

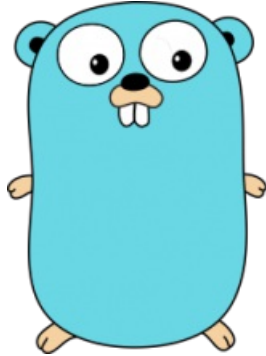
Primeros pasos para el programador de software:

Golang en Windows para sistemas Linux ARM

Sinopsis de Go

Sinopsis de pure.box

Desarrollo bajo Linux



Este tutorial sirve para la iniciación en la programación de soluciones individuales de comunicación y automatización sobre la base de sistemas Linux ARM en [Go](#).

El lenguaje desarrollado por Google se aprende con facilidad y rapidez, en contraposición con los lenguajes de script, pero además es muy potente y eficaz. Trae consigo un Cross-Compiler para los sistemas Linux ARM, facilitando así el flujo de trabajo en la programación de [pure.box](#), el polifacético servidor de comunicación y automatización de Wiesemann & Theis.

Tras la instalación y configuración de un entorno de trabajo, en este tutorial se muestra el manejo básico con Gotool a través de la barra de comandos en base a un sencillo programa Hello-World para el [pure.box](#).

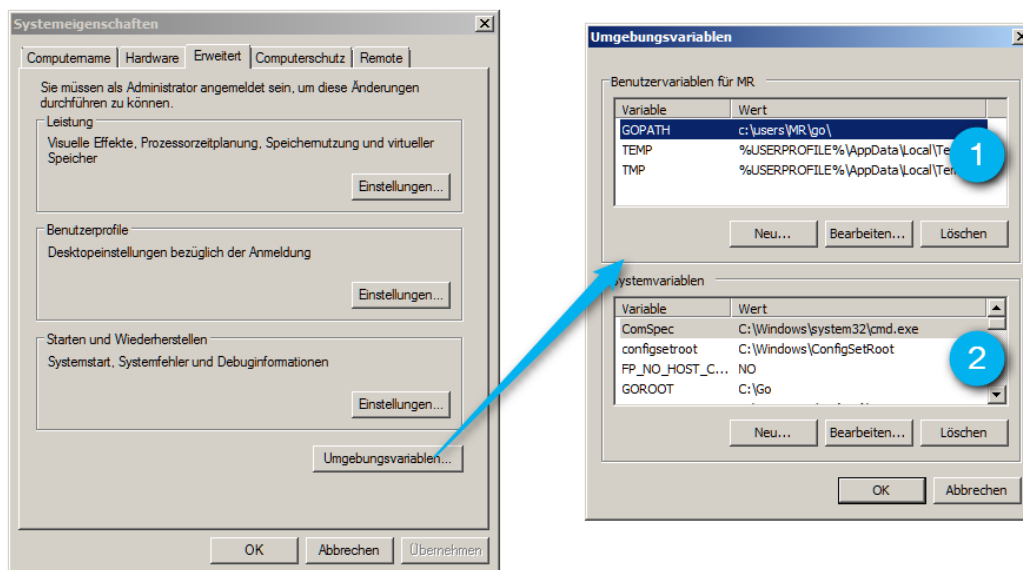
Paso 1: instalar Go

Go es un software libre y puede ser descargado desde [la página del proyecto](#). Tras la instalación hay que aplicar o ajustar algunas variables del entorno:

- **PATH**
La variable del entorno PATH tiene que contener el subdirectorio **bin** en el directorio de GO, para poder ejecutar Go en todo el sistema.
- **GOROOT**
La variable GOROOT del entorno contiene la ruta para el directorio de instalación de Go. Esta variable solo se aplica cuando se haya instalado Go en un directorio diferente al estándar **C:\Go bajo Windows**.
- **GOPATH**
La variable GOPATH contiene la ruta para el directorio de trabajo del usuario. Este directorio contendrá luego los subdirectorios **src** para fuentes, **lib** para bibliotecas precompiladas y **bin** para los archivos ejecutables.

Aplicación de las variables del entorno

Hacer clic dentro del **Panel de control** en **Sistema**, luego en **Ajustes avanzados del sistema**. Aquí se encuentra la opción **Variables del entorno**.



En la parte superior se encuentran las variables del entorno para el usuario actual **1**. En la parte inferior se encuentran las variables del entorno para todo el sistema **2**. Establecer la variable del usuario **GOPATH** e introducir la ruta al directorio de trabajo. Crear, si es necesario, la variable **GOROOT** en las variables del sistema. Ampliar ahora la variable del sistema **PATH** con el directorio **bin** en su **GOROOT**, por ejemplo **C:\Go\bin**.

Instalar Git

Para instalar ampliaciones de idiomas y paquetes de terceros sin demasiado esfuerzo, Go emplea Git, el software desarrollado por Linus Torvalds para la gestión repartida de versiones. Este puede ser descargado de [www.git-scm.com](#).

Verificar la instalación

La herramienta de Go es como una navaja suiza para la programación en Go. Para verificar si la instalación ha sido correcta se puede visualizar con ella el entorno de Go.

