30
BPF_JGT                         = 0x20
BPF_JMP                         = 0x5
BPF_JSET                        = 0x40
BPF_K                           = 0x0
BPF_LD                          = 0x0
BPF_LDX                         = 0x1
BPF_LEN                         = 0x80
BPF_LSH                         = 0x60
BPF_MAJOR_VERSION               = 0x1
BPF_MAXINSNS                    = 0x1000
BPF_MEM                         = 0x60
BPF_MEMWORDS                    = 0x10
BPF_MINOR_VERSION               = 0x1
BPF_MISC                        = 0x7
BPF_MSH                         = 0xa0
BPF_MUL                         = 0x20
BPF_NEG                         = 0x80
BPF_OR                          = 0x40
```

```
BPF_RET                         = 0x6
BPF_RSH                         = 0x70
BPF_ST                          = 0x2
BPF_STX                         = 0x3
BPF_SUB                         = 0x10
BPF_TAX                         = 0x0
BPF_TXA                         = 0x80
BPF_W                           = 0x0
BPF_X                           = 0x8
CLONE_CHILD_CLEARTID            = 0x200000
CLONE_CHILD_SETTID              = 0x1000000
CLONE_DETACHED                  = 0x400000
CLONE_FILES                     = 0x400
CLONE_FS                        = 0x200
CLONE_IO                        = 0x80000000
CLONE_NEWIPC                    = 0x8000000
CLONE_NEWNET                    = 0x40000000
CLONE_NEWNS                     = 0x20000
CLONE_NEWPID                    = 0x20000000
CLONE_NEWUSER                   = 0x10000000
CLONE_NEWUTS                    = 0x4000000
CLONE_PARENT                    = 0x8000
CLONE_PARENT_SETTID             = 0x100000
CLONE_PTRACE                    = 0x2000
CLONE_SETTLS                    = 0x80000
CLONE_SIGHAND                   = 0x800
CLONE_SYSVSEM                   = 0x40000
CLONE_THREAD                    = 0x10000
CLONE_UNTRACED                  = 0x800000
CLONE_VFORK                     = 0x4000
CLONE_VM                        = 0x100
DT_BLK                          = 0x6
DT_CHR                          = 0x2
DT_DIR                          = 0x4
DT_FIFO                         = 0x1
DT_LNK                          = 0xa
DT_REG                          = 0x8
DT_SOCK                         = 0xc
DT_UNKNOWN                      = 0x0
DT_WHT                          = 0xe
EPOLLERR                        = 0x8
EPOLLET                         = -0x80000000
EPOLLHUP                        = 0x10
EPOLLIN                         = 0x1
EPOLLMSG                        = 0x400
EPOLLONESHOT                    = 0x40000000
EPOLLOUT                        = 0x4
EPOLLPRI                        = 0x2
EPOLLRDBAND                     = 0x80
EPOLLRDHUP                      = 0x2000
EPOLLRDNORM                     = 0x40
EPOLLWRBAND                     = 0x200
EPOLLWRNORM                     = 0x100
EPOLL_CLOEXEC                   = 0x80000
```

```
EPOLL_CTL_ADD                   = 0x1
EPOLL_CTL_DEL                   = 0x2
EPOLL_CTL_MOD                   = 0x3
EPOLL_NONBLOCK                  = 0x800
ETH_P_1588                      = 0x88f7
ETH_P_8021Q                     = 0x8100
ETH_P_802_2                     = 0x4
ETH_P_802_3                     = 0x1
ETH_P_AARP                      = 0x80f3
ETH_P_ALL                       = 0x3
ETH_P_AOE                       = 0x88a2
ETH_P_ARCNET                    = 0x1a
ETH_P_ARP                       = 0x806
ETH_P_ATALK                     = 0x809b
ETH_P_ATMFATE                   = 0x8884
ETH_P_ATMMPOA                   = 0x884c
ETH_P_AX25                      = 0x2
ETH_P_BPQ                       = 0x8ff
ETH_P_CAIF                      = 0xf7
ETH_P_CAN                       = 0xc
ETH_P_CONTROL                   = 0x16
ETH_P_CUST                      = 0x6006
ETH_P_DDCMP                     = 0x6
ETH_P_DEC                       = 0x6000
ETH_P_DIAG                      = 0x6005
ETH_P_DNA_DL                    = 0x6001
ETH_P_DNA_RC                    = 0x6002
ETH_P_DNA_RT                    = 0x6003
ETH_P_DSA                       = 0x1b
ETH_P_ECONET                    = 0x18
ETH_P_EDSA                      = 0xdada
ETH_P_FCOE                      = 0x8906
ETH_P_FIP                       = 0x8914
ETH_P_HDLC                      = 0x19
ETH_P_IEEE802154                = 0xf6
ETH_P_IEEEPUP                   = 0xa00
ETH_P_IEEEPUPAT                 = 0xa01
ETH_P_IP                        = 0x800
ETH_P_IPV6                      = 0x86dd
ETH_P_IPX                       = 0x8137
ETH_P_IRDA                      = 0x17
ETH_P_LAT                       = 0x6004
ETH_P_LINK_CTL                  = 0x886c
ETH_P_LOCALTALK                 = 0x9
ETH_P_LOOP                      = 0x60
ETH_P_MOBITEX                   = 0x15
ETH_P_MPLS_MC                   = 0x8848
ETH_P_MPLS_UC                   = 0x8847
ETH_P_PAE                       = 0x888e
ETH_P_PAUSE                     = 0x8808
ETH_P_PHONET                    = 0xf5
ETH_P_PPPTALK                   = 0x10
ETH_P_PPP_DISC                  = 0x8863
ETH_P_PPP_MP                    = 0x8
```

```
ETH_P_PPP_SES                   = 0x8864
ETH_P_PUP                       = 0x200
ETH_P_PUPAT                     = 0x201
ETH_P_RARP                      = 0x8035
ETH_P_SCA                       = 0x6007
ETH_P_SLOW                      = 0x8809
ETH_P_SNAP                      = 0x5
ETH_P_TEB                       = 0x6558
ETH_P_TIPC                      = 0x88ca
ETH_P_TRAILER                   = 0x1c
ETH_P_TR_802_2                  = 0x11
ETH_P_WAN_PPP                   = 0x7
ETH_P_WCCP                      = 0x883e
ETH_P_X25                       = 0x805
FD_CLOEXEC                      = 0x1
FD_SETSIZE                      = 0x400
F_DUPFD                         = 0x0
F_DUPFD_CLOEXEC                 = 0x406
F_EXLCK                         = 0x4
F_GETFD                         = 0x1
F_GETFL                         = 0x3
F_GETLEASE                      = 0x401
F_GETLK                         = 0x5
F_GETLK64                       = 0x5
F_GETOWN                        = 0x9
F_GETOWN_EX                     = 0x10
F_GETPIPE_SZ                    = 0x408
F_GETSIG                        = 0xb
F_LOCK                          = 0x1
F_NOTIFY                        = 0x402
F_OK                            = 0x0
F_RDLCK                         = 0x0
F_SETFD                         = 0x2
F_SETFL                         = 0x4
F_SETLEASE                      = 0x400
F_SETLK                         = 0x6
F_SETLK64                       = 0x6
F_SETLKW                        = 0x7
F_SETLKW64                      = 0x7
F_SETOWN                        = 0x8
F_SETOWN_EX                     = 0xf
F_SETPIPE_SZ                    = 0x407
F_SETSIG                        = 0xa
F_SHLCK                         = 0x8
F_TEST                          = 0x3
F_TLOCK                         = 0x2
F_ULOCK                         = 0x0
F_UNLCK                         = 0x2
F_WRLCK                         = 0x1
ICMPV6_FILTER                   = 0x1
IFA_F_DADFAILED                 = 0x8
IFA_F_DEPRECATED                = 0x20
IFA_F_HOMEADDRESS               = 0x10
IFA_F_NODAD                     = 0x2
```

```
IFA_F_OPTIMISTIC                = 0x4
IFA_F_PERMANENT                 = 0x80
IFA_F_SECONDARY                 = 0x1
IFA_F_TEMPORARY                 = 0x1
IFA_F_TENTATIVE                 = 0x40
IFA_MAX                         = 0x7
IFF_ALLMULTI                    = 0x200
IFF_AUTOMEDIA                   = 0x4000
IFF_BROADCAST                   = 0x2
IFF_DEBUG                       = 0x4
IFF_DYNAMIC                     = 0x8000
IFF_LOOPBACK                    = 0x8
IFF_MASTER                      = 0x400
IFF_MULTICAST                   = 0x1000
IFF_NOARP                       = 0x80
IFF_NOTRAILERS                  = 0x20
IFF_NO_PI                       = 0x1000
IFF_ONE_QUEUE                   = 0x2000
IFF_POINTOPOINT                 = 0x10
IFF_PORTSEL                     = 0x2000
IFF_PROMISC                     = 0x100
IFF_RUNNING                     = 0x40
IFF_SLAVE                       = 0x800
IFF_TAP                         = 0x2
IFF_TUN                         = 0x1
IFF_TUN_EXCL                    = 0x8000
IFF_UP                          = 0x1
IFF_VNET_HDR                    = 0x4000
IFNAMSIZ                        = 0x10
IN_ACCESS                       = 0x1
IN_ALL_EVENTS                   = 0xfff
IN_ATTRIB                       = 0x4
IN_CLASSA_HOST                  = 0xffffff
IN_CLASSA_MAX                   = 0x80
IN_CLASSA_NET                   = 0xff000000
IN_CLASSA_NSHIFT                = 0x18
IN_CLASSB_HOST                  = 0xffff
IN_CLASSB_MAX                   = 0x10000
IN_CLASSB_NET                   = 0xffff0000
IN_CLASSB_NSHIFT                = 0x10
IN_CLASSC_HOST                  = 0xff
IN_CLASSC_NET                   = 0xffffff00
IN_CLASSC_NSHIFT                = 0x8
IN_CLOEXEC                      = 0x80000
IN_CLOSE                        = 0x18
IN_CLOSE_NOWRITE                = 0x10
IN_CLOSE_WRITE                  = 0x8
IN_CREATE                       = 0x100
IN_DELETE                       = 0x200
IN_DELETE_SELF                  = 0x400
IN_DONT_FOLLOW                  = 0x2000000
IN_EXCL_UNLINK                  = 0x4000000
IN_IGNORED                      = 0x8000
IN_ISDIR                        = 0x40000000
```

```
IN_LOOPBACKNET                  = 0x7f
IN_MASK_ADD                     = 0x20000000
IN_MODIFY                       = 0x2
IN_MOVE                         = 0xc0
IN_MOVED_FROM                   = 0x40
IN_MOVED_TO                     = 0x80
IN_MOVE_SELF                    = 0x800
IN_NONBLOCK                     = 0x800
IN_ONESHOT                      = 0x80000000
IN_ONLYDIR                      = 0x1000000
IN_OPEN                         = 0x20
IN_Q_OVERFLOW                   = 0x4000
IN_UNMOUNT                      = 0x2000
IPPROTO_AH                      = 0x33
IPPROTO_COMP                    = 0x6c
IPPROTO_DCCP                    = 0x21
IPPROTO_DSTOPTS                 = 0x3c
IPPROTO_EGP                     = 0x8
IPPROTO_ENCAP                   = 0x62
IPPROTO_ESP                     = 0x32
IPPROTO_FRAGMENT                = 0x2c
IPPROTO_GRE                     = 0x2f
IPPROTO_HOPOPTS                 = 0x0
IPPROTO_ICMP                    = 0x1
IPPROTO_ICMPV6                  = 0x3a
IPPROTO_IDP                     = 0x16
IPPROTO_IGMP                    = 0x2
IPPROTO_IP                      = 0x0
IPPROTO_IPIP                    = 0x4
IPPROTO_IPV6                    = 0x29
IPPROTO_MTP                     = 0x5c
IPPROTO_NONE                    = 0x3b
IPPROTO_PIM                     = 0x67
IPPROTO_PUP                     = 0xc
IPPROTO_RAW                     = 0xff
IPPROTO_ROUTING                 = 0x2b
IPPROTO_RSVP                    = 0x2e
IPPROTO_SCTP                    = 0x84
IPPROTO_TCP                     = 0x6
IPPROTO_TP                      = 0x1d
IPPROTO_UDP                     = 0x11
IPPROTO_UDPLITE                 = 0x88
IPV6_2292DSTOPTS                = 0x4
IPV6_2292HOPLIMIT               = 0x8
IPV6_2292HOPOPTS                = 0x3
IPV6_2292PKTINFO                = 0x2
IPV6_2292PKTOPTIONS             = 0x6
IPV6_2292RTHDR                  = 0x5
IPV6_ADDRFORM                   = 0x1
IPV6_ADD_MEMBERSHIP             = 0x14
IPV6_AUTHHDR                    = 0xa
IPV6_CHECKSUM                   = 0x7
IPV6_DROP_MEMBERSHIP            = 0x15
IPV6_DSTOPTS                    = 0x3b
```

```
IPV6_HOPLIMIT                   = 0x34
IPV6_HOPOPTS                    = 0x36
IPV6_IPSEC_POLICY               = 0x22
IPV6_JOIN_ANYCAST               = 0x1b
IPV6_JOIN_GROUP                 = 0x14
IPV6_LEAVE_ANYCAST              = 0x1c
IPV6_LEAVE_GROUP                = 0x15
IPV6_MTU                        = 0x18
IPV6_MTU_DISCOVER               = 0x17
IPV6_MULTICAST_HOPS             = 0x12
IPV6_MULTICAST_IF               = 0x11
IPV6_MULTICAST_LOOP             = 0x13
IPV6_NEXTHOP                    = 0x9
IPV6_PKTINFO                    = 0x32
IPV6_PMTUDISC_DO                = 0x2
IPV6_PMTUDISC_DONT              = 0x0
IPV6_PMTUDISC_PROBE             = 0x3
IPV6_PMTUDISC_WANT              = 0x1
IPV6_RECVDSTOPTS                = 0x3a
IPV6_RECVERR                    = 0x19
IPV6_RECVHOPLIMIT               = 0x33
IPV6_RECVHOPOPTS                = 0x35
IPV6_RECVPKTINFO                = 0x31
IPV6_RECVRTHDR                  = 0x38
IPV6_RECVTCLASS                 = 0x42
IPV6_ROUTER_ALERT               = 0x16
IPV6_RTHDR                      = 0x39
IPV6_RTHDRDSTOPTS               = 0x37
IPV6_RTHDR_LOOSE                = 0x0
IPV6_RTHDR_STRICT               = 0x1
IPV6_RTHDR_TYPE_0               = 0x0
IPV6_RXDSTOPTS                  = 0x3b
IPV6_RXHOPOPTS                  = 0x36
IPV6_TCLASS                     = 0x43
IPV6_UNICAST_HOPS               = 0x10
IPV6_V6ONLY                     = 0x1a
IPV6_XFRM_POLICY                = 0x23
IP_ADD_MEMBERSHIP               = 0x23
IP_ADD_SOURCE_MEMBERSHIP        = 0x27
IP_BLOCK_SOURCE                 = 0x26
IP_DEFAULT_MULTICAST_LOOP       = 0x1
IP_DEFAULT_MULTICAST_TTL        = 0x1
IP_DF                           = 0x4000
IP_DROP_MEMBERSHIP              = 0x24
IP_DROP_SOURCE_MEMBERSHIP       = 0x28
IP_FREEBIND                     = 0xf
IP_HDRINCL                      = 0x3
IP_IPSEC_POLICY                 = 0x10
IP_MAXPACKET                    = 0xffff
IP_MAX_MEMBERSHIPS              = 0x14
IP_MF                           = 0x2000
IP_MINTTL                       = 0x15
IP_MSFILTER                     = 0x29
IP_MSS                          = 0x240
```

