Introduction

An experimental port of Go is available for the Plan 9 from Bell Labs operating system.

Supported architectures

The Plan 9 port of Go is available for the following architectures:

- 386
- amd64
- arm

Supported kernels

The current Go distribution has been tested successfully on:

- Plan 9 from Bell Labs (386 and arm) kernel
- 9front amd64 kernel
- Bell Labs 9k (amd64) kernel

Requirements

Go on Plan 9 requires a kernel providing the following system calls:

- tsemacquire (available since 2012-06-30)
- nsec (available since 2014-05-15)

A fix to the pread system call is required, so pread will not update the channel offset when reading a file:

- 9-pread-offset

The TCP connection control file must handle the "close" message to be able close a TCP connection gracefully, while waking up the readers:

- 9-tcp-close

A fix to Fossil is required to increment Qid.vers after a wstat, so a truncate followed by a read will return the content of the new file instead of the cached content.

- fossil-wstat-qid

You will need support for SHA-2 signatures in X.509 certificates, so you could download sources from repositories hosted on GoogleSource, GitHub and so on.

- libsec-x509-sha2

If you're running Plan 9 on Raspberry Pi, you'll require the latest bcm kernel from Richard Miller.

CA certificates for `crypto/x509` package needs to be installed at `/sys/lib/tls/ca.pem`. You can download it from `https://curl.haxx.se/ca/cacert.pem` or copy from another system.

For 9front, you may need to configure loopback addresses for standard library tests:

```
ip/ipconfig -P loopback /dev/null 127.1
ip/ipconfig -P loopback /dev/null ::1
```

Installation

Since Go is written in Go, you may want to bootstrap Go with Go 1.4.3, which is the latest release of Go written in C.

However, since the Go 1.4 runtime wasn't mature enough on Plan 9, we recommend you to bootstrap from a more recent version of Go, using another operating system.

Bootstrapping from Plan 9

First, install Go 1.4:

```
cd /tmp
git clone -b go1.4.3 https://go.googlesource.com/go go1.4
cd go1.4/src
hget http://9legacy.org/go/patch/syscall-exec.diff | ape/patch -p2
make.rc
```

The syscall-exec patch is required if you're running a machine with SMP enabled.

Then, set the `GOROOT_BOOTSTRAP` environment variable:

```
GOROOT_BOOTSTRAP=/tmp/go1.4
```

Finally, install the latest version of Go:

```
cd /tmp
git clone https://go.googlesource.com/go
cd go/src
all.rc
bind -a /tmp/go/bin /bin
```

Go is now ready to use.

Bootstrapping from another operating system

First, you must have installed Go on this operating system, following the Go installation instructions.

Then, you can cross-compile a Go toolchain for Plan 9:

```
cd $GOROOT/src
GOOS=plan9 GOARCH=386 ./bootstrap.bash
```

Then, the bootstrap toolchain will be available in `../../go-plan9-386-bootstrap.tbz`.

Finally, you can extract this archive to your Plan 9 machine.

For example:

```
cd /tmp
tar xzf go-plan9-386-bootstrap.tbz
bind -a /tmp/go-plan9-386-bootstrap/bin /bin
```

Go is now ready to use.

You'll be able to use this installation of Go to bootstrap future Go releases, by setting the `GOROOT_BOOTSTRAP` environment variable:

```
GOROOT_BOOTSTRAP=/tmp/go-plan9-386-bootstrap
```

Bootstraping from binaries

A binary package for plan9/386 is available.

This binary package is used to bootstrap Go on the plan9/386 builder.

```
cd /tmp
hget -o gobootstrap-plan9-386.tar.gz https://storage.googleapis.com/go-builder-
data/gobootstrap-plan9-386.tar.gz
mkdir gobootstrap-plan9-386
cd  gobootstrap-plan9-386
tar xzf ../gobootstrap-plan9-386.tar.gz
```

You'll be able to use this binary package of Go to bootstrap Go, by setting the `GOROOT_BOOTSTRAP` environment variable:

```
GOROOT_BOOTSTRAP=/tmp/gobootstrap-plan9-386
```

Other binary packages are available here and here.

Git

Git is not available on Plan 9. However a Git wrapper is available as a simple rc script. It includes everything you need to use the go tool.

Builders

Three Plan 9 builders are currently running and reporting the results to the Go Dashboard:

- plan9-386 is running Plan 9 from Bell Labs (386 kernel) on a virtual machine
- plan9-amd64-9front is running Plan 9 from Bell Labs (amd64 kernel) on a virtual machine
- plan9-arm is running Plan 9 from Bell Labs (arm kernel) on a Raspberry Pi 3

Status

The Plan 9 port of Go is considered experimental and is still a work-in-progress. There is a number of known issues available on the Go issue tracker.

Issues

Many issues are currently open on the Go issue tracker with the OS-Plan9 label.

Help

The Plan 9 port of Go is a community-driven port. Any help to improve the Go port on Plan 9 would be highly appreciated.

Maintainer

Many people have contributed to the Plan 9 port of Go. Since December 2013, the Plan 9 port of Go is maintained by David du Colombier.

Trivia

Many parts of Go are directly influenced by the Plan 9 system, as two of its three main designers worked on Plan 9 at Bell labs. Some of these connections are:

- The Gopher is designed by Renée French, who also designed Glenda, the Plan 9 bunny.
- The Gc compiler hosted in this repository uses a Plan 9-style loader.
- Go's Assembler is based on Plan 9's syntax. The Go 1.x stack-based ABI is borrowed from Plan 9.
- Although not recommended, there was also some support for [[Plan 9 C|GcToolchainTricks]], with the original standard library being descended from Plan 9's libc.