Go 💖 MIDI

Exploration of Linux's System Interfaces

Tracks

0. Presentation Header
1. MIDI and SMF
2. Linux
3. Hardware
4. Demo
5. Conclusion

Track 1:
MIDI and SMF

Musical Instrument Digital Interface

- Protocol and Electrical standard for connecting musical instruments, computers, and related audio devices
- Defined in 1983, continues to be in active use
- Baud rate: 31250
- Supports up to 16 channels multiplexed
- Commands to trigger notes, no audio data is transmitted

Live Messages

- Channel Messages
    - Voice (note on/off, key pressure, pitch bends, bank select)
    - Mode (silence, polyphonic, monophonic)
- System Messages
    - Real Time (clock, start/stop, reset)
    - Common (MIDI timing code, song select)
    - Exclusive (stream of bytes)

Standard MIDI Files

- Set of instructions for MIDI devices
  - Sequencers, DAW, video games
- Storages messages along with their position
- Versions
  - Single track
  - Multitrack synchronous
  - Multitrack asynchronous
- Sounds only as good as the hardware synthesizing

SMF Messages

- Meta Messages
  - Sequence and track names, copyrights, lyrics
  - Time signature, tempo, key
- Channel Messages
- System Common and Exclusive Messages

BREAKING NEWS BREAKING NEWS BREAKING NEWS

MIDI 2.0

- Draft announced January 2019
- Increased resolution (8-bit to 32-bit in many cases)
- Supports up to 256 channels
- Backwards compatible

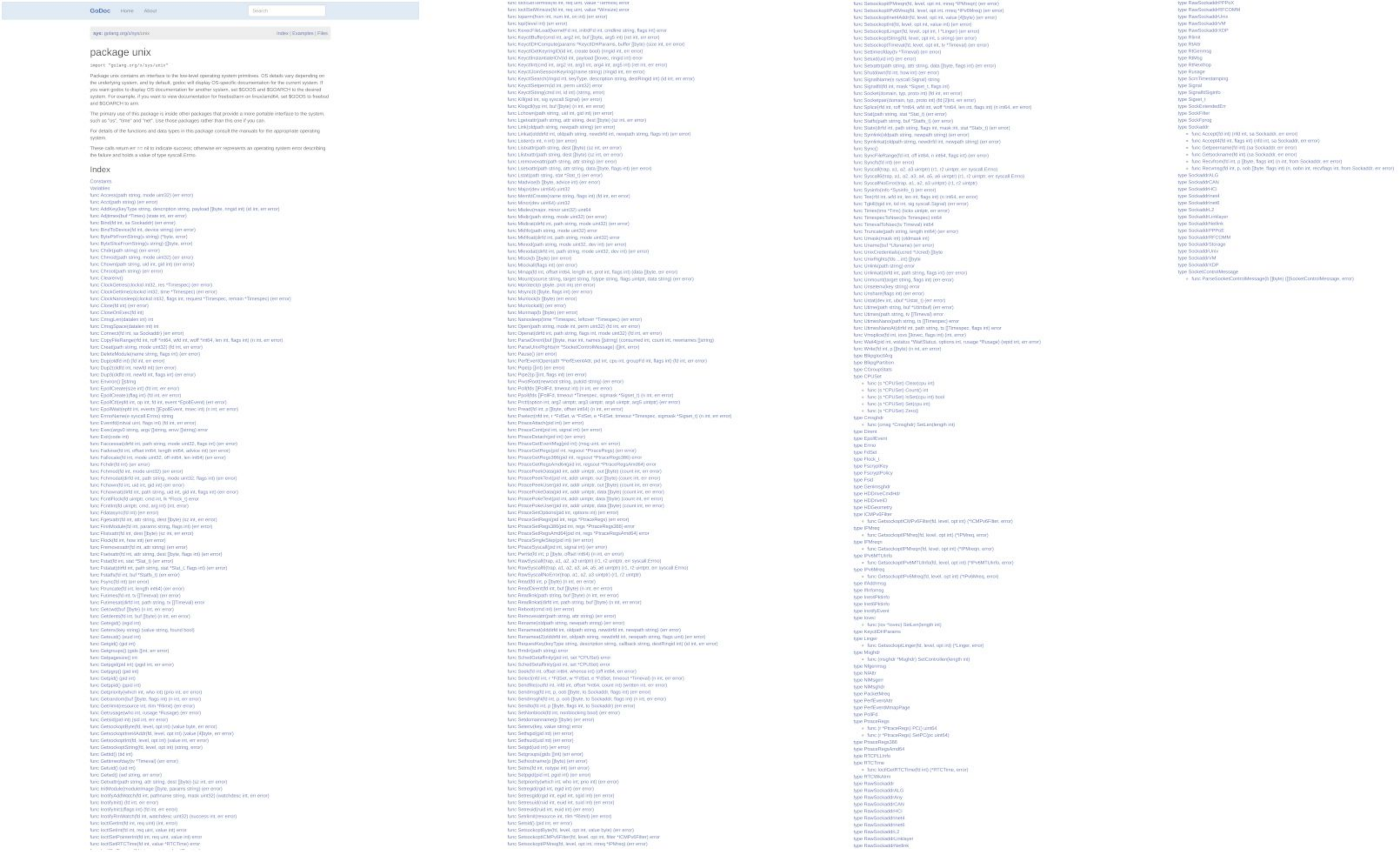
Track 2:
Linux

gokrazy

- Pure-Go Linux userland for applications
  - Small init written in Go
  - No C runtime
- `gokr-packer` CLI tool to build the distribution, including our Go applications

Syscall

- Directly make function calls into the kernel
- A lot of functionality is exposed!