```
IP_MTU                          = 0xe
IP_MTU_DISCOVER                 = 0xa
IP_MULTICAST_IF                 = 0x20
IP_MULTICAST_LOOP               = 0x22
IP_MULTICAST_TTL                = 0x21
IP_OFFMASK                      = 0x1fff
IP_OPTIONS                      = 0x4
IP_ORIGDSTADDR                  = 0x14
IP_PASSSEC                      = 0x12
IP_PKTINFO                      = 0x8
IP_PKTOPTIONS                   = 0x9
IP_PMTUDISC                     = 0xa
IP_PMTUDISC_DO                  = 0x2
IP_PMTUDISC_DONT                = 0x0
IP_PMTUDISC_PROBE               = 0x3
IP_PMTUDISC_WANT                = 0x1
IP_RECVERR                      = 0xb
IP_RECVOPTS                     = 0x6
IP_RECVORIGDSTADDR              = 0x14
IP_RECVRETOPTS                  = 0x7
IP_RECVTOS                      = 0xd
IP_RECVTTL                      = 0xc
IP_RETOPTS                      = 0x7
IP_RF                           = 0x8000
IP_ROUTER_ALERT                 = 0x5
IP_TOS                          = 0x1
IP_TRANSPARENT                  = 0x13
IP_TTL                          = 0x2
IP_UNBLOCK_SOURCE               = 0x25
IP_XFRM_POLICY                  = 0x11
LINUX_REBOOT_CMD_CAD_OFF        = 0x0
LINUX_REBOOT_CMD_CAD_ON         = 0x89abcdef
LINUX_REBOOT_CMD_HALT           = 0xcdef0123
LINUX_REBOOT_CMD_KEXEC          = 0x45584543
LINUX_REBOOT_CMD_POWER_OFF      = 0x4321fedc
LINUX_REBOOT_CMD_RESTART        = 0x1234567
LINUX_REBOOT_CMD_RESTART2       = 0xa1b2c3d4
LINUX_REBOOT_CMD_SW_SUSPEND     = 0xd000fce2
LINUX_REBOOT_MAGIC1             = 0xfee1dead
LINUX_REBOOT_MAGIC2             = 0x28121969
LOCK_EX                         = 0x2
LOCK_NB                         = 0x4
LOCK_SH                         = 0x1
LOCK_UN                         = 0x8
MADV_DOFORK                     = 0xb
MADV_DONTFORK                   = 0xa
MADV_DONTNEED                   = 0x4
MADV_HUGEPAGE                   = 0xe
MADV_HWPOISON                   = 0x64
MADV_MERGEABLE                  = 0xc
MADV_NOHUGEPAGE                 = 0xf
MADV_NORMAL                     = 0x0
MADV_RANDOM                     = 0x1
MADV_REMOVE                     = 0x9
```

```
MADV_SEQUENTIAL                 = 0x2
MADV_UNMERGEABLE                = 0xd
MADV_WILLNEED                   = 0x3
MAP_32BIT                       = 0x40
MAP_ANON                        = 0x20
MAP_ANONYMOUS                   = 0x20
MAP_DENYWRITE                   = 0x800
MAP_EXECUTABLE                  = 0x1000
MAP_FILE                        = 0x0
MAP_FIXED                       = 0x10
MAP_GROWSDOWN                   = 0x100
MAP_HUGETLB                     = 0x40000
MAP_LOCKED                      = 0x2000
MAP_NONBLOCK                    = 0x10000
MAP_NORESERVE                   = 0x4000
MAP_POPULATE                    = 0x8000
MAP_PRIVATE                     = 0x2
MAP_SHARED                      = 0x1
MAP_STACK                       = 0x20000
MAP_TYPE                        = 0xf
MCL_CURRENT                     = 0x1
MCL_FUTURE                      = 0x2
MNT_DETACH                      = 0x2
MNT_EXPIRE                      = 0x4
MNT_FORCE                       = 0x1
MSG_CMSG_CLOEXEC                = 0x40000000
MSG_CONFIRM                     = 0x800
MSG_CTRUNC                      = 0x8
MSG_DONTROUTE                   = 0x4
MSG_DONTWAIT                    = 0x40
MSG_EOR                         = 0x80
MSG_ERRQUEUE                    = 0x2000
MSG_FASTOPEN                    = 0x20000000
MSG_FIN                         = 0x200
MSG_MORE                        = 0x8000
MSG_NOSIGNAL                    = 0x4000
MSG_OOB                         = 0x1
MSG_PEEK                        = 0x2
MSG_PROXY                       = 0x10
MSG_RST                         = 0x1000
MSG_SYN                         = 0x400
MSG_TRUNC                       = 0x20
MSG_TRYHARD                     = 0x4
MSG_WAITALL                     = 0x100
MSG_WAITFORONE                  = 0x10000
MS_ACTIVE                       = 0x40000000
MS_ASYNC                        = 0x1
MS_BIND                         = 0x1000
MS_DIRSYNC                      = 0x80
MS_INVALIDATE                   = 0x2
MS_I_VERSION                    = 0x800000
MS_KERNMOUNT                    = 0x400000
MS_MANDLOCK                     = 0x40
MS_MGC_MSK                      = 0xffff0000
```

```
MS_MGC_VAL                      = 0xc0ed0000
MS_MOVE                         = 0x2000
MS_NOATIME                      = 0x400
MS_NODEV                        = 0x4
MS_NODIRATIME                   = 0x800
MS_NOEXEC                       = 0x8
MS_NOSUID                       = 0x2
MS_NOUSER                       = -0x80000000
MS_POSIXACL                     = 0x10000
MS_PRIVATE                      = 0x40000
MS_RDONLY                       = 0x1
MS_REC                          = 0x4000
MS_RELATIME                     = 0x200000
MS_REMOUNT                      = 0x20
MS_RMT_MASK                     = 0x800051
MS_SHARED                       = 0x100000
MS_SILENT                       = 0x8000
MS_SLAVE                        = 0x80000
MS_STRICTATIME                  = 0x1000000
MS_SYNC                         = 0x4
MS_SYNCHRONOUS                  = 0x10
MS_UNBINDABLE                   = 0x20000
NAME_MAX                        = 0xff
NETLINK_ADD_MEMBERSHIP          = 0x1
NETLINK_AUDIT                   = 0x9
NETLINK_BROADCAST_ERROR         = 0x4
NETLINK_CONNECTOR               = 0xb
NETLINK_DNRTMSG                 = 0xe
NETLINK_DROP_MEMBERSHIP         = 0x2
NETLINK_ECRYPTFS                = 0x13
NETLINK_FIB_LOOKUP              = 0xa
NETLINK_FIREWALL                = 0x3
NETLINK_GENERIC                 = 0x10
NETLINK_INET_DIAG               = 0x4
NETLINK_IP6_FW                  = 0xd
NETLINK_ISCSI                   = 0x8
NETLINK_KOBJECT_UEVENT          = 0xf
NETLINK_NETFILTER               = 0xc
NETLINK_NFLOG                   = 0x5
NETLINK_NO_ENOBUFS              = 0x5
NETLINK_PKTINFO                 = 0x3
NETLINK_ROUTE                   = 0x0
NETLINK_SCSITRANSPORT           = 0x12
NETLINK_SELINUX                 = 0x7
NETLINK_UNUSED                  = 0x1
NETLINK_USERSOCK                = 0x2
NETLINK_XFRM                    = 0x6
NLA_ALIGNTO                     = 0x4
NLA_F_NESTED                    = 0x8000
NLA_F_NET_BYTEORDER             = 0x4000
NLA_HDRLEN                      = 0x4
NLMSG_ALIGNTO                   = 0x4
NLMSG_DONE                      = 0x3
NLMSG_ERROR                     = 0x2
```

```
NLMSG_HDRLEN                    = 0x10
NLMSG_MIN_TYPE                  = 0x10
NLMSG_NOOP                      = 0x1
NLMSG_OVERRUN                   = 0x4
NLM_F_ACK                       = 0x4
NLM_F_APPEND                    = 0x800
NLM_F_ATOMIC                    = 0x400
NLM_F_CREATE                    = 0x400
NLM_F_DUMP                      = 0x300
NLM_F_ECHO                      = 0x8
NLM_F_EXCL                      = 0x200
NLM_F_MATCH                     = 0x200
NLM_F_MULTI                     = 0x2
NLM_F_REPLACE                   = 0x100
NLM_F_REQUEST                   = 0x1
NLM_F_ROOT                      = 0x100
O_ACCMODE                       = 0x3
O_APPEND                        = 0x400
O_ASYNC                         = 0x2000
O_CLOEXEC                       = 0x80000
O_CREAT                         = 0x40
O_DIRECT                        = 0x4000
O_DIRECTORY                     = 0x10000
O_DSYNC                         = 0x1000
O_EXCL                          = 0x80
O_FSYNC                         = 0x101000
O_LARGEFILE                     = 0x0
O_NDELAY                        = 0x800
O_NOATIME                       = 0x40000
O_NOCTTY                        = 0x100
O_NOFOLLOW                      = 0x20000
O_NONBLOCK                      = 0x800
O_RDONLY                        = 0x0
O_RDWR                          = 0x2
O_RSYNC                         = 0x101000
O_SYNC                          = 0x101000
O_TRUNC                         = 0x200
O_WRONLY                        = 0x1
PACKET_ADD_MEMBERSHIP           = 0x1
PACKET_BROADCAST                = 0x1
PACKET_DROP_MEMBERSHIP          = 0x2
PACKET_FASTROUTE                = 0x6
PACKET_HOST                     = 0x0
PACKET_LOOPBACK                 = 0x5
PACKET_MR_ALLMULTI              = 0x2
PACKET_MR_MULTICAST             = 0x0
PACKET_MR_PROMISC               = 0x1
PACKET_MULTICAST                = 0x2
PACKET_OTHERHOST                = 0x3
PACKET_OUTGOING                 = 0x4
PACKET_RECV_OUTPUT              = 0x3
PACKET_RX_RING                  = 0x5
PACKET_STATISTICS               = 0x6
PRIO_PGRP                       = 0x1
```

```
PRIO_PROCESS                    = 0x0
PRIO_USER                       = 0x2
PROT_EXEC                       = 0x4
PROT_GROWSDOWN                  = 0x1000000
PROT_GROWSUP                    = 0x2000000
PROT_NONE                       = 0x0
PROT_READ                       = 0x1
PROT_WRITE                      = 0x2
PR_CAPBSET_DROP                 = 0x18
PR_CAPBSET_READ                 = 0x17
PR_ENDIAN_BIG                   = 0x0
PR_ENDIAN_LITTLE                = 0x1
PR_ENDIAN_PPC_LITTLE            = 0x2
PR_FPEMU_NOPRINT                = 0x1
PR_FPEMU_SIGFPE                 = 0x2
PR_FP_EXC_ASYNC                 = 0x2
PR_FP_EXC_DISABLED              = 0x0
PR_FP_EXC_DIV                   = 0x10000
PR_FP_EXC_INV                   = 0x100000
PR_FP_EXC_NONRECOV              = 0x1
PR_FP_EXC_OVF                   = 0x20000
PR_FP_EXC_PRECISE               = 0x3
PR_FP_EXC_RES                   = 0x80000
PR_FP_EXC_SW_ENABLE             = 0x80
PR_FP_EXC_UND                   = 0x40000
PR_GET_DUMPABLE                 = 0x3
PR_GET_ENDIAN                   = 0x13
PR_GET_FPEMU                    = 0x9
PR_GET_FPEXC                    = 0xb
PR_GET_KEEPCAPS                 = 0x7
PR_GET_NAME                     = 0x10
PR_GET_PDEATHSIG                = 0x2
PR_GET_SECCOMP                  = 0x15
PR_GET_SECUREBITS               = 0x1b
PR_GET_TIMERSLACK               = 0x1e
PR_GET_TIMING                   = 0xd
PR_GET_TSC                      = 0x19
PR_GET_UNALIGN                  = 0x5
PR_MCE_KILL                     = 0x21
PR_MCE_KILL_CLEAR               = 0x0
PR_MCE_KILL_DEFAULT             = 0x2
PR_MCE_KILL_EARLY               = 0x1
PR_MCE_KILL_GET                 = 0x22
PR_MCE_KILL_LATE                = 0x0
PR_MCE_KILL_SET                 = 0x1
PR_SET_DUMPABLE                 = 0x4
PR_SET_ENDIAN                   = 0x14
PR_SET_FPEMU                    = 0xa
PR_SET_FPEXC                    = 0xc
PR_SET_KEEPCAPS                 = 0x8
PR_SET_NAME                     = 0xf
PR_SET_PDEATHSIG                = 0x1
PR_SET_PTRACER                  = 0x59616d61
PR_SET_SECCOMP                  = 0x16
```

```
PR_SET_SECUREBITS               = 0x1c
PR_SET_TIMERSLACK               = 0x1d
PR_SET_TIMING                   = 0xe
PR_SET_TSC                      = 0x1a
PR_SET_UNALIGN                  = 0x6
PR_TASK_PERF_EVENTS_DISABLE     = 0x1f
PR_TASK_PERF_EVENTS_ENABLE      = 0x20
PR_TIMING_STATISTICAL           = 0x0
PR_TIMING_TIMESTAMP             = 0x1
PR_TSC_ENABLE                   = 0x1
PR_TSC_SIGSEGV                  = 0x2
PR_UNALIGN_NOPRINT              = 0x1
PR_UNALIGN_SIGBUS               = 0x2
PTRACE_ARCH_PRCTL               = 0x1e
PTRACE_ATTACH                   = 0x10
PTRACE_CONT                     = 0x7
PTRACE_DETACH                   = 0x11
PTRACE_EVENT_CLONE              = 0x3
PTRACE_EVENT_EXEC               = 0x4
PTRACE_EVENT_EXIT               = 0x6
PTRACE_EVENT_FORK               = 0x1
PTRACE_EVENT_VFORK              = 0x2
PTRACE_EVENT_VFORK_DONE         = 0x5
PTRACE_GETEVENTMSG              = 0x4201
PTRACE_GETFPREGS                = 0xe
PTRACE_GETFPXREGS               = 0x12
PTRACE_GETREGS                  = 0xc
PTRACE_GETREGSET                = 0x4204
PTRACE_GETSIGINFO               = 0x4202
PTRACE_GET_THREAD_AREA          = 0x19
PTRACE_KILL                     = 0x8
PTRACE_OLDSETOPTIONS            = 0x15
PTRACE_O_MASK                   = 0x7f
PTRACE_O_TRACECLONE             = 0x8
PTRACE_O_TRACEEXEC              = 0x10
PTRACE_O_TRACEEXIT              = 0x40
PTRACE_O_TRACEFORK              = 0x2
PTRACE_O_TRACESYSGOOD           = 0x1
PTRACE_O_TRACEVFORK             = 0x4
PTRACE_O_TRACEVFORKDONE         = 0x20
PTRACE_PEEKDATA                 = 0x2
PTRACE_PEEKTEXT                 = 0x1
PTRACE_PEEKUSR                  = 0x3
PTRACE_POKEDATA                 = 0x5
PTRACE_POKETEXT                 = 0x4
PTRACE_POKEUSR                  = 0x6
PTRACE_SETFPREGS                = 0xf
PTRACE_SETFPXREGS               = 0x13
PTRACE_SETOPTIONS               = 0x4200
PTRACE_SETREGS                  = 0xd
PTRACE_SETREGSET                = 0x4205
PTRACE_SETSIGINFO               = 0x4203
PTRACE_SET_THREAD_AREA          = 0x1a
PTRACE_SINGLEBLOCK              = 0x21
```