Abra el indicador de comandos e introduzca el comando siguiente:

```
go env
```

A continuación se obtiene una relación de todas las variables del entorno relevantes para Go:

```
C:\Users\Gopher\go>go env
set GOARCH=amd64
set GOBIN=
set GOEXE=.exe
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOOS=windows
set GOPATH=c:\users\Gopher\go\
set GORACE=
set GOROOT=C:\Go
set GOTOOOLDIR=C:\Go\pkg\tool\windows_amd64
set CC=gcc
set GOGCCFLAGS=-m64 -mthreads -fmessage-length=0 -fdebug-prefix-map=C:\Users\Gopher\AppData\Local\Temp\go-build015282382=/tmp/go-build -gno-record-gcc-switches
set CXX=g++
set CGO_ENABLED=1
```

Si aparece un mensaje de error, seguramente el directorio bin en **GOROOT** no se encuentra en la ruta de búsqueda. Analice la variable del entorno **PATH** con el comando **echo %PATH%**. Compruebe ahora si **GOPATH** se corresponde con su directorio de trabajo.

Paso 2: Hello World

El desarrollo de programas en Go se realiza dentro del GOPATH en el directorio `src`. Cree aquí el directorio `hello` y dentro de ese el archivo `hello.go`.

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
```

Línea 1:

Si desea crear un archivo ejecutable en Go, trabaje en el paquete `main`.

Línea 3:

El paquete `fmt` pone a disposición funciones de entrada y salida.

Línea 5:

La función `main()` es el punto de iniciación en el programa.

Línea 6:

`fmt.Println()` ofrece un string, seguido de un carácter Newline.

Para compilar el programa, introduzca ahora el comando siguiente:

```
go install hello
```

Si no se emite ningún mensaje de error, el proceso de traducción se ha ejecutado correctamente. El archivo ejecutable `hello.exe` se encuentra ahora en el directorio bin de su Go-Path.

```
./hello Hello World!
```

Paso 3: Hello pure.box - Cross-Compilation para ARM de Linux

Go facilita especialmente la compilación de aplicaciones para otras plataformas destino. Para traducir el ejemplo Hello World para pure.box solo tiene que aplicar temporalmente algunas variables del entorno:

- **GOARCH**
Con la variable GOARCH se establece la arquitectura destino. Para pure.box es **arm**.
- **GOOS**
Es el sistema operativo de la plataforma destino. Para pure.box hay que fijar esta variable en **linux**.
- **GOARM**
Con GOARM se comunica al compilador el modo de proceder con las operaciones de coma flotante (Floatingpoint). El valor para pure.box es **5**.

Introduzca lo siguiente en el indicador de comandos:

```
set GOOS=linux
set GOARCH=arm
set GOARM=5
```

Para iniciar la traducción para todas las plataformas, cambie al directorio del código fuente y ejecute **algo build**:

```
go build

dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: DADE-DEDA

Verzeichnis von C:\Users\gopher\go\src\hello

05.04.2017 10:51 <DIR> .
05.04.2017 10:51 <DIR> ..
09.12.2016 12:54 78 hello.go
05.04.2017 10:51 1.706.496 purebox

2 Datei(en), 1.706.574 Bytes
2 Verzeichnis(se), 61.440.692.224 Bytes frei
```

Como puede ver, en su directorio de trabajo existe ahora el archivo ejecutable **hello**. ¡La compilación cruzada con Go es realmente así de fácil!

Paso 4: Ejecutar en pure.box

Para transferir Binary a la Box existen varias opciones: Además del acceso por FTP, también se puede integrar la Box directamente en su red de Windows a través de SMB.

El modo más sencillo de probar una aplicación es ejecutarla manualmente en una sesión del terminal, para ello se debería activar SSH en la Box. Con ello, la transferencia de archivos por SCP se sobreentiende.

Si aún no tiene ningún cliente SSH instalado, descárguese por ejemplo [Putty](#). Para SCP necesita los parámetros siguientes: El **nombre de usuario** es "admin", la **contraseña** es la que haya configurado como tal en la administración. Introduzca como **nombre del host** el nombre del host o la dirección IP de su pure.box.

Copiar archivo:

```
pscp hello admin@hostname:/data/userfiles
```

Ejecutar programa:

Conéctese con la pure.box mediante Putty o con el comando ssh.

```
ssh admin@hostname
```

Ejecute a continuación el archivo **hello** en el directorio **/data/userfiles**.

```
./hello  
Hello pure.box
```

¿Alguna pregunta? ¡Llámenos!

¿Tiene alguna pregunta sobre estas instrucciones? Nuestro [Support](#) le atenderá con mucho gusto.



Le atendemos personalmente:

Wiesemann & Theis
GmbH
Porschestr. 12
42279 Wuppertal
Tel: +49 202/2680-110 (lu-vi de 8-17
horas)
Fax: +49-202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, salvo errores y modificaciones: como podemos cometer errores, no se debe utilizar nuestros enunciados sin verificarlos. Por favor, notifíquenos todas las erratas y malentendidos que detecte, para que podamos localizarlo y solucionarlo lo antes posible.

[Protección de datos](#)