GoDoc Home About Search

sys: golang.org/x/sys/unix Index | Examples | Files

package unix

```
import "golang.org/x/sys/unix"
```

Package unix contains an interface to the low-level operating system primitives. OS details vary depending on the underlying system, and by default, godoc will display OS-specific documentation for the current system. If you want godoc to display OS 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 this package 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 represents an operating system error describing the failure and holds a value of type syscall.Errno.

Index

Constants
Variables
func Access(path string, mode uint32) (err error)
func Acct(path string) (err error)
func AddKey(keyType string, description string, payload []byte, ringid int) (id int, err error)
func Adjtimex(buf *Timex) (state int, err error)
func Bind(fd int, sa Sockaddr) (err error)
func BindToDevice(fd int, device string) (err error)
func BytePtrFromString(s string) (*byte, error)
func ByteSliceFromString(s string) ([]byte, error)
func Chdir(path string) (err error)
func Chmod(path string, mode uint32) (err error)
func Chown(path string, uid int, gid int) (err error)
func Chroot(path string) (err error)
func Clearenv()
func ClockGetres(clockid int32, res *Timespec) (err error)
func ClockGettime(clockid int32, time *Timespec) (err error)
func ClockNanosleep(clockid int32, flags int, request *Timespec, remain *Timespec) (err error)
func Close(fd int) (err error)
func CloseOnExec(fd int)
func CmsgLen(datalen int) int
func CmsgSpace(datalen int) int
func Connect(fd int, sa Sockaddr) (err error)
func CopyFileRange(rfd int, roff *int64, wfd int, woff *int64, len int, flags int) (n int, err error)
func Creat(path string, mode uint32) (fd int, err error)
func DeleteModule(name string, flags int) (err error)
func Dup(oldfd int) (fd int, err error)
func Dup2(oldfd int, newfd int) (err error)
func Dup3(oldfd int, newfd int, flags int) (err error)
func Environ() []string
func EpollCreate(size int) (fd int, err error)
func EpollCreate1(flag int) (fd int, err error)
func EpollCtl(epfd int, op int, fd int, event *EpollEvent) (err error)
func EpollWait(epfd int, events []EpollEvent, msec int) (n int, err error)
func ErrnoName(e syscall.Errno) string
func Eventfd(initval uint, flags int) (fd int, err error)
func Exec(argv0 string, argv []string, envv []string) error
func Exit(code int)
func Faccessat(dirfd int, path string, mode uint32, flags int) (err error)
func Fadvise(fd int, offset int64, length int64, advice int) (err error)
func Fallocate(fd int, mode uint32, off int64, len int64) (err error)
func Fchdir(fd int) (err error)
func Fchmod(fd int, mode uint32) (err error)
func Fchmodat(dirfd int, path string, mode uint32, flags int) (err error)
func Fchown(fd int, uid int, gid int) (err error)
func Fchownat(dirfd int, path string, uid int, gid int, flags int) (err error)
func FcntlFlock(fd uintptr, cmd int, lk *Flock_t) error
func FcntlInt(fd uintptr, cmd, arg int) (int, error)
func Fdatasync(fd int) (err error)
func Fgetxattr(fd int, attr string, dest []byte) (sz int, err error)
func FinitModule(fd int, params string, flags int) (err error)
func Flistxattr(fd int, dest []byte) (sz int, err error)
func Flock(fd int, how int) (err error)
func Fremovexattr(fd int, attr string) (err error)
func Fsetxattr(fd int, attr string, dest []byte, flags int) (err error)
func Fstat(fd int, stat *Stat_t) (err error)
func Fstatat(dirfd int, path string, stat *Stat_t, flags int) (err error)
func Fstatfs(fd int, buf *Statfs_t) (err error)
func Fsync(fd int) (err error)
func Ftruncate(fd int, length int64) (err error)
func Futimes(fd int, tv []Timeval) (err error)
func Futimesat(dirfd int, path string, tv []Timeval) error
func Getcwd(buf []byte) (n int, err error)
func Getdents(fd int, buf []byte) (n int, err error)
func Getegid() (egid int)
func Getenv(key string) (value string, found bool)
func Geteuid() (euid int)
func Getgid() (gid int)
func Getgroups() (gids []int, err error)
func Getpagesize() int
func Getpgid(pid int) (pgid int, err error)
func Getpgrp() (pid int)
func Getpid() (pid int)
func Getppid() (ppid int)
func Getpriority(which int, who int) (prio int, err error)
func Getrandom(buf []byte, flags int) (n int, err error)
func Getrlimit(resource int, rlim *Rlimit) (err error)
func Getrusage(who int, rusage *Rusage) (err error)
func Getsid(pid int) (sid int, err error)
func GetsockoptByte(fd, level, opt int) (value byte, err error)
func GetsockoptInet4Addr(fd, level, opt int) (value [4]byte, err error)
func GetsockoptInt(fd, level, opt int) (value int, err error)
func GetsockoptString(fd, level, opt int) (string, error)
func Gettid() (tid int)
func Gettimeofday(tv *Timeval) (err error)
func Getuid() (uid int)
func Getwd() (wd string, err error)
func Getxattr(path string, attr string, dest []byte) (sz int, err error)
func InitModule(moduleImage []byte, params string) (err error)
func InotifyAddWatch(fd int, pathname string, mask uint32) (watchdesc int, err error)
func InotifyInit() (fd int, err error)
func InotifyInit1(flags int) (fd int, err error)
func InotifyRmWatch(fd int, watchdesc uint32) (success int, err error)
func IoctlGetInt(fd int, req uint) (int, error)