```
PTRACE_SINGLESTEP               = 0x9
PTRACE_SYSCALL                  = 0x18
PTRACE_SYSEMU                   = 0x1f
PTRACE_SYSEMU_SINGLESTEP        = 0x20
PTRACE_TRACEME                  = 0x0
RLIMIT_AS                       = 0x9
RLIMIT_CORE                     = 0x4
RLIMIT_CPU                      = 0x0
RLIMIT_DATA                     = 0x2
RLIMIT_FSIZE                    = 0x1
RLIMIT_NOFILE                   = 0x7
RLIMIT_STACK                    = 0x3
RLIM_INFINITY                   = -0x1
RTAX_ADVMSS                     = 0x8
RTAX_CWND                       = 0x7
RTAX_FEATURES                   = 0xc
RTAX_FEATURE_ALLFRAG            = 0x8
RTAX_FEATURE_ECN                = 0x1
RTAX_FEATURE_SACK               = 0x2
RTAX_FEATURE_TIMESTAMP          = 0x4
RTAX_HOPLIMIT                   = 0xa
RTAX_INITCWND                   = 0xb
RTAX_INITRWND                   = 0xe
RTAX_LOCK                       = 0x1
RTAX_MAX                        = 0xe
RTAX_MTU                        = 0x2
RTAX_REORDERING                 = 0x9
RTAX_RTO_MIN                    = 0xd
RTAX_RTT                        = 0x4
RTAX_RTTVAR                     = 0x5
RTAX_SSTHRESH                   = 0x6
RTAX_UNSPEC                     = 0x0
RTAX_WINDOW                     = 0x3
RTA_ALIGNTO                     = 0x4
RTA_MAX                         = 0x10
RTCF_DIRECTSRC                  = 0x4000000
RTCF_DOREDIRECT                 = 0x1000000
RTCF_LOG                        = 0x2000000
RTCF_MASQ                       = 0x400000
RTCF_NAT                        = 0x800000
RTCF_VALVE                      = 0x200000
RTF_ADDRCLASSMASK               = 0xf8000000
RTF_ADDRCONF                    = 0x40000
RTF_ALLONLINK                   = 0x20000
RTF_BROADCAST                   = 0x10000000
RTF_CACHE                       = 0x1000000
RTF_DEFAULT                     = 0x10000
RTF_DYNAMIC                     = 0x10
RTF_FLOW                        = 0x2000000
RTF_GATEWAY                     = 0x2
RTF_HOST                        = 0x4
RTF_INTERFACE                   = 0x40000000
RTF_IRTT                        = 0x100
RTF_LINKRT                      = 0x100000
```

```
RTF_LOCAL                       = 0x80000000
RTF_MODIFIED                    = 0x20
RTF_MSS                         = 0x40
RTF_MTU                         = 0x40
RTF_MULTICAST                   = 0x20000000
RTF_NAT                         = 0x8000000
RTF_NOFORWARD                   = 0x1000
RTF_NONEXTHOP                   = 0x200000
RTF_NOPMTUDISC                  = 0x4000
RTF_POLICY                      = 0x4000000
RTF_REINSTATE                   = 0x8
RTF_REJECT                      = 0x200
RTF_STATIC                      = 0x400
RTF_THROW                       = 0x2000
RTF_UP                          = 0x1
RTF_WINDOW                      = 0x80
RTF_XRESOLVE                    = 0x800
RTM_BASE                        = 0x10
RTM_DELACTION                   = 0x31
RTM_DELADDR                     = 0x15
RTM_DELADDRLABEL                = 0x49
RTM_DELLINK                     = 0x11
RTM_DELNEIGH                    = 0x1d
RTM_DELQDISC                    = 0x25
RTM_DELROUTE                    = 0x19
RTM_DELRULE                     = 0x21
RTM_DELTCLASS                   = 0x29
RTM_DELTFILTER                  = 0x2d
RTM_F_CLONED                    = 0x200
RTM_F_EQUALIZE                  = 0x400
RTM_F_NOTIFY                    = 0x100
RTM_F_PREFIX                    = 0x800
RTM_GETACTION                   = 0x32
RTM_GETADDR                     = 0x16
RTM_GETADDRLABEL                = 0x4a
RTM_GETANYCAST                  = 0x3e
RTM_GETDCB                      = 0x4e
RTM_GETLINK                     = 0x12
RTM_GETMULTICAST                = 0x3a
RTM_GETNEIGH                    = 0x1e
RTM_GETNEIGHTBL                 = 0x42
RTM_GETQDISC                    = 0x26
RTM_GETROUTE                    = 0x1a
RTM_GETRULE                     = 0x22
RTM_GETTCLASS                   = 0x2a
RTM_GETTFILTER                  = 0x2e
RTM_MAX                         = 0x4f
RTM_NEWACTION                   = 0x30
RTM_NEWADDR                     = 0x14
RTM_NEWADDRLABEL                = 0x48
RTM_NEWLINK                     = 0x10
RTM_NEWNDUSEROPT                = 0x44
RTM_NEWNEIGH                    = 0x1c
RTM_NEWNEIGHTBL                 = 0x40
```

```
RTM_NEWPREFIX                   = 0x34
RTM_NEWQDISC                    = 0x24
RTM_NEWROUTE                    = 0x18
RTM_NEWRULE                     = 0x20
RTM_NEWTCLASS                   = 0x28
RTM_NEWTFILTER                  = 0x2c
RTM_NR_FAMILIES                 = 0x10
RTM_NR_MSGTYPES                 = 0x40
RTM_SETDCB                      = 0x4f
RTM_SETLINK                     = 0x13
RTM_SETNEIGHTBL                 = 0x43
RTNH_ALIGNTO                    = 0x4
RTNH_F_DEAD                     = 0x1
RTNH_F_ONLINK                   = 0x4
RTNH_F_PERVASIVE                = 0x2
RTN_MAX                         = 0xb
RTPROT_BIRD                     = 0xc
RTPROT_BOOT                     = 0x3
RTPROT_DHCP                     = 0x10
RTPROT_DNROUTED                 = 0xd
RTPROT_GATED                    = 0x8
RTPROT_KERNEL                   = 0x2
RTPROT_MRT                      = 0xa
RTPROT_NTK                      = 0xf
RTPROT_RA                       = 0x9
RTPROT_REDIRECT                 = 0x1
RTPROT_STATIC                   = 0x4
RTPROT_UNSPEC                   = 0x0
RTPROT_XORP                     = 0xe
RTPROT_ZEBRA                    = 0xb
RT_CLASS_DEFAULT                = 0xfd
RT_CLASS_LOCAL                  = 0xff
RT_CLASS_MAIN                   = 0xfe
RT_CLASS_MAX                    = 0xff
RT_CLASS_UNSPEC                 = 0x0
RUSAGE_CHILDREN                 = -0x1
RUSAGE_SELF                     = 0x0
RUSAGE_THREAD                   = 0x1
SCM_CREDENTIALS                 = 0x2
SCM_RIGHTS                      = 0x1
SCM_TIMESTAMP                   = 0x1d
SCM_TIMESTAMPING                = 0x25
SCM_TIMESTAMPNS                 = 0x23
SHUT_RD                         = 0x0
SHUT_RDWR                       = 0x2
SHUT_WR                         = 0x1
SIOCADDDLCI                     = 0x8980
SIOCADDMULTI                    = 0x8931
SIOCADDRT                       = 0x890b
SIOCATMARK                      = 0x8905
SIOCDARP                        = 0x8953
SIOCDELDLCI                     = 0x8981
SIOCDELMULTI                    = 0x8932
SIOCDELRT                       = 0x890c
```

```
SIOCDEVPRIVATE                  = 0x89f0
SIOCDIFADDR                     = 0x8936
SIOCDRARP                       = 0x8960
SIOCGARP                        = 0x8954
SIOCGIFADDR                     = 0x8915
SIOCGIFBR                       = 0x8940
SIOCGIFBRDADDR                  = 0x8919
SIOCGIFCONF                     = 0x8912
SIOCGIFCOUNT                    = 0x8938
SIOCGIFDSTADDR                  = 0x8917
SIOCGIFENCAP                    = 0x8925
SIOCGIFFLAGS                    = 0x8913
SIOCGIFHWADDR                   = 0x8927
SIOCGIFINDEX                    = 0x8933
SIOCGIFMAP                      = 0x8970
SIOCGIFMEM                      = 0x891f
SIOCGIFMETRIC                   = 0x891d
SIOCGIFMTU                      = 0x8921
SIOCGIFNAME                     = 0x8910
SIOCGIFNETMASK                  = 0x891b
SIOCGIFPFLAGS                   = 0x8935
SIOCGIFSLAVE                    = 0x8929
SIOCGIFTXQLEN                   = 0x8942
SIOCGPGRP                       = 0x8904
SIOCGRARP                       = 0x8961
SIOCGSTAMP                      = 0x8906
SIOCGSTAMPNS                    = 0x8907
SIOCPROTOPRIVATE                = 0x89e0
SIOCRTMSG                       = 0x890d
SIOCSARP                        = 0x8955
SIOCSIFADDR                     = 0x8916
SIOCSIFBR                       = 0x8941
SIOCSIFBRDADDR                  = 0x891a
SIOCSIFDSTADDR                  = 0x8918
SIOCSIFENCAP                    = 0x8926
SIOCSIFFLAGS                    = 0x8914
SIOCSIFHWADDR                   = 0x8924
SIOCSIFHWBROADCAST              = 0x8937
SIOCSIFLINK                     = 0x8911
SIOCSIFMAP                      = 0x8971
SIOCSIFMEM                      = 0x8920
SIOCSIFMETRIC                   = 0x891e
SIOCSIFMTU                      = 0x8922
SIOCSIFNAME                     = 0x8923
SIOCSIFNETMASK                  = 0x891c
SIOCSIFPFLAGS                   = 0x8934
SIOCSIFSLAVE                    = 0x8930
SIOCSIFTXQLEN                   = 0x8943
SIOCSPGRP                       = 0x8902
SIOCSRARP                       = 0x8962
SOCK_CLOEXEC                    = 0x80000
SOCK_DCCP                       = 0x6
SOCK_DGRAM                      = 0x2
SOCK_NONBLOCK                   = 0x800
```

```
SOCK_PACKET                     = 0xa
SOCK_RAW                        = 0x3
SOCK_RDM                        = 0x4
SOCK_SEQPACKET                  = 0x5
SOCK_STREAM                     = 0x1
SOL_AAL                         = 0x109
SOL_ATM                         = 0x108
SOL_DECNET                      = 0x105
SOL_ICMPV6                      = 0x3a
SOL_IP                          = 0x0
SOL_IPV6                        = 0x29
SOL_IRDA                        = 0x10a
SOL_PACKET                      = 0x107
SOL_RAW                         = 0xff
SOL_SOCKET                      = 0x1
SOL_TCP                         = 0x6
SOL_X25                         = 0x106
SOMAXCONN                       = 0x80
SO_ACCEPTCONN                   = 0x1e
SO_ATTACH_FILTER                = 0x1a
SO_BINDTODEVICE                 = 0x19
SO_BROADCAST                    = 0x6
SO_BSDCOMPAT                    = 0xe
SO_DEBUG                        = 0x1
SO_DETACH_FILTER                = 0x1b
SO_DOMAIN                       = 0x27
SO_DONTROUTE                    = 0x5
SO_ERROR                        = 0x4
SO_KEEPALIVE                    = 0x9
SO_LINGER                       = 0xd
SO_MARK                         = 0x24
SO_NO_CHECK                     = 0xb
SO_OOBINLINE                    = 0xa
SO_PASSCRED                     = 0x10
SO_PASSSEC                      = 0x22
SO_PEERCRED                     = 0x11
SO_PEERNAME                     = 0x1c
SO_PEERSEC                      = 0x1f
SO_PRIORITY                     = 0xc
SO_PROTOCOL                     = 0x26
SO_RCVBUF                       = 0x8
SO_RCVBUFFORCE                  = 0x21
SO_RCVLOWAT                     = 0x12
SO_RCVTIMEO                     = 0x14
SO_REUSEADDR                    = 0x2
SO_RXQ_OVFL                     = 0x28
SO_SECURITY_AUTHENTICATION      = 0x16
SO_SECURITY_ENCRYPTION_NETWORK  = 0x18
SO_SECURITY_ENCRYPTION_TRANSPORT = 0x17
SO_SNDBUF                       = 0x7
SO_SNDBUFFORCE                  = 0x20
SO_SNDLOWAT                     = 0x13
SO_SNDTIMEO                     = 0x15
SO_TIMESTAMP                    = 0x1d
```

```
SO_TIMESTAMPING                 = 0x25
SO_TIMESTAMPNS                  = 0x23
SO_TYPE                         = 0x3
S_BLKSIZE                       = 0x200
S_IEXEC                         = 0x40
S_IFBLK                         = 0x6000
S_IFCHR                         = 0x2000
S_IFDIR                         = 0x4000
S_IFIFO                         = 0x1000
S_IFLNK                         = 0xa000
S_IFMT                          = 0xf000
S_IFREG                         = 0x8000
S_IFSOCK                        = 0xc000
S_IREAD                         = 0x100
S_IRGRP                         = 0x20
S_IROTH                         = 0x4
S_IRUSR                         = 0x100
S_IRWXG                         = 0x38
S_IRWXO                         = 0x7
S_IRWXU                         = 0x1c0
S_ISGID                         = 0x400
S_ISUID                         = 0x800
S_ISVTX                         = 0x200
S_IWGRP                         = 0x10
S_IWOTH                         = 0x2
S_IWRITE                        = 0x80
S_IWUSR                         = 0x80
S_IXGRP                         = 0x8
S_IXOTH                         = 0x1
S_IXUSR                         = 0x40
TCIFLUSH                        = 0x0
TCIOFLUSH                       = 0x2
TCOFLUSH                        = 0x1
TCP_CONGESTION                  = 0xd
TCP_CORK                        = 0x3
TCP_DEFER_ACCEPT                = 0x9
TCP_INFO                        = 0xb
TCP_KEEPCNT                     = 0x6
TCP_KEEPIDLE                    = 0x4
TCP_KEEPINTVL                   = 0x5
TCP_LINGER2                     = 0x8
TCP_MAXSEG                      = 0x2
TCP_MAXWIN                      = 0xffff
TCP_MAX_WINSHIFT                = 0xe
TCP_MD5SIG                      = 0xe
TCP_MD5SIG_MAXKEYLEN            = 0x50
TCP_MSS                         = 0x200
TCP_NODELAY                     = 0x1
TCP_QUICKACK                    = 0xc
TCP_SYNCNT                      = 0x7
TCP_WINDOW_CLAMP                = 0xa
TIOCCBRK                        = 0x5428
TIOCCONS                        = 0x541d
TIOCEXCL                        = 0x540c
```

```
TIOCGDEV                        = 0x80045432
TIOCGETD                        = 0x5424
TIOCGICOUNT                     = 0x545d
TIOCGLCKTRMIOS                  = 0x5456
TIOCGPGRP                       = 0x540f
TIOCGPTN                        = 0x80045430
TIOCGRS485                      = 0x542e
TIOCGSERIAL                     = 0x541e
TIOCGSID                        = 0x5429
TIOCGSOFTCAR                    = 0x5419
TIOCGWINSZ                      = 0x5413
TIOCINQ                         = 0x541b
TIOCLINUX                       = 0x541c
TIOCMBIC                        = 0x5417
TIOCMBIS                        = 0x5416
TIOCMGET                        = 0x5415
TIOCMIWAIT                      = 0x545c
TIOCMSET                        = 0x5418
TIOCM_CAR                       = 0x40
TIOCM_CD                        = 0x40
TIOCM_CTS                       = 0x20
TIOCM_DSR                       = 0x100
TIOCM_DTR                       = 0x2
TIOCM_LE                        = 0x1
TIOCM_RI                        = 0x80
TIOCM_RNG                       = 0x80
TIOCM_RTS                       = 0x4
TIOCM_SR                        = 0x10
TIOCM_ST                        = 0x8
TIOCNOTTY                       = 0x5422
TIOCNXCL                        = 0x540d
TIOCOUTQ                        = 0x5411
TIOCPKT                         = 0x5420
TIOCPKT_DATA                    = 0x0
TIOCPKT_DOSTOP                  = 0x20
TIOCPKT_FLUSHREAD               = 0x1
TIOCPKT_FLUSHWRITE              = 0x2
TIOCPKT_IOCTL                   = 0x40
TIOCPKT_NOSTOP                  = 0x10
TIOCPKT_START                   = 0x8
TIOCPKT_STOP                    = 0x4
TIOCSBRK                        = 0x5427
TIOCSCTTY                       = 0x540e
TIOCSERCONFIG                   = 0x5453
TIOCSERGETLSR                   = 0x5459
TIOCSERGETMULTI                 = 0x545a
TIOCSERGSTRUCT                  = 0x5458
TIOCSERGWILD                    = 0x5454
TIOCSERSETMULTI                 = 0x545b
TIOCSERSWILD                    = 0x5455
TIOCSER_TEMT                    = 0x1
TIOCSETD                        = 0x5423
TIOCSIG                         = 0x40045436
TIOCSLCKTRMIOS                  = 0x5457
```

```
    TIOCSPGRP                       = 0x5410
    TIOCSPTLCK                      = 0x40045431
    TIOCSRS485                      = 0x542f
    TIOCSSERIAL                     = 0x541f
    TIOCSSOFTCAR                    = 0x541a
    TIOCSTI                         = 0x5412
    TIOCSWINSZ                      = 0x5414
    TUNATTACHFILTER                 = 0x401054d5
    TUNDETACHFILTER                 = 0x401054d6
    TUNGETFEATURES                  = 0x800454cf
    TUNGETIFF                       = 0x800454d2
    TUNGETSNDBUF                    = 0x800454d3
    TUNGETVNETHDRSZ                 = 0x800454d7
    TUNSETDEBUG                     = 0x400454c9
    TUNSETGROUP                     = 0x400454ce
    TUNSETIFF                       = 0x400454ca
    TUNSETLINK                      = 0x400454cd
    TUNSETNOCSUM                    = 0x400454c8
    TUNSETOFFLOAD                   = 0x400454d0
    TUNSETOWNER                     = 0x400454cc
    TUNSETPERSIST                   = 0x400454cb
    TUNSETSNDBUF                    = 0x400454d4
    TUNSETTXFILTER                  = 0x400454d1
    TUNSETVNETHDRSZ                 = 0x400454d8
    WALL                            = 0x40000000
    WCLONE                          = 0x80000000
    WCONTINUED                      = 0x8
    WEXITED                         = 0x4
    WNOHANG                         = 0x1
    WNOTHREAD                       = 0x20000000
    WNOWAIT                         = 0x1000000
    WORDSIZE                        = 0x40
    WSTOPPED                        = 0x2
    WUNTRACED                       = 0x2
)
```

```
const (
    E2BIG           = Errno(0x7)
    EACCES          = Errno(0xd)
    EADDRINUSE      = Errno(0x62)
    EADDRNOTAVAIL   = Errno(0x63)
    EADV            = Errno(0x44)
    EAFNOSUPPORT    = Errno(0x61)
    EAGAIN          = Errno(0xb)
    EALREADY        = Errno(0x72)
    EBADE           = Errno(0x34)
    EBADF           = Errno(0x9)
    EBADFD          = Errno(0x4d)
    EBADMSG         = Errno(0x4a)
    EBADR           = Errno(0x35)
    EBADRQC         = Errno(0x38)
    EBADSLT         = Errno(0x39)
    EBFONT          = Errno(0x3b)
```

```
    EBUSY           = Errno(0x10)
    ECANCELED       = Errno(0x7d)
    ECHILD          = Errno(0xa)
    ECHRNG          = Errno(0x2c)
    ECOMM           = Errno(0x46)
    ECONNABORTED    = Errno(0x67)
    ECONNREFUSED    = Errno(0x6f)
    ECONNRESET      = Errno(0x68)
    EDEADLK         = Errno(0x23)
    EDEADLOCK       = Errno(0x23)
    EDESTADDRREQ    = Errno(0x59)
    EDOM            = Errno(0x21)
    EDOTDOT         = Errno(0x49)
    EDQUOT          = Errno(0x7a)
    EEXIST          = Errno(0x11)
    EFAULT          = Errno(0xe)
    EFBIG           = Errno(0x1b)
    EHOSTDOWN       = Errno(0x70)
    EHOSTUNREACH    = Errno(0x71)
    EIDRM           = Errno(0x2b)
    EILSEQ          = Errno(0x54)
    EINPROGRESS     = Errno(0x73)
    EINTR           = Errno(0x4)
    EINVAL          = Errno(0x16)
    EIO             = Errno(0x5)
    EISCONN         = Errno(0x6a)
    EISDIR          = Errno(0x15)
    EISNAM          = Errno(0x78)
    EKEYEXPIRED     = Errno(0x7f)
    EKEYREJECTED    = Errno(0x81)
    EKEYREVOKED     = Errno(0x80)
    EL2HLT          = Errno(0x33)
    EL2NSYNC        = Errno(0x2d)
    EL3HLT          = Errno(0x2e)
    EL3RST          = Errno(0x2f)
    ELIBACC         = Errno(0x4f)
    ELIBBAD         = Errno(0x50)
    ELIBEXEC        = Errno(0x53)
    ELIBMAX         = Errno(0x52)
    ELIBSCN         = Errno(0x51)
    ELNRNG          = Errno(0x30)
    ELOOP           = Errno(0x28)
    EMEDIUMTYPE     = Errno(0x7c)
    EMFILE          = Errno(0x18)
    EMLINK          = Errno(0x1f)
    EMSGSIZE        = Errno(0x5a)
    EMULTIHOP       = Errno(0x48)
    ENAMETOOLONG    = Errno(0x24)
    ENAVAIL         = Errno(0x77)
    ENETDOWN        = Errno(0x64)
    ENETRESET       = Errno(0x66)
    ENETUNREACH     = Errno(0x65)
    ENFILE          = Errno(0x17)
    ENOANO          = Errno(0x37)
```

```
    ENOBUFS         = Errno(0x69)
    ENOCSI          = Errno(0x32)
    ENODATA         = Errno(0x3d)
    ENODEV          = Errno(0x13)
    ENOENT          = Errno(0x2)
    ENOEXEC         = Errno(0x8)
    ENOKEY          = Errno(0x7e)
    ENOLCK          = Errno(0x25)
    ENOLINK         = Errno(0x43)
    ENOMEDIUM       = Errno(0x7b)
    ENOMEM          = Errno(0xc)
    ENOMSG          = Errno(0x2a)
    ENONET          = Errno(0x40)
    ENOPKG          = Errno(0x41)
    ENOPROTOOPT     = Errno(0x5c)
    ENOSPC          = Errno(0x1c)
    ENOSR           = Errno(0x3f)
    ENOSTR          = Errno(0x3c)
    ENOSYS          = Errno(0x26)
    ENOTBLK         = Errno(0xf)
    ENOTCONN        = Errno(0x6b)
    ENOTDIR         = Errno(0x14)
    ENOTEMPTY       = Errno(0x27)
    ENOTNAM         = Errno(0x76)
    ENOTRECOVERABLE = Errno(0x83)
    ENOTSOCK        = Errno(0x58)
    ENOTSUP         = Errno(0x5f)
    ENOTTY          = Errno(0x19)
    ENOTUNIQ        = Errno(0x4c)
    ENXIO           = Errno(0x6)
    EOPNOTSUPP      = Errno(0x5f)
    EOVERFLOW       = Errno(0x4b)
    EOWNERDEAD      = Errno(0x82)
    EPERM           = Errno(0x1)
    EPFNOSUPPORT    = Errno(0x60)
    EPIPE           = Errno(0x20)
    EPROTO          = Errno(0x47)
    EPROTONOSUPPORT = Errno(0x5d)
    EPROTOTYPE      = Errno(0x5b)
    ERANGE          = Errno(0x22)
    EREMCHG         = Errno(0x4e)
    EREMOTE         = Errno(0x42)
    EREMOTEIO       = Errno(0x79)
    ERESTART        = Errno(0x55)
    ERFKILL         = Errno(0x84)
    EROFS           = Errno(0x1e)
    ESHUTDOWN       = Errno(0x6c)
    ESOCKTNOSUPPORT = Errno(0x5e)
    ESPIPE          = Errno(0x1d)
    ESRCH           = Errno(0x3)
    ESRMNT          = Errno(0x45)
    ESTALE          = Errno(0x74)
    ESTRPIPE        = Errno(0x56)
    ETIME           = Errno(0x3e)
```

```
    ETIMEDOUT       = Errno(0x6e)
    ETOOMANYREFS    = Errno(0x6d)
    ETXTBSY         = Errno(0x1a)
    EUCLEAN         = Errno(0x75)
    EUNATCH         = Errno(0x31)
    EUSERS          = Errno(0x57)
    EWOULDBLOCK     = Errno(0xb)
    EXDEV           = Errno(0x12)
    EXFULL          = Errno(0x36)
)
```

Errors

```
const (
    SIGABRT   = Signal(0x6)
    SIGALRM   = Signal(0xe)
    SIGBUS    = Signal(0x7)
    SIGCHLD   = Signal(0x11)
    SIGCLD    = Signal(0x11)
    SIGCONT   = Signal(0x12)
    SIGFPE    = Signal(0x8)
    SIGHUP    = Signal(0x1)
    SIGILL    = Signal(0x4)
    SIGINT    = Signal(0x2)
    SIGIO     = Signal(0x1d)
    SIGIOT    = Signal(0x6)
    SIGKILL   = Signal(0x9)
    SIGPIPE   = Signal(0xd)
    SIGPOLL   = Signal(0x1d)
    SIGPROF   = Signal(0x1b)
    SIGPWR    = Signal(0x1e)
    SIGQUIT   = Signal(0x3)
    SIGSEGV   = Signal(0xb)
    SIGSTKFLT = Signal(0x10)
    SIGSTOP   = Signal(0x13)
    SIGSYS    = Signal(0x1f)
    SIGTERM   = Signal(0xf)
    SIGTRAP   = Signal(0x5)
    SIGTSTP   = Signal(0x14)
    SIGTTIN   = Signal(0x15)
    SIGTTOU   = Signal(0x16)
    SIGUNUSED = Signal(0x1f)
    SIGURG    = Signal(0x17)
    SIGUSR1   = Signal(0xa)
    SIGUSR2   = Signal(0xc)
    SIGVTALRM = Signal(0x1a)
    SIGWINCH  = Signal(0x1c)
    SIGXCPU   = Signal(0x18)
    SIGXFSZ   = Signal(0x19)
)
```

Signals

```
const (
    SYS_READ                = 0
    SYS_WRITE               = 1
    SYS_OPEN                = 2
    SYS_CLOSE               = 3
    SYS_STAT                = 4
    SYS_FSTAT               = 5
    SYS_LSTAT               = 6
    SYS_POLL                = 7
    SYS_LSEEK               = 8
    SYS_MMAP                = 9
    SYS_MPROTECT            = 10
    SYS_MUNMAP              = 11
    SYS_BRK                 = 12
    SYS_RT_SIGACTION        = 13
    SYS_RT_SIGPROCMASK      = 14
    SYS_RT_SIGRETURN        = 15
    SYS_IOCTL               = 16
    SYS_PREAD64             = 17
    SYS_PWRITE64            = 18
    SYS_READV               = 19
    SYS_WRITEV              = 20
    SYS_ACCESS              = 21
    SYS_PIPE                = 22
    SYS_SELECT              = 23
    SYS_SCHED_YIELD         = 24
    SYS_MREMAP              = 25
    SYS_MSYNC               = 26
    SYS_MINCORE             = 27
    SYS_MADVISE             = 28
    SYS_SHMGET              = 29
    SYS_SHMAT               = 30
    SYS_SHMCTL              = 31
    SYS_DUP                 = 32
    SYS_DUP2                = 33
    SYS_PAUSE               = 34
    SYS_NANOSLEEP           = 35
    SYS_GETITIMER           = 36
    SYS_ALARM               = 37
    SYS_SETITIMER           = 38
    SYS_GETPID              = 39
    SYS_SENDFILE            = 40
    SYS_SOCKET              = 41
    SYS_CONNECT             = 42
    SYS_ACCEPT              = 43
    SYS_SENDTO              = 44
    SYS_RECVFROM            = 45
    SYS_SENDMSG             = 46
    SYS_RECVMSG             = 47
    SYS_SHUTDOWN            = 48
    SYS_BIND                = 49
    SYS_LISTEN              = 50
    SYS_GETSOCKNAME         = 51
```