func IoctlSetInt(fd int, req uint, value int) error
func IoctlSetPointerInt(fd int, req uint, value int) error
func IoctlSetRTCTime(fd int, value *RTCTime) error
func IoctlSetTermios(fd int, req uint, value *Termios) error
func IoctlSetWinsize(fd int, req uint, value *Winsize) error
func Ioperm(from int, num int, on int) (err error)
func Iopl(level int) (err error)
func KexecFileLoad(kernelFd int, initrdFd int, cmdline string, flags int) error
func KeyctlBuffer(cmd int, arg2 int, buf []byte, arg5 int) (ret int, err error)
func KeyctlDHCompute(params *KeyctlDHParams, buffer []byte) (size int, err error)
func KeyctlGetKeyringID(id int, create bool) (ringid int, err error)
func KeyctlInstantiateIOV(id int, payload []Iovec, ringid int) error
func KeyctlInt(cmd int, arg2 int, arg3 int, arg4 int, arg5 int) (ret int, err error)
func KeyctlJoinSessionKeyring(name string) (ringid int, err error)
func KeyctlSearch(ringid int, keyType, description string, destRingid int) (id int, err error)
func KeyctlSetperm(id int, perm uint32) error
func KeyctlString(cmd int, id int) (string, error)
func Kill(pid int, sig syscall.Signal) (err error)
func Klogctl(typ int, buf []byte) (n int, err error)
func Lchown(path string, uid int, gid int) (err error)
func Lgetxattr(path string, attr string, dest []byte) (sz int, err error)
func Link(oldpath string, newpath string) (err error)
func Linkat(olddirfd int, oldpath string, newdirfd int, newpath string, flags int) (err error)
func Listen(s int, n int) (err error)
func Listxattr(path string, dest []byte) (sz int, err error)
func Llistxattr(path string, dest []byte) (sz int, err error)
func Lremovexattr(path string, attr string) (err error)
func Lsetxattr(path string, attr string, data []byte, flags int) (err error)
func Lstat(path string, stat *Stat_t) (err error)
func Madvise(b []byte, advice int) (err error)
func Major(dev uint64) uint32
func MemfdCreate(name string, flags int) (fd int, err error)
func Minor(dev uint64) uint32
func Mkdev(major, minor uint32) uint64
func Mkdir(path string, mode uint32) (err error)
func Mkdirat(dirfd int, path string, mode uint32) (err error)
func Mkfifo(path string, mode uint32) error
func Mkfifoat(dirfd int, path string, mode uint32) error
func Mknod(path string, mode uint32, dev int) (err error)
func Mknodat(dirfd int, path string, mode uint32, dev int) (err error)
func Mlock(b []byte) (err error)
func Mlockall(flags int) (err error)
func Mmap(fd int, offset int64, length int, prot int, flags int) (data []byte, err error)
func Mount(source string, target string, fstype string, flags uintptr, data string) (err error)
func Mprotect(b []byte, prot int) (err error)
func Msync(b []byte, flags int) (err error)
func Munlock(b []byte) (err error)
func Munlockall() (err error)
func Munmap(b []byte) (err error)
func Nanosleep(time *Timespec, leftover *Timespec) (err error)
func Open(path string, mode int, perm uint32) (fd int, err error)
func Openat(dirfd int, path string, flags int, mode uint32) (fd int, err error)
func ParseDirent(buf []byte, max int, names []string) (consumed int, count int, newnames []string)
func ParseUnixRights(m *SocketControlMessage) ([]int, error)
func Pause() (err error)
func PerfEventOpen(attr *PerfEventAttr, pid int, cpu int, groupFd int, flags int) (fd int, err error)
func Pipe(p []int) (err error)
func Pipe2(p []int, flags int) (err error)
func PivotRoot(newroot string, putold string) (err error)
func Poll(fds []PollFd, timeout int) (n int, err error)
func Ppoll(fds []PollFd, timeout *Timespec, sigmask *Sigset_t) (n int, err error)
func Prctl(option int, arg2 uintptr, arg3 uintptr, arg4 uintptr, arg5 uintptr) (err error)
func Pread(fd int, p []byte, offset int64) (n int, err error)
func Pselect(nfd int, r *FdSet, w *FdSet, e *FdSet, timeout *Timespec, sigmask *Sigset_t) (n int, err error)
func PtraceAttach(pid int) (err error)
func PtraceCont(pid int, signal int) (err error)
func PtraceDetach(pid int) (err error)
func PtraceGetEventMsg(pid int) (msg uint, err error)
func PtraceGetRegs(pid int, regsout *PtraceRegs) (err error)
func PtraceGetRegs386(pid int, regsout *PtraceRegs386) error
func PtraceGetRegsAmd64(pid int, regsout *PtraceRegsAmd64) error
func PtracePeekData(pid int, addr uintptr, out []byte) (count int, err error)
func PtracePeekText(pid int, addr uintptr, out []byte) (count int, err error)
func PtracePeekUser(pid int, addr uintptr, out []byte) (count int, err error)
func PtracePokeData(pid int, addr uintptr, data []byte) (count int, err error)
func PtracePokeText(pid int, addr uintptr, data []byte) (count int, err error)
func PtracePokeUser(pid int, addr uintptr, data []byte) (count int, err error)
func PtraceSetOptions(pid int, options int) (err error)
func PtraceSetRegs(pid int, regs *PtraceRegs) (err error)
func PtraceSetRegs386(pid int, regs *PtraceRegs386) error
func PtraceSetRegsAmd64(pid int, regs *PtraceRegsAmd64) error
func PtraceSingleStep(pid int) (err error)
func PtraceSyscall(pid int, signal int) (err error)
func Pwrite(fd int, p []byte, offset int64) (n int, err error)
func RawSyscall(trap, a1, a2, a3 uintptr) (r1, r2 uintptr, err syscall.Errno)
func RawSyscall6(trap, a1, a2, a3, a4, a5, a6 uintptr) (r1, r2 uintptr, err syscall.Errno)
func RawSyscallNoError(trap, a1, a2, a3 uintptr) (r1, r2 uintptr)