```
SYS_GETPEERNAME         = 52
SYS_SOCKETPAIR          = 53
SYS_SETSOCKOPT          = 54
SYS_GETSOCKOPT          = 55
SYS_CLONE               = 56
SYS_FORK                = 57
SYS_VFORK               = 58
SYS_EXECVE              = 59
SYS_EXIT                = 60
SYS_WAIT4               = 61
SYS_KILL                = 62
SYS_UNAME               = 63
SYS_SEMGET              = 64
SYS_SEMOP               = 65
SYS_SEMCTL              = 66
SYS_SHMDT               = 67
SYS_MSGGET              = 68
SYS_MSGSND              = 69
SYS_MSGRCV              = 70
SYS_MSGCTL              = 71
SYS_FCNTL               = 72
SYS_FLOCK               = 73
SYS_FSYNC               = 74
SYS_FDATASYNC           = 75
SYS_TRUNCATE            = 76
SYS_FTRUNCATE           = 77
SYS_GETDENTS            = 78
SYS_GETCWD              = 79
SYS_CHDIR               = 80
SYS_FCHDIR              = 81
SYS_RENAME              = 82
SYS_MKDIR               = 83
SYS_RMDIR               = 84
SYS_CREAT               = 85
SYS_LINK                = 86
SYS_UNLINK              = 87
SYS_SYMLINK             = 88
SYS_READLINK            = 89
SYS_CHMOD               = 90
SYS_FCHMOD              = 91
SYS_CHOWN               = 92
SYS_FCHOWN              = 93
SYS_LCHOWN              = 94
SYS_UMASK               = 95
SYS_GETTIMEOFDAY        = 96
SYS_GETRLIMIT           = 97
SYS_GETRUSAGE           = 98
SYS_SYSINFO             = 99
SYS_TIMES               = 100
SYS_PTRACE              = 101
SYS_GETUID              = 102
SYS_SYSLOG              = 103
SYS_GETGID              = 104
SYS_SETUID              = 105
```

```
SYS_SETGID                = 106
SYS_GETEUID               = 107
SYS_GETEGID               = 108
SYS_SETPGID               = 109
SYS_GETPPID               = 110
SYS_GETPGRP               = 111
SYS_SETSID                = 112
SYS_SETREUID              = 113
SYS_SETREGID              = 114
SYS_GETGROUPS             = 115
SYS_SETGROUPS             = 116
SYS_SETRESUID             = 117
SYS_GETRESUID             = 118
SYS_SETRESGID             = 119
SYS_GETRESGID             = 120
SYS_GETPGID               = 121
SYS_SETFSUID              = 122
SYS_SETFSGID              = 123
SYS_GETSID                = 124
SYS_CAPGET                = 125
SYS_CAPSET                = 126
SYS_RT_SIGPENDING         = 127
SYS_RT_SIGTIMEDWAIT       = 128
SYS_RT_SIGQUEUEINFO       = 129
SYS_RT_SIGSUSPEND         = 130
SYS_SIGALTSTACK           = 131
SYS_UTIME                 = 132
SYS_MKNOD                 = 133
SYS_USELIB                = 134
SYS_PERSONALITY           = 135
SYS_USTAT                 = 136
SYS_STATFS                = 137
SYS_FSTATFS               = 138
SYS_SYSFS                 = 139
SYS_GETPRIORITY           = 140
SYS_SETPRIORITY           = 141
SYS_SCHED_SETPARAM        = 142
SYS_SCHED_GETPARAM        = 143
SYS_SCHED_SETSCHEDULER    = 144
SYS_SCHED_GETSCHEDULER    = 145
SYS_SCHED_GET_PRIORITY_MAX = 146
SYS_SCHED_GET_PRIORITY_MIN = 147
SYS_SCHED_RR_GET_INTERVAL  = 148
SYS_MLOCK                 = 149
SYS_MUNLOCK               = 150
SYS_MLOCKALL              = 151
SYS_MUNLOCKALL            = 152
SYS_VHANGUP               = 153
SYS_MODIFY_LDT            = 154
SYS_PIVOT_ROOT            = 155
SYS__SYSCTL               = 156
SYS_PRCTL                 = 157
SYS_ARCH_PRCTL            = 158
SYS_ADJTIMEX              = 159
```

```
SYS_SETRLIMIT              = 160
SYS_CHROOT                 = 161
SYS_SYNC                   = 162
SYS_ACCT                   = 163
SYS_SETTIMEOFDAY           = 164
SYS_MOUNT                  = 165
SYS_UMOUNT2                = 166
SYS_SWAPON                 = 167
SYS_SWAPOFF                = 168
SYS_REBOOT                 = 169
SYS_SETHOSTNAME            = 170
SYS_SETDOMAINNAME          = 171
SYS_IOPL                   = 172
SYS_IOPERM                 = 173
SYS_CREATE_MODULE          = 174
SYS_INIT_MODULE            = 175
SYS_DELETE_MODULE          = 176
SYS_GET_KERNEL_SYMS        = 177
SYS_QUERY_MODULE           = 178
SYS_QUOTACTL               = 179
SYS_NFSSERVCTL             = 180
SYS_GETPMSG                = 181
SYS_PUTPMSG                = 182
SYS_AFS_SYSCALL            = 183
SYS_TUXCALL                = 184
SYS_SECURITY               = 185
SYS_GETTID                 = 186
SYS_READAHEAD              = 187
SYS_SETXATTR               = 188
SYS_LSETXATTR              = 189
SYS_FSETXATTR              = 190
SYS_GETXATTR               = 191
SYS_LGETXATTR              = 192
SYS_FGETXATTR              = 193
SYS_LISTXATTR              = 194
SYS_LLISTXATTR             = 195
SYS_FLISTXATTR             = 196
SYS_REMOVEXATTR            = 197
SYS_LREMOVEXATTR           = 198
SYS_FREMOVEXATTR           = 199
SYS_TKILL                  = 200
SYS_TIME                   = 201
SYS_FUTEX                  = 202
SYS_SCHED_SETAFFINITY      = 203
SYS_SCHED_GETAFFINITY      = 204
SYS_SET_THREAD_AREA        = 205
SYS_IO_SETUP               = 206
SYS_IO_DESTROY             = 207
SYS_IO_GETEVENTS           = 208
SYS_IO_SUBMIT              = 209
SYS_IO_CANCEL              = 210
SYS_GET_THREAD_AREA        = 211
SYS_LOOKUP_DCOOKIE         = 212
SYS_EPOLL_CREATE           = 213
```

```
SYS_EPOLL_CTL_OLD          = 214
SYS_EPOLL_WAIT_OLD         = 215
SYS_REMAP_FILE_PAGES       = 216
SYS_GETDENTS64             = 217
SYS_SET_TID_ADDRESS        = 218
SYS_RESTART_SYSCALL        = 219
SYS_SEMTIMEDOP             = 220
SYS_FADVISE64              = 221
SYS_TIMER_CREATE           = 222
SYS_TIMER_SETTIME          = 223
SYS_TIMER_GETTIME          = 224
SYS_TIMER_GETOVERRUN       = 225
SYS_TIMER_DELETE           = 226
SYS_CLOCK_SETTIME          = 227
SYS_CLOCK_GETTIME          = 228
SYS_CLOCK_GETRES           = 229
SYS_CLOCK_NANOSLEEP        = 230
SYS_EXIT_GROUP             = 231
SYS_EPOLL_WAIT             = 232
SYS_EPOLL_CTL              = 233
SYS_TGKILL                 = 234
SYS_UTIMES                 = 235
SYS_VSERVER                = 236
SYS_MBIND                  = 237
SYS_SET_MEMPOLICY          = 238
SYS_GET_MEMPOLICY          = 239
SYS_MQ_OPEN                = 240
SYS_MQ_UNLINK              = 241
SYS_MQ_TIMEDSEND           = 242
SYS_MQ_TIMEDRECEIVE        = 243
SYS_MQ_NOTIFY              = 244
SYS_MQ_GETSETATTR          = 245
SYS_KEXEC_LOAD             = 246
SYS_WAITID                 = 247
SYS_ADD_KEY                = 248
SYS_REQUEST_KEY            = 249
SYS_KEYCTL                 = 250
SYS_IOPRIO_SET             = 251
SYS_IOPRIO_GET             = 252
SYS_INOTIFY_INIT           = 253
SYS_INOTIFY_ADD_WATCH      = 254
SYS_INOTIFY_RM_WATCH       = 255
SYS_MIGRATE_PAGES          = 256
SYS_OPENAT                 = 257
SYS_MKDIRAT                = 258
SYS_MKNODAT                = 259
SYS_FCHOWNAT               = 260
SYS_FUTIMESAT              = 261
SYS_NEWFSTATAT             = 262
SYS_UNLINKAT               = 263
SYS_RENAMEAT               = 264
SYS_LINKAT                 = 265
SYS_SYMLINKAT              = 266
SYS_READLINKAT             = 267
```

```
    SYS_FCHMODAT               = 268
    SYS_FACCESSAT              = 269
    SYS_PSELECT6               = 270
    SYS_PPOLL                  = 271
    SYS_UNSHARE                = 272
    SYS_SET_ROBUST_LIST        = 273
    SYS_GET_ROBUST_LIST        = 274
    SYS_SPLICE                 = 275
    SYS_TEE                    = 276
    SYS_SYNC_FILE_RANGE        = 277
    SYS_VMSPLICE               = 278
    SYS_MOVE_PAGES             = 279
    SYS_UTIMENSAT              = 280
    SYS_EPOLL_PWAIT            = 281
    SYS_SIGNALFD               = 282
    SYS_TIMERFD_CREATE         = 283
    SYS_EVENTFD                = 284
    SYS_FALLOCATE              = 285
    SYS_TIMERFD_SETTIME        = 286
    SYS_TIMERFD_GETTIME        = 287
    SYS_ACCEPT4                = 288
    SYS_SIGNALFD4              = 289
    SYS_EVENTFD2               = 290
    SYS_EPOLL_CREATE1          = 291
    SYS_DUP3                   = 292
    SYS_PIPE2                  = 293
    SYS_INOTIFY_INIT1          = 294
    SYS_PREADV                 = 295
    SYS_PWRITEV                = 296
    SYS_RT_TGSIGQUEUEINFO      = 297
    SYS_PERF_EVENT_OPEN        = 298
    SYS_RECVMMSG               = 299
    SYS_FANOTIFY_INIT          = 300
    SYS_FANOTIFY_MARK          = 301
    SYS_PRLIMIT64              = 302
)
```

```
const (
    SizeofSockaddrInet4     = 0x10
    SizeofSockaddrInet6     = 0x1c
    SizeofSockaddrAny       = 0x70
    SizeofSockaddrUnix      = 0x6e
    SizeofSockaddrLinklayer = 0x14
    SizeofSockaddrNetlink   = 0xc
    SizeofLinger            = 0x8
    SizeofIPMreq            = 0x8
    SizeofIPMreqn           = 0xc
    SizeofIPv6Mreq          = 0x14
    SizeofMsghdr            = 0x38
    SizeofCmsghdr           = 0x10
    SizeofInet4Pktinfo      = 0xc
    SizeofInet6Pktinfo      = 0x14
    SizeofIPv6MTUInfo       = 0x20
```

```
    SizeofICMPv6Filter      = 0x20
    SizeofUcred             = 0xc
    SizeofTCPInfo           = 0x68
)
```

```
const (
    IFA_UNSPEC          = 0x0
    IFA_ADDRESS         = 0x1
    IFA_LOCAL           = 0x2
    IFA_LABEL           = 0x3
    IFA_BROADCAST       = 0x4
    IFA_ANYCAST         = 0x5
    IFA_CACHEINFO       = 0x6
    IFA_MULTICAST       = 0x7
    IFLA_UNSPEC         = 0x0
    IFLA_ADDRESS        = 0x1
    IFLA_BROADCAST      = 0x2
    IFLA_IFNAME         = 0x3
    IFLA_MTU            = 0x4
    IFLA_LINK           = 0x5
    IFLA_QDISC          = 0x6
    IFLA_STATS          = 0x7
    IFLA_COST           = 0x8
    IFLA_PRIORITY       = 0x9
    IFLA_MASTER         = 0xa
    IFLA_WIRELESS       = 0xb
    IFLA_PROTINFO       = 0xc
    IFLA_TXQLEN         = 0xd
    IFLA_MAP            = 0xe
    IFLA_WEIGHT         = 0xf
    IFLA_OPERSTATE      = 0x10
    IFLA_LINKMODE       = 0x11
    IFLA_LINKINFO       = 0x12
    IFLA_NET_NS_PID     = 0x13
    IFLA_IFALIAS        = 0x14
    IFLA_MAX            = 0x1d
    RT_SCOPE_UNIVERSE   = 0x0
    RT_SCOPE_SITE       = 0xc8
    RT_SCOPE_LINK       = 0xfd
    RT_SCOPE_HOST       = 0xfe
    RT_SCOPE_NOWHERE    = 0xff
    RT_TABLE_UNSPEC     = 0x0
    RT_TABLE_COMPAT     = 0xfc
    RT_TABLE_DEFAULT    = 0xfd
    RT_TABLE_MAIN       = 0xfe
    RT_TABLE_LOCAL      = 0xff
    RT_TABLE_MAX        = 0xffffffff
    RTA_UNSPEC          = 0x0
    RTA_DST             = 0x1
    RTA_SRC             = 0x2
    RTA_IIF             = 0x3
    RTA_OIF             = 0x4
    RTA_GATEWAY         = 0x5
```

```
    RTA_PRIORITY        = 0x6
    RTA_PREFSRC         = 0x7
    RTA_METRICS         = 0x8
    RTA_MULTIPATH       = 0x9
    RTA_FLOW            = 0xb
    RTA_CACHEINFO       = 0xc
    RTA_TABLE           = 0xf
    RTN_UNSPEC          = 0x0
    RTN_UNICAST         = 0x1
    RTN_LOCAL           = 0x2
    RTN_BROADCAST       = 0x3
    RTN_ANYCAST         = 0x4
    RTN_MULTICAST       = 0x5
    RTN_BLACKHOLE       = 0x6
    RTN_UNREACHABLE     = 0x7
    RTN_PROHIBIT        = 0x8
    RTN_THROW           = 0x9
    RTN_NAT             = 0xa
    RTN_XRESOLVE        = 0xb
    RTNLGRP_NONE        = 0x0
    RTNLGRP_LINK        = 0x1
    RTNLGRP_NOTIFY      = 0x2
    RTNLGRP_NEIGH       = 0x3
    RTNLGRP_TC          = 0x4
    RTNLGRP_IPV4_IFADDR = 0x5
    RTNLGRP_IPV4_MROUTE = 0x6
    RTNLGRP_IPV4_ROUTE  = 0x7
    RTNLGRP_IPV4_RULE   = 0x8
    RTNLGRP_IPV6_IFADDR = 0x9
    RTNLGRP_IPV6_MROUTE = 0xa
    RTNLGRP_IPV6_ROUTE  = 0xb
    RTNLGRP_IPV6_IFINFO = 0xc
    RTNLGRP_IPV6_PREFIX = 0x12
    RTNLGRP_IPV6_RULE   = 0x13
    RTNLGRP_ND_USEROPT  = 0x14
    SizeofNlMsghdr      = 0x10
    SizeofNlMsgerr      = 0x14
    SizeofRtGenmsg      = 0x1
    SizeofNlAttr        = 0x4
    SizeofRtAttr        = 0x4
    SizeofIfInfomsg     = 0x10
    SizeofIfAddrmsg     = 0x8
    SizeofRtMsg         = 0xc
    SizeofRtNexthop     = 0x8
)
```