func Read(fd int, p []byte) (n int, err error)
func ReadDirent(fd int, buf []byte) (n int, err error)
func Readlink(path string, buf []byte) (n int, err error)
func Readlinkat(dirfd int, path string, buf []byte) (n int, err error)
func Reboot(cmd int) (err error)
func Removexattr(path string, attr string) (err error)
func Rename(oldpath string, newpath string) (err error)
func Renameat(olddirfd int, oldpath string, newdirfd int, newpath string) (err error)
func Renameat2(olddirfd int, oldpath string, newdirfd int, newpath string, flags uint) (err error)
func RequestKey(keyType string, description string, callback string, destRingid int) (id int, err error)
func Rmdir(path string) error
func SchedGetaffinity(pid int, set *CPUSet) error
func SchedSetaffinity(pid int, set *CPUSet) error
func Seek(fd int, offset int64, whence int) (off int64, err error)
func Select(nfd int, r *FdSet, w *FdSet, e *FdSet, timeout *Timeval) (n int, err error)
func Sendfile(outfd int, infd int, offset *int64, count int) (written int, err error)
func Sendmsg(fd int, p, oob []byte, to Sockaddr, flags int) (err error)
func SendmsgN(fd int, p, oob []byte, to Sockaddr, flags int) (n int, err error)
func Sendto(fd int, p []byte, flags int, to Sockaddr) (err error)
func SetNonblock(fd int, nonblocking bool) (err error)
func Setdomainname(p []byte) (err error)
func Setenv(key, value string) error
func Setfsgid(gid int) (err error)
func Setfsuid(uid int) (err error)
func Setgid(uid int) (err error)
func Setgroups(gids []int) (err error)
func Sethostname(p []byte) (err error)
func Setns(fd int, nstype int) (err error)
func Setpgid(pid int, pgid int) (err error)
func Setpriority(which int, who int, prio int) (err error)
func Setregid(rgid int, egid int) (err error)
func Setresgid(rgid int, egid int, sgid int) (err error)
func Setresuid(ruid int, euid int, suid int) (err error)
func Setreuid(ruid int, euid int) (err error)
func Setrlimit(resource int, rlim *Rlimit) (err error)
func Setsid() (pid int, err error)
func SetsockoptByte(fd, level, opt int, value byte) (err error)
func SetsockoptICMPv6Filter(fd, level, opt int, filter *ICMPv6Filter) error
func SetsockoptIPMreq(fd, level, opt int, mreq *IPMreq) (err error)
func SetsockoptIPMreqn(fd, level, opt int, mreq *IPMreqn) (err error)
func SetsockoptIPv6Mreq(fd, level, opt int, mreq *IPv6Mreq) (err error)
func SetsockoptInet4Addr(fd, level, opt int, value [4]byte) (err error)
func SetsockoptInt(fd, level, opt int, value int) (err error)
func SetsockoptLinger(fd, level, opt int, l *Linger) (err error)
func SetsockoptString(fd, level, opt int, s string) (err error)
func SetsockoptTimeval(fd, level, opt int, tv *Timeval) (err error)
func Settimeofday(tv *Timeval) (err error)
func Setuid(uid int) (err error)
func Setxattr(path string, attr string, data []byte, flags int) (err error)
func Shutdown(fd int, how int) (err error)
func SignalName(s syscall.Signal) string
func Signalfd(fd int, mask *Sigset_t, flags int)
func Socket(domain, typ, proto int) (fd int, err error)
func Socketpair(domain, typ, proto int) (fd [2]int, err error)
func Splice(rfd int, roff *int64, wfd int, woff *int64, len int, flags int) (n int64, err error)
func Stat(path string, stat *Stat_t) (err error)
func Statfs(path string, buf *Statfs_t) (err error)
func Statx(dirfd int, path string, flags int, mask int, stat *Statx_t) (err error)
func Symlink(oldpath string, newpath string) (err error)
func Symlinkat(oldpath string, newdirfd int, newpath string) (err error)
func Sync()
func SyncFileRange(fd int, off int64, n int64, flags int) (err error)
func Syncfs(fd int) (err error)
func Syscall(trap, a1, a2, a3 uintptr) (r1, r2 uintptr, err syscall.Errno)
func Syscall6(trap, a1, a2, a3, a4, a5, a6 uintptr) (r1, r2 uintptr, err syscall.Errno)
func SyscallNoError(trap, a1, a2, a3 uintptr) (r1, r2 uintptr)
func Sysinfo(info *Sysinfo_t) (err error)
func Tee(rfd int, wfd int, len int, flags int) (n int64, err error)
func Tgkill(tgid int, tid int, sig syscall.Signal) (err error)
func Times(tms *Tms) (ticks uintptr, err error)
func TimespecToNsec(ts Timespec) int64
func TimevalToNsec(tv Timeval) int64
func Truncate(path string, length int64) (err error)
func Umask(mask int) (oldmask int)
func Uname(buf *Utsname) (err error)
func UnixCredentials(ucred *Ucred) []byte
func UnixRights(fds ...int) []byte
func Unlink(path string) error
func Unlinkat(dirfd int, path string, flags int) (err error)
func Unmount(target string, flags int) (err error)
func Unsetenv(key string) error
func Unshare(flags int) (err error)
func Ustat(dev int, ubuf *Ustat_t) (err error)
func Utime(path string, buf *Utimbuf) (err error)
func Utimes(path string, tv []Timeval) error
func UtimesNano(path string, ts []Timespec) error
func UtimesNanoAt(dirfd int, path string, ts []Timespec, flags int) error
func Vmsplice(fd int, iovs []Iovec, flags int) (int, error)
func Wait4(pid int, wstatus *WaitStatus, options int, rusage *Rusage) (wpid int, err error)
func Write(fd int, p []byte) (n int, err error)
type BlkpgIoctlArg
type BlkpgPartition
type CGroupStats
type CPUSet
- func (s *CPUSet) Clear(cpu int)
- func (s *CPUSet) Count() int
- func (s *CPUSet) IsSet(cpu int) bool
- func (s *CPUSet) Set(cpu int)
- func (s *CPUSet) Zero()

type Cmsghdr
- func (cmsg *Cmsghdr) SetLen(length int)

type Dirent
type EpollEvent
type Errno
type FdSet
type Flock_t
type FscryptKey
type FscryptPolicy
type Fsid
type Genlmsghdr
type HDDriveCmdHdr
type HDDriveID
type HDGeometry
type ICMPv6Filter
- func GetsockoptICMPv6Filter(fd, level, opt int) (*ICMPv6Filter, error)

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

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

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

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

type IfAddrmsg
type IfInfomsg
type Inet4Pktinfo
type Inet6Pktinfo
type InotifyEvent
type Iovec
- func (iov *Iovec) SetLen(length int)

type KeyctlDHParams
type Linger
- func GetsockoptLinger(fd, level, opt int) (*Linger, error)

type Msghdr
- func (msghdr *Msghdr) SetControllen(length int)

type Nfgenmsg
type NlAttr
type NlMsgerr
type NlMsghdr
type PacketMreq
type PerfEventAttr
type PerfEventMmapPage
type PollFd
type PtraceRegs
- func (r *PtraceRegs) PC() uint64
- func (r *PtraceRegs) SetPC(pc uint64)

type PtraceRegs386
type PtraceRegsAmd64
type RTCPLLInfo
type RTCTime
- func IoctlGetRTCTime(fd int) (*RTCTime, error)

type RTCWkAlrm
type RawSockaddr
type RawSockaddrALG
type RawSockaddrAny
type RawSockaddrCAN
type RawSockaddrHCI
type RawSockaddrInet4
type RawSockaddrInet6
type RawSockaddrL2
type RawSockaddrLinklayer
type RawSockaddrNetlink
type RawSockaddrPPPoX
type RawSockaddrRFCOMM
type RawSockaddrUnix
type RawSockaddrVM
type RawSockaddrXDP
type Rlimit
type RtAttr
type RtGenmsg
type RtMsg
type RtNexthop
type Rusage
type ScmTimestamping
type Signal
type SignalfdSiginfo
type Sigset_t
type SockExtendedErr
type SockFilter
type SockFprog
type Sockaddr
- func Accept(fd int) (nfd int, sa Sockaddr, err error)
- func Accept4(fd int, flags int) (nfd int, sa Sockaddr, err error)
- func Getpeername(fd int) (sa Sockaddr, err error)
- func Getsockname(fd int) (sa Sockaddr, err error)
- func Recvfrom(fd int, p []byte, flags int) (n int, from Sockaddr, err error)
- func Recvmsg(fd int, p, oob []byte, flags int) (n, oobn int, recvflags int, from Sockaddr, err error)

type SockaddrALG
type SockaddrCAN
type SockaddrHCI
type SockaddrInet4
type SockaddrInet6
type SockaddrL2
type SockaddrLinklayer
type SockaddrNetlink
type SockaddrPPPoE
type SockaddrRFCOMM
type SockaddrStorage
type SockaddrUnix
type SockaddrVM
type SockaddrXDP
type SocketControlMessage
- func ParseSocketControlMessage(b []byte) ([]SocketControlMessage, error)

golang.org/x/sys/unix

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

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

```
func SyscallNoError(trap, a1, a2, a3 uintptr) (r1, r2 uintptr)
```

And a bunch of wrapping functions to deal with architecture differences

Memory Backed File

- Option 1: Create a temporary file on /tmp
  - Program or user might not have permissions to write to this directory
  - Directory isn't guaranteed to exist
  - Harder to track memory usage
  - Must remember to remove the file after use

Memory Backed File

- Option 2: Ask the Linux Kernel to create a file backed by memory
  - Kernel automatically clears the memory when all references to the file are closed
- `func MemfdCreate(name string, flags int) (fd int, err error)`
  - Create a file descriptor backed by memory
- `func Ftruncate(fd int, length int64) (err error)`
  - Truncate the file to the size of our data
- `func Mmap(fd int, offset int64, length int, prot int, flags int) (data []byte, err error)`
  - Map the memory of the file into our memory space

ioctl

Interact with special devices

```
cmd := ethtool_cmd{cmd: 0x00000001} // ETHTOOL_GSET
dev := []byte{"eth0"}
ifr := ifreq{name: uintptr(unsafe.Pointer(&dev[0])), data: &cmd}
f, _ := unix.Socket(unix.AF_INET, unix.SOCK_STREAM, 0)
_, _, errno := syscall.Syscall(unix.SYS_IOCTL, uintptr(f.Fd()),
         uintptr(unix.SIOCETHTOOL), uintptr(unsafe.Pointer(&ifr)))
```