```
const (
    SizeofSockFilter = 0x8
    SizeofSockFprog  = 0x10
)
```

```
const (
    VINTR    = 0x0
```

```
VQUIT    = 0x1
VERASE   = 0x2
VKILL    = 0x3
VEOF     = 0x4
VTIME    = 0x5
VMIN     = 0x6
VSWTC    = 0x7
VSTART   = 0x8
VSTOP    = 0x9
VSUSP    = 0xa
VEOL     = 0xb
VREPRINT = 0xc
VDISCARD = 0xd
VWERASE  = 0xe
VLNEXT   = 0xf
VEOL2    = 0x10
IGNBRK   = 0x1
BRKINT   = 0x2
IGNPAR   = 0x4
PARMRK   = 0x8
INPCK    = 0x10
ISTRIP   = 0x20
INLCR    = 0x40
IGNCR    = 0x80
ICRNL    = 0x100
IUCLC    = 0x200
IXON     = 0x400
IXANY    = 0x800
IXOFF    = 0x1000
IMAXBEL  = 0x2000
IUTF8    = 0x4000
OPOST    = 0x1
OLCUC    = 0x2
ONLCR    = 0x4
OCRNL    = 0x8
ONOCR    = 0x10
ONLRET   = 0x20
OFILL    = 0x40
OFDEL    = 0x80
B0       = 0x0
B50      = 0x1
B75      = 0x2
B110     = 0x3
B134     = 0x4
B150     = 0x5
B200     = 0x6
B300     = 0x7
B600     = 0x8
B1200    = 0x9
B1800    = 0xa
B2400    = 0xb
B4800    = 0xc
B9600    = 0xd
B19200   = 0xe
```

```
    B38400   = 0xf
    CSIZE    = 0x30
    CS5      = 0x0
    CS6      = 0x10
    CS7      = 0x20
    CS8      = 0x30
    CSTOPB   = 0x40
    CREAD    = 0x80
    PARENB   = 0x100
    PARODD   = 0x200
    HUPCL    = 0x400
    CLOCAL   = 0x800
    B57600   = 0x1001
    B115200  = 0x1002
    B230400  = 0x1003
    B460800  = 0x1004
    B500000  = 0x1005
    B576000  = 0x1006
    B921600  = 0x1007
    B1000000 = 0x1008
    B1152000 = 0x1009
    B1500000 = 0x100a
    B2000000 = 0x100b
    B2500000 = 0x100c
    B3000000 = 0x100d
    B3500000 = 0x100e
    B4000000 = 0x100f
    ISIG     = 0x1
    ICANON   = 0x2
    XCASE    = 0x4
    ECHO     = 0x8
    ECHOE    = 0x10
    ECHOK    = 0x20
    ECHONL   = 0x40
    NOFLSH   = 0x80
    TOSTOP   = 0x100
    ECHOCTL  = 0x200
    ECHOPRT  = 0x400
    ECHOKE   = 0x800
    FLUSHO   = 0x1000
    PENDIN   = 0x4000
    IEXTEN   = 0x8000
    TCGETS   = 0x5401
    TCSETS   = 0x5402
)
```

```
const ImplementsGetwd = true
```

```
const (
    PathMax = 0x1000
)
```

```
const SizeofInotifyEvent = 0x10
```

Variables

```
var (
    Stdin  = 0
    Stdout = 1
    Stderr = 2
)
```

```
var ForkLock sync.RWMutex
```

```
var SocketDisableIPv6 bool
```

For testing: clients can set this flag to force creation of IPv6 sockets to return EAFNOSUPPORT.

func Access

```
func Access(path string, mode uint32) (err error)
```

func Acct

```
func Acct(path string) (err error)
```

func Adjtimex

```
func Adjtimex(buf *Timex) (state int, err error)
```

func AttachLsf

```
func AttachLsf(fd int, i []SockFilter) error
```

Deprecated: Use golang.org/x/net/bpf instead.

func Bind

```
func Bind(fd int, sa Sockaddr) (err error)
```

func BindToDevice

```
func BindToDevice(fd int, device string) (err error)
```

BindToDevice binds the socket associated with fd to device.

func BytePtrFromString

```
func BytePtrFromString(s string) (*byte, error)
```

BytePtrFromString returns a pointer to a NUL-terminated array of bytes containing the text of s. If s contains a NUL byte at any location, it returns (nil, EINVAL).

func ByteSliceFromString

```
func ByteSliceFromString(s string) ([]byte, error)
```

ByteSliceFromString returns a NUL-terminated slice of bytes containing the text of s. If s contains a NUL byte at any location, it returns (nil, EINVAL).

func Chdir

```
func Chdir(path string) (err error)
```

func Chmod

```
func Chmod(path string, mode uint32) (err error)
```

func Chown

```
func Chown(path string, uid int, gid int) (err error)
```

func Chroot

```
func Chroot(path string) (err error)
```

func Clearenv

```
func Clearenv()
```

func Close

```
func Close(fd int) (err error)
```

func CloseOnExec

```
func CloseOnExec(fd int)
```

func CmsgLen

```
func CmsgLen(datalen int) int
```

CmsgLen returns the value to store in the Len field of the Cmsghdr structure, taking into account any necessary alignment.

func CmsgSpace

```
func CmsgSpace(datalen int) int
```

CmsgSpace returns the number of bytes an ancillary element with payload of the passed data length occupies.

func Connect

```
func Connect(fd int, sa Sockaddr) (err error)
```

func Creat

```
func Creat(path string, mode uint32) (fd int, err error)
```

func DetachLsf

```
func DetachLsf(fd int) error
```

Deprecated: Use golang.org/x/net/bpf instead.

func Dup

```
func Dup(oldfd int) (fd int, err error)
```

func Dup2

```
func Dup2(oldfd int, newfd int) (err error)
```

func Dup3

```
func Dup3(oldfd int, newfd int, flags int) (err error)
```

func Environ

```
func Environ() []string
```

func EpollCreate

```
func EpollCreate(size int) (fd int, err error)
```

func EpollCreate1

```
func EpollCreate1(flag int) (fd int, err error)
```

func EpollCtl

```
func EpollCtl(epfd int, op int, fd int, event *EpollEvent) (err error)
```

func EpollWait

```
func EpollWait(epfd int, events []EpollEvent, msec int) (n int, err error)
```

func Exec

```
func Exec(argv0 string, argv []string, envv []string) (err error)
```

Exec invokes the execve(2) system call.

func Exit

```
func Exit(code int)
```

func Faccessat

```
func Faccessat(dirfd int, path string, mode uint32, flags int) (err error)
```

func Fallocate

```
func Fallocate(fd int, mode uint32, off int64, len int64) (err error)
```

func Fchdir

```
func Fchdir(fd int) (err error)
```

func Fchmod

```
func Fchmod(fd int, mode uint32) (err error)
```

func Fchmodat

```
func Fchmodat(dirfd int, path string, mode uint32, flags int) (err error)
```

func Fchown

```
func Fchown(fd int, uid int, gid int) (err error)
```

func Fchownat

```
func Fchownat(dirfd int, path string, uid int, gid int, flags int) (err error)
```

func FcntlFlock

```
func FcntlFlock(fd uintptr, cmd int, lk *Flock_t) error
```

FcntlFlock performs a fcntl syscall for the F_GETLK, F_SETLK or F_SETLKW command.

func Fdatasync

```
func Fdatasync(fd int) (err error)
```

func Flock

```
func Flock(fd int, how int) (err error)
```

func ForkExec

```
func ForkExec(argv0 string, argv []string, attr *ProcAttr) (pid int, err error)
```

Combination of fork and exec, careful to be thread safe.

func Fstat

```
func Fstat(fd int, stat *Stat_t) (err error)
```

func Fstatfs

```
func Fstatfs(fd int, buf *Statfs_t) (err error)
```

func Fsync

```
func Fsync(fd int) (err error)
```

func Ftruncate

```
func Ftruncate(fd int, length int64) (err error)
```

func Futimes

```
func Futimes(fd int, tv []Timeval) (err error)
```

func Futimesat

```
func Futimesat(dirfd int, path string, tv []Timeval) (err error)
```

func Getcwd

```
func Getcwd(buf []byte) (n int, err error)
```

func Getdents

```
func Getdents(fd int, buf []byte) (n int, err error)
```

func Getegid

```
func Getegid() (egid int)
```

func Getenv

```
func Getenv(key string) (value string, found bool)
```

func Geteuid

```
func Geteuid() (euid int)
```

func Getgid

```
func Getgid() (gid int)
```

func Getgroups

```
func Getgroups() (gids []int, err error)
```

func Getpagesize

```
func Getpagesize() int
```

func Getpgid

```
func Getpgid(pid int) (pgid int, err error)
```

func Getpgrp

```
func Getpgrp() (pid int)
```

func Getpid

```
func Getpid() (pid int)
```

func Getppid

```
func Getppid() (ppid int)
```

func Getpriority

```
func Getpriority(which int, who int) (prio int, err error)
```

func Getrlimit

```
func Getrlimit(resource int, rlim *Rlimit) (err error)
```

func Getrusage

```
func Getrusage(who int, rusage *Rusage) (err error)
```

func GetsockoptInet4Addr

```
func GetsockoptInet4Addr(fd, level, opt int) (value [4]byte, err error)
```

func GetsockoptInt

```
func GetsockoptInt(fd, level, opt int) (value int, err error)
```

func Gettid

```
func Gettid() (tid int)
```

func Gettimeofday

```
func Gettimeofday(tv *Timeval) (err error)
```

func Getuid

```
func Getuid() (uid int)
```

func Getwd

```
func Getwd() (wd string, err error)
```

func Getxattr

```
func Getxattr(path string, attr string, dest []byte) (sz int, err error)
```

func InotifyAddWatch

```
func InotifyAddWatch(fd int, pathname string, mask uint32) (watchdesc int, err error)
```

func InotifyInit

```
func InotifyInit() (fd int, err error)
```

func InotifyInit1

```
func InotifyInit1(flags int) (fd int, err error)
```

func InotifyRmWatch

```
func InotifyRmWatch(fd int, watchdesc uint32) (success int, err error)
```

func Ioperm

```
func Ioperm(from int, num int, on int) (err error)
```

func Iopl

```
func Iopl(level int) (err error)
```

func Kill

```
func Kill(pid int, sig Signal) (err error)
```

func Klogctl

```
func Klogctl(typ int, buf []byte) (n int, err error)
```

func Lchown

```
func Lchown(path string, uid int, gid int) (err error)
```

func Link

```
func Link(oldpath string, newpath string) (err error)
```

func Listen

```
func Listen(s int, n int) (err error)
```

func Listxattr

```
func Listxattr(path string, dest []byte) (sz int, err error)
```

func LsfSocket

```
func LsfSocket(ifindex, proto int) (int, error)
```

Deprecated: Use golang.org/x/net/bpf instead.

func Lstat

```
func Lstat(path string, stat *Stat_t) (err error)
```

func Madvise

```
func Madvise(b []byte, advice int) (err error)
```

func Mkdir

```
func Mkdir(path string, mode uint32) (err error)
```

func Mkdirat

```
func Mkdirat(dirfd int, path string, mode uint32) (err error)
```

func Mkfifo

```
func Mkfifo(path string, mode uint32) (err error)
```

func Mknod

```
func Mknod(path string, mode uint32, dev int) (err error)
```

func Mknodat

```
func Mknodat(dirfd int, path string, mode uint32, dev int) (err error)
```

func Mlock

```
func Mlock(b []byte) (err error)
```

func Mlockall

```
func Mlockall(flags int) (err error)
```

func Mmap

```
func Mmap(fd int, offset int64, length int, prot int, flags int) (data []byte, err er
```

func Mount

```
func Mount(source string, target string, fstype string, flags uintptr, data string) (
```

func Mprotect

```
func Mprotect(b []byte, prot int) (err error)
```

func Munlock

```
func Munlock(b []byte) (err error)
```

func Munlockall

```
func Munlockall() (err error)
```

func Munmap

```
func Munmap(b []byte) (err error)
```

func Nanosleep

```
func Nanosleep(time *Timespec, leftover *Timespec) (err error)
```

func NetlinkRIB

```
func NetlinkRIB(proto, family int) ([]byte, error)
```

NetlinkRIB returns routing information base, as known as RIB, which consists of network facility information, states and parameters.

func Open

```
func Open(path string, mode int, perm uint32) (fd int, err error)
```

func Openat

```
func Openat(dirfd int, path string, flags int, mode uint32) (fd int, err error)
```

func ParseDirent

```
func ParseDirent(buf []byte, max int, names []string) (consumed int, count int, newna
```

ParseDirent parses up to max directory entries in buf, appending the names to names. It returns the number of bytes consumed from buf, the number of entries added to names, and the new names slice.

func ParseUnixRights

```
func ParseUnixRights(m *SocketControlMessage) ([]int, error)
```

ParseUnixRights decodes a socket control message that contains an integer array of open file descriptors from another process.

func Pause

```
func Pause() (err error)
```

func Pipe

```
func Pipe(p []int) (err error)
```

func Pipe2

```
func Pipe2(p []int, flags int) (err error)
```

func PivotRoot

```
func PivotRoot(newroot string, putold string) (err error)
```

func Pread

```
func Pread(fd int, p []byte, offset int64) (n int, err error)
```

func PtraceAttach

```
func PtraceAttach(pid int) (err error)
```

func PtraceCont

```
func PtraceCont(pid int, signal int) (err error)
```

func PtraceDetach

```
func PtraceDetach(pid int) (err error)
```

func PtraceGetEventMsg

```
func PtraceGetEventMsg(pid int) (msg uint, err error)
```

func PtraceGetRegs

```
func PtraceGetRegs(pid int, regsout *PtraceRegs) (err error)
```

func PtracePeekData

```
func PtracePeekData(pid int, addr uintptr, out []byte) (count int, err error)
```

func PtracePeekText

```
func PtracePeekText(pid int, addr uintptr, out []byte) (count int, err error)
```

func PtracePokeData

```
func PtracePokeData(pid int, addr uintptr, data []byte) (count int, err error)
```

func PtracePokeText

```
func PtracePokeText(pid int, addr uintptr, data []byte) (count int, err error)
```

func PtraceSetOptions

```
func PtraceSetOptions(pid int, options int) (err error)
```

func PtraceSetRegs

```
func PtraceSetRegs(pid int, regs *PtraceRegs) (err error)
```

func PtraceSingleStep

```
func PtraceSingleStep(pid int) (err error)
```

func PtraceSyscall

```
func PtraceSyscall(pid int, signal int) (err error)
```

func Pwrite

```
func Pwrite(fd int, p []byte, offset int64) (n int, err error)
```

func Read

```
func Read(fd int, p []byte) (n int, err error)
```

func ReadDirent

```
func ReadDirent(fd int, buf []byte) (n int, err error)
```

func Readlink

```
func Readlink(path string, buf []byte) (n int, err error)
```

func Reboot

```
func Reboot(cmd int) (err error)
```

func Removexattr

```
func Removexattr(path string, attr string) (err error)
```

func Rename

```
func Rename(oldpath string, newpath string) (err error)
```

func Renameat

```
func Renameat(olddirfd int, oldpath string, newdirfd int, newpath string) (err error)
```

func Rmdir

```
func Rmdir(path string) error
```

func Seek

```
func Seek(fd int, offset int64, whence int) (off int64, err error)
```

func Select

```
func Select(nfd int, r *FdSet, w *FdSet, e *FdSet, timeout *Timeval) (n int, err erro
```

func Sendfile

```
func Sendfile(outfd int, infd int, offset *int64, count int) (written int, err error)
```

func Sendmsg

```
func Sendmsg(fd int, p, oob []byte, to Sockaddr, flags int) (err error)
```

func SendmsgN

```
func SendmsgN(fd int, p, oob []byte, to Sockaddr, flags int) (n int, err error)
```

func Sendto

```
func Sendto(fd int, p []byte, flags int, to Sockaddr) (err error)
```

func SetLsfPromisc

```
func SetLsfPromisc(name string, m bool) error
```

Deprecated: Use golang.org/x/net/bpf instead.

func SetNonblock

```
func SetNonblock(fd int, nonblocking bool) (err error)
```

func Setdomainname

```
func Setdomainname(p []byte) (err error)
```

func Setenv

```
func Setenv(key, value string) error
```

func Setfsgid

```
func Setfsgid(gid int) (err error)
```

func Setfsuid

```
func Setfsuid(uid int) (err error)
```

func Setgid

```
func Setgid(gid int) (err error)
```

func Setgroups

```
func Setgroups(gids []int) (err error)
```

func Sethostname

```
func Sethostname(p []byte) (err error)
```

func Setpgid

```
func Setpgid(pid int, pgid int) (err error)
```

func Setpriority

```
func Setpriority(which int, who int, prio int) (err error)
```

func Setregid

```
func Setregid(rgid int, egid int) (err error)
```

func Setresgid

```
func Setresgid(rgid int, egid int, sgid int) (err error)
```

func Setresuid

```
func Setresuid(ruid int, euid int, suid int) (err error)
```

func Setreuid

```
func Setreuid(ruid int, euid int) (err error)
```

func Setrlimit

```
func Setrlimit(resource int, rlim *Rlimit) (err error)
```

func Setsid

```
func Setsid() (pid int, err error)
```

func SetsockoptByte

```
func SetsockoptByte(fd, level, opt int, value byte) (err error)
```

func SetsockoptICMPv6Filter

```
func SetsockoptICMPv6Filter(fd, level, opt int, filter *ICMPv6Filter) error
```

func SetsockoptIPMreq

```
func SetsockoptIPMreq(fd, level, opt int, mreq *IPMreq) (err error)
```

func SetsockoptIPMreqn

```
func SetsockoptIPMreqn(fd, level, opt int, mreq *IPMreqn) (err error)
```

func SetsockoptIPv6Mreq

```
func SetsockoptIPv6Mreq(fd, level, opt int, mreq *IPv6Mreq) (err error)
```

func SetsockoptInet4Addr

```
func SetsockoptInet4Addr(fd, level, opt int, value [4]byte) (err error)
```

func SetsockoptInt

```
func SetsockoptInt(fd, level, opt int, value int) (err error)
```

func SetsockoptLinger

```
func SetsockoptLinger(fd, level, opt int, l *Linger) (err error)
```

func SetsockoptString

```
func SetsockoptString(fd, level, opt int, s string) (err error)
```

func SetsockoptTimeval

```
func SetsockoptTimeval(fd, level, opt int, tv *Timeval) (err error)
```

func Settimeofday

```
func Settimeofday(tv *Timeval) (err error)
```

func Setuid

```
func Setuid(uid int) (err error)
```

func Setxattr

```
func Setxattr(path string, attr string, data []byte, flags int) (err error)
```

func Shutdown

```
func Shutdown(fd int, how int) (err error)
```

func SlicePtrFromStrings

```
func SlicePtrFromStrings(ss []string) ([]*byte, error)
```

SlicePtrFromStrings converts a slice of strings to a slice of pointers to NUL-terminated byte arrays. If any string contains a NUL byte, it returns (nil, EINVAL).

func Socket

```
func Socket(domain, typ, proto int) (fd int, err error)
```

func Socketpair

```
func Socketpair(domain, typ, proto int) (fd [2]int, err error)
```

func Splice

```
func Splice(rfd int, roff *int64, wfd int, woff *int64, len int, flags int) (n int64,
```

func StartProcess

```
func StartProcess(argv0 string, argv []string, attr *ProcAttr) (pid int, handle uintp
```

StartProcess wraps ForkExec for package os.

func Stat

```
func Stat(path string, stat *Stat_t) (err error)
```

func Statfs

```
func Statfs(path string, buf *Statfs_t) (err error)
```

func StringBytePtr

```
func StringBytePtr(s string) *byte
```

StringBytePtr returns a pointer to a NUL-terminated array of bytes. If s contains a NUL byte this function panics instead of returning an error.

Deprecated: Use BytePtrFromString instead.

func StringByteSlice

```
func StringByteSlice(s string) []byte
```

StringByteSlice converts a string to a NUL-terminated []byte, If s contains a NUL byte this function panics instead of returning an error.

Deprecated: Use ByteSliceFromString instead.

func StringSlicePtr

```
func StringSlicePtr(ss []string) []*byte
```

StringSlicePtr converts a slice of strings to a slice of pointers to NUL-terminated byte arrays. If any string contains a NUL byte this function panics instead of returning an error.

Deprecated: Use SlicePtrFromStrings instead.

func Symlink

```
func Symlink(oldpath string, newpath string) (err error)
```

func Sync

```
func Sync()
```

func SyncFileRange

```
func SyncFileRange(fd int, off int64, n int64, flags int) (err error)
```

func Sysinfo

```
func Sysinfo(info *Sysinfo_t) (err error)
```

func Tee

```
func Tee(rfd int, wfd int, len int, flags int) (n int64, err error)
```

func Tgkill

```
func Tgkill(tgid int, tid int, sig Signal) (err error)
```

func Times

```
func Times(tms *Tms) (ticks uintptr, err error)
```

func TimespecToNsec

```
func TimespecToNsec(ts Timespec) int64
```

TimespecToNsec converts a Timespec value into a number of nanoseconds since the Unix epoch.

func TimevalToNsec

```
func TimevalToNsec(tv Timeval) int64
```

TimevalToNsec converts a Timeval value into a number of nanoseconds since the Unix epoch.

func Truncate

```
func Truncate(path string, length int64) (err error)
```

func Umask

```
func Umask(mask int) (oldmask int)
```

func Uname

```
func Uname(buf *Utsname) (err error)
```

func UnixCredentials

```
func UnixCredentials(ucred *Ucred) []byte
```

UnixCredentials encodes credentials into a socket control message for sending to another process. This can be used for authentication.

func UnixRights

```
func UnixRights(fds ...int) []byte
```

UnixRights encodes a set of open file descriptors into a socket control message for sending to another process.

func Unlink

```
func Unlink(path string) error
```

func Unlinkat

```
func Unlinkat(dirfd int, path string) error
```

func Unmount

```
func Unmount(target string, flags int) (err error)
```

func Unsetenv

```
func Unsetenv(key string) error
```

func Unshare

```
func Unshare(flags int) (err error)
```

func Ustat

```
func Ustat(dev int, ubuf *Ustat_t) (err error)
```

func Utime

```
func Utime(path string, buf *Utimbuf) (err error)
```

func Utimes

```
func Utimes(path string, tv []Timeval) (err error)
```

func UtimesNano

```
func UtimesNano(path string, ts []Timespec) (err error)
```

func Wait4

```
func Wait4(pid int, wstatus *WaitStatus, options int, rusage *Rusage) (wpid int, err
```

func Write

```
func Write(fd int, p []byte) (n int, err error)
```

type Cmsghdr

```
type Cmsghdr struct {
    Len   uint64
```

```
    Level int32
    Type  int32
}
```

func (*Cmsghdr) SetLen

```
func (cmsg *Cmsghdr) SetLen(length int)
```

type Conn

```
type Conn interface {
    // SyscallConn returns a raw network connection.
    SyscallConn() (RawConn, error)
}
```

Conn is implemented by some types in the net and os packages to provide access to the underlying file descriptor or handle.

type Credential

```
type Credential struct {
    Uid         uint32   // User ID.
    Gid         uint32   // Group ID.
    Groups      []uint32 // Supplementary group IDs.
    NoSetGroups bool     // If true, don't set supplementary groups
}
```

Credential holds user and group identities to be assumed by a child process started by StartProcess.

type Dirent

```
type Dirent struct {
    Ino       uint64
    Off       int64
    Reclen    uint16
    Type      uint8
    Name      [256]int8
    Pad_cgo_0 [5]byte
}
```

type EpollEvent

```
type EpollEvent struct {
    Events uint32
    Fd     int32
    Pad    int32
}
```

type Errno

```
type Errno uintptr
```

An Errno is an unsigned number describing an error condition. It implements the error interface. The zero Errno is by convention a non-error, so code to convert from Errno to error should use:

```
err = nil
if errno != 0 {
    err = errno
}
```

Errno values can be tested against error values from the os package using errors.Is. For example:

```
_, _, err := syscall.Syscall(...)
if errors.Is(err, os.ErrNotExist) ...
```

func RawSyscall

```
func RawSyscall(trap, a1, a2, a3 uintptr) (r1, r2 uintptr, err Errno)
```

func RawSyscall6

```
func RawSyscall6(trap, a1, a2, a3, a4, a5, a6 uintptr) (r1, r2 uintptr, err Errno)
```

func Syscall

```
func Syscall(trap, a1, a2, a3 uintptr) (r1, r2 uintptr, err Errno)
```

func Syscall6

```
func Syscall6(trap, a1, a2, a3, a4, a5, a6 uintptr) (r1, r2 uintptr, err Errno)
```

func (Errno) Error

```
func (e Errno) Error() string
```

func (Errno) Is

```
func (e Errno) Is(target error) bool
```

func (Errno) Temporary

```
func (e Errno) Temporary() bool
```

func (Errno) Timeout

```
func (e Errno) Timeout() bool
```

type FdSet

```
type FdSet struct {
    Bits [16]int64
}
```

type Flock_t

```
type Flock_t struct {
    Type      int16
    Whence    int16
    Pad_cgo_0 [4]byte
    Start     int64
    Len       int64
    Pid       int32
    Pad_cgo_1 [4]byte
}
```

type Fsid

```
type Fsid struct {
    X__val [2]int32
}
```

type ICMPv6Filter

```
type ICMPv6Filter struct {
    Data [8]uint32
}
```

func GetsockoptICMPv6Filter

```
func GetsockoptICMPv6Filter(fd, level, opt int) (*ICMPv6Filter, error)
```

type IPMreq

```
type IPMreq struct {
    Multiaddr [4]byte /* in_addr */
    Interface [4]byte /* in_addr */
}
```

func GetsockoptIPMreq

```
func GetsockoptIPMreq(fd, level, opt int) (*IPMreq, error)
```

type IPMreqn

```
type IPMreqn struct {
    Multiaddr [4]byte /* in_addr */
    Address   [4]byte /* in_addr */
    Ifindex   int32
}
```

func GetsockoptIPMreqn

```
func GetsockoptIPMreqn(fd, level, opt int) (*IPMreqn, error)
```

type IPv6MTUInfo

```
type IPv6MTUInfo struct {
    Addr RawSockaddrInet6
    Mtu  uint32
}
```

func GetsockoptIPv6MTUInfo

```
func GetsockoptIPv6MTUInfo(fd, level, opt int) (*IPv6MTUInfo, error)
```

type IPv6Mreq

```
type IPv6Mreq struct {
    Multiaddr [16]byte /* in6_addr */
    Interface uint32
}
```

func GetsockoptIPv6Mreq

```
func GetsockoptIPv6Mreq(fd, level, opt int) (*IPv6Mreq, error)
```

type IfAddrmsg

```
type IfAddrmsg struct {
    Family    uint8
    Prefixlen uint8
    Flags     uint8
    Scope     uint8
```

```
    Index     uint32
}
```

type IfInfomsg

```
type IfInfomsg struct {
    Family     uint8
    X__ifi_pad uint8
    Type       uint16
    Index      int32
    Flags      uint32
    Change     uint32
}
```

type Inet4Pktinfo

```
type Inet4Pktinfo struct {
    Ifindex  int32
    Spec_dst [4]byte /* in_addr */
    Addr     [4]byte /* in_addr */
}
```

type Inet6Pktinfo

```
type Inet6Pktinfo struct {
    Addr    [16]byte /* in6_addr */
    Ifindex uint32
}
```

type InotifyEvent

```
type InotifyEvent struct {
    Wd     int32
    Mask   uint32
    Cookie uint32
    Len    uint32
    Name   [0]uint8
}
```

type Iovec

```
type Iovec struct {
    Base *byte
    Len  uint64
}
```

func (*Iovec) SetLen

```
func (iov *Iovec) SetLen(length int)
```

type Linger

```
type Linger struct {
    Onoff  int32
    Linger int32
}
```

type Msghdr

```
type Msghdr struct {
    Name       *byte
    Namelen    uint32
    Pad_cgo_0  [4]byte
    Iov        *Iovec
    Iovlen     uint64
    Control    *byte
    Controllen uint64
    Flags      int32
    Pad_cgo_1  [4]byte
}
```

func (*Msghdr) SetControllen

```
func (msghdr *Msghdr) SetControllen(length int)
```

type NetlinkMessage

```
type NetlinkMessage struct {
    Header NlMsghdr
    Data   []byte
}
```

NetlinkMessage represents a netlink message.

func ParseNetlinkMessage

```
func ParseNetlinkMessage(b []byte) ([]NetlinkMessage, error)
```

ParseNetlinkMessage parses b as an array of netlink messages and returns the slice containing the NetlinkMessage structures.

type NetlinkRouteAttr

```
type NetlinkRouteAttr struct {
    Attr  RtAttr
    Value []byte
}
```

NetlinkRouteAttr represents a netlink route attribute.

func ParseNetlinkRouteAttr

```
func ParseNetlinkRouteAttr(m *NetlinkMessage) ([]NetlinkRouteAttr, error)
```

ParseNetlinkRouteAttr parses m's payload as an array of netlink route attributes and returns the slice containing the NetlinkRouteAttr structures.

type NetlinkRouteRequest

```
type NetlinkRouteRequest struct {
    Header NlMsghdr
    Data   RtGenmsg
}
```

NetlinkRouteRequest represents a request message to receive routing and link states from the kernel.

type NlAttr

```
type NlAttr struct {
    Len  uint16
    Type uint16
}
```

type NlMsgerr

```
type NlMsgerr struct {
    Error int32
    Msg   NlMsghdr
}
```

type NlMsghdr

```
type NlMsghdr struct {
    Len   uint32
    Type  uint16
    Flags uint16
    Seq   uint32
    Pid   uint32
}
```

type ProcAttr

```
type ProcAttr struct {
    Dir   string    // Current working directory.
    Env   []string  // Environment.
    Files []uintptr // File descriptors.
    Sys   *SysProcAttr
}
```

ProcAttr holds attributes that will be applied to a new process started by StartProcess.

type PtraceRegs