Sockets

Similar to ioctl, passing structures between Go and Linux

```
info := iptGetInfo{name: []byte{"nat"}}
f, _ := unix.Socket(unix.AF_INET, unix.SOCK_RAW, unix.IPPROTO_RAW)
_, _, err := syscall.Syscall6(unix.SYS_GETSOCKOPT,
       uintptr(f.Fd()), uintptr(unix.IPPROTO_IP), uintptr(64),
       uintptr(unsafe.Pointer(&info)),
       unsafe.Sizeof(iptGetInfo), 0)
fmt.Printf("table has %d entries.\n", info.numEntries)
```

Problems

- The previous ways of interacting with the kernel rely on passing memory.
- Fine for small values, but annoying for structs.

Linux Netlink

- An IPC mechanism enabling communication between kernel and userspace (or between multiple userspace programs)
- Netlink communications never leave the local host
- Body of Netlink messages are encoded as list of attributes (Length, Type, Value)
    - Attributes may be nested in Values
    - Value is 4-byte aligned
- Bus for each family of messages

github.com/mdlayher/netlink

- Attribute Encoder and Decoders
- Creates and manages underlying sockets
- Manages sequence IDs for you :)

Generic Netlink

- Make it easier for kernel modules to support Netlink, a generic bus was defined, with its own families.
- Attributes of a Generic Netlink family
  - ID (change with every reboot, must look up)
  - Name
  - Version
  - Multicast group

github.com/mdlayher/genetlink

- Helps creates valid Netlink messages for the Generic Netlink family

Getting WiFI interfaces

```
c, err := genetlink.Dial(nil)
family, err := c.GetFamily(name)
req := genetlink.Message{
    Header: genetlink.Header{
        Command: nl80211CommandGetInterface,
        Version: family.Version,
    },
}
flags := netlink.HeaderFlagsRequest | netlink.HeaderFlagsDump
msgs, err := c.Execute(req, family.ID, flags)
```

Parsing Interface fields

```
func (ifi *Interface) parseAttributes(attrs []netlink.Attribute)
error {
    for _, a := range attrs {
        switch a.Type {
        case nl80211.AttrIfindex:
            ifi.Index = int(nlenc.Uint32(a.Data))
        case nl80211.AttrIfname:
            ifi.Name = nlenc.String(a.Data)
    }

    return nil
}
```

Aside to Desktop Linux

- A userspace daemon would listen to these events
    - Apply policies
    - Create symlinks on devtmpfs
    - Maintain device metadata and emit enriched events
- systemd, eudev, vdev
    - Maintain /run directories of metadata
    - Emit enriched events to dbus
- (optionally) Another daemon farther process these events
    - Udisks2 listens on dbus, and exposes RPCs specifically for working with disks
    - Udevil automatically mounts disks by listening to udev events

uevent and kobject

- Linux creates and manages kobjects that tracks the addition, modification, and removal of devices
- Changes to the kobject result in uevents passed from the kernel to userspace over a netlink family
- The values for the last uevent are exposed by Linux as a file in sysfs

github.com/mdlayher/kobject

```
c, _ := kobject.New()
for {
    event, _ := c.Receive()
    // process event
}
```

Coldplug

- uevent has no buffering, only new events
- Can process existing events by walking sysfs
  - filepath.Walk("/sys", filepath.WalkFunc)
  - Read "uevent" file
  - Parse key-value pair from each line

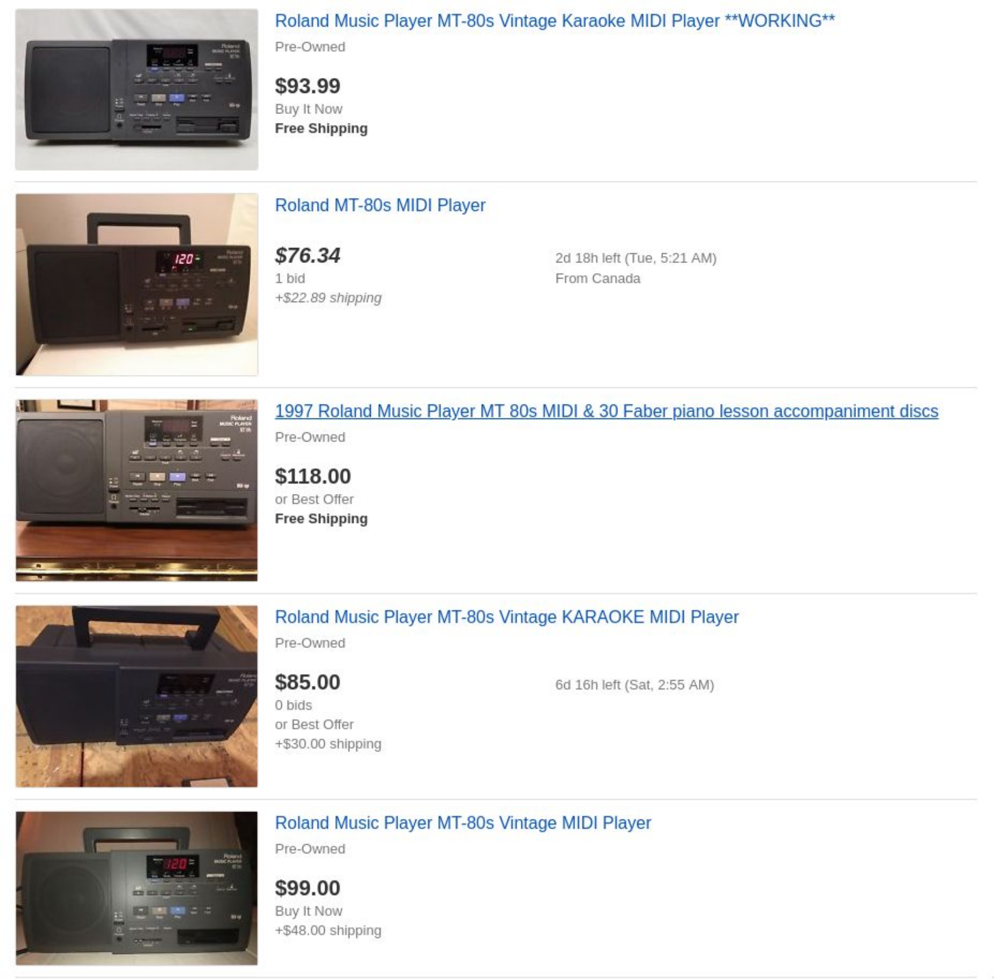
```
func ParseEvent(f io.Reader) map[string]string {
    event := make(map[string]string, 0)
    buf := bufio.NewScanner(f)
    for buf.Scan() {
        fields := strings.SplitN(buf.Text(), "=", 2)
        if len(fields) < 2 {
            return event
        }
        event[fields[0]] = fields[1]
    }
    return event
}
```

Track 3:
Hardware

Back in 90s

- Roland made a "music tutor" that played back MIDI from 3.5" floppies

In 2019



Roland Music Player MT-80s Vintage Karaoke MIDI Player **WORKING**

Pre-Owned

$93.99

Buy It Now

Free Shipping

Roland MT-80s MIDI Player

$76.34

1 bid

+$22.89 shipping

2d 18h left (Tue, 5:21 AM)

From Canada

1997 Roland Music Player MT 80s MIDI & 30 Faber piano lesson accompaniment discs

Pre-Owned

$118.00

or Best Offer

Free Shipping

Roland Music Player MT-80s Vintage KARAOKE MIDI Player

Pre-Owned

$85.00

0 bids

or Best Offer

+$30.00 shipping

6d 16h left (Sat, 2:55 AM)

Roland Music Player MT-80s Vintage MIDI Player

Pre-Owned

$99.00

Buy It Now

+$48.00 shipping

Building My Own

- Raspberry Pi 3+
- MIDI synthesizer attached to serial port
- Floppy drive over USB
- Player written in Go

Why Go?

JBD says (https://github.com/rakyll/go-hardware):

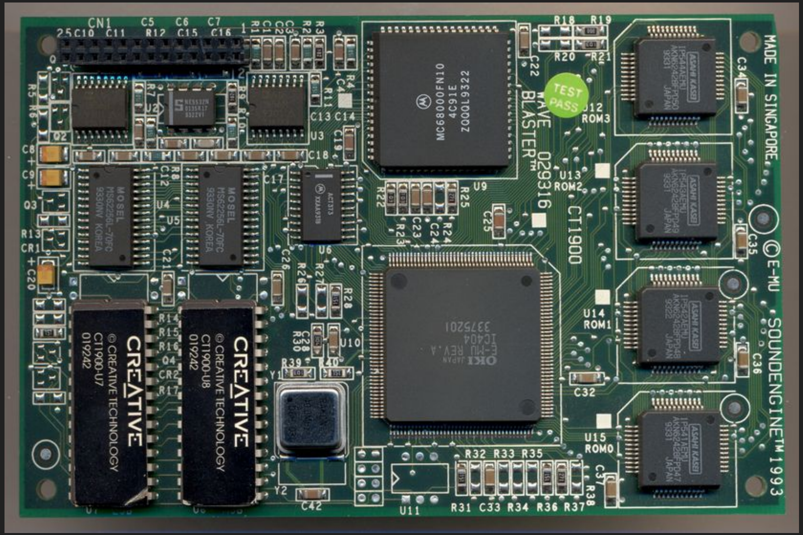
- "Out-of-the-box cross compilation"
- "Built-in concurrency primitives"
- "Go programs compile to static binaries"
- "Go is efficient, fast and has low memory footprint"
- "Code reuse"

I says:

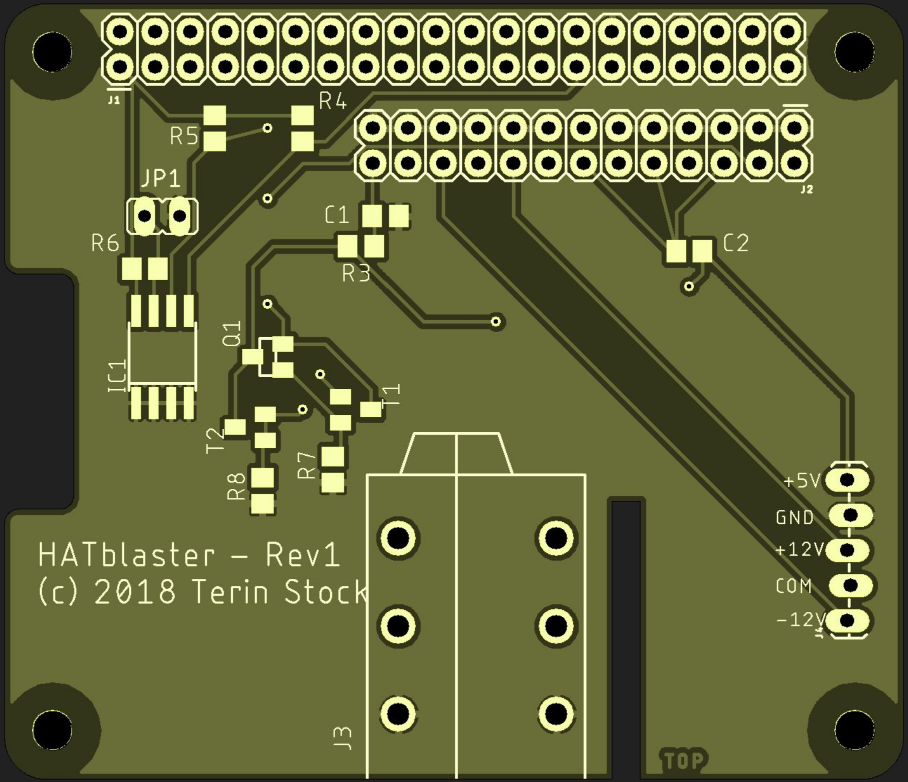
- "Not C"
- "Out of the box toolchain"

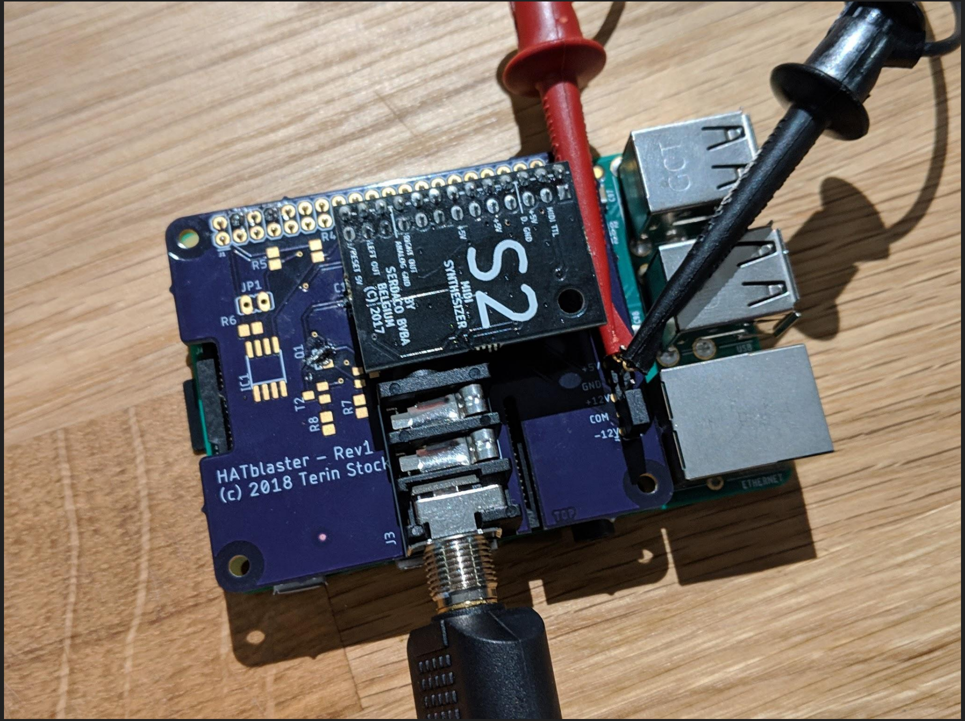
MIDI Synthesizer

- Many classic MIDI synths are available in WaveBlaster-compatible daughterboards
  - Creative WaveBlaster
  - Yamaha DB60XG
  - Roland SCB-55
  - Korg Ai20
- A few modern synthesizers are available
  - Serdaco Dreamblaster X2
  - Serdaco Dreamblaster S2



WaveBlaster Hat




HATblaster – Rev1
(c) 2018 Terin Stock
S2
MIDI SYNTHESIZER
BY SERDACO BVBA BELGIUM
(C) 2017
/RESET 5V
LEFT OUT
RIGHT OUT
ANALOG GND
+5V
+5V
D. GND
+5V
MIDI TTL
R4
R5
JP1
R6
C1
Q1
IC1
T2
R7
R8
J3
+5V
GND
+12V
COM
-12V
GCT
USB
ETHERNET

Track 4:
Demo

Demo gods got me :(

Track 5:
Conclusion

We're Hiring

SAY NO! TO SLOW LOAD TIMES!
CLOUDFLARE
BETTERINTERNET.COM
SECURE ALL SITES!
#BETTERINTERNET
CLOUDFLARE
THE GLOBAL CLOUD NETWORK
A SAFE INTERNET IS YOUR RIGHT
IMPEACH LATENCY
BETTERINTERNET.COM
PERFORMANCE AND SECURITY UNITED
JOIN 10 MILLION DOMAINS ON CLOUDFLARE
DATA BREACHES GOTTA GO!
A BETTER INTERNET IS HERE
KEEP HACKER HANDS OFF YOUR NETWORK!

Acknowledgements

- Matt Layher (https://mdlayher.com/)
  - Many libraries for working with Netlink and system administration
- Michael Stapelberg (https://michael.stapelberg.de/)
  - Main developer on gokrazy
- Chiel Kersten (http://members.home.nl/c.kersten/)
  - Large catalog of WaveBlaster-compatible daughterboards

Thanks!

- Blog: https://terinstock.com
- Code: https://git.terinstock.com
- Twitter: @terinjokes