```
type PtraceRegs struct {
    R15      uint64
    R14      uint64
    R13      uint64
    R12      uint64
    Rbp      uint64
    Rbx      uint64
    R11      uint64
    R10      uint64
    R9       uint64
    R8       uint64
    Rax      uint64
    Rcx      uint64
    Rdx      uint64
    Rsi      uint64
    Rdi      uint64
    Orig_rax uint64
    Rip      uint64
    Cs       uint64
    Eflags   uint64
    Rsp      uint64
    Ss       uint64
    Fs_base  uint64
    Gs_base  uint64
    Ds       uint64
    Es       uint64
    Fs       uint64
    Gs       uint64
}
```

func (*PtraceRegs) PC

```
func (r *PtraceRegs) PC() uint64
```

func (*PtraceRegs) SetPC

```
func (r *PtraceRegs) SetPC(pc uint64)
```

type RawConn

```
type RawConn interface {
    // Control invokes f on the underlying connection's file
    // descriptor or handle.
    // The file descriptor fd is guaranteed to remain valid while
    // f executes but not after f returns.
    Control(f func(fd uintptr)) error

    // Read invokes f on the underlying connection's file
    // descriptor or handle; f is expected to try to read from the
    // file descriptor.
    // If f returns true, Read returns. Otherwise Read blocks
    // waiting for the connection to be ready for reading and
    // tries again repeatedly.
    // The file descriptor is guaranteed to remain valid while f
    // executes but not after f returns.
    Read(f func(fd uintptr) (done bool)) error

    // Write is like Read but for writing.
    Write(f func(fd uintptr) (done bool)) error
}
```

A RawConn is a raw network connection.

type RawSockaddr

```
type RawSockaddr struct {
    Family uint16
    Data   [14]int8
}
```

type RawSockaddrAny

```
type RawSockaddrAny struct {
    Addr RawSockaddr
    Pad  [96]int8
}
```

type RawSockaddrInet4

```
type RawSockaddrInet4 struct {
    Family uint16
    Port   uint16
    Addr   [4]byte /* in_addr */
    Zero   [8]uint8
}
```

type RawSockaddrInet6

```
type RawSockaddrInet6 struct {
    Family   uint16
    Port     uint16
    Flowinfo uint32
    Addr     [16]byte /* in6_addr */
    Scope_id uint32
}
```

type RawSockaddrLinklayer

```
type RawSockaddrLinklayer struct {
    Family   uint16
    Protocol uint16
    Ifindex  int32
    Hatype   uint16
    Pkttype  uint8
    Halen    uint8
    Addr     [8]uint8
}
```

type RawSockaddrNetlink

```
type RawSockaddrNetlink struct {
    Family uint16
    Pad    uint16
    Pid    uint32
    Groups uint32
}
```

type RawSockaddrUnix

```
type RawSockaddrUnix struct {
    Family uint16
    Path   [108]int8
}
```

type Rlimit

```
type Rlimit struct {
    Cur uint64
    Max uint64
}
```

type RtAttr

```
type RtAttr struct {
    Len  uint16
```

```
    Type uint16
}
```

type RtGenmsg

```
type RtGenmsg struct {
    Family uint8
}
```

type RtMsg

```
type RtMsg struct {
    Family   uint8
    Dst_len  uint8
    Src_len  uint8
    Tos      uint8
    Table    uint8
    Protocol uint8
    Scope    uint8
    Type     uint8
    Flags    uint32
}
```

type RtNexthop

```
type RtNexthop struct {
    Len     uint16
    Flags   uint8
    Hops    uint8
    Ifindex int32
}
```

type Rusage

```
type Rusage struct {
    Utime    Timeval
    Stime    Timeval
    Maxrss   int64
    Ixrss    int64
    Idrss    int64
    Isrss    int64
    Minflt   int64
    Majflt   int64
    Nswap    int64
    Inblock  int64
    Oublock  int64
    Msgsnd   int64
    Msgrcv   int64
    Nsignals int64
```

```
    Nvcsw    int64
    Nivcsw   int64
}
```

type Signal

```
type Signal int
```

A Signal is a number describing a process signal. It implements the os.Signal interface.

func (Signal) Signal

```
func (s Signal) Signal()
```

func (Signal) String

```
func (s Signal) String() string
```

type SockFilter

```
type SockFilter struct {
    Code uint16
    Jt   uint8
    Jf   uint8
    K    uint32
}
```

func LsfJump

```
func LsfJump(code, k, jt, jf int) *SockFilter
```

Deprecated: Use golang.org/x/net/bpf instead.

func LsfStmt

```
func LsfStmt(code, k int) *SockFilter
```

Deprecated: Use golang.org/x/net/bpf instead.

type SockFprog

```
type SockFprog struct {
    Len       uint16
    Pad_cgo_0 [6]byte
    Filter    *SockFilter
}
```

type Sockaddr

```
type Sockaddr interface {
    // contains filtered or unexported methods
}
```

func Accept

```
func Accept(fd int) (nfd int, sa Sockaddr, err error)
```

func Accept4

```
func Accept4(fd int, flags int) (nfd int, sa Sockaddr, err error)
```

func Getpeername

```
func Getpeername(fd int) (sa Sockaddr, err error)
```

func Getsockname

```
func Getsockname(fd int) (sa Sockaddr, err error)
```

func Recvfrom

```
func Recvfrom(fd int, p []byte, flags int) (n int, from Sockaddr, err error)
```

func Recvmsg

```
func Recvmsg(fd int, p, oob []byte, flags int) (n, oobn int, recvflags int, from Sock
```

type SockaddrInet4

```
type SockaddrInet4 struct {
    Port int
    Addr [4]byte
    // contains filtered or unexported fields
}
```

type SockaddrInet6

```
type SockaddrInet6 struct {
    Port   int
    ZoneId uint32
    Addr   [16]byte
```

```
    // contains filtered or unexported fields
}
```

type SockaddrLinklayer

```
type SockaddrLinklayer struct {
    Protocol uint16
    Ifindex  int
    Hatype   uint16
    Pkttype  uint8
    Halen    uint8
    Addr     [8]byte
    // contains filtered or unexported fields
}
```

type SockaddrNetlink

```
type SockaddrNetlink struct {
    Family uint16
    Pad    uint16
    Pid    uint32
    Groups uint32
    // contains filtered or unexported fields
}
```

type SockaddrUnix

```
type SockaddrUnix struct {
    Name string
    // contains filtered or unexported fields
}
```

type SocketControlMessage

```
type SocketControlMessage struct {
    Header Cmsghdr
    Data   []byte
}
```

SocketControlMessage represents a socket control message.

func ParseSocketControlMessage

```
func ParseSocketControlMessage(b []byte) ([]SocketControlMessage, error)
```

ParseSocketControlMessage parses b as an array of socket control messages.

type Stat_t

```
type Stat_t struct {
    Dev       uint64
    Ino       uint64
    Nlink     uint64
    Mode      uint32
    Uid       uint32
    Gid       uint32
    X__pad0   int32
    Rdev      uint64
    Size      int64
    Blksize   int64
    Blocks    int64
    Atim      Timespec
    Mtim      Timespec
    Ctim      Timespec
    X__unused [3]int64
}
```

type Statfs_t

```
type Statfs_t struct {
    Type    int64
    Bsize   int64
    Blocks  uint64
    Bfree   uint64
    Bavail  uint64
    Files   uint64
    Ffree   uint64
    Fsid    Fsid
    Namelen int64
    Frsize  int64
    Flags   int64
    Spare   [4]int64
}
```

type SysProcAttr

```
type SysProcAttr struct {
    Chroot     string      // Chroot.
    Credential *Credential // Credential.
    // Ptrace tells the child to call ptrace(PTRACE_TRACEME).
    // Call runtime.LockOSThread before starting a process with this set,
    // and don't call UnlockOSThread until done with PtraceSyscall calls.
    Ptrace bool
    Setsid bool // Create session.
    // Setpgid sets the process group ID of the child to Pgid,
    // or, if Pgid == 0, to the new child's process ID.
    Setpgid bool
    // Setctty sets the controlling terminal of the child to
```

```
    // file descriptor Ctty. Ctty must be a descriptor number
    // in the child process: an index into ProcAttr.Files.
    // This is only meaningful if Setsid is true.
    Setctty bool
    Noctty  bool // Detach fd 0 from controlling terminal
    Ctty    int  // Controlling TTY fd
    // Foreground places the child process group in the foreground.
    // This implies Setpgid. The Ctty field must be set to
    // the descriptor of the controlling TTY.
    // Unlike Setctty, in this case Ctty must be a descriptor
    // number in the parent process.
    Foreground   bool
    Pgid         int            // Child's process group ID if Setpgid.
    Pdeathsig    Signal         // Signal that the process will get when its parent d
    Cloneflags   uintptr        // Flags for clone calls (Linux only)
    Unshareflags uintptr        // Flags for unshare calls (Linux only)
    UidMappings  []SysProcIDMap // User ID mappings for user namespaces.
    GidMappings  []SysProcIDMap // Group ID mappings for user namespaces.
    // GidMappingsEnableSetgroups enabling setgroups syscall.
    // If false, then setgroups syscall will be disabled for the child process.
    // This parameter is no-op if GidMappings == nil. Otherwise for unprivileged
    // users this should be set to false for mappings work.
    GidMappingsEnableSetgroups bool
    AmbientCaps                []uintptr // Ambient capabilities (Linux only)
}
```

type SysProcIDMap

```
type SysProcIDMap struct {
    ContainerID int // Container ID.
    HostID      int // Host ID.
    Size        int // Size.
}
```

SysProcIDMap holds Container ID to Host ID mappings used for User Namespaces in Linux. See user_namespaces(7).

type Sysinfo_t

```
type Sysinfo_t struct {
    Uptime    int64
    Loads     [3]uint64
    Totalram  uint64
    Freeram   uint64
    Sharedram uint64
    Bufferram uint64
    Totalswap uint64
    Freeswap  uint64
    Procs     uint16
    Pad       uint16
    Pad_cgo_0 [4]byte
```

```
    Totalhigh uint64
    Freehigh  uint64
    Unit      uint32
    X_f       [0]byte
    Pad_cgo_1 [4]byte
}
```

type TCPInfo

```
type TCPInfo struct {
    State          uint8
    Ca_state       uint8
    Retransmits    uint8
    Probes         uint8
    Backoff        uint8
    Options        uint8
    Pad_cgo_0      [2]byte
    Rto            uint32
    Ato            uint32
    Snd_mss        uint32
    Rcv_mss        uint32
    Unacked        uint32
    Sacked         uint32
    Lost           uint32
    Retrans        uint32
    Fackets        uint32
    Last_data_sent uint32
    Last_ack_sent  uint32
    Last_data_recv uint32
    Last_ack_recv  uint32
    Pmtu           uint32
    Rcv_ssthresh   uint32
    Rtt            uint32
    Rttvar         uint32
    Snd_ssthresh   uint32
    Snd_cwnd       uint32
    Advmss         uint32
    Reordering     uint32
    Rcv_rtt        uint32
    Rcv_space      uint32
    Total_retrans  uint32
}
```

type Termios

```
type Termios struct {
    Iflag     uint32
    Oflag     uint32
    Cflag     uint32
    Lflag     uint32
    Line      uint8
    Cc        [32]uint8
```

```
    Pad_cgo_0 [3]byte
    Ispeed    uint32
    Ospeed    uint32
}
```

type Time_t

```
type Time_t int64
```

func Time

```
func Time(t *Time_t) (tt Time_t, err error)
```

type Timespec

```
type Timespec struct {
    Sec  int64
    Nsec int64
}
```

func NsecToTimespec

```
func NsecToTimespec(nsec int64) Timespec
```

NsecToTimespec takes a number of nanoseconds since the Unix epoch and returns the corresponding Timespec value.

func (*Timespec) Nano

```
func (ts *Timespec) Nano() int64
```

Nano returns ts as the number of nanoseconds elapsed since the Unix epoch.

func (*Timespec) Unix

```
func (ts *Timespec) Unix() (sec int64, nsec int64)
```

Unix returns ts as the number of seconds and nanoseconds elapsed since the Unix epoch.

type Timeval

```
type Timeval struct {
    Sec  int64
    Usec int64
}
```

func NsecToTimeval

```
func NsecToTimeval(nsec int64) Timeval
```

NsecToTimeval takes a number of nanoseconds since the Unix epoch and returns the corresponding Timeval value.

func (*Timeval) Nano

```
func (tv *Timeval) Nano() int64
```

Nano returns tv as the number of nanoseconds elapsed since the Unix epoch.

func (*Timeval) Unix

```
func (tv *Timeval) Unix() (sec int64, nsec int64)
```

Unix returns tv as the number of seconds and nanoseconds elapsed since the Unix epoch.

type Timex

```
type Timex struct {
    Modes     uint32
    Pad_cgo_0 [4]byte
    Offset    int64
    Freq      int64
    Maxerror  int64
    Esterror  int64
    Status    int32
    Pad_cgo_1 [4]byte
    Constant  int64
    Precision int64
    Tolerance int64
    Time      Timeval
    Tick      int64
    Ppsfreq   int64
    Jitter    int64
    Shift     int32
    Pad_cgo_2 [4]byte
    Stabil    int64
    Jitcnt    int64
    Calcnt    int64
    Errcnt    int64
    Stbcnt    int64
    Tai       int32
    Pad_cgo_3 [44]byte
}
```

type Tms

```
type Tms struct {
    Utime  int64
    Stime  int64
    Cutime int64
    Cstime int64
}
```

type Ucred

```
type Ucred struct {
    Pid int32
    Uid uint32
    Gid uint32
}
```

func GetsockoptUcred

```
func GetsockoptUcred(fd, level, opt int) (*Ucred, error)
```

func ParseUnixCredentials

```
func ParseUnixCredentials(m *SocketControlMessage) (*Ucred, error)
```

ParseUnixCredentials decodes a socket control message that contains credentials in a Ucred structure. To receive such a message, the SO_PASSCRED option must be enabled on the socket.

type Ustat_t

```
type Ustat_t struct {
    Tfree     int32
    Pad_cgo_0 [4]byte
    Tinode    uint64
    Fname     [6]int8
    Fpack     [6]int8
    Pad_cgo_1 [4]byte
}
```

type Utimbuf

```
type Utimbuf struct {
    Actime  int64
    Modtime int64
}
```

type Utsname

```
type Utsname struct {
    Sysname    [65]int8
    Nodename   [65]int8
    Release    [65]int8
    Version    [65]int8
    Machine    [65]int8
    Domainname [65]int8
}
```

type WaitStatus

```
type WaitStatus uint32
```

func (WaitStatus) Continued

```
func (w WaitStatus) Continued() bool
```

func (WaitStatus) CoreDump

```
func (w WaitStatus) CoreDump() bool
```

func (WaitStatus) ExitStatus

```
func (w WaitStatus) ExitStatus() int
```

func (WaitStatus) Exited

```
func (w WaitStatus) Exited() bool
```

func (WaitStatus) Signal

```
func (w WaitStatus) Signal() Signal
```

func (WaitStatus) Signaled

```
func (w WaitStatus) Signaled() bool
```

func (WaitStatus) StopSignal

```
func (w WaitStatus) StopSignal() Signal
```

func (WaitStatus) Stopped

```
func (w WaitStatus) Stopped() bool
```

func (WaitStatus) TrapCause

```
func (w WaitStatus) TrapCause() int